

Merge Toolkit

1.1

(offline documentation is available at the package path “MergeIt/Documentation”)

Generated on Wed Jul 3 2024 18:06:59 for Merge Toolkit by Doxygen 1.10.0

Wed Jul 3 2024 18:06:59

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	13
3.1 Class List	13
4 File Index	21
4.1 File List	21
5 Namespace Documentation	29
5.1 Mergelt Namespace Reference	29
5.2 Mergelt.Core Namespace Reference	29
5.3 Mergelt.Core.Animations Namespace Reference	29
5.4 Mergelt.Core.Commands Namespace Reference	29
5.5 Mergelt.Core.Configs Namespace Reference	30
5.6 Mergelt.Core.Configs.Data Namespace Reference	30
5.7 Mergelt.Core.Configs.Effects Namespace Reference	30
5.8 Mergelt.Core.Configs.Elements Namespace Reference	30
5.9 Mergelt.Core.Configs.Hints Namespace Reference	30
5.10 Mergelt.Core.Configs.Icons Namespace Reference	30
5.11 Mergelt.Core.Configs.Inventory Namespace Reference	31
5.12 Mergelt.Core.Configs.LevelUp Namespace Reference	31
5.13 Mergelt.Core.Configs.Types Namespace Reference	31
5.13.1 Enumeration Type Documentation	31
5.13.1.1 ConsumableOperationType	31
5.13.1.2 ConsumableTargetType	31
5.13.1.3 CurrencyType	32
5.13.1.4 ElementType	32
5.14 Mergelt.Core.Configs.Windows Namespace Reference	32
5.15 Mergelt.Core.ElementsStock Namespace Reference	32
5.16 Mergelt.Core.Evolutions Namespace Reference	32
5.17 Mergelt.Core.FieldElements Namespace Reference	33
5.17.1 Enumeration Type Documentation	33
5.17.1.1 FieldElementState	33
5.18 Mergelt.Core.Helpers Namespace Reference	34
5.19 Mergelt.Core.Inventory Namespace Reference	34
5.20 Mergelt.Core.Messages Namespace Reference	34
5.21 Mergelt.Core.MVP Namespace Reference	34
5.22 Mergelt.Core.Saves Namespace Reference	34
5.22.1 Enumeration Type Documentation	35
5.22.1.1 GameSaveType	35

5.23 Mergelt.Core.Schemes Namespace Reference	35
5.24 Mergelt.Core.Schemes.Data Namespace Reference	35
5.25 Mergelt.Core.Services Namespace Reference	35
5.26 Mergelt.Core.User Namespace Reference	36
5.27 Mergelt.Core.Utils Namespace Reference	36
5.28 Mergelt.Core.WindowSystem Namespace Reference	36
5.29 Mergelt.Core.WindowSystem.Commands Namespace Reference	36
5.30 Mergelt.Core.WindowSystem.Data Namespace Reference	37
5.31 Mergelt.Core.WindowSystem.Factory Namespace Reference	37
5.32 Mergelt.Core.WindowSystem.Messages Namespace Reference	37
5.33 Mergelt.Core.WindowSystem.Windows Namespace Reference	37
5.33.1 Enumeration Type Documentation	37
5.33.1.1 WindowState	37
5.34 Mergelt.Editor Namespace Reference	38
5.35 Mergelt.Editor.Configs Namespace Reference	38
5.36 Mergelt.Editor.Configs.ItemsDrawers Namespace Reference	38
5.37 Mergelt.Editor.Core Namespace Reference	38
5.38 Mergelt.Editor.Core.LevelEditor Namespace Reference	38
5.39 Mergelt.Editor.Core.LevelEditor.Commands Namespace Reference	38
5.40 Mergelt.Editor.EvolutionsEditor Namespace Reference	39
5.41 Mergelt.Editor.Helpers Namespace Reference	39
5.42 Mergelt.Editor.LevelEditor Namespace Reference	39
5.43 Mergelt.Editor.LevelEditor.Commands Namespace Reference	39
5.44 Mergelt.Game Namespace Reference	40
5.45 Mergelt.Game.Commands Namespace Reference	40
5.46 Mergelt.Game.Configs Namespace Reference	40
5.47 Mergelt.Game.Configs.Services Namespace Reference	40
5.48 Mergelt.Game.Converters Namespace Reference	40
5.49 Mergelt.Game.Effects Namespace Reference	41
5.50 Mergelt.Game.Effects.Configs Namespace Reference	41
5.51 Mergelt.Game.Effects.Controllers Namespace Reference	41
5.52 Mergelt.Game.Effects.Parameters Namespace Reference	41
5.53 Mergelt.Game.ElementsStock Namespace Reference	41
5.54 Mergelt.Game.Enums Namespace Reference	42
5.54.1 Enumeration Type Documentation	42
5.54.1.1 HudPanelType	42
5.55 Mergelt.Game.Evolutions Namespace Reference	42
5.56 Mergelt.Game.Factories Namespace Reference	42
5.57 Mergelt.Game.Factories.ElementInfo Namespace Reference	42
5.57.1 Enumeration Type Documentation	43
5.57.1.1 ElementInfoType	43
5.58 Mergelt.Game.Factories.Field Namespace Reference	43

5.59 Mergelt.Game.Factories.FieldElement Namespace Reference	43
5.60 Mergelt.Game.Factories.Icons Namespace Reference	43
5.61 Mergelt.Game.Factories.Inventory Namespace Reference	43
5.62 Mergelt.Game.Field Namespace Reference	44
5.63 Mergelt.Game.Field.Actions Namespace Reference	44
5.64 Mergelt.Game.Field.Elements Namespace Reference	44
5.65 Mergelt.Game.Field.Elements.Animations Namespace Reference	44
5.66 Mergelt.Game.Field.Elements.Generator Namespace Reference	44
5.67 Mergelt.Game.Helpers Namespace Reference	45
5.68 Mergelt.Game.HUD Namespace Reference	45
5.69 Mergelt.Game.Managers Namespace Reference	45
5.70 Mergelt.Game.Messages Namespace Reference	45
5.71 Mergelt.Game.Services Namespace Reference	46
5.72 Mergelt.Game.Services.Saves Namespace Reference	46
5.73 Mergelt.Game.Services.Saves.Strategies Namespace Reference	47
5.74 Mergelt.Game.TopPanel Namespace Reference	47
5.75 Mergelt.Game.UI Namespace Reference	47
5.76 Mergelt.Game.UI.InfoPanel Namespace Reference	47
5.76.1 Enumeration Type Documentation	47
5.76.1.1 ElementActionType	47
5.77 Mergelt.Game.UI.InventoryPanel Namespace Reference	48
5.78 Mergelt.Game.User Namespace Reference	48
5.79 Mergelt.Game.Windows Namespace Reference	48
5.80 Mergelt.Game.Windows.ElementInfo Namespace Reference	48
5.81 Mergelt.Game.Windows.Inventory Namespace Reference	48
5.82 Mergelt.Game.Windows.UserProgress Namespace Reference	48
5.83 Mergelt.SimpleDI Namespace Reference	49
5.84 Mergelt.SimpleDI.ReservedInterfaces Namespace Reference	49
6 Class Documentation	51
6.1 Mergelt.Editor.LevelEditor.Commands.ActionCommandBase Class Reference	51
6.1.1 Detailed Description	51
6.1.2 Constructor & Destructor Documentation	51
6.1.2.1 ActionCommandBase()	51
6.1.3 Member Function Documentation	52
6.1.3.1 Execute()	52
6.1.3.2 Undo()	52
6.1.4 Member Data Documentation	52
6.1.4.1 Window	52
6.2 Mergelt.Editor.LevelEditor.Commands.ActionCommandManager Class Reference	52
6.2.1 Detailed Description	52
6.2.2 Member Function Documentation	53

6.2.2.1 ExecuteCommand()	53
6.2.2.2 Redo()	53
6.2.2.3 Undo()	53
6.3 Mergelt.Game.Messages.ActivateHintsMessage Class Reference	53
6.3.1 Detailed Description	54
6.3.2 Property Documentation	54
6.3.2.1 Active	54
6.4 Mergelt.Core.Animations.AnimationControllerBase Class Reference	54
6.4.1 Detailed Description	54
6.4.2 Member Function Documentation	55
6.4.2.1 Initialize()	55
6.4.2.2 SetState() [1/2]	55
6.4.2.3 SetState() [2/2]	55
6.4.2.4 SetState< T >()	55
6.5 Mergelt.Core.Animations.AnimationListener Class Reference	56
6.5.1 Detailed Description	56
6.5.2 Member Function Documentation	56
6.5.2.1 ResetAnimationState()	56
6.6 Mergelt.Core.Animations.AnimatorStateMachineBehaviour Class Reference	56
6.6.1 Detailed Description	57
6.6.2 Member Function Documentation	57
6.6.2.1 OnStateEnter()	57
6.6.2.2 OnStateUpdate()	57
6.7 Mergelt.Editor.LevelEditor.Commands.ApplyCellCommand Class Reference	58
6.7.1 Detailed Description	58
6.7.2 Constructor & Destructor Documentation	58
6.7.2.1 ApplyCellCommand()	58
6.7.3 Member Function Documentation	59
6.7.3.1 Execute()	59
6.7.3.2 Undo()	59
6.8 Mergelt.Game.Effects.Controllers.BaseEffect Class Reference	59
6.8.1 Detailed Description	60
6.8.2 Member Function Documentation	60
6.8.2.1 Finish()	60
6.8.2.2 OnFinished()	60
6.8.2.3 OnStarted()	60
6.8.2.4 Setup()	61
6.8.2.5 Start()	61
6.8.2.6 Update()	61
6.8.3 Member Data Documentation	61
6.8.3.1 Animator	61
6.8.3.2 ConfigsService	61

6.8.3.3 FinishedCallback	62
6.8.3.4 Target	62
6.8.4 Property Documentation	62
6.8.4.1 Started	62
6.9 Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter Class Reference	62
6.9.1 Detailed Description	63
6.9.2 Member Function Documentation	63
6.9.2.1 FinishCallbackHandler()	63
6.9.2.2 GetPresenter()	64
6.9.2.3 Setup()	64
6.9.3 Member Data Documentation	64
6.9.3.1 Presenter	64
6.10 Mergelt.Editor.EvolutionsEditor.BaseNode Class Reference	64
6.10.1 Detailed Description	65
6.10.2 Member Function Documentation	65
6.10.2.1 Dispose()	65
6.10.2.2 GetPort()	65
6.10.2.3 SaveData()	65
6.10.3 Property Documentation	66
6.10.3.1 Id	66
6.11 Mergelt.Game.Services.Saves.Strategies.BaseSerializeStrategy Class Reference	66
6.11.1 Detailed Description	66
6.11.2 Member Function Documentation	67
6.11.2.1 Load< T >()	67
6.11.2.2 Save< T >()	67
6.11.3 Property Documentation	67
6.11.3.1 SaveDir	67
6.12 Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer Class Reference	67
6.12.1 Detailed Description	68
6.12.2 Constructor & Destructor Documentation	68
6.12.2.1 BaseSettingsDrawer()	68
6.12.3 Member Function Documentation	68
6.12.3.1 Dispose()	68
6.12.3.2 Expand()	69
6.12.3.3 Hide()	69
6.12.3.4 RotateArrow()	69
6.12.3.5 Show()	69
6.12.4 Member Data Documentation	69
6.12.4.1 Arrow	69
6.12.4.2 Config	69
6.12.4.3 Parent	70
6.12.4.4 Root	70

6.13 Mergelt.Game.Services.Saves.Strategies.BinaryFormatterStrategy Class Reference	70
6.13.1 Detailed Description	71
6.13.2 Member Function Documentation	71
6.13.2.1 Load< T >()	71
6.13.2.2 Save< T >()	71
6.14 Mergelt.Core.Utils.Bindable< T > Class Template Reference	72
6.14.1 Detailed Description	72
6.14.2 Member Function Documentation	72
6.14.2.1 Hold()	72
6.14.2.2 Release()	72
6.14.2.3 SetValueSilently()	73
6.14.2.4 Subscribe() [1/2]	73
6.14.2.5 Subscribe() [2/2]	73
6.14.2.6 Unsubscribe() [1/2]	73
6.14.2.7 Unsubscribe() [2/2]	73
6.14.3 Property Documentation	74
6.14.3.1 Value	74
6.15 Mergelt.Core.WindowSystem.BlackoutComponent Class Reference	74
6.15.1 Detailed Description	74
6.15.2 Member Function Documentation	75
6.15.2.1 SetActive()	75
6.15.2.2 SetLayer()	75
6.16 Mergelt.Editor.LevelEditor.Commands.ChangeCellCommand Class Reference	75
6.16.1 Detailed Description	76
6.16.2 Constructor & Destructor Documentation	76
6.16.2.1 ChangeCellCommand()	76
6.16.3 Member Function Documentation	76
6.16.3.1 Execute()	76
6.16.3.2 Undo()	77
6.17 Mergelt.Game.Commands.CheckEvolutionsProgressCommand Class Reference	77
6.17.1 Detailed Description	78
6.17.2 Member Function Documentation	78
6.17.2.1 Execute()	78
6.18 Mergelt.Game.Messages.CheckGeneratorMessage Class Reference	78
6.18.1 Detailed Description	79
6.18.2 Property Documentation	79
6.18.2.1 GeneratorPoint	79
6.19 Mergelt.Game.Commands.CheckOpeningGeneratorsCommand Class Reference	79
6.19.1 Detailed Description	80
6.19.2 Member Function Documentation	80
6.19.2.1 Execute()	80
6.20 Mergelt.Editor.LevelEditor.Commands.ClearCellCommand Class Reference	80

6.20.1 Detailed Description	81
6.20.2 Constructor & Destructor Documentation	81
6.20.2.1 ClearCellCommand()	81
6.20.3 Member Function Documentation	81
6.20.3.1 Execute()	81
6.20.3.2 Undo()	82
6.21 Mergelt.Game.Messages.ClickElementMessage Class Reference	82
6.21.1 Detailed Description	82
6.21.2 Property Documentation	82
6.21.2.1 Cell	82
6.22 Mergelt.Core.WindowSystem.Messages.CloseWindowMessage Class Reference	83
6.22.1 Detailed Description	83
6.22.2 Property Documentation	83
6.22.2.1 Presenter	83
6.23 Mergelt.Core.Commands.Command Class Reference	83
6.23.1 Detailed Description	84
6.23.2 Member Function Documentation	84
6.23.2.1 Dispose()	84
6.23.2.2 Execute()	84
6.23.2.3 ExecuteAsync()	85
6.23.2.4 Finish()	85
6.23.2.5 OnDispose()	85
6.23.3 Event Documentation	85
6.23.3.1 Finished	85
6.24 Mergelt.Core.Commands.CommandManager Class Reference	85
6.24.1 Detailed Description	86
6.24.2 Member Function Documentation	86
6.24.2.1 Add()	86
6.24.2.2 Run()	86
6.24.2.3 RunAsync()	87
6.24.2.4 RunSimultaneously()	87
6.24.3 Property Documentation	87
6.24.3.1 Executing	87
6.24.4 Event Documentation	87
6.24.4.1 Finished	87
6.25 Mergelt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer Class Reference	88
6.25.1 Detailed Description	89
6.25.2 Constructor & Destructor Documentation	89
6.25.2.1 CommonSettingsDrawer()	89
6.25.3 Member Function Documentation	89
6.25.3.1 Dispose()	89
6.25.3.2 Draw()	89

6.25.3.3 GetUxml()	90
6.25.3.4 OnExpand()	90
6.26 Mergelt.Core.FieldElements.ConfigParameters Class Reference	90
6.26.1 Detailed Description	91
6.26.2 Property Documentation	91
6.26.2.1 ElementConfig	91
6.26.2.2 EvolutionData	91
6.27 Mergelt.Game.Converters.ConfigProcessor Class Reference	91
6.27.1 Detailed Description	92
6.27.2 Member Function Documentation	92
6.27.2.1 BuildLevel()	92
6.27.2.2 Convert()	92
6.27.2.3 ConvertToFieldElement()	93
6.27.2.4 ConvertToFieldElementData() [1/2]	93
6.27.2.5 ConvertToFieldElementData() [2/2]	94
6.28 Mergelt.Game.Configs.ConfigsContainer Class Reference	95
6.28.1 Detailed Description	95
6.28.2 Property Documentation	95
6.28.2.1 EffectConfigs	95
6.28.2.2 GameConfig	95
6.28.2.3 HintsConfig	95
6.28.2.4 IconsConfig	96
6.28.2.5 InventoryConfig	96
6.28.2.6 LevelConfig	96
6.28.2.7 LevelUpConfig	96
6.28.2.8 Prefabs	96
6.29 Mergelt.Editor.Helpers.ConfigsPostprocessor Class Reference	97
6.29.1 Detailed Description	97
6.30 Mergelt.Game.Configs.Services.ConfigsService Class Reference	97
6.30.1 Detailed Description	98
6.30.2 Member Function Documentation	98
6.30.2.1 GetConfig()	98
6.30.2.2 GetConfigs()	98
6.30.2.3 GetCurrencyIcon()	98
6.30.2.4 GetEffectConfig< T >()	99
6.30.2.5 GetElementPrefab()	99
6.30.2.6 GetEvolutionByElement()	99
6.30.2.7 GetEvolutionData()	99
6.30.2.8 GetEvolutionIdByElement()	100
6.30.2.9 GetLevelUpData()	100
6.30.2.10 Load()	100
6.30.3 Property Documentation	100

6.30.3.1 GameConfig	100
6.30.3.2 HintsConfig	101
6.30.3.3 InventoryConfig	101
6.30.3.4 LevelConfig	101
6.31 Mergelt.Game.Configs.Services.ConfigsServiceModel Class Reference	101
6.31.1 Detailed Description	102
6.31.2 Member Function Documentation	102
6.31.2.1 StoreEffectsConfigs()	102
6.31.2.2 StoreElementsPrefabs()	102
6.31.2.3 StoreGameConfig()	102
6.31.2.4 StoreHintsConfig()	103
6.31.2.5 StoreIconsConfig()	103
6.31.2.6 StoreInventoryConfig()	103
6.31.2.7 StoreLevelConfig()	103
6.31.2.8 StoreLevelUpConfig()	103
6.31.3 Property Documentation	104
6.31.3.1 EffectConfigs	104
6.31.3.2 ElementConfigs	104
6.31.3.3 GameConfig	104
6.31.3.4 HintsConfig	104
6.31.3.5 IconsConfig	104
6.31.3.6 InventoryConfig	104
6.31.3.7 LevelConfig	105
6.31.3.8 LevelUpConfig	105
6.31.3.9 Prefabs	105
6.32 Mergelt.Core.Configs.Elements.ConsumableSettings Struct Reference	105
6.32.1 Detailed Description	105
6.32.2 Property Documentation	105
6.32.2.1 Operation	105
6.32.2.2 Target	106
6.32.2.3 Value	106
6.33 Mergelt.Game.Messages.CreateElementMessage Class Reference	106
6.33.1 Detailed Description	106
6.33.2 Property Documentation	106
6.33.2.1 FromPosition	106
6.33.2.2 NewElement	107
6.33.2.3 ToPoint	107
6.34 Mergelt.Editor.Configs.CurrencyComponent Class Reference	107
6.34.1 Detailed Description	107
6.34.2 Constructor & Destructor Documentation	108
6.34.2.1 CurrencyComponent()	108
6.34.3 Property Documentation	108

6.34.3.1 AmountBinding	108
6.34.3.2 Title	108
6.34.3.3 TypeBinding	108
6.35 Mergelt.Editor.Configs.CurrencyComponentsTraits Class Reference	108
6.35.1 Detailed Description	109
6.35.2 Member Function Documentation	109
6.35.2.1 Init()	109
6.36 Mergelt.Game.Services.CurrencyService Class Reference	109
6.36.1 Detailed Description	110
6.36.2 Member Function Documentation	110
6.36.2.1 Sell()	110
6.36.2.2 TryPay()	110
6.37 Mergelt.Core.Configs.Elements.CurrencySettings Struct Reference	111
6.37.1 Detailed Description	111
6.37.2 Property Documentation	111
6.37.2.1 Amount	111
6.37.2.2 Currency	111
6.38 Mergelt.SimpleDI.DiContainer Class Reference	112
6.38.1 Detailed Description	112
6.38.2 Member Function Documentation	112
6.38.2.1 Clear()	112
6.38.2.2 Drop< TInterface >()	112
6.38.2.3 Get< TInterface >()	113
6.38.2.4 Register< TInterface, TType >()	113
6.38.2.5 Register< TType >()	113
6.38.2.6 RegisterInterfacesFor< TType >()	114
6.39 Mergelt.SimpleDI.DiHandler Class Reference	114
6.39.1 Detailed Description	114
6.39.2 Member Function Documentation	114
6.39.2.1 AddDependency< TType >()	114
6.39.2.2 AsSingleton()	115
6.39.2.3 AsSingleton< TType >()	115
6.40 Mergelt.SimpleDI.DiRoot Class Reference	115
6.40.1 Detailed Description	116
6.40.2 Member Function Documentation	116
6.40.2.1 Run()	116
6.41 Mergelt.Core.Schemes.Data.EdgeData Class Reference	116
6.41.1 Detailed Description	116
6.41.2 Member Data Documentation	116
6.41.2.1 InputNodeID	116
6.41.2.2 InputPortName	116
6.41.2.3 OutputNodeID	117

6.41.2.4 OutputPortName	117
6.42 Mergelt.Game.Commands.EffectCommand Class Reference	117
6.42.1 Detailed Description	118
6.42.2 Constructor & Destructor Documentation	118
6.42.2.1 EffectCommand()	118
6.42.3 Member Function Documentation	118
6.42.3.1 Execute()	118
6.43 Mergelt.Core.Configs.Effects.EffectConfig Class Reference	118
6.43.1 Detailed Description	119
6.43.2 Member Data Documentation	119
6.43.2.1 Duration	119
6.43.2.2 Name	119
6.44 Mergelt.Game.Effects.EffectsFactory Class Reference	119
6.44.1 Detailed Description	120
6.44.2 Member Function Documentation	120
6.44.2.1 CreateEffect< T >() [1/2]	120
6.44.2.2 CreateEffect< T >() [2/2]	120
6.45 Mergelt.Game.Effects.EffectsManager Class Reference	121
6.45.1 Detailed Description	121
6.45.2 Member Function Documentation	121
6.45.2.1 AddEffect()	121
6.45.2.2 Update()	121
6.46 Mergelt.Game.Messages.ElementActionMessage Class Reference	122
6.46.1 Detailed Description	122
6.46.2 Property Documentation	122
6.46.2.1 ActionType	122
6.46.2.2 Element	122
6.47 Mergelt.Game.UI.InfoPanel.ElementCommonInfo Class Reference	122
6.47.1 Detailed Description	123
6.47.2 Member Function Documentation	123
6.47.2.1 OnTrySetup()	123
6.47.3 Property Documentation	124
6.47.3.1 ActionType	124
6.48 Mergelt.Core.Configs.Elements.ElementCommonSettings Class Reference	124
6.48.1 Detailed Description	124
6.48.2 Property Documentation	124
6.48.2.1 ConsumableSettings	124
6.48.2.2 Description	125
6.48.2.3 Expanded	125
6.48.2.4 Icon	125
6.48.2.5 IsConsumable	125
6.48.2.6 Name	125

6.48.2.7 SellCostSettings	125
6.48.2.8 SplitCostSettings	126
6.48.2.9 UnlockCostSettings	126
6.49 Mergelt.Core.Configs.Elements.ElementConfig Class Reference	126
6.49.1 Detailed Description	127
6.49.2 Member Function Documentation	127
6.49.2.1 GenerateGuid()	127
6.49.2.2 GetIconComponent()	127
6.49.3 Property Documentation	127
6.49.3.1 CommonSettings	127
6.49.3.2 GeneratorSettings	127
6.49.3.3 Id	128
6.49.3.4 Type	128
6.50 Mergelt.Editor.Configs.ItemsDrawers.ElementConfigInspector Class Reference	128
6.50.1 Detailed Description	128
6.50.2 Member Function Documentation	129
6.50.2.1 CreateInspectorGUI()	129
6.50.3 Property Documentation	129
6.50.3.1 Root	129
6.50.3.2 Target	129
6.51 Mergelt.Core.Configs.Elements.ElementGeneratorSettings Class Reference	130
6.51.1 Detailed Description	130
6.51.2 Property Documentation	130
6.51.2.1 AdSkip	130
6.51.2.2 Charged	130
6.51.2.3 ChargeTime	130
6.51.2.4 DropCount	131
6.51.2.5 EnergyCost	131
6.51.2.6 Expanded	131
6.51.2.7 GenerateBlocked	131
6.51.2.8 GenerateItems	131
6.51.2.9 IsUnlimited	131
6.51.2.10 MaxDrop	132
6.51.2.11 MinDrop	132
6.51.2.12 NeedOpen	132
6.51.2.13 OpenTime	132
6.51.2.14 SkipChargeCostSettings	132
6.51.2.15 SkipOpenCostSettings	132
6.51.2.16 Stack	133
6.52 Mergelt.Game.UI.InfoPanel.ElementInfo Class Reference	133
6.52.1 Detailed Description	133
6.52.2 Member Function Documentation	134

6.52.2.1 ActionButtonClick()	134
6.52.2.2 Clear()	134
6.52.2.3 Destroy()	134
6.52.2.4 OnActionButtonClick()	134
6.52.2.5 OnClear()	134
6.52.2.6 OnStart()	134
6.52.2.7 SendActionMessage()	135
6.52.2.8 TrySetup()	135
6.52.3 Member Data Documentation	135
6.52.3.1 ConfigsService	135
6.52.3.2 MessageBus	135
6.52.3.3 SelectedElement	135
6.52.4 Property Documentation	136
6.52.4.1 ActionType	136
6.53 Mergelt.Game.Windows.ElementInfo.ElementInfoArgs Class Reference	136
6.53.1 Detailed Description	136
6.53.2 Property Documentation	136
6.53.2.1 ElementConfig	136
6.54 Mergelt.Game.Factories.ElementInfo.ElementInfoFactory Class Reference	137
6.54.1 Detailed Description	137
6.54.2 Member Function Documentation	137
6.54.2.1 CreateElementWindowItem()	137
6.54.2.2 CreateUnknownElementWindowItem()	138
6.55 Mergelt.Game.Windows.ElementInfo.ElementInfoItemComponent Class Reference	138
6.55.1 Detailed Description	139
6.55.2 Member Function Documentation	139
6.55.2.1 Awake()	139
6.55.2.2 OnDestroy()	139
6.55.2.3 SetIcon()	139
6.55.2.4 SetUnknown()	139
6.55.2.5 Setup()	140
6.55.3 Member Data Documentation	140
6.55.3.1 ClickEvent	140
6.55.4 Property Documentation	140
6.55.4.1 RectTransform	140
6.56 Mergelt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent Class Reference	140
6.56.1 Detailed Description	141
6.56.2 Member Function Documentation	141
6.56.2.1 SetItem()	141
6.56.2.2 SetSpacing()	141
6.56.2.3 SetTitle()	141
6.56.3 Property Documentation	142

6.56.3.1 ItemsGrid	142
6.56.3.2 PanelTitle	142
6.56.3.3 RectTransform	142
6.57 Mergelt.Game.Windows.ElementInfo.ElementInfoModel Class Reference	142
6.57.1 Detailed Description	143
6.58 Mergelt.Game.UI.InfoPanel.ElementInfoPanelComponent Class Reference	143
6.58.1 Detailed Description	143
6.59 Mergelt.Game.Windows.ElementInfo.ElementInfoPresenter Class Reference	143
6.59.1 Detailed Description	144
6.59.2 Member Function Documentation	144
6.59.2.1 OnDestroyWindow()	144
6.59.2.2 OnInitialize()	145
6.60 Mergelt.Game.Windows.ElementInfo.ElementInfoWindow Class Reference	145
6.60.1 Detailed Description	147
6.60.2 Property Documentation	147
6.60.2.1 CreatedFromPanel	147
6.60.2.2 EvolutionPanel	147
6.60.2.3 GeneratesPanel	147
6.60.2.4 ScrollContent	148
6.60.2.5 SpacingBetweenPanels	148
6.60.2.6 SpacingInsidePanels	148
6.60.2.7 WindowTitleText	148
6.61 Mergelt.Game.Windows.ElementInfo.ElementInfoWindowContent Class Reference	148
6.61.1 Detailed Description	149
6.62 Mergelt.Editor.EvolutionsEditor.ElementNode Class Reference	149
6.62.1 Detailed Description	149
6.62.2 Constructor & Destructor Documentation	150
6.62.2.1 ElementNode()	150
6.62.3 Member Function Documentation	150
6.62.3.1 BuildContextualMenu()	150
6.62.3.2 Dispose()	151
6.62.3.3 SaveData()	151
6.62.3.4 Validate()	151
6.62.4 Property Documentation	152
6.62.4.1 Config	152
6.62.4.2 Data	152
6.62.4.3 Id	152
6.63 Mergelt.Core.Schemes.Data.ElementNodeData Class Reference	152
6.63.1 Detailed Description	153
6.63.2 Member Function Documentation	153
6.63.2.1 Copy< T >()	153
6.63.3 Property Documentation	153

6.63.3.1 ElementConfig	153
6.63.3.2 Id	154
6.63.3.3 Position	154
6.64 Mergelt.Game.UI.InfoPanel.ElementOpenInfo Class Reference	154
6.64.1 Detailed Description	155
6.64.2 Member Function Documentation	155
6.64.2.1 Destroy()	155
6.64.2.2 OnStart()	155
6.64.2.3 OnTrySetup()	156
6.64.3 Property Documentation	156
6.64.3.1 ActionType	156
6.65 Mergelt.Game.Configs.ElementPrefab Class Reference	156
6.65.1 Detailed Description	156
6.65.2 Property Documentation	156
6.65.2.1 Prefab	156
6.65.2.2 Type	157
6.66 Mergelt.Game.UI.InfoPanel.ElementSellInfo Class Reference	157
6.66.1 Detailed Description	158
6.66.2 Member Function Documentation	158
6.66.2.1 Destroy()	158
6.66.2.2 OnStart()	158
6.66.2.3 OnTrySetup()	159
6.66.3 Property Documentation	159
6.66.3.1 ActionType	159
6.67 Mergelt.Game.Services.ElementService Class Reference	159
6.67.1 Detailed Description	160
6.67.2 Member Function Documentation	160
6.67.2.1 TrySell()	160
6.67.2.2 TrySplit()	160
6.67.2.3 TryUnlock()	161
6.68 Mergelt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo Class Reference	161
6.68.1 Detailed Description	162
6.68.2 Member Function Documentation	162
6.68.2.1 Destroy()	162
6.68.2.2 OnStart()	162
6.68.2.3 OnTrySetup()	163
6.68.3 Property Documentation	163
6.68.3.1 ActionType	163
6.69 Mergelt.Game.UI.InfoPanel.ElementSpeedUpOpeningInfo Class Reference	163
6.69.1 Detailed Description	164
6.69.2 Member Function Documentation	164
6.69.2.1 OnClear()	164

6.69.2.2 OnStart()	165
6.69.2.3 OnTrySetup()	165
6.69.3 Property Documentation	165
6.69.3.1 ActionType	165
6.70 Mergelt.Game.UI.InfoPanel.ElementSplitInfo Class Reference	166
6.70.1 Detailed Description	167
6.70.2 Member Function Documentation	167
6.70.2.1 Destroy()	167
6.70.2.2 OnStart()	167
6.70.2.3 OnTrySetup()	167
6.70.3 Property Documentation	168
6.70.3.1 ActionType	168
6.71 Mergelt.Game.ElementsStock.ElementsStockComponent Class Reference	168
6.71.1 Detailed Description	168
6.71.2 Member Function Documentation	168
6.71.2.1 Initialize()	168
6.71.2.2 OnDestroy()	169
6.71.2.3 PopElement()	169
6.71.2.4 SetupElement()	169
6.72 Mergelt.Game.ElementsStock.ElementsStockData Class Reference	170
6.72.1 Detailed Description	170
6.72.2 Property Documentation	170
6.72.2.1 Elements	170
6.73 Mergelt.Game.Services.ElementsStockService Class Reference	170
6.73.1 Detailed Description	171
6.73.2 Member Function Documentation	171
6.73.2.1 Add()	171
6.73.2.2 CreateStock()	171
6.73.2.3 GetCurrent()	171
6.73.2.4 GetData()	172
6.73.2.5 GetNext()	172
6.73.2.6 Remove()	172
6.73.2.7 SetupStock()	172
6.74 Mergelt.Game.Services.ElementsStockServiceModel Class Reference	173
6.74.1 Detailed Description	173
6.74.2 Property Documentation	173
6.74.2.1 Elements	173
6.75 Mergelt.Game.UI.InfoPanel.ElementUnlockInfo Class Reference	173
6.75.1 Detailed Description	174
6.75.2 Member Function Documentation	174
6.75.2.1 Destroy()	174
6.75.2.2 OnStart()	175

6.75.2.3 OnTrySetup()	175
6.75.3 Property Documentation	175
6.75.3.1 ActionType	175
6.76 Mergelt.Game.Messages.EndDragElementMessage Class Reference	175
6.76.1 Detailed Description	176
6.76.2 Property Documentation	176
6.76.2.1 FromPoint	176
6.76.2.2 Position	176
6.76.2.3 ToGameObject	176
6.77 Mergelt.Game.HUD.EnergyComponent Class Reference	176
6.77.1 Detailed Description	177
6.77.2 Member Function Documentation	177
6.77.2.1 OnApplyModel()	177
6.77.2.2 Start()	178
6.77.3 Property Documentation	178
6.77.3.1 Type	178
6.78 Mergelt.Game.Messages.EnergyRestoredMessage Class Reference	178
6.78.1 Detailed Description	178
6.79 Mergelt.Game.Services.EnergyService Class Reference	179
6.79.1 Detailed Description	179
6.79.2 Member Function Documentation	179
6.79.2.1 Dispose()	179
6.79.2.2 Initialize()	179
6.80 Mergelt.Core.Configs.Data.EvolutionData Class Reference	180
6.80.1 Detailed Description	180
6.80.2 Constructor & Destructor Documentation	180
6.80.2.1 EvolutionData()	180
6.80.3 Member Function Documentation	181
6.80.3.1 Equals() [1/2]	181
6.80.3.2 Equals() [2/2]	181
6.80.3.3 GetHashCode()	181
6.80.4 Property Documentation	181
6.80.4.1 Chain	181
6.80.4.2 Description	181
6.80.4.3 Discovered	182
6.80.4.4 Id	182
6.80.4.5 Name	182
6.81 Mergelt.Editor.EvolutionsEditor.EvolutionNode Class Reference	182
6.81.1 Detailed Description	183
6.81.2 Constructor & Destructor Documentation	183
6.81.2.1 EvolutionNode()	183
6.81.3 Member Function Documentation	184

6.81.3.1 Dispose()	184
6.81.3.2 SaveData()	184
6.81.4 Property Documentation	184
6.81.4.1 Data	184
6.81.4.2 Description	184
6.81.4.3 Discovered	184
6.81.4.4 Id	185
6.81.4.5 Name	185
6.82 Mergelt.Core.Schemes.Data.EvolutionNodeData Class Reference	185
6.82.1 Detailed Description	186
6.82.2 Member Function Documentation	186
6.82.2.1 Copy< T >()	186
6.82.3 Property Documentation	186
6.82.3.1 Description	186
6.82.3.2 Discovered	186
6.82.3.3 Id	187
6.82.3.4 Name	187
6.82.3.5 Position	187
6.83 Mergelt.Core.Evolutions.EvolutionProgressData Class Reference	187
6.83.1 Detailed Description	188
6.83.2 Property Documentation	188
6.83.2.1 EvolutionId	188
6.83.2.2 Progress	188
6.84 Mergelt.Editor.EvolutionsEditor.EvolutionsEditorWindow Class Reference	188
6.84.1 Detailed Description	189
6.84.2 Member Function Documentation	189
6.84.2.1 Initialize()	189
6.84.2.2 OnDestroy()	189
6.84.2.3 OnDisable()	190
6.84.2.4 OnEnable()	190
6.84.2.5 OnFocus()	190
6.84.2.6 Reload()	190
6.84.2.7 ShowWindow()	190
6.84.3 Property Documentation	191
6.84.3.1 Initialized	191
6.85 Mergelt.Editor.LevelEditor.EvolutionSelection Class Reference	191
6.85.1 Detailed Description	191
6.85.2 Member Function Documentation	191
6.85.2.1 ClearSelection()	191
6.85.2.2 Init()	192
6.85.2.3 LockCell()	192
6.85.2.4 SelectCell()	192

6.85.2.5 SelectElement()	192
6.85.2.6 SelectEvolution()	193
6.85.2.7 UpdateCell()	193
6.85.3 Property Documentation	193
6.85.3.1 ElementData	193
6.85.3.2 Evoltem	193
6.85.3.3 Evolution	193
6.86 Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView Class Reference	194
6.86.1 Detailed Description	194
6.86.2 Constructor & Destructor Documentation	194
6.86.2.1 EvolutionsGraphView()	194
6.86.3 Member Function Documentation	195
6.86.3.1 BakeEvolutionsData()	195
6.86.3.2 BuildContextualMenu()	195
6.86.3.3 ClearNodes()	196
6.86.3.4 GetCompatiblePorts()	196
6.86.3.5 HandleEvent()	196
6.86.3.6 Restore()	197
6.86.3.7 Save()	198
6.86.3.8 Validate()	199
6.86.4 Property Documentation	199
6.86.4.1 canCopySelection	199
6.86.4.2 canDuplicateSelection	199
6.87 Mergelt.Editor.EvolutionsEditor.EvolutionsGraphViewTraits Class Reference	200
6.87.1 Detailed Description	200
6.87.2 Member Function Documentation	200
6.87.2.1 Init()	200
6.88 Mergelt.Game.Evolutions.EvolutionsProgressData Class Reference	200
6.88.1 Detailed Description	201
6.88.2 Property Documentation	201
6.88.2.1 EvolutionsProgress	201
6.89 Mergelt.Game.Services.EvolutionsService Class Reference	201
6.89.1 Detailed Description	202
6.89.2 Member Function Documentation	202
6.89.2.1 CreateEvolutionsProgress()	202
6.89.2.2 GetData()	202
6.89.2.3 GetEvolutionChain()	202
6.89.2.4 GetEvolutionProgress()	203
6.89.2.5 GetGeneratedBy()	203
6.89.2.6 GetGenerates()	203
6.89.2.7 SetupEvolutionsProgress()	204
6.89.2.8 UpdateProgress()	204

6.90 Mergelt.Game.Services.EvolutionsServiceModel Class Reference	204
6.90.1 Detailed Description	205
6.90.2 Property Documentation	205
6.90.2.1 EvolutionsProgress	205
6.91 Mergelt.Game.Field.Actions.FieldActionProcessorBase Class Reference	205
6.91.1 Detailed Description	205
6.91.2 Member Function Documentation	206
6.91.2.1 ProcessClick()	206
6.91.2.2 ProcessEndDrag()	206
6.91.3 Member Data Documentation	206
6.91.3.1 FieldLogicModel	206
6.91.3.2 MessageBus	206
6.91.3.3 SaveService	206
6.92 Mergelt.Game.Field.FieldCellComponent Class Reference	207
6.92.1 Detailed Description	207
6.92.2 Member Function Documentation	207
6.92.2.1 AttachObject()	207
6.92.2.2 DetachObject()	208
6.92.2.3 Initialize()	208
6.92.2.4 OnBeginDrag()	208
6.92.2.5 OnDrag()	208
6.92.2.6 OnEndDrag()	209
6.92.2.7 OnPointerDown()	209
6.92.2.8 OnPointerUp()	209
6.92.2.9 ToggleSelector()	209
6.92.3 Property Documentation	210
6.92.3.1 FieldElementPresenter	210
6.92.3.2 ObjectContainer	210
6.92.3.3 Point	210
6.92.3.4 RectTransform	210
6.93 Mergelt.Game.Field.Actions.FieldConsumableProcessor Class Reference	210
6.93.1 Detailed Description	211
6.93.2 Member Function Documentation	211
6.93.2.1 ProcessClick()	211
6.94 Mergelt.Game.Field.FieldData Class Reference	212
6.94.1 Detailed Description	213
6.94.2 Member Data Documentation	213
6.94.2.1 FieldHeight	213
6.94.2.2 FieldWidth	213
6.94.2.3 SavedElementsData	213
6.95 Mergelt.Core.FieldElements.FieldElement Class Reference	213
6.95.1 Detailed Description	214

6.95.2 Member Function Documentation	214
6.95.2.1 ToString()	214
6.95.3 Property Documentation	214
6.95.3.1 ConfigParameters	214
6.95.3.2 GeneratorOpenParameters	214
6.95.3.3 GeneratorParameters	215
6.95.3.4 InfoParameters	215
6.95.3.5 ProducedByParameters	215
6.95.3.6 ProduceParameters	215
6.96 Mergelt.Game.Field.Elements.Animations.FieldElementAnimationController Class Reference	215
6.96.1 Detailed Description	216
6.96.2 Member Function Documentation	216
6.96.2.1 GetFieldElementState()	216
6.96.2.2 Initialize()	216
6.96.2.3 SetState< T >()	217
6.97 Mergelt.Core.FieldElements.FieldElementData Class Reference	217
6.97.1 Detailed Description	217
6.97.2 Member Data Documentation	217
6.97.2.1 ConfigParameters	217
6.97.2.2 GeneratorOpenParameters	218
6.97.2.3 GeneratorParameters	218
6.97.2.4 InfoParameters	218
6.98 Mergelt.Game.Factories.FieldElement.FieldElementFactory Class Reference	218
6.98.1 Detailed Description	218
6.98.2 Member Function Documentation	219
6.98.2.1 CreateFieldElement() [1/3]	219
6.98.2.2 CreateFieldElement() [2/3]	219
6.98.2.3 CreateFieldElement() [3/3]	219
6.99 Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorModel Class Reference	219
6.99.1 Detailed Description	220
6.100 Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorPresenter Class Reference	220
6.100.1 Detailed Description	221
6.100.2 Constructor & Destructor Documentation	221
6.100.2.1 FieldElementGeneratorPresenter()	221
6.100.3 Member Function Documentation	222
6.100.3.1 OnDispose()	222
6.100.3.2 Release()	222
6.100.3.3 Update()	222
6.101 Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorView Class Reference	222
6.101.1 Detailed Description	224
6.101.2 Member Function Documentation	224
6.101.2.1 HideTimer()	224

6.101.2.2 SetTimer()	224
6.102 Mergelt.Core.Configs.Elements.FieldElementIconComponent Class Reference	224
6.102.1 Detailed Description	225
6.102.2 Member Function Documentation	225
6.102.2.1 GetImage()	225
6.102.2.2 SetBlocked()	225
6.102.3 Property Documentation	225
6.102.3.1 RectTransform	225
6.103 Mergelt.Game.Field.Elements.FieldElementModel Class Reference	226
6.103.1 Detailed Description	226
6.103.2 Property Documentation	226
6.103.2.1 ClicksCount	226
6.103.2.2 IsBusy	226
6.103.2.3 IsLocked	227
6.103.2.4 Point	227
6.103.2.5 Selected	227
6.104 Mergelt.Game.Field.Elements.FieldElementPresenter Class Reference	227
6.104.1 Detailed Description	228
6.105 Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel > Class Template Reference	229
6.105.1 Detailed Description	230
6.105.2 Member Function Documentation	230
6.105.2.1 Activate()	230
6.105.2.2 EndDrag()	230
6.105.2.3 GetModel()	231
6.105.2.4 GetView()	231
6.105.2.5 OnInitialize()	231
6.105.2.6 Release()	231
6.105.2.7 Remove()	232
6.105.2.8 ResetAnimationState()	232
6.105.2.9 ResetPosition()	232
6.105.2.10 Select()	232
6.105.2.11 SetBusy()	232
6.105.2.12 SetLock()	233
6.105.2.13 SetParent()	233
6.105.2.14 SetPoint()	233
6.105.2.15 SetState()	233
6.105.2.16 StartDrag()	234
6.105.2.17 Update()	234
6.105.3 Member Data Documentation	234
6.105.3.1 FieldElement	234
6.105.4 Property Documentation	234

6.105.4.1 Canvas	234
6.105.4.2 IsAvailable	235
6.105.4.3 IsBusy	235
6.105.4.4 IsLocked	235
6.105.4.5 RectTransform	235
6.105.4.6 State	236
6.105.4.7 Transform	236
6.106 Mergelt.Game.Field.Elements.Animations.FieldElementStateMachineBehaviour Class Reference .	236
6.106.1 Detailed Description	236
6.106.2 Member Function Documentation	237
6.106.2.1 OnStateExit()	237
6.107 Mergelt.Game.Field.Elements.FieldElementView Class Reference	237
6.107.1 Detailed Description	238
6.107.2 Member Function Documentation	238
6.107.2.1 Lock()	238
6.107.2.2 ResetState()	239
6.107.3 Member Data Documentation	239
6.107.3.1 ResetStateEvent	239
6.107.4 Property Documentation	239
6.107.4.1 AnimationController	239
6.107.4.2 Canvas	239
6.107.4.3 GraphicRaycaster	239
6.107.4.4 RectTransform	240
6.108 Mergelt.Game.Factories.FieldElement.FieldElementVisualFactory Class Reference	240
6.108.1 Detailed Description	240
6.108.2 Member Function Documentation	241
6.108.2.1 CreateFieldElement()	241
6.109 Mergelt.Game.Factories.Field.FieldFactory Class Reference	242
6.109.1 Detailed Description	242
6.109.2 Member Function Documentation	242
6.109.2.1 CreateField()	242
6.109.2.2 CreateFieldCell()	243
6.109.3 Member Data Documentation	243
6.109.3.1 FieldCellObjectEvenPath	243
6.109.3.2 FieldCellObjectOddPath	243
6.109.3.3 FieldGameObjectPath	243
6.110 Mergelt.Game.Messages.FieldFullMessage Class Reference	244
6.110.1 Detailed Description	244
6.111 Mergelt.Game.Field.Actions.FieldGenerationProcessor Class Reference	244
6.111.1 Detailed Description	245
6.111.2 Member Function Documentation	245
6.111.2.1 ProcessClick()	245

6.112 Mergelt.Game.Field.FieldLogic Class Reference	246
6.112.1 Detailed Description	246
6.112.2 Member Function Documentation	246
6.112.2.1 Dispose()	246
6.112.2.2 Initialize()	247
6.113 Mergelt.Game.Field.FieldLogicModel Class Reference	247
6.113.1 Detailed Description	247
6.113.2 Member Function Documentation	247
6.113.2.1 GetPosition()	247
6.113.3 Property Documentation	248
6.113.3.1 AllGenerators	248
6.113.3.2 CellComponents	248
6.113.3.3 CellSize	248
6.113.3.4 CellSpacing	248
6.113.3.5 FieldElements	248
6.113.3.6 FieldHeight	248
6.113.3.7 FieldRect	249
6.113.3.8 FieldWidth	249
6.113.3.9 OpeningGenerator	249
6.113.3.10 PositionOffset	249
6.114 Mergelt.Game.Field.Actions.FieldMergeProcessor Class Reference	249
6.114.1 Detailed Description	250
6.114.2 Member Function Documentation	250
6.114.2.1 ProcessClick()	250
6.114.2.2 ProcessEndDrag()	250
6.115 Mergelt.Game.Field.FieldModel Class Reference	251
6.115.1 Detailed Description	251
6.116 Mergelt.Game.Field.FieldPresenter Class Reference	252
6.116.1 Detailed Description	252
6.116.2 Constructor & Destructor Documentation	253
6.116.2.1 FieldPresenter()	253
6.116.3 Member Function Documentation	253
6.116.3.1 Initialize()	253
6.116.3.2 OnDispose()	254
6.117 Mergelt.Game.Field.FieldView Class Reference	254
6.117.1 Detailed Description	255
6.117.2 Property Documentation	255
6.117.2.1 CellsContainer	255
6.117.2.2 CellSpacing	255
6.118 Mergelt.Core.Configs.GameConfig Class Reference	256
6.118.1 Detailed Description	256
6.118.2 Member Data Documentation	256

6.118.2.1 EnergyCap	256
6.118.2.2 EnergyRestoreTime	256
6.118.2.3 InitialHardCurrency	256
6.118.2.4 InitialSoftCurrency	257
6.118.2.5 InitialSplittersCount	257
6.118.2.6 LevelString	257
6.118.2.7 SaveInterval	257
6.119 Mergelt.Game.Services.GameFieldActionsService Class Reference	257
6.119.1 Detailed Description	257
6.119.2 Member Function Documentation	258
6.119.2.1 Dispose()	258
6.119.2.2 Initialize()	258
6.120 Mergelt.Game.Services.GameFieldService Class Reference	258
6.120.1 Detailed Description	258
6.120.2 Member Function Documentation	259
6.120.2.1 CreateNewElement()	259
6.120.2.2 Dispose()	259
6.120.2.3 GetFreeCell()	259
6.120.2.4 Initialize()	260
6.121 Mergelt.Core.Configs.Icons.GameliconsConfig Class Reference	260
6.121.1 Detailed Description	260
6.121.2 Property Documentation	260
6.121.2.1 InfoPanelCurrencyIcons	260
6.122 Mergelt.Game.Services.Saves.GameLoadService Class Reference	261
6.122.1 Detailed Description	261
6.122.2 Member Function Documentation	261
6.122.2.1 Load< T >()	261
6.123 Mergelt.Game.GameRoot Class Reference	261
6.123.1 Detailed Description	262
6.123.2 Member Function Documentation	262
6.123.2.1 OnInstall()	262
6.123.2.2 Run()	263
6.124 Mergelt.Game.Services.Saves.GameSaveEveryIntervalService Class Reference	263
6.124.1 Detailed Description	263
6.124.2 Member Function Documentation	263
6.124.2.1 Dispose()	263
6.124.2.2 Initialize()	264
6.124.2.3 OnApplicationQuit()	264
6.124.2.4 Save()	264
6.124.2.5 Update()	264
6.125 Mergelt.Game.Services.Saves.GameSaveOnEveryActionService Class Reference	265
6.125.1 Detailed Description	265

6.125.2 Member Function Documentation	265
6.125.2.1 Dispose()	265
6.125.2.2 Initialize()	265
6.125.2.3 OnApplicationQuit()	265
6.125.2.4 Save()	266
6.126 Mergelt.Game.Services.Saves.GameSaveOnQuitService Class Reference	266
6.126.1 Detailed Description	266
6.126.2 Member Function Documentation	267
6.126.2.1 Dispose()	267
6.126.2.2 Initialize()	267
6.126.2.3 OnApplicationQuit()	267
6.126.2.4 Save()	267
6.127 Mergelt.Game.Services.GameService Class Reference	268
6.127.1 Detailed Description	268
6.127.2 Member Function Documentation	268
6.127.2.1 Dispose()	268
6.127.2.2 Initialize()	268
6.127.3 Member Data Documentation	269
6.127.3.1 _messageBus	269
6.128 Mergelt.Game.Services.GameServiceModel Class Reference	269
6.128.1 Detailed Description	269
6.128.2 Property Documentation	269
6.128.2.1 LoadedLevel	269
6.128.2.2 MainCamera	269
6.128.2.3 MainCanvas	270
6.129 Mergelt.Core.Configs.Elements.GeneratableElement Class Reference	270
6.129.1 Detailed Description	270
6.129.2 Member Data Documentation	270
6.129.2.1 Element	270
6.129.2.2 Possibility	270
6.130 Mergelt.Core.FieldElements.GeneratableFieldElement Class Reference	271
6.130.1 Detailed Description	271
6.130.2 Property Documentation	271
6.130.2.1 Config	271
6.130.2.2 Possibility	271
6.131 Mergelt.Editor.GeneratedItemsCreator Class Reference	271
6.131.1 Detailed Description	272
6.131.2 Constructor & Destructor Documentation	272
6.131.2.1 GeneratedItemsCreator()	272
6.131.3 Member Function Documentation	272
6.131.3.1 Clear()	272
6.131.3.2 Dispose()	273

6.131.3.3 Generate()	273
6.132 Mergelt.Editor.GeneratedItemUI Class Reference	273
6.132.1 Detailed Description	273
6.132.2 Constructor & Destructor Documentation	274
6.132.2.1 GeneratedItemUI()	274
6.132.3 Member Function Documentation	275
6.132.3.1 Dispose()	275
6.132.4 Property Documentation	275
6.132.4.1 Index	275
6.132.4.2 RemoveGeneratedItemEvent	275
6.133 Mergelt.Game.Messages.GeneratorBusyMessage Class Reference	275
6.133.1 Detailed Description	276
6.133.2 Property Documentation	276
6.133.2.1 Point	276
6.134 Mergelt.Game.Messages.GeneratorOpenedMessage Class Reference	276
6.134.1 Detailed Description	276
6.134.2 Property Documentation	276
6.134.2.1 GeneratorPoint	276
6.135 Mergelt.Core.FieldElements.GeneratorOpenParameters Class Reference	277
6.135.1 Detailed Description	277
6.135.2 Member Function Documentation	277
6.135.2.1 CopyFrom()	277
6.135.3 Property Documentation	277
6.135.3.1 IsOpening	277
6.135.3.2 RemainingTime	278
6.135.3.3 StartOpeningTime	278
6.136 Mergelt.Game.Messages.GeneratorOpenStartMessage Class Reference	278
6.136.1 Detailed Description	278
6.136.2 Property Documentation	279
6.136.2.1 GeneratorPoint	279
6.137 Mergelt.Core.FieldElements.GeneratorParameters Class Reference	279
6.137.1 Detailed Description	279
6.137.2 Member Function Documentation	280
6.137.2.1 CopyFrom()	280
6.137.3 Property Documentation	280
6.137.3.1 AvailableToDrop	280
6.137.3.2 ChargedCount	280
6.137.3.3 Charging	280
6.137.3.4 DroppedElements	280
6.137.3.5 MinDropChargeTime	281
6.137.3.6 MinDropFullChargeTime	281
6.137.3.7 RemainChargeTime	281

6.137.3.8 StartChargingTime	281
6.138 Mergelt.Game.Messages.GeneratorRestoredMessage Class Reference	281
6.138.1 Detailed Description	282
6.138.2 Property Documentation	282
6.138.2.1 GeneratorPoint	282
6.139 Mergelt.Game.Services.GeneratorService Class Reference	282
6.139.1 Detailed Description	282
6.139.2 Member Function Documentation	283
6.139.2.1 Dispose()	283
6.139.2.2 Initialize()	283
6.139.2.3 TryOpen()	283
6.139.2.4 TrySkipCharging()	284
6.139.2.5 TrySkipOpening()	284
6.139.2.6 Update()	284
6.140 Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer Class Reference	285
6.140.1 Detailed Description	286
6.140.2 Constructor & Destructor Documentation	286
6.140.2.1 GeneratorSettingsBaseDrawer()	286
6.140.3 Member Function Documentation	287
6.140.3.1 Clear()	287
6.140.3.2 Dispose()	287
6.140.3.3 Draw()	287
6.140.3.4 GetUxml()	288
6.140.3.5 OnClear()	288
6.140.3.6 OnExpand()	288
6.141 Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsEvolutionDrawer Class Reference	288
6.141.1 Detailed Description	290
6.141.2 Constructor & Destructor Documentation	290
6.141.2.1 GeneratorSettingsEvolutionDrawer()	290
6.141.3 Member Function Documentation	290
6.141.3.1 Draw()	290
6.141.3.2 OnClear()	290
6.142 Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsInspectorDrawer Class Reference	291
6.142.1 Detailed Description	292
6.142.2 Constructor & Destructor Documentation	292
6.142.2.1 GeneratorSettingsInspectorDrawer()	292
6.142.3 Member Function Documentation	292
6.142.3.1 Draw()	292
6.143 Mergelt.Game.Field.Elements.Generator.GeneratorTimerComponent Class Reference	293
6.143.1 Detailed Description	293
6.143.2 Member Function Documentation	293
6.143.2.1 StartTimer()	293

6.144 Mergelt.Core.FieldElements.GridPoint Struct Reference	293
6.144.1 Detailed Description	294
6.144.2 Constructor & Destructor Documentation	294
6.144.2.1 GridPoint()	294
6.144.3 Member Function Documentation	294
6.144.3.1 CompareTo()	294
6.144.3.2 Copy()	295
6.144.3.3 Create()	295
6.144.3.4 Equals() [1/2]	295
6.144.3.5 Equals() [2/2]	295
6.144.3.6 GetHashCode()	295
6.144.3.7 operator"!="()	296
6.144.3.8 operator==()	296
6.144.3.9 ToString()	296
6.144.4 Member Data Documentation	296
6.144.4.1 Default	296
6.144.5 Property Documentation	296
6.144.5.1 X	296
6.144.5.2 Y	297
6.145 Mergelt.Game.HUD.HardCurrencyComponent Class Reference	297
6.145.1 Detailed Description	298
6.145.2 Member Function Documentation	298
6.145.2.1 OnApplyModel()	298
6.145.3 Property Documentation	298
6.145.3.1 Type	298
6.146 Mergelt.Core.Configs.Hints.HintsConfig Class Reference	298
6.146.1 Detailed Description	299
6.146.2 Property Documentation	299
6.146.2.1 Interval	299
6.147 Mergelt.Game.Managers.HintsManager Class Reference	299
6.147.1 Detailed Description	299
6.147.2 Member Function Documentation	300
6.147.2.1 Dispose()	300
6.147.2.2 Initialize()	300
6.147.2.3 Update()	300
6.148 Mergelt.Game.HUD.HudComponent Class Reference	301
6.148.1 Detailed Description	301
6.149 Mergelt.Game.HUD.HudPanelBase Class Reference	301
6.149.1 Detailed Description	302
6.149.2 Member Function Documentation	302
6.149.2.1 Start()	302
6.149.3 Property Documentation	302

6.149.3.1 Type	302
6.150 Mergelt.Game.Helpers.HudTargetComponent Class Reference	302
6.150.1 Detailed Description	303
6.151 Mergelt.Editor.LevelEditor.Commands.IActionCommand Interface Reference	303
6.151.1 Detailed Description	303
6.152 Mergelt.Editor.LevelEditor.Commands.IActionCommandManager Interface Reference	303
6.152.1 Detailed Description	304
6.153 Mergelt.Core.Animations.IAnimationController Interface Reference	304
6.153.1 Detailed Description	304
6.153.2 Member Function Documentation	304
6.153.2.1 SetState< T >()	304
6.154 Mergelt.Core.Animations.IAnimationListener Interface Reference	305
6.154.1 Detailed Description	305
6.155 Mergelt.Core.Commands.ICommand Interface Reference	305
6.155.1 Detailed Description	306
6.155.2 Event Documentation	306
6.155.2.1 Finished	306
6.156 Mergelt.Core.Commands.ICommandManager Interface Reference	306
6.156.1 Detailed Description	307
6.156.2 Property Documentation	307
6.156.2.1 Executing	307
6.156.3 Event Documentation	307
6.156.3.1 Finished	307
6.157 Mergelt.Game.Factories.Icons.IconFactory Class Reference	307
6.157.1 Detailed Description	307
6.157.2 Member Function Documentation	308
6.157.2.1 CreateIcon()	308
6.158 Mergelt.Core.FieldElements.IConfigParameters Interface Reference	308
6.158.1 Detailed Description	309
6.158.2 Property Documentation	309
6.158.2.1 ElementConfig	309
6.158.2.2 EvolutionData	309
6.159 Mergelt.Game.Converters.IConfigProcessor Interface Reference	309
6.159.1 Detailed Description	309
6.160 Mergelt.Core.Services.IConfigsService Interface Reference	310
6.160.1 Detailed Description	310
6.160.2 Member Function Documentation	310
6.160.2.1 GetEffectConfig< T >()	310
6.160.3 Property Documentation	311
6.160.3.1 GameConfig	311
6.160.3.2 HintsConfig	311
6.160.3.3 InventoryConfig	311

6.160.3.4 LevelConfig	311
6.161 Mergelt.Core.Configs.Icons.IconParameters Class Reference	311
6.161.1 Detailed Description	311
6.161.2 Property Documentation	312
6.161.2.1 CurrencyIcon	312
6.161.2.2 CurrencyType	312
6.162 Mergelt.Core.Services.ICurrencyService Interface Reference	312
6.162.1 Detailed Description	312
6.163 Mergelt.Game.Effects.Controllers.IEffect Interface Reference	313
6.163.1 Detailed Description	313
6.164 Mergelt.Game.Effects.Parameters.IEffectParameters Interface Reference	313
6.164.1 Detailed Description	313
6.164.2 Property Documentation	314
6.164.2.1 OverriddenName	314
6.165 Mergelt.Game.Effects.IEffectsFactory Interface Reference	314
6.165.1 Detailed Description	314
6.165.2 Member Function Documentation	314
6.165.2.1 CreateEffect< T >() [1/2]	314
6.165.2.2 CreateEffect< T >() [2/2]	315
6.166 Mergelt.Game.Effects.IEffectsManager Interface Reference	315
6.166.1 Detailed Description	315
6.167 Mergelt.Game.Effects.Controllers.IEffectWithPresenter Interface Reference	315
6.167.1 Detailed Description	316
6.168 Mergelt.Game.Factories.ElementInfo.IElementInfoFactory Interface Reference	316
6.168.1 Detailed Description	316
6.169 Mergelt.Editor.EvolutionsEditor.IElementNode Interface Reference	317
6.169.1 Detailed Description	317
6.169.2 Property Documentation	317
6.169.2.1 Config	317
6.169.2.2 Data	318
6.170 Mergelt.Core.Schemes.Data.IElementNodeData Interface Reference	318
6.170.1 Detailed Description	318
6.170.2 Member Function Documentation	318
6.170.2.1 Copy< T >()	318
6.170.3 Property Documentation	319
6.170.3.1 ElementConfig	319
6.170.3.2 Id	319
6.170.3.3 Position	319
6.171 Mergelt.Core.Services.IElementService Interface Reference	319
6.171.1 Detailed Description	319
6.172 Mergelt.Core.ElementsStock.IElementsStockComponent Interface Reference	320
6.172.1 Detailed Description	320

6.173 Mergelt.Core.ElementsStock.IElementsStockData Interface Reference	320
6.173.1 Detailed Description	320
6.173.2 Property Documentation	321
6.173.2.1 Elements	321
6.174 Mergelt.Core.Services.IElementsStockService Interface Reference	321
6.174.1 Detailed Description	321
6.175 Mergelt.Core.Services.IEnergyService Interface Reference	321
6.175.1 Detailed Description	322
6.176 Mergelt.Editor.EvolutionsEditor.IEvolutionNode Interface Reference	322
6.176.1 Detailed Description	322
6.176.2 Property Documentation	323
6.176.2.1 Data	323
6.176.2.2 Description	323
6.176.2.3 Discovered	323
6.176.2.4 Name	323
6.177 Mergelt.Core.Schemes.Data.IEvolutionNodeData Interface Reference	323
6.177.1 Detailed Description	324
6.177.2 Member Function Documentation	324
6.177.2.1 Copy< T >()	324
6.177.3 Property Documentation	324
6.177.3.1 Description	324
6.177.3.2 Discovered	324
6.177.3.3 Id	324
6.177.3.4 Name	325
6.177.3.5 Position	325
6.178 Mergelt.Core.Evolutions.IEvolutionsProgressData Interface Reference	325
6.178.1 Detailed Description	325
6.178.2 Property Documentation	325
6.178.2.1 EvolutionsProgress	325
6.179 Mergelt.Core.Services.IEvolutionsService Interface Reference	326
6.179.1 Detailed Description	326
6.180 Mergelt.Game.Field.Actions.IFieldActionProcessor Interface Reference	326
6.180.1 Detailed Description	327
6.181 Mergelt.Core.FieldElements.IFieldElement Interface Reference	327
6.181.1 Detailed Description	327
6.181.2 Property Documentation	327
6.181.2.1 ConfigParameters	327
6.181.2.2 GeneratorOpenParameters	327
6.181.2.3 GeneratorParameters	328
6.181.2.4 InfoParameters	328
6.181.2.5 ProducedByParameters	328
6.181.2.6 ProduceParameters	328

6.182 Mergelt.Game.Factories.IFieldElement Interface Reference	328
6.182.1 Detailed Description	329
6.183 Mergelt.Core.FieldElements.IFieldElementModel Interface Reference	329
6.183.1 Detailed Description	329
6.183.2 Property Documentation	329
6.183.2.1 ClicksCount	329
6.183.2.2 IsBusy	329
6.183.2.3 IsLocked	330
6.183.2.4 Point	330
6.183.2.5 Selected	330
6.184 Mergelt.Core.FieldElements.IFieldElementPresenter Interface Reference	330
6.184.1 Detailed Description	331
6.184.2 Property Documentation	331
6.184.2.1 Canvas	331
6.184.2.2 IsAvailable	331
6.184.2.3 IsBusy	331
6.184.2.4 IsLocked	331
6.184.2.5 RectTransform	332
6.184.2.6 State	332
6.184.2.7 Transform	332
6.185 Mergelt.Core.FieldElements.IFieldElementView Interface Reference	332
6.185.1 Detailed Description	333
6.185.2 Property Documentation	333
6.185.2.1 Canvas	333
6.185.2.2 GraphicRaycaster	333
6.185.2.3 RectTransform	333
6.186 Mergelt.Game.Factories.IFieldElementVisualFactory Interface Reference	334
6.186.1 Detailed Description	334
6.187 Mergelt.Game.Factories.Field.IFieldFactory Interface Reference	334
6.187.1 Detailed Description	334
6.188 Mergelt.Game.Field.IFieldLogic Interface Reference	335
6.188.1 Detailed Description	335
6.189 Mergelt.Core.Services.IGameFieldActionsService Interface Reference	335
6.189.1 Detailed Description	335
6.190 Mergelt.Core.Services.IGameFieldService Interface Reference	335
6.190.1 Detailed Description	336
6.191 Mergelt.Core.Services.IGameLoadService Interface Reference	336
6.191.1 Detailed Description	336
6.191.2 Member Function Documentation	336
6.191.2.1 Load< T >()	336
6.192 Mergelt.Core.Services.IGameSaveService Interface Reference	337
6.192.1 Detailed Description	337

6.193 Mergelt.Core.Services.IGameService Interface Reference	337
6.193.1 Detailed Description	337
6.194 Mergelt.Core.FieldElements.IGeneratorOpenParameters Interface Reference	337
6.194.1 Detailed Description	338
6.194.2 Property Documentation	338
6.194.2.1 IsOpening	338
6.194.2.2 RemainingTime	338
6.194.2.3 StartOpeningTime	338
6.195 Mergelt.Core.FieldElements.IGeneratorParameters Interface Reference	339
6.195.1 Detailed Description	339
6.195.2 Property Documentation	339
6.195.2.1 AvailableToDrop	339
6.195.2.2 ChargedCount	339
6.195.2.3 Charging	340
6.195.2.4 DroppedElements	340
6.195.2.5 MinDropChargeTime	340
6.195.2.6 MinDropFullChargeTime	340
6.195.2.7 RemainChargeTime	340
6.195.2.8 StartChargingTime	340
6.196 Mergelt.Core.Services.IGeneratorsService Interface Reference	341
6.196.1 Detailed Description	341
6.197 Mergelt.Game.Managers.IHintsManager Interface Reference	341
6.197.1 Detailed Description	341
6.198 Mergelt.Game.HUD.IHudPanel Interface Reference	342
6.198.1 Detailed Description	342
6.198.2 Property Documentation	342
6.198.2.1 Type	342
6.199 Mergelt.Game.Factories(Icons).IIconFactory Interface Reference	342
6.199.1 Detailed Description	343
6.200 Mergelt.Core.Services.IInfoPanelService Interface Reference	343
6.200.1 Detailed Description	343
6.201 Mergelt.Core.FieldElements.IInfoParameters Interface Reference	343
6.201.1 Detailed Description	344
6.201.2 Property Documentation	344
6.201.2.1 Description	344
6.201.2.2 IsBlocked	344
6.201.2.3 LogicPosition	344
6.201.2.4 Name	344
6.201.2.5 Type	344
6.202 Mergelt.SimpleDI.ReservedInterfaces.IInitializable Interface Reference	345
6.202.1 Detailed Description	345
6.203 Mergelt.Core.Inventory.IInventoryData Interface Reference	345

6.203.1 Detailed Description	346
6.203.2 Property Documentation	346
6.203.2.1 InventoryElements	346
6.203.2.2 InventorySize	346
6.204 Mergelt.Game.Factories.Inventory.IInventoryFactory Interface Reference	346
6.204.1 Detailed Description	347
6.205 Mergelt.Core.Services.IInventoryService Interface Reference	347
6.205.1 Detailed Description	347
6.206 Mergelt.Core.Messages.IMessage Interface Reference	348
6.206.1 Detailed Description	348
6.207 Mergelt.Core.Messages.IMessageBus Interface Reference	348
6.207.1 Detailed Description	349
6.207.2 Member Function Documentation	349
6.207.2.1 AddListener< T >()	349
6.207.2.2 DefineMessage< T >()	349
6.207.2.3 Fire< T >() [1/2]	349
6.207.2.4 Fire< T >() [2/2]	349
6.207.2.5 RemoveListener< T >()	350
6.208 Mergelt.Core.MVP.IModel Interface Reference	350
6.208.1 Detailed Description	350
6.209 Mergelt.Core.Helpers.IMonoApplicationQuitHandler Interface Reference	350
6.209.1 Detailed Description	351
6.210 Mergelt.Core.Helpers.IMonoUpdateHandler Interface Reference	351
6.210.1 Detailed Description	351
6.211 Mergelt.Game.Services.InfoPanelService Class Reference	351
6.211.1 Detailed Description	351
6.211.2 Member Function Documentation	352
6.211.2.1 Dispose()	352
6.211.2.2 Initialize()	352
6.212 Mergelt.Core.FieldElements.InfoParameters Class Reference	352
6.212.1 Detailed Description	352
6.212.2 Property Documentation	353
6.212.2.1 Description	353
6.212.2.2 IsBlocked	353
6.212.2.3 LogicPosition	353
6.212.2.4 Name	353
6.212.2.5 Type	353
6.213 Mergelt.SimpleDI.IntroduceAttribute Class Reference	354
6.213.1 Detailed Description	354
6.213.2 Constructor & Destructor Documentation	354
6.213.2.1 IntroduceAttribute()	354
6.214 Mergelt.Core.Configs.Inventory.InventoryConfig Class Reference	354

6.214.1 Detailed Description	355
6.214.2 Member Data Documentation	355
6.214.2.1 InitialCapacity	355
6.214.2.2 PaidCells	355
6.215 Mergelt.Game.Factories.Inventory.InventoryData Class Reference	355
6.215.1 Detailed Description	355
6.215.2 Property Documentation	356
6.215.2.1 InventoryElements	356
6.215.2.2 InventorySize	356
6.216 Mergelt.Game.Factories.Inventory.InventoryFactory Class Reference	356
6.216.1 Detailed Description	356
6.216.2 Member Function Documentation	357
6.216.2.1 CreateInventoryPanellItem()	357
6.216.2.2 CreateInventoryWindowItem()	357
6.216.2.3 CreateWindowEmptyCell()	358
6.216.2.4 CreateWindowPaidCell()	358
6.217 Mergelt.Game.Windows.Inventory.InventoryModel Class Reference	358
6.217.1 Detailed Description	359
6.218 Mergelt.Game.UI.InventoryPanel.InventoryPanelComponent Class Reference	359
6.218.1 Detailed Description	359
6.219 Mergelt.Game.UI.InventoryPanel.InventoryPanelItemComponent Class Reference	359
6.219.1 Detailed Description	360
6.219.2 Property Documentation	360
6.219.2.1 RectTransform	360
6.219.3 Event Documentation	360
6.219.3.1 ClickEvent	360
6.220 Mergelt.Game.UI.InventoryPanel.InventoryPanelItemPair Class Reference	360
6.220.1 Detailed Description	360
6.220.2 Constructor & Destructor Documentation	361
6.220.2.1 InventoryPanellItemPair()	361
6.220.3 Property Documentation	361
6.220.3.1 Component	361
6.220.3.2 Element	361
6.221 Mergelt.Game.Windows.Inventory.InventoryPresenter Class Reference	361
6.221.1 Detailed Description	362
6.221.2 Member Function Documentation	363
6.221.2.1 OnDestroyWindow()	363
6.221.2.2 OnInitialize()	363
6.222 Mergelt.Game.Services.InventoryService Class Reference	363
6.222.1 Detailed Description	364
6.222.2 Member Function Documentation	364
6.222.2.1 Add()	364

6.222.2.2 CreateInventory()	364
6.222.2.3 GetData()	365
6.222.2.4 OpenWindow()	365
6.222.2.5 Remove()	365
6.222.2.6 SetupInventory()	366
6.223 Mergelt.Game.Services.InventoryServiceModel Class Reference	366
6.223.1 Detailed Description	366
6.223.2 Member Function Documentation	366
6.223.2.1 IsFull()	366
6.223.3 Property Documentation	367
6.223.3.1 InventoryElements	367
6.223.3.2 InventorySize	367
6.224 Mergelt.Game.Windows.Inventory.InventoryWindow Class Reference	367
6.224.1 Detailed Description	369
6.224.2 Property Documentation	369
6.224.2.1 ItemsGrid	369
6.225 Mergelt.Game.Windows.Inventory.InventoryWidgetItemComponent Class Reference	369
6.225.1 Detailed Description	370
6.225.2 Member Function Documentation	370
6.225.2.1 Awake()	370
6.225.2.2 OnDestroy()	370
6.225.2.3 SetIcon()	370
6.225.3 Event Documentation	370
6.225.3.1 InfoClickEvent	370
6.225.3.2 ItemClickEvent	370
6.226 Mergelt.Game.Windows.Inventory.InventoryWidgetItemPair Class Reference	371
6.226.1 Detailed Description	371
6.226.2 Constructor & Destructor Documentation	371
6.226.2.1 InventoryWidgetItemPair()	371
6.226.3 Property Documentation	371
6.226.3.1 Component	371
6.226.3.2 Element	371
6.227 Mergelt.Game.Windows.Inventory.InventoryWindowPaidCellComponent Class Reference	372
6.227.1 Detailed Description	372
6.227.2 Member Function Documentation	372
6.227.2.1 Awake()	372
6.227.2.2 OnDestroy()	372
6.227.2.3 Setup()	373
6.227.3 Event Documentation	373
6.227.3.1 BuyCellEvent	373
6.228 Mergelt.Core.MVP.IPresenter Interface Reference	373
6.228.1 Detailed Description	373

6.229 Mergelt.Core.FieldElements.IProducedByParameters Interface Reference	374
6.229.1 Detailed Description	374
6.229.2 Property Documentation	374
6.229.2.1 Elements	374
6.230 Mergelt.Core.FieldElements.IProduceParameters Interface Reference	374
6.230.1 Detailed Description	375
6.230.2 Property Documentation	375
6.230.2.1 Elements	375
6.231 Mergelt.Core.Services.IResourcesLoaderService Interface Reference	375
6.231.1 Detailed Description	375
6.231.2 Member Function Documentation	375
6.231.2.1 GetObject< T >()	375
6.232 Mergelt.Core.Saves.ISavable Interface Reference	376
6.232.1 Detailed Description	376
6.233 Mergelt.Editor.EvolutionsEditor.ISaveNode Interface Reference	376
6.233.1 Detailed Description	376
6.233.2 Property Documentation	376
6.233.2.1 Id	376
6.234 Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy Interface Reference	377
6.234.1 Detailed Description	377
6.234.2 Member Function Documentation	377
6.234.2.1 Load< T >()	377
6.234.2.2 Save< T >()	377
6.234.3 Property Documentation	378
6.234.3.1 SaveDir	378
6.235 Mergelt.SimpleDI.ReservedInterfaces.IUpdatable Interface Reference	378
6.235.1 Detailed Description	378
6.236 Mergelt.Core.User.IUserData Interface Reference	378
6.236.1 Detailed Description	379
6.236.2 Property Documentation	379
6.236.2.1 Energy	379
6.236.2.2 EnergyRestoringStartTime	379
6.236.2.3 Experience	379
6.236.2.4 HardCurrency	379
6.236.2.5 Level	380
6.236.2.6 Name	380
6.236.2.7 SoftCurrency	380
6.236.2.8 Splitters	380
6.237 Mergelt.Game.HUD.IUserListener Interface Reference	380
6.237.1 Detailed Description	381
6.238 Mergelt.Core.Services.IUserProgressService Interface Reference	381
6.238.1 Detailed Description	381

6.239 Mergelt.Core.Services.IUserService Interface Reference	381
6.239.1 Detailed Description	382
6.240 Mergelt.Core.MVP.IView Interface Reference	382
6.240.1 Detailed Description	382
6.240.2 Property Documentation	382
6.240.2.1 GameObject	382
6.240.3 Event Documentation	383
6.240.3.1 DestroyEvent	383
6.240.3.2 InitializeEvent	383
6.241 Mergelt.Core.WindowSystem.Windows.IWindow Interface Reference	383
6.241.1 Detailed Description	384
6.241.2 Event Documentation	384
6.241.2.1 CloseEndEvent	384
6.241.2.2 CloseStartEvent	384
6.241.2.3 InitiateCloseEvent	384
6.241.2.4 ShowEndEvent	384
6.241.2.5 ShowStartEvent	385
6.242 Mergelt.Core.Animations.IWindowAnimationController Interface Reference	385
6.242.1 Detailed Description	385
6.243 Mergelt.Core.Animations.IWindowAnimationListener Interface Reference	385
6.243.1 Detailed Description	386
6.244 Mergelt.Core.WindowSystem.Data.IWindowArgs Interface Reference	386
6.244.1 Detailed Description	386
6.245 Mergelt.Core.WindowSystem.Factory.IWindowFactory Interface Reference	386
6.245.1 Detailed Description	387
6.245.2 Member Function Documentation	387
6.245.2.1 CreateWindow< TPresenter >()	387
6.246 Mergelt.Core.WindowSystem.Data.IWindowOpenParameters Interface Reference	387
6.246.1 Detailed Description	388
6.246.2 Property Documentation	388
6.246.2.1 ClosePrevious	388
6.246.2.2 NeedBlackout	388
6.246.2.3 Presenter	388
6.247 Mergelt.Core.WindowSystem.Windows.IWindowPresenter Interface Reference	388
6.247.1 Detailed Description	389
6.247.2 Property Documentation	389
6.247.2.1 Layer	389
6.247.2.2 State	389
6.248 Mergelt.Core.WindowSystem.IWindowSystem Interface Reference	390
6.248.1 Detailed Description	390
6.248.2 Member Function Documentation	390
6.248.2.1 OpenWindow< TPresenter >()	390

6.249 Mergelt.Game.Services.Saves.Strategies.JsonSerializeStrategy Class Reference	390
6.249.1 Detailed Description	391
6.249.2 Member Function Documentation	391
6.249.2.1 Load< T >()	391
6.249.2.2 Save< T >()	392
6.250 Mergelt.Game.Effects.Controllers.JumpEffect Class Reference	392
6.250.1 Detailed Description	393
6.250.2 Constructor & Destructor Documentation	393
6.250.2.1 JumpEffect()	393
6.250.3 Member Function Documentation	393
6.250.3.1 Setup()	393
6.250.3.2 Update()	394
6.251 Mergelt.Game.Effects.Configs.JumpEffectConfig Class Reference	394
6.251.1 Detailed Description	395
6.251.2 Member Data Documentation	395
6.251.2.1 CurveY	395
6.251.2.2 ScaleCurve	395
6.252 Mergelt.Game.Effects.Controllers.JumpEffectWithPresenter Class Reference	395
6.252.1 Detailed Description	396
6.252.2 Member Function Documentation	397
6.252.2.1 OnStarted()	397
6.252.2.2 Setup()	397
6.252.2.3 Start()	397
6.252.2.4 Update()	397
6.253 Mergelt.Core.Configs.Windows.LayerParameters Class Reference	398
6.253.1 Detailed Description	398
6.253.2 Member Function Documentation	398
6.253.2.1 CompareTo()	398
6.253.2.2 Equals() [1/2]	398
6.253.2.3 Equals() [2/2]	399
6.253.2.4 GetHashCode()	399
6.253.3 Property Documentation	399
6.253.3.1 Name	399
6.253.3.2 Order	399
6.254 Mergelt.Core.Configs.Windows.LayersConfig Class Reference	399
6.254.1 Detailed Description	400
6.254.2 Property Documentation	400
6.254.2.1 Layers	400
6.255 Mergelt.Core.Configs.LevelConfig Class Reference	400
6.255.1 Detailed Description	400
6.255.2 Member Data Documentation	401
6.255.2.1 EvolutionsScheme	401

6.255.2.2 FieldElementsData	401
6.255.2.3 FieldHeight	401
6.255.2.4 FieldWidth	401
6.256 Mergelt.Editor.LevelEditor.LevelConfigEditor Class Reference	401
6.256.1 Detailed Description	402
6.256.2 Member Function Documentation	402
6.256.2.1 OnInspectorGUI()	402
6.257 Mergelt.Editor.LevelEditor.LevelEditorWindow Class Reference	402
6.257.1 Detailed Description	402
6.257.2 Member Function Documentation	403
6.257.2.1 ApplyCell()	403
6.257.2.2 ClearCell()	403
6.257.2.3 CreateGUI()	403
6.257.2.4 Show()	405
6.257.2.5 ShowWindow()	405
6.257.2.6 UndoApplyCell()	405
6.257.2.7 UndoClearCell()	406
6.258 Mergelt.Core.Configs.Data.LevelElementData Class Reference	406
6.258.1 Detailed Description	406
6.258.2 Member Function Documentation	406
6.258.2.1 CopyFrom()	406
6.258.2.2 GetClone()	407
6.258.3 Member Data Documentation	407
6.258.3.1 Element	407
6.258.3.2 EvolutionId	407
6.258.3.3 IsBlocked	407
6.258.3.4 Position	407
6.259 Mergelt.Core.Configs.LevelUp.LevelUpConfig Class Reference	407
6.259.1 Detailed Description	408
6.259.2 Member Data Documentation	408
6.259.2.1 LevelUp	408
6.260 Mergelt.Game.Messages.LevelUpdatedMessage Class Reference	408
6.260.1 Detailed Description	408
6.260.2 Property Documentation	408
6.260.2.1 NextLevelExp	408
6.261 Mergelt.Game.Messages.LevelUpMessage Class Reference	409
6.261.1 Detailed Description	409
6.262 Mergelt.Core.Configs.LevelUp.LevelUpParameters Class Reference	409
6.262.1 Detailed Description	409
6.262.2 Member Data Documentation	409
6.262.2.1 Bonuses	409
6.262.2.2 Experience	409

6.263 Mergelt.Editor.Core.LevelEditor.Commands.LimitedStack< T > Class Template Reference	410
6.263.1 Detailed Description	410
6.263.2 Constructor & Destructor Documentation	410
6.263.2.1 LimitedStack()	410
6.263.3 Member Function Documentation	410
6.263.3.1 Any()	410
6.263.3.2 Clear()	410
6.263.3.3 Peek()	411
6.263.3.4 Pop()	411
6.263.3.5 Push()	411
6.263.4 Property Documentation	411
6.263.4.1 Count	411
6.264 Mergelt.Game.Commands.LoadConfigsCommand Class Reference	412
6.264.1 Detailed Description	412
6.264.2 Member Function Documentation	413
6.264.2.1 ExecuteAsync()	413
6.265 Mergelt.Game.Messages.LoadedGameMessage Class Reference	413
6.265.1 Detailed Description	413
6.266 Mergelt.Core.Helpers.MainThreadDispatcher Class Reference	413
6.266.1 Detailed Description	414
6.266.2 Member Function Documentation	414
6.266.2.1 Enqueue()	414
6.266.2.2 IsMainThread()	414
6.266.2.3 RunOnMainThread()	414
6.266.3 Property Documentation	415
6.266.3.1 Instance	415
6.267 Mergelt.Core.Animations.MecanimController Class Reference	415
6.267.1 Detailed Description	415
6.267.2 Member Function Documentation	416
6.267.2.1 Initialize()	416
6.267.2.2 SetState() [1/2]	416
6.267.2.3 SetState() [2/2]	416
6.267.2.4 SetState< T >()	416
6.268 Mergelt.Game.Messages.MergeElementsMessage Class Reference	417
6.268.1 Detailed Description	417
6.268.2 Property Documentation	417
6.268.2.1 From	417
6.268.2.2 NewElement	417
6.269 Mergelt.Core.Messages.MessageBus Class Reference	417
6.269.1 Detailed Description	418
6.269.2 Member Function Documentation	418
6.269.2.1 AddListener< T >()	418

6.269.2.2 DefineMessage< T >()	418
6.269.2.3 Fire< T >() [1/2]	419
6.269.2.4 Fire< T >() [2/2]	419
6.269.2.5 Initialize()	420
6.269.2.6 RemoveListener< T >()	420
6.270 Mergelt.Core.MVP.Model Class Reference	420
6.270.1 Detailed Description	421
6.271 Mergelt.Core.Helpers.MonoEventsListener Class Reference	421
6.271.1 Detailed Description	421
6.271.2 Member Function Documentation	421
6.271.2.1 SubscribeOnApplicationQuit()	421
6.271.2.2 SubscribeOnUpdate()	422
6.271.2.3 UnsubscribeFromApplicationQuit()	422
6.271.2.4 UnsubscribeFromUpdate()	422
6.271.3 Property Documentation	422
6.271.3.1 Instance	422
6.272 Mergelt.Game.Effects.Controllers.MoveEffect Class Reference	423
6.272.1 Detailed Description	423
6.272.2 Constructor & Destructor Documentation	424
6.272.2.1 MoveEffect()	424
6.272.3 Member Function Documentation	424
6.272.3.1 Setup() [1/2]	424
6.272.3.2 Setup() [2/2]	424
6.272.3.3 Update()	425
6.273 Mergelt.Game.Effects.Configs.MoveEffectConfig Class Reference	425
6.273.1 Detailed Description	426
6.273.2 Member Data Documentation	426
6.273.2.1 PositionXCurve	426
6.273.2.2 PositionYCurve	426
6.273.2.3 ScaleCurve	426
6.274 Mergelt.Game.Effects.Parameters.MoveEffectParameters Class Reference	426
6.274.1 Detailed Description	427
6.274.2 Constructor & Destructor Documentation	427
6.274.2.1 MoveEffectParameters()	427
6.274.3 Property Documentation	427
6.274.3.1 FromPosition	427
6.274.3.2 OverriddenName	427
6.274.3.3 TargetInHud	428
6.274.3.4 ToPosition	428
6.275 Mergelt.Game.Effects.Controllers.MoveEffectWithPresenter Class Reference	428
6.275.1 Detailed Description	429
6.275.2 Member Function Documentation	430

6.275.2.1 OnFinished()	430
6.275.2.2 OnStarted()	430
6.275.2.3 Setup()	430
6.275.2.4 Start()	430
6.275.2.5 Update()	431
6.276 Mergelt.Game.Messages.MoveElementMessage Class Reference	431
6.276.1 Detailed Description	431
6.276.2 Property Documentation	431
6.276.2.1 FieldElementView	431
6.276.2.2 FromPoint	432
6.276.2.3 ToPoint	432
6.277 Mergelt.Game.Effects.Controllers.MoveToUiEffect Class Reference	432
6.277.1 Detailed Description	433
6.277.2 Member Function Documentation	433
6.277.2.1 Setup()	433
6.277.2.2 Update()	434
6.278 Mergelt.Game.Effects.Configs.MoveToUiEffectConfig Class Reference	434
6.278.1 Detailed Description	435
6.278.2 Member Data Documentation	435
6.278.2.1 PositionXCurve	435
6.278.2.2 PositionYCurve	435
6.278.2.3 ScaleCurve	435
6.279 Mergelt.Game.Effects.Controllers.MoveToUiEffectWithPresenter Class Reference	435
6.279.1 Detailed Description	436
6.279.2 Member Function Documentation	437
6.279.2.1 OnStarted()	437
6.279.2.2 Setup()	437
6.279.2.3 Start()	437
6.279.2.4 Update()	437
6.280 Mergelt.Game.Commands.PrepareEnergyCommand Class Reference	438
6.280.1 Detailed Description	438
6.280.2 Member Function Documentation	439
6.280.2.1 Execute()	439
6.281 Mergelt.Game.Commands.PrepareFieldCommand Class Reference	439
6.281.1 Detailed Description	440
6.281.2 Member Function Documentation	440
6.281.2.1 Execute()	440
6.282 Mergelt.Game.Messages.PrepareGameMessage Class Reference	441
6.282.1 Detailed Description	441
6.283 Mergelt.Game.Commands.PrepareInventoryCommand Class Reference	441
6.283.1 Detailed Description	442
6.283.2 Member Function Documentation	442

6.283.2.1 Execute()	442
6.284 Mergelt.Game.Commands.PrepareStockCommand Class Reference	442
6.284.1 Detailed Description	443
6.284.2 Member Function Documentation	443
6.284.2.1 Execute()	443
6.285 Mergelt.Game.Commands.PrepareUserCommand Class Reference	444
6.285.1 Detailed Description	444
6.285.2 Member Function Documentation	445
6.285.2.1 Execute()	445
6.286 Mergelt.Core.MVP.Presenter< TView, TModel > Class Template Reference	445
6.286.1 Detailed Description	446
6.286.2 Member Function Documentation	446
6.286.2.1 Dispose()	446
6.286.2.2 Initialize()	446
6.286.2.3 OnDispose()	446
6.286.2.4 OnInitialize()	447
6.286.3 Property Documentation	447
6.286.3.1 Model	447
6.286.3.2 View	447
6.287 Mergelt.Core.FieldElements.ProducedByParameters Class Reference	447
6.287.1 Detailed Description	448
6.287.2 Property Documentation	448
6.287.2.1 Elements	448
6.288 Mergelt.Core.FieldElements.ProduceParameters Class Reference	448
6.288.1 Detailed Description	448
6.288.2 Property Documentation	449
6.288.2.1 Elements	449
6.289 Mergelt.Game.HUD.ProgressComponent Class Reference	449
6.289.1 Detailed Description	450
6.289.2 Member Function Documentation	450
6.289.2.1 Initialize()	450
6.289.2.2 OnApplyModel()	450
6.289.2.3 OnDestroy()	450
6.289.2.4 SetMaxProgress()	451
6.289.2.5 Start()	451
6.289.2.6 UpdateProgress()	451
6.289.3 Property Documentation	451
6.289.3.1 Type	451
6.290 Mergelt.Game.Messages.RemoveElementMessage Class Reference	452
6.290.1 Detailed Description	452
6.290.2 Property Documentation	452
6.290.2.1 RemoveAtPoint	452

6.291 Mergelt.Game.Messages.RemoveFromInventoryMessage Class Reference	452
6.291.1 Detailed Description	453
6.291.2 Property Documentation	453
6.291.2.1 FieldElement	453
6.292 Mergelt.Game.Messages.ResetHintsMessage Class Reference	453
6.292.1 Detailed Description	453
6.293 Mergelt.Game.Messages.ResetPositionMessage Class Reference	453
6.293.1 Detailed Description	454
6.293.2 Property Documentation	454
6.293.2.1 From	454
6.294 Mergelt.Game.Messages.ResetSelectionMessage Class Reference	454
6.294.1 Detailed Description	454
6.295 Mergelt.Game.Services.ResourcesLoaderService Class Reference	454
6.295.1 Detailed Description	455
6.295.2 Member Function Documentation	455
6.295.2.1 GetObject< TResource >()	455
6.296 Mergelt.Core.Utils.ResponsiveGridLayout Class Reference	455
6.296.1 Detailed Description	456
6.297 Mergelt.Core.Saves.SavableAttribute Class Reference	456
6.297.1 Detailed Description	456
6.297.2 Constructor & Destructor Documentation	456
6.297.2.1 SavableAttribute()	456
6.297.3 Property Documentation	457
6.297.3.1 Extension	457
6.297.3.2 Name	457
6.298 Mergelt.Core.FieldElements.SavedConfigParameters Class Reference	457
6.298.1 Detailed Description	457
6.298.2 Property Documentation	457
6.298.2.1 ElementId	457
6.298.2.2 EvolutionId	457
6.299 Mergelt.Core.FieldElements.SavedGeneratorOpenParameters Class Reference	458
6.299.1 Detailed Description	458
6.299.2 Member Function Documentation	458
6.299.2.1 CopyFrom()	458
6.299.3 Property Documentation	458
6.299.3.1 StartOpeningTime	458
6.300 Mergelt.Core.FieldElements.SavedGeneratorParameters Class Reference	458
6.300.1 Detailed Description	459
6.300.2 Member Function Documentation	459
6.300.2.1 CopyFrom()	459
6.300.3 Property Documentation	459
6.300.3.1 AvailableToDelete	459

6.300.3.2 ChargedCount	459
6.300.3.3 DroppedElements	459
6.300.3.4 StartChargingTime	460
6.301 Mergelt.Core.FieldElements.SavedInfoParameters Class Reference	460
6.301.1 Detailed Description	460
6.301.2 Property Documentation	460
6.301.2.1 IsBlocked	460
6.301.2.2 LogicPosition	460
6.302 Mergelt.Editor.Helpers.SavesHelper Class Reference	461
6.302.1 Detailed Description	461
6.302.2 Member Function Documentation	461
6.302.2.1 ClearSaves()	461
6.303 Mergelt.Core.Schemes.SchemeData Class Reference	461
6.303.1 Detailed Description	462
6.303.2 Member Data Documentation	462
6.303.2.1 Edges	462
6.303.2.2 ElementsNodesData	462
6.303.2.3 EvolutionsData	462
6.303.2.4 EvolutionsNodesData	462
6.304 Mergelt.Core.Schemes.SchemeObject Class Reference	462
6.304.1 Detailed Description	463
6.304.2 Member Data Documentation	463
6.304.2.1 SchemeData	463
6.304.3 Property Documentation	463
6.304.3.1 Evolution	463
6.305 Mergelt.Editor.EvolutionsEditor.SchemeObjectEditor Class Reference	463
6.305.1 Detailed Description	464
6.305.2 Member Function Documentation	464
6.305.2.1 OnInspectorGUI()	464
6.306 Mergelt.Game.Messages.SelectedElementMessage Class Reference	464
6.306.1 Detailed Description	464
6.306.2 Property Documentation	465
6.306.2.1 Point	465
6.307 Mergelt.Editor.EvolutionsEditor.SerializedElementNodes Class Reference	465
6.307.1 Detailed Description	465
6.307.2 Member Data Documentation	465
6.307.2.1 EvolutionNodesData	465
6.308 Mergelt.Game.Messages.SkipTimeMessage Class Reference	465
6.308.1 Detailed Description	466
6.308.2 Property Documentation	466
6.308.2.1 Seconds	466
6.309 Mergelt.Core.Helpers.SlicedFilledImage Class Reference	466

6.309.1 Detailed Description	467
6.309.2 Member Enumeration Documentation	467
6.309.2.1 FillDirection	467
6.309.3 Constructor & Destructor Documentation	467
6.309.3.1 SlicedFilledImage()	467
6.309.4 Member Function Documentation	467
6.309.4.1 OnDisable()	467
6.309.4.2 OnEnable()	468
6.309.4.3 OnPopulateMesh()	468
6.309.4.4 UpdateMaterial()	468
6.309.5 Property Documentation	468
6.309.5.1 alphaHitTestMinimumThreshold	468
6.309.5.2 fillAmount	469
6.309.5.3 fillCenter	469
6.309.5.4 fillDirection	469
6.309.5.5 hasBorder	469
6.309.5.6 mainTexture	470
6.309.5.7 material	470
6.309.5.8 overrideSprite	470
6.309.5.9 pixelsPerUnit	470
6.309.5.10 pixelsPerUnitMultiplier	471
6.309.5.11 sprite	471
6.310 Mergelt.Editor.Helpers.SlicedFilledImageEditor Class Reference	471
6.310.1 Detailed Description	471
6.310.2 Member Function Documentation	472
6.310.2.1 OnInspectorGUI()	472
6.311 Mergelt.Game.HUD.SoftCurrencyComponent Class Reference	472
6.311.1 Detailed Description	473
6.311.2 Member Function Documentation	473
6.311.2.1 OnApplyModel()	473
6.311.3 Property Documentation	473
6.311.3.1 Type	473
6.312 Mergelt.Game.Messages.SplitElementMessage Class Reference	474
6.312.1 Detailed Description	474
6.312.2 Property Documentation	474
6.312.2.1 SplitElement1	474
6.312.2.2 SplitElement2	474
6.313 Mergelt.Game.Messages.StartChargingMessage Class Reference	474
6.313.1 Detailed Description	475
6.313.2 Property Documentation	475
6.313.2.1 RemainingTime	475
6.314 Mergelt.Game.Messages.StartGameMessage Class Reference	475

6.314.1 Detailed Description	475
6.315 Mergelt.Game.Messages.SwapElementsMessage Class Reference	475
6.315.1 Detailed Description	476
6.315.2 Property Documentation	476
6.315.2.1 From	476
6.315.2.2 To	476
6.316 Mergelt.Game.TopPanel.TopPanelComponent Class Reference	476
6.316.1 Detailed Description	476
6.317 Mergelt.Game.Messages.UnlockElementMessage Class Reference	477
6.317.1 Detailed Description	477
6.317.2 Property Documentation	477
6.317.2.1 Element	477
6.318 Mergelt.Game.Messages.UpdateStockMessage Class Reference	477
6.318.1 Detailed Description	477
6.319 Mergelt.Game.User(userData Class Reference	478
6.319.1 Detailed Description	478
6.319.2 Property Documentation	478
6.319.2.1 Energy	478
6.319.2.2 EnergyRestoringStartTime	478
6.319.2.3 Experience	479
6.319.2.4 HardCurrency	479
6.319.2.5 Level	479
6.319.2.6 Name	479
6.319.2.7 SoftCurrency	479
6.319.2.8 Splitters	480
6.320 Mergelt.Game.Windows.UserProgress.userInfo Model Class Reference	480
6.320.1 Detailed Description	480
6.321 Mergelt.Game.Windows.UserProgress.userInfoPresenter Class Reference	480
6.321.1 Detailed Description	481
6.321.2 Member Function Documentation	482
6.321.2.1 OnDestroyWindow()	482
6.321.2.2 OnInitialize()	482
6.322 Mergelt.Game.Windows.UserProgress.userInfoWindow Class Reference	482
6.322.1 Detailed Description	484
6.322.2 Member Function Documentation	484
6.322.2.1 Close()	484
6.322.2.2 EnableLevelUpButton()	485
6.322.2.3 OnOpenFinished()	485
6.322.2.4 UpdateProgress()	485
6.322.3 Member Data Documentation	485
6.322.3.1 LevelUpEvent	485
6.322.4 Property Documentation	485

6.322.4.1 PrizesContainer	485
6.323 Mergelt.Game.HUD.UserListenerComponent Class Reference	486
6.323.1 Detailed Description	486
6.323.2 Member Function Documentation	486
6.323.2.1 ApplyModel()	486
6.323.3 Property Documentation	486
6.323.3.1 UserServiceModel	486
6.324 Mergelt.Game.Windows.UserProgress.UserProgressComponent Class Reference	487
6.324.1 Detailed Description	487
6.324.2 Member Function Documentation	487
6.324.2.1 UpdateProgress()	487
6.325 Mergelt.Game.Services.UserProgressService Class Reference	487
6.325.1 Detailed Description	488
6.325.2 Member Function Documentation	488
6.325.2.1 CanLevelUp()	488
6.325.2.2 Dispose()	488
6.325.2.3 GetCurrentLevelMaxExp()	488
6.325.2.4 GetLevelUpPrizes()	489
6.325.2.5 Initialize()	489
6.326 Mergelt.Game.Services.UserService Class Reference	489
6.326.1 Detailed Description	489
6.326.2 Member Function Documentation	490
6.326.2.1 CreateUser()	490
6.326.2.2 SetupUser()	490
6.327 Mergelt.Game.Services.UserServiceModel Class Reference	490
6.327.1 Detailed Description	491
6.327.2 Member Function Documentation	491
6.327.2.1 GetData()	491
6.327.2.2 Set()	491
6.327.3 Property Documentation	491
6.327.3.1 Energy	491
6.327.3.2 EnergyRestoringStartTime	491
6.327.3.3 Experience	492
6.327.3.4 HardCurrency	492
6.327.3.5 Level	492
6.327.3.6 Name	492
6.327.3.7 SoftCurrency	492
6.327.3.8 Splitters	492
6.328 Mergelt.Editor.Configs.CurrencyComponent.UxmlFactory Class Reference	493
6.328.1 Detailed Description	493
6.329 Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView.UxmlFactory Class Reference	493
6.329.1 Detailed Description	493

6.330 Mergelt.Core.MVP.View Class Reference	493
6.330.1 Detailed Description	494
6.330.2 Member Function Documentation	494
6.330.2.1 Initialize()	494
6.330.2.2 OnDestroy()	494
6.330.3 Property Documentation	494
6.330.3.1 GameObject	494
6.330.4 Event Documentation	495
6.330.4.1 DestroyEvent	495
6.330.4.2 InitializeEvent	495
6.331 Mergelt.Core.WindowSystem.Data.WindowArgs Class Reference	495
6.331.1 Detailed Description	495
6.332 Mergelt.Core.WindowSystem.Windows.WindowBase Class Reference	495
6.332.1 Detailed Description	497
6.332.2 Member Function Documentation	497
6.332.2.1 Close()	497
6.332.2.2 OnCloseFinished()	497
6.332.2.3 OnCloseStarted()	497
6.332.2.4 OnDestroy()	498
6.332.2.5 OnOpenFinished()	498
6.332.2.6 OnOpenStarted()	498
6.332.2.7 SetLayer()	498
6.332.2.8 Show()	499
6.332.3 Member Data Documentation	499
6.332.3.1 ClickOutsideArea	499
6.332.3.2 CloseButton	499
6.332.4 Property Documentation	499
6.332.4.1 AnimationController	499
6.332.4.2 Canvas	499
6.332.5 Event Documentation	500
6.332.5.1 CloseEndEvent	500
6.332.5.2 CloseStartEvent	500
6.332.5.3 InitiateCloseEvent	500
6.332.5.4 ShowEndEvent	500
6.332.5.5 ShowStartEvent	500
6.333 Mergelt.Core.WindowSystem.Commands.WindowCloseCommand Class Reference	500
6.333.1 Detailed Description	501
6.333.2 Constructor & Destructor Documentation	501
6.333.2.1 WindowCloseCommand()	501
6.333.3 Member Function Documentation	502
6.333.3.1 OnDispose()	502
6.333.3.2 Update()	502

6.334 Mergelt.Core.WindowSystem.Windows.WindowCreateInfo Struct Reference	502
6.334.1 Detailed Description	503
6.334.2 Member Function Documentation	503
6.334.2.1 Equals() [1/2]	503
6.334.2.2 Equals() [2/2]	503
6.334.2.3 GetHashCode()	503
6.334.3 Property Documentation	503
6.334.3.1 LayerName	503
6.334.3.2 PrefabPath	504
6.334.3.3 Type	504
6.335 Mergelt.Core.Animations.WindowDefaultAnimationController Class Reference	504
6.335.1 Detailed Description	504
6.335.2 Member Function Documentation	504
6.335.2.1 CloseWindow()	504
6.335.2.2 Initialize()	505
6.335.2.3 OnCloseEnd()	505
6.335.2.4 OnOpenEnd()	505
6.335.2.5 OpenWindow()	505
6.336 Mergelt.Core.WindowSystem.Factory.WindowFactory Class Reference	506
6.336.1 Detailed Description	506
6.336.2 Member Function Documentation	506
6.336.2.1 CreateWindow< TPresenter >()	506
6.336.2.2 GetBlackout()	507
6.336.2.3 GetRoot()	507
6.337 Mergelt.Core.WindowSystem.Commands.WindowHideCommand Class Reference	507
6.337.1 Detailed Description	508
6.337.2 Constructor & Destructor Documentation	508
6.337.2.1 WindowHideCommand()	508
6.337.3 Member Function Documentation	509
6.337.3.1 OnDispose()	509
6.337.3.2 Update()	509
6.338 Mergelt.Core.Animations.WindowMecanimAnimationController Class Reference	509
6.338.1 Detailed Description	510
6.338.2 Member Function Documentation	510
6.338.2.1 CloseWindow()	510
6.338.2.2 Initialize()	510
6.338.2.3 OnCloseEnd()	510
6.338.2.4 OnOpenEnd()	510
6.338.2.5 OpenWindow()	511
6.339 Mergelt.Core.WindowSystem.Windows.WindowModel Class Reference	511
6.339.1 Detailed Description	511
6.340 Mergelt.Core.WindowSystem.Commands.WindowOpenCommand Class Reference	511

6.340.1 Detailed Description	512
6.340.2 Constructor & Destructor Documentation	512
6.340.2.1 WindowOpenCommand()	512
6.340.3 Member Function Documentation	513
6.340.3.1 OnDispose()	513
6.340.3.2 Update()	513
6.341 Mergelt.Core.WindowSystem.Data.WindowOpenParameters Class Reference	513
6.341.1 Detailed Description	514
6.341.2 Property Documentation	514
6.341.2.1 ClosePrevious	514
6.341.2.2 NeedBlackout	514
6.341.2.3 Presenter	514
6.342 Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel > Class Template Reference	514
6.342.1 Detailed Description	516
6.342.2 Member Function Documentation	516
6.342.2.1 Close()	516
6.342.2.2 DestroyWindow()	516
6.342.2.3 Hide()	516
6.342.2.4 Initialize()	517
6.342.2.5 OnDestroyWindow()	517
6.342.2.6 OnDispose()	517
6.342.2.7 OnInitialize()	517
6.342.2.8 OnWindowClosed()	518
6.342.2.9 OnWindowShown()	518
6.342.2.10 OnWindowStartClosing()	518
6.342.2.11 OnWindowStartShowing()	518
6.342.2.12 SetWindowActive()	519
6.342.2.13 SetWindowLayer()	519
6.342.2.14 Show()	519
6.342.3 Member Data Documentation	519
6.342.3.1 MessageBus	519
6.342.3.2 WindowSystem	519
6.342.4 Property Documentation	520
6.342.4.1 Layer	520
6.342.4.2 State	520
6.343 Mergelt.Core.WindowSystem.WindowsSystem Class Reference	520
6.343.1 Detailed Description	521
6.343.2 Member Function Documentation	521
6.343.2.1 CloseWindow()	521
6.343.2.2 Dispose()	521
6.343.2.3 Initialize()	521

6.343.2.4 OpenWindow< TPresenter >()	522
6.343.3 Property Documentation	522
6.343.3.1 Blackout	522
6.343.3.2 Root	523
7 File Documentation	525
7.1 CurrencyComponent.cs	525
7.2 CurrencyComponentsTraits.cs	525
7.3 BaseSettingsDrawer.cs	526
7.4 CommonSettingsDrawer.cs	527
7.5 ElementConfigInspector.cs	528
7.6 GeneratorSettingsBaseDrawer.cs	530
7.7 GeneratorSettingsEvolutionDrawer.cs	531
7.8 GeneratorSettingsInspectorDrawer.cs	532
7.9 StylesConstants.cs	532
7.10 Constants.cs	533
7.11 EvolutionsGraphView.cs	533
7.12 EvolutionsGraphViewTraits.cs	539
7.13 EvolutionsEditorWindow.cs	540
7.14 BaseNode.cs	542
7.15 SerializedElementNodes.cs	543
7.16 ElementNode.cs	543
7.17 EvolutionNode.cs	545
7.18 IElementNode.cs	546
7.19 IEvolutionNode.cs	547
7.20 ISaveNode.cs	547
7.21 SchemeObjectEditor.cs	547
7.22 GeneratedItemsCreator.cs	548
7.23 GeneratedItemUI.cs	550
7.24 ActionCommandBase.cs	552
7.25 ActionCommandManager.cs	552
7.26 ApplyCellCommand.cs	553
7.27 ChangeCellCommand.cs	553
7.28 ClearCellCommand.cs	554
7.29 IActionCommand.cs	554
7.30 IActionCommandManager.cs	554
7.31 LimitedStack.cs	555
7.32 EvolutionSelection.cs	555
7.33 LevelConfigEditor.cs	556
7.34 LevelEditorWindow.cs	557
7.35 ConfigsPostprocessor.cs	567
7.36 SlicedFilledImageEditor.cs	568

7.37 AnimationControllerBase.cs	569
7.38 AnimationListener.cs	569
7.39 AnimationWindowStates.cs	569
7.40 AnimatorStateMachineBehaviour.cs	569
7.41 IAnimationController.cs	570
7.42 IAnimationListener.cs	570
7.43 IWindowAnimationController.cs	571
7.44 IWindowAnimationListener.cs	571
7.45 MecanimController.cs	571
7.46 WindowDefaultAnimationController.cs	571
7.47 WindowMecanimAnimationController.cs	572
7.48 Command.cs	573
7.49 CommandManager.cs	573
7.50 ICommand.cs	575
7.51 ICommandManager.cs	575
7.52 ConfigsConstants.cs	575
7.53 EvolutionData.cs	575
7.54 LevelElementData.cs	576
7.55 EffectConfig.cs	577
7.56 ElementConfig.cs	577
7.57 FieldElementIconComponent.cs	578
7.58 ConsumableSettings.cs	578
7.59 CurrencySettings.cs	579
7.60 ElementCommonSettings.cs	579
7.61 ElementGeneratorSettings.cs	580
7.62 GeneratableElement.cs	582
7.63 GameConfig.cs	582
7.64 HintsConfig.cs	582
7.65 GameIconsConfig.cs	582
7.66 IconParameters.cs	583
7.67 InventoryConfig.cs	583
7.68 LevelConfig.cs	583
7.69 LevelUpConfig.cs	584
7.70 LevelUpParameters.cs	584
7.71 ConsumableOperationType.cs	584
7.72 ConsumableTargetType.cs	584
7.73 CurrencyType.cs	585
7.74 ElementType.cs	585
7.75 LayerParameters.cs	585
7.76 LayersConfig.cs	586
7.77 IElementsStockComponent.cs	586
7.78 IElementsStockData.cs	586

7.79 EvolutionProgressData.cs	586
7.80 IEvolutionsProgressData.cs	587
7.81 FieldElementState.cs	587
7.82 FieldElement.cs	587
7.83 FieldElementData.cs	587
7.84 GeneratableFieldElement.cs	588
7.85 GridPoint.cs	588
7.86 IFieldElement.cs	589
7.87 IFieldElementModel.cs	589
7.88 IFieldElementPresenter.cs	589
7.89 IFieldElementView.cs	590
7.90 ConfigParameters.cs	590
7.91 GeneratorOpenParameters.cs	590
7.92 GeneratorParameters.cs	591
7.93 IConfigParameters.cs	591
7.94 IGeneratorOpenParameters.cs	591
7.95 IGeneratorParameters.cs	591
7.96 IInfoParameters.cs	592
7.97 InfoParameters.cs	592
7.98 IProducedByParameters.cs	592
7.99 IProduceParameters.cs	592
7.100 ProducedByParameters.cs	593
7.101 ProduceParameters.cs	593
7.102 SavedConfigParameters.cs	593
7.103 SavedGeneratorOpenParameters.cs	593
7.104 SavedGeneratorParameters.cs	594
7.105 SavedInfoParameters.cs	594
7.106 CurvesHelper.cs	594
7.107 ListExtensions.cs	595
7.108 NumericExtensions.cs	595
7.109 RectTransformExtensions.cs	596
7.110 SavesHelper.cs	596
7.111 SavesHelper.cs	597
7.112 TimeExtensions.cs	597
7.113 IInventoryData.cs	598
7.114 IMessage.cs	599
7.115 IMessageBus.cs	599
7.116 MessageBus.cs	599
7.117 IModel.cs	600
7.118 IPresenter.cs	601
7.119 IView.cs	601
7.120 Model.cs	601

7.121 Presenter.cs	601
7.122 View.cs	602
7.123 GameSaveType.cs	602
7.124 ISavable.cs	602
7.125 SavableAttribute.cs	603
7.126 EdgeData.cs	603
7.127 ElementNodeData.cs	603
7.128 EvolutionNodeData.cs	604
7.129 IElementNodeData.cs	605
7.130 IEvolutionNodeData.cs	605
7.131 SchemeData.cs	605
7.132 SchemeObject.cs	605
7.133 IConfigsService.cs	606
7.134 ICurrencyService.cs	606
7.135 IElementService.cs	606
7.136 IElementsStockService.cs	607
7.137 IEnergyService.cs	607
7.138 IEvolutionsService.cs	607
7.139 IGameFieldActionsService.cs	607
7.140 IGameFieldService.cs	608
7.141 IGameLoadService.cs	608
7.142 IGameSaveService.cs	608
7.143 IGameService.cs	608
7.144 IGeneratorsService.cs	608
7.145 IInfoPanelService.cs	609
7.146 IInventoryService.cs	609
7.147 IResourcesLoaderService.cs	609
7.148 IUserProgressService.cs	609
7.149 IUserService.cs	609
7.150 IUserData.cs	610
7.151 BlackoutComponent.cs	610
7.152 WindowCloseCommand.cs	610
7.153 WindowHideCommand.cs	611
7.154 WindowOpenCommand.cs	612
7.155 IWindowArgs.cs	612
7.156 IWindowOpenParameters.cs	612
7.157 WindowArgs.cs	613
7.158 WindowOpenParameters.cs	613
7.159 IWindowFactory.cs	613
7.160 WindowFactory.cs	613
7.161 IWindowSystem.cs	614
7.162 CloseWindowMessage.cs	614

7.163 SortingLayers.cs	615
7.164 SortingLayers.cs	615
7.165 WindowName.cs	615
7.166 IWindow.cs	615
7.167 IWindowPresenter.cs	616
7.168 WindowBase.cs	616
7.169 WindowCreateInfo.cs	617
7.170 WindowModel.cs	618
7.171 WindowPresenter.cs	618
7.172WindowState.cs	620
7.173 WindowsStorage.cs	620
7.174 WindowsSystem.cs	621
7.175 CheckEvolutionsProgressCommand.cs	624
7.176 CheckOpeningGeneratorsCommand.cs	625
7.177 EffectCommand.cs	625
7.178 LoadConfigsCommand.cs	625
7.179 PrepareEnergyCommand.cs	626
7.180 PrepareFieldCommand.cs	626
7.181 PrepareInventoryCommand.cs	627
7.182 PrepareStockCommand.cs	628
7.183 PrepareUserCommand.cs	628
7.184 ConfigsContainer.cs	629
7.185 ElementPrefab.cs	630
7.186 ConfigsService.cs	630
7.187 ConfigsServiceModel.cs	632
7.188 ConfigProcessor.cs	633
7.189 IConfigProcessor.cs	637
7.190 JumpEffectConfig.cs	637
7.191 MoveEffectConfig.cs	637
7.192 MoveToUiEffectConfig.cs	637
7.193 BaseEffect.cs	638
7.194 BaseEffectWithPresenter.cs	638
7.195 IEffect.cs	639
7.196 IEffectWithPresenter.cs	639
7.197 JumpEffect.cs	639
7.198 JumpEffectWithPresenter.cs	640
7.199 MoveEffect.cs	641
7.200 MoveEffectWithPresenter.cs	642
7.201 MoveToUiEffect.cs	643
7.202 MoveToUiEffectWithPresenter.cs	643
7.203 EffectName.cs	644
7.204 EffectsFactory.cs	644

7.205 EffectsManager.cs	645
7.206 IEffectsFactory.cs	645
7.207 IEffectsManager.cs	646
7.208 IEffectParameters.cs	646
7.209 MoveEffectParameters.cs	646
7.210 ElementsStockComponent.cs	647
7.211 ElementsStockData.cs	648
7.212 HudPanelType.cs	649
7.213 EvolutionsProgressData.cs	649
7.214 ElementInfoFactory.cs	649
7.215 ElementInfoType.cs	650
7.216 IElementInfoFactory.cs	650
7.217 FieldFactory.cs	650
7.218 IFieldFactory.cs	651
7.219 FieldElementFactory.cs	651
7.220 FieldElementVisualFactory.cs	653
7.221 IFieldElementFactory.cs	654
7.222 IFieldElementVisualFactory.cs	654
7.223 IconFactory.cs	654
7.224 IIconFactory.cs	655
7.225 IInventoryFactory.cs	655
7.226 InventoryData.cs	656
7.227 InventoryFactory.cs	656
7.228 FieldActionProcessorBase.cs	657
7.229 FieldConsumableProcessor.cs	658
7.230 FieldGenerationProcessor.cs	659
7.231 FieldMergeProcessor.cs	661
7.232 IFieldActionProcessor.cs	664
7.233 FieldElementAnimationController.cs	664
7.234 FieldElementStateMachineBehaviour.cs	665
7.235 FieldElementModel.cs	665
7.236 FieldElementPresenter.cs	666
7.237 FieldElementPresenterBase.cs	666
7.238 FieldElementView.cs	668
7.239 FieldElementGeneratorModel.cs	669
7.240 FieldElementGeneratorPresenter.cs	670
7.241 FieldElementGeneratorView.cs	671
7.242 GeneratorTimerComponent.cs	671
7.243 FieldCellComponent.cs	672
7.244 FieldData.cs	674
7.245 FieldLogic.cs	675
7.246 FieldLogicModel.cs	676

7.247 FieldModel.cs	676
7.248 FieldPresenter.cs	676
7.249 FieldView.cs	680
7.250 IFieldLogic.cs	680
7.251 GameRoot.cs	680
7.252 FieldElementExtensions.cs	681
7.253 FieldGeneratorElementExtensions.cs	682
7.254 HudTargetComponent.cs	686
7.255 HudTargets.cs	686
7.256 EnergyComponent.cs	686
7.257 HardCurrencyComponent.cs	688
7.258 HudComponent.cs	688
7.259 HudPanelBase.cs	689
7.260 IHudPanel.cs	689
7.261 IUserListener.cs	689
7.262 ProgressComponent.cs	690
7.263 SoftCurrencyComponent.cs	691
7.264 UserListenerComponent.cs	692
7.265 HintsManager.cs	692
7.266 IHintsManager.cs	694
7.267 ActivateHintsMessage.cs	694
7.268 CheckGeneratorMessage.cs	694
7.269 ClickElementMessage.cs	695
7.270 CreateElementMessage.cs	695
7.271 ElementActionMessage.cs	695
7.272 EndDragElementMessage.cs	695
7.273 EnergyRestoredMessage.cs	696
7.274 FieldFullMessage.cs	696
7.275 GeneratorBusyMessage.cs	696
7.276 GeneratorOpenedMessage.cs	696
7.277 GeneratorOpenStartMessage.cs	696
7.278 GeneratorRestoredMessage.cs	697
7.279 LevelUpdatedMessage.cs	697
7.280 LevelUpMessage.cs	697
7.281 LoadedGameMessage.cs	697
7.282 MergeElementsMessage.cs	697
7.283 MoveElementMessage.cs	698
7.284 PrepareGameMessage.cs	698
7.285 RemoveElementMessage.cs	698
7.286 RemoveFromInventoryMessage.cs	698
7.287 ResetHintsMessage.cs	698
7.288 ResetPositionMessage.cs	699

7.289 ResetSelectionMessage.cs	699
7.290 SelectedElementMessage.cs	699
7.291 SkipTimeMessage.cs	699
7.292 SplitElementMessage.cs	699
7.293 StartChargingMessage.cs	700
7.294 StartGameMessage.cs	700
7.295 SwapElementsMessage.cs	700
7.296 UnlockElementMessage.cs	700
7.297 UpdateStockMessage.cs	700
7.298 CurrencyService.cs	701
7.299 ElementService.cs	702
7.300 ElementsStockService.cs	703
7.301 ElementsStockServiceModel.cs	704
7.302 EnergyService.cs	704
7.303 EvolutionsService.cs	706
7.304 EvolutionsServiceModel.cs	708
7.305 GameFieldActionsService.cs	708
7.306 GameFieldService.cs	709
7.307 GameService.cs	710
7.308 GameServiceModel.cs	711
7.309 GeneratorService.cs	711
7.310 InfoPanelService.cs	714
7.311 InventoryService.cs	715
7.312 InventoryServiceModel.cs	717
7.313 ResourcesLoaderService.cs	717
7.314 GameLoadService.cs	717
7.315 GameSaveEveryIntervalService.cs	718
7.316 GameSaveOnEveryActionService.cs	720
7.317 GameSaveOnQuitService.cs	722
7.318 BaseSerializeStrategy.cs	723
7.319 BinaryFormatterStrategy.cs	724
7.320 ISerializeStrategy.cs	724
7.321 JsonSerializerStrategy.cs	725
7.322 UserProgressService.cs	725
7.323 UserService.cs	727
7.324 UserServiceModel.cs	727
7.325 TopPanelComponent.cs	728
7.326 ElementActionType.cs	728
7.327 ElementCommonInfo.cs	729
7.328 ElementInfo.cs	729
7.329 ElementInfoPanelComponent.cs	730
7.330 ElementOpenInfo.cs	732

7.331 ElementSellInfo.cs	733
7.332 ElementSpeedUpChargingInfo.cs	734
7.333 ElementSpeedUpOpeningInfo.cs	735
7.334 ElementSplitInfo.cs	736
7.335 ElementUnlockInfo.cs	737
7.336 InventoryPanelComponent.cs	738
7.337 InventoryPanelItemComponent.cs	740
7.338 InventoryPanelItemPair.cs	740
7.339 UserData.cs	741
7.340 ElementInfoArgs.cs	741
7.341 ElementInfoItemComponent.cs	741
7.342 ElementInfoItemsPanelComponent.cs	743
7.343 ElementInfoModel.cs	743
7.344 ElementInfoPresenter.cs	744
7.345 ElementInfoWindow.cs	746
7.346 ElementInfoWindowContent.cs	747
7.347 InventoryModel.cs	747
7.348 InventoryPresenter.cs	747
7.349 InventoryWindow.cs	750
7.350 InventoryWidgetItemComponent.cs	750
7.351 InventoryWidgetItemPair.cs	751
7.352 InventoryWindowPaidCellComponent.cs	751
7.353 UserInfoModel.cs	752
7.354 UserInfoPresenter.cs	752
7.355 UserInfoWindow.cs	754
7.356 UserProgressComponent.cs	755
7.357 DiContainer.cs	755
7.358 DiHandler.cs	757
7.359 DiRoot.cs	758
7.360 IntroduceAttribute.cs	759
7.361 IInitializable.cs	759
7.362 IUpdatable.cs	759
7.363 Bindable.cs	760
7.364 IMonoApplicationQuitHandler.cs	761
7.365 IMonoUpdateHandler.cs	761
7.366 MainThreadDispatcher.cs	761
7.367 MonoEventsListener.cs	762
7.368 ResponsiveGridLayout.cs	763
7.369 SlicedFilledImage.cs	764
Index	773

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Mergelt	29
Mergelt.Core	29
Mergelt.Core.Animations	29
Mergelt.Core.Commands	29
Mergelt.Core.Configs	30
Mergelt.Core.Configs.Data	30
Mergelt.Core.Configs.Effects	30
Mergelt.Core.Configs.Elements	30
Mergelt.Core.Configs.Hints	30
Mergelt.Core.Configs.Icons	30
Mergelt.Core.Configs.Inventory	31
Mergelt.Core.Configs.LevelUp	31
Mergelt.Core.Configs.Types	31
Mergelt.Core.Configs.Windows	32
Mergelt.Core.ElementsStock	32
Mergelt.Core.Evolutions	32
Mergelt.Core.FieldElements	33
Mergelt.Core.Helpers	34
Mergelt.Core.Inventory	34
Mergelt.Core.Messages	34
Mergelt.Core.MVP	34
Mergelt.Core.Saves	34
Mergelt.Core.Schemes	35
Mergelt.Core.Schemes.Data	35
Mergelt.Core.Services	35
Mergelt.Core.User	36
Mergelt.Core.Utils	36
Mergelt.Core.WindowSystem	36
Mergelt.Core.WindowSystem.Commands	36
Mergelt.Core.WindowSystem.Data	37
Mergelt.Core.WindowSystem.Factory	37
Mergelt.Core.WindowSystem.Messages	37
Mergelt.Core.WindowSystem.Windows	37
Mergelt.Editor	38
Mergelt.Editor.Configs	38

Mergelt.Editor.Configs.ItemsDrawers	38
Mergelt.Editor.Core	38
Mergelt.Editor.Core.LevelEditor	38
Mergelt.Editor.Core.LevelEditor.Commands	38
Mergelt.Editor.EvolutionsEditor	39
Mergelt.Editor.Helpers	39
Mergelt.Editor.LevelEditor	39
Mergelt.Editor.LevelEditor.Commands	39
Mergelt.Game	40
Mergelt.Game.Commands	40
Mergelt.Game.Configs	40
Mergelt.Game.Configs.Services	40
Mergelt.Game.Converters	40
Mergelt.Game.Effects	41
Mergelt.Game.Effects.Configs	41
Mergelt.Game.Effects.Controllers	41
Mergelt.Game.Effects.Parameters	41
Mergelt.Game.ElementsStock	41
Mergelt.Game.Enums	42
Mergelt.Game.Evolutions	42
Mergelt.Game.Factories	42
Mergelt.Game.Factories.ElementInfo	42
Mergelt.Game.Factories.Field	43
Mergelt.Game.Factories.FieldElement	43
Mergelt.Game.Factories.Icons	43
Mergelt.Game.Factories.Inventory	43
Mergelt.Game.Field	44
Mergelt.Game.Field.Actions	44
Mergelt.Game.Field.Elements	44
Mergelt.Game.Field.Elements.Animations	44
Mergelt.Game.Field.Elements.Generator	44
Mergelt.Game.Helpers	45
Mergelt.Game.HUD	45
Mergelt.Game.Managers	45
Mergelt.Game.Messages	45
Mergelt.Game.Services	46
Mergelt.Game.Services.Saves	46
Mergelt.Game.Services.Saves.Strategies	47
Mergelt.Game.TopPanel	47
Mergelt.Game.UI	47
Mergelt.Game.UI.InfoPanel	47
Mergelt.Game.UI.InventoryPanel	48
Mergelt.Game.User	48
Mergelt.Game.Windows	48
Mergelt.Game.Windows.ElementInfo	48
Mergelt.Game.Windows.Inventory	48
Mergelt.Game.Windows.UserProgress	48
Mergelt.SimpleDI	49
Mergelt.SimpleDI.ReservedInterfaces	49

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

AssetModificationProcessor	
Mergelt.Editor.Helpers.ConfigsPostprocessor	97
Attribute	
Mergelt.Core.Saves.SavableAttribute	456
Mergelt.SimpleDI.IntroduceAttribute	354
Mergelt.Core.Utils.Bindable< T >	72
Mergelt.Core.Utils.Bindable< float >	72
Mergelt.Game.Configs.Services.ConfigsServiceModel	101
Mergelt.Core.Configs.Elements.ConsumableSettings	105
Mergelt.Core.Configs.Elements.CurrencySettings	111
Mergelt.SimpleDI.DiContainer	112
Mergelt.SimpleDI.DiHandler	114
Mergelt.Core.Schemes.Data.EdgeData	116
UnityEditor.Editor	
Mergelt.Editor.Configs.ItemsDrawers.ElementConfigInspector	128
Mergelt.Editor.EvolutionsEditor.SchemeObjectEditor	463
Mergelt.Editor.Helpers.SavesHelper	461
Mergelt.Editor.Helpers.SlicedFilledImageEditor	471
Mergelt.Editor.LevelEditor.LevelConfigEditor	401
EditorWindow	
Mergelt.Editor.EvolutionsEditor.EvolutionsEditorWindow	188
Mergelt.Editor.LevelEditor.LevelEditorWindow	402
Mergelt.Core.Configs.Elements.ElementCommonSettings	124
Mergelt.Core.Configs.Elements.ElementGeneratorSettings	130
Mergelt.Game.Configs.ElementPrefab	156
Mergelt.Game.Services.ElementsStockServiceModel	173
Mergelt.Core.Evolutions.EvolutionProgressData	187
Mergelt.Editor.LevelEditor.EvolutionSelection	191
Mergelt.Game.Services.EvolutionsServiceModel	204
Mergelt.Core.FieldElements.FieldElementData	217
Mergelt.Game.Field.Elements.FieldElementPresenterBase< FieldElementGeneratorView, FieldElementGeneratorModel >	229
Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorPresenter	220
Mergelt.Game.Field.Elements.FieldElementPresenterBase< FieldElementView, FieldElementModel >	229
Mergelt.Game.Field.Elements.FieldElementPresenter	227

Mergelt.Game.Field.FieldLogicModel	247
Mergelt.Game.Services.GameServiceModel	269
Mergelt.Core.Configs.Elements.GeneratableElement	270
Mergelt.Core.FieldElements.GeneratableFieldElement	271
Mergelt.Editor.LevelEditor.Commands.IActionCommand	303
Mergelt.Editor.LevelEditor.Commands.ActionCommandBase	51
Mergelt.Editor.LevelEditor.Commands.ApplyCellCommand	58
Mergelt.Editor.LevelEditor.Commands.ChangeCellCommand	75
Mergelt.Editor.LevelEditor.Commands.ClearCellCommand	80
Mergelt.Editor.LevelEditor.Commands.IActionCommandManager	303
Mergelt.Editor.LevelEditor.Commands.ActionCommandManager	52
Mergelt.Core.Animations.IAnimationController	304
Mergelt.Core.Animations.AnimationControllerBase	54
Mergelt.Core.Animations.MecanimController	415
Mergelt.Game.Field.Elements.Animations.FieldElementAnimationController	215
Mergelt.Core.Animations.IAnimationListener	305
Mergelt.Core.Animations.AnimationListener	56
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >	229
IBeginDragHandler	
Mergelt.Game.Field.FieldCellComponent	207
ICanvasRaycastFilter	
Mergelt.Core.Helpers.SlicedFilledImage	466
Mergelt.Core.Commands.ICommand	305
Mergelt.Core.Commands.Command	83
Mergelt.Core.WindowSystem.Commands.WindowCloseCommand	500
Mergelt.Core.WindowSystem.Commands.WindowHideCommand	507
Mergelt.Core.WindowSystem.Commands.WindowOpenCommand	511
Mergelt.Game.Commands.CheckEvolutionsProgressCommand	77
Mergelt.Game.Commands.CheckOpeningGeneratorsCommand	79
Mergelt.Game.Commands.EffectCommand	117
Mergelt.Game.Commands.LoadConfigsCommand	412
Mergelt.Game.Commands.PrepareEnergyCommand	438
Mergelt.Game.Commands.PrepareFieldCommand	439
Mergelt.Game.Commands.PrepareInventoryCommand	441
Mergelt.Game.Commands.PrepareStockCommand	442
Mergelt.Game.Commands.PrepareUserCommand	444
Mergelt.Core.Commands.ICommandManager	306
Mergelt.Core.Commands.CommandManager	85
IComparable	
Mergelt.Core.Configs.Windows.LayerParameters	398
Mergelt.Core.FieldElements.GridPoint	293
Mergelt.Core.FieldElements.IConfigParameters	308
Mergelt.Core.FieldElements.ConfigParameters	90
Mergelt.Game.Converters.IConfigProcessor	309
Mergelt.Game.Converters.ConfigProcessor	91
Mergelt.Core.Services.IConfigsService	310
Mergelt.Game.Configs.Services.ConfigsService	97
Mergelt.Core.Configs.Icons.IconParameters	311
Mergelt.Core.Services.ICurrencyService	312
Mergelt.Game.Services.CurrencyService	109
IDisposable	
Mergelt.Core.Commands.Command	83
Mergelt.Core.MVP.IPresenter	373
Mergelt.Core.FieldElements.IFieldElementPresenter	330
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >	229

Mergelt.Core.MVP.Presenter< TView, TModel >	445
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >	229
Mergelt.Core.WindowSystem.Windows.IWindowPresenter	388
Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >	514
Mergelt.Core.WindowSystem.WindowsSystem	520
Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer	67
Mergelt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer	88
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer	285
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsEvolutionDrawer	288
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsInspectorDrawer	291
Mergelt.Editor.EvolutionsEditor.BaseNode	64
Mergelt.Editor.EvolutionsEditor.ElementNode	149
Mergelt.Editor.EvolutionsEditor.EvolutionNode	182
Mergelt.Editor.GeneratedItemUI	273
Mergelt.Editor.GeneratedItemsCreator	271
Mergelt.Game.Field.FieldLogic	246
Mergelt.Game.Managers.HintsManager	299
Mergelt.Game.Services.EnergyService	179
Mergelt.Game.Services.GameFieldActionsService	257
Mergelt.Game.Services.GameFieldService	258
Mergelt.Game.Services.GameService	268
Mergelt.Game.Services.GeneratorService	282
Mergelt.Game.Services.InfoPanelService	351
Mergelt.Game.Services.Saves.GameSaveEveryIntervalService	263
Mergelt.Game.Services.Saves.GameSaveOnEveryActionService	265
Mergelt.Game.Services.Saves.GameSaveOnQuitService	266
Mergelt.Game.Services.UserProgressService	487
IDragHandler	
Mergelt.Game.Field.FieldCellComponent	207
Mergelt.Game.Effects.Controllers.IEffect	313
Mergelt.Game.Effects.Controllers.BaseEffect	59
Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter	62
Mergelt.Game.Effects.Controllers.JumpEffectWithPresenter	395
Mergelt.Game.Effects.Controllers.MoveEffectWithPresenter	428
Mergelt.Game.Effects.Controllers.MoveToUiEffectWithPresenter	435
Mergelt.Game.Effects.Controllers.JumpEffect	392
Mergelt.Game.Effects.Controllers.MoveEffect	423
Mergelt.Game.Effects.Controllers.MoveToUiEffect	432
Mergelt.Game.Effects.Parameters.IEffectParameters	313
Mergelt.Game.Effects.Parameters.MoveEffectParameters	426
Mergelt.Game.Effects.IEffectsFactory	314
Mergelt.Game.Effects.EffectsFactory	119
Mergelt.Game.Effects.IEffectsManager	315
Mergelt.Game.Effects.EffectsManager	121
Mergelt.Game.Effects.Controllers.IEffectWithPresenter	315
Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter	62
Mergelt.Game.Factories.ElementInfo.IElementInfoFactory	316
Mergelt.Game.Factories.ElementInfo.ElementInfoFactory	137
Mergelt.Core.Schemes.Data.IElementNodeData	318
Mergelt.Core.Schemes.Data.ElementNodeData	152
Mergelt.Core.Services.IElementService	319
Mergelt.Game.Services.ElementService	159
Mergelt.Core.ElementsStock.IElementsStockComponent	320
Mergelt.Game.ElementsStock.ElementsStockComponent	168
Mergelt.Core.Services.IElementsStockService	321

Mergelt.Game.Services.ElementsStockService	170
IEndDragHandler	
Mergelt.Game.Field.FieldCellComponent	207
Mergelt.Core.Services.IEnergyService	321
Mergelt.Game.Services.EnergyService	179
IEquatable	
Mergelt.Core.Configs.Data.EvolutionData	180
Mergelt.Core.Configs.Windows.LayerParameters	398
Mergelt.Core.FieldElements.GridPoint	293
Mergelt.Core.WindowSystem.Windows.WindowCreateInfo	502
Mergelt.Core.Schemes.Data.IEvolutionNodeData	323
Mergelt.Core.Schemes.Data.EvolutionNodeData	185
Mergelt.Core.Services.IEvolutionsService	326
Mergelt.Game.Services.EvolutionsService	201
Mergelt.Game.Field.Actions.IFieldActionProcessor	326
Mergelt.Game.Field.Actions.FieldActionProcessorBase	205
Mergelt.Game.Field.Actions.FieldConsumableProcessor	210
Mergelt.Game.Field.Actions.FieldGenerationProcessor	244
Mergelt.Game.Field.Actions.FieldMergeProcessor	249
Mergelt.Core.FieldElements.IFieldElement	327
Mergelt.Core.FieldElements.FieldElement	213
Mergelt.Game.Factories.FieldElement.IFieldElementFactory	328
Mergelt.Game.Factories.FieldElement.FieldElementFactory	218
Mergelt.Core.FieldElements.IFieldElementModel	329
Mergelt.Game.Field.Elements.FieldElementModel	226
Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorModel	219
Mergelt.Game.Factories.FieldElement.IFieldElementVisualFactory	334
Mergelt.Game.Factories.FieldElement.FieldElementVisualFactory	240
Mergelt.Game.Factories.Field.IFieldFactory	334
Mergelt.Game.Factories.Field.FieldFactory	242
Mergelt.Game.Field.IFieldLogic	335
Mergelt.Game.Field.FieldLogic	246
Mergelt.Core.Services.IGameFieldActionsService	335
Mergelt.Game.Services.GameFieldActionsService	257
Mergelt.Core.Services.IGameFieldService	335
Mergelt.Game.Services.GameFieldService	258
Mergelt.Core.Services.IGameLoadService	336
Mergelt.Game.Services.Saves.GameLoadService	261
Mergelt.Core.Services.IGameSaveService	337
Mergelt.Game.Services.Saves.GameSaveEveryIntervalService	263
Mergelt.Game.Services.Saves.GameSaveOnEveryActionService	265
Mergelt.Game.Services.Saves.GameSaveOnQuitService	266
Mergelt.Core.Services.IGameService	337
Mergelt.Game.Services.GameService	268
Mergelt.Core.FieldElements.IGeneratorOpenParameters	337
Mergelt.Core.FieldElements.GeneratorOpenParameters	277
Mergelt.Core.FieldElements.IGeneratorParameters	339
Mergelt.Core.FieldElements.GeneratorParameters	279
Mergelt.Core.Services.IGeneratorsService	341
Mergelt.Game.Services.GeneratorService	282
Mergelt.Game.Managers.IHintsManager	341
Mergelt.Game.Managers.HintsManager	299

Mergelt.Game.HUD.IHudPanel	342
Mergelt.Game.HUD.HudPanelBase	301
Mergelt.Game.HUD.EnergyComponent	176
Mergelt.Game.HUD.HardCurrencyComponent	297
Mergelt.Game.HUD.ProgressComponent	449
Mergelt.Game.HUD.SoftCurrencyComponent	472
Mergelt.Game.Factories.Icons.IIconFactory	342
Mergelt.Game.Factories.Icons.IconFactory	307
Mergelt.Core.Services.IInfoPanelService	343
Mergelt.Game.Services.InfoPanelService	351
Mergelt.Core.FieldElements.IInfoParameters	343
Mergelt.Core.FieldElements.InfoParameters	352
Mergelt.SimpleDI.ReservedInterfaces.IInitializable	345
Mergelt.Core.Messages.MessageBus	417
Mergelt.Core.WindowSystem.WindowsSystem	520
Mergelt.Game.Field.FieldLogic	246
Mergelt.Game.Managers.HintsManager	299
Mergelt.Game.Services.EnergyService	179
Mergelt.Game.Services.GameFieldActionsService	257
Mergelt.Game.Services.GameFieldService	258
Mergelt.Game.Services.GameService	268
Mergelt.Game.Services.GeneratorService	282
Mergelt.Game.Services.InfoPanelService	351
Mergelt.Game.Services.Saves.GameSaveEveryIntervalService	263
Mergelt.Game.Services.Saves.GameSaveOnEveryActionService	265
Mergelt.Game.Services.Saves.GameSaveOnQuitService	266
Mergelt.Game.Services.UserProgressService	487
Mergelt.Game.Factories.Inventory.IInventoryFactory	346
Mergelt.Game.Factories.Inventory.InventoryFactory	356
Mergelt.Core.Services.IInventoryService	347
Mergelt.Game.Services.InventoryService	363
ILayoutElement	
Mergelt.Core.Helpers.SlicedFilledImage	466
Mergelt.Core.Messages.IMessage	348
Mergelt.Core.WindowSystem.Messages.CloseWindowMessage	83
Mergelt.Game.Messages.ActivateHintsMessage	53
Mergelt.Game.Messages.CheckGeneratorMessage	78
Mergelt.Game.Messages.ClickElementMessage	82
Mergelt.Game.Messages.CreateElementMessage	106
Mergelt.Game.Messages.ElementActionMessage	122
Mergelt.Game.Messages.EndDragElementMessage	175
Mergelt.Game.Messages.EnergyRestoredMessage	178
Mergelt.Game.Messages.FieldFullMessage	244
Mergelt.Game.Messages.GeneratorBusyMessage	275
Mergelt.Game.Messages.GeneratorOpenStartMessage	278
Mergelt.Game.Messages.GeneratorOpenedMessage	276
Mergelt.Game.Messages.GeneratorRestoredMessage	281
Mergelt.Game.Messages.LevelUpMessage	409
Mergelt.Game.Messages.LevelUpdatedMessage	408
Mergelt.Game.Messages.LoadedGameMessage	413
Mergelt.Game.Messages.MergeElementsMessage	417
Mergelt.Game.Messages.MoveElementMessage	431
Mergelt.Game.Messages.PrepareGameMessage	441
Mergelt.Game.Messages.RemoveElementMessage	452
Mergelt.Game.Messages.RemoveFromInventoryMessage	452
Mergelt.Game.Messages.ResetHintsMessage	453

Mergelt.Game.Messages.ResetPositionMessage	453
Mergelt.Game.Messages.ResetSelectionMessage	454
Mergelt.Game.Messages.SelectedElementMessage	464
Mergelt.Game.Messages.SkipTimeMessage	465
Mergelt.Game.Messages.SplitElementMessage	474
Mergelt.Game.Messages.StartChargingMessage	474
Mergelt.Game.Messages.StartGameMessage	475
Mergelt.Game.Messages.SwapElementsMessage	475
Mergelt.Game.Messages.UnlockElementMessage	477
Mergelt.Game.Messages.UpdateStockMessage	477
Mergelt.Core.Messages.IMessageBus	348
Mergelt.Core.Messages.MessageBus	417
Mergelt.Core.MVP.IModel	350
Mergelt.Core.MVP.Model	420
Mergelt.Game.Field.FieldModel	251
Mergelt.Core.WindowSystem.Windows.WindowModel	511
Mergelt.Game.Windows.ElementInfo.ElementInfoModel	142
Mergelt.Game.Windows.Inventory.InventoryModel	358
Mergelt.Game.Windows.UserProgress.UserInfoModel	480
Mergelt.Game.Field.Elements.FieldElementModel	226
Mergelt.Core.Helpers.IMonoApplicationQuitHandler	350
Mergelt.Game.Services.Saves.GameSaveEveryIntervalService	263
Mergelt.Game.Services.Saves.GameSaveOnEveryActionService	265
Mergelt.Game.Services.Saves.GameSaveOnQuitService	266
Mergelt.Core.Helpers.IMonoUpdateHandler	351
Mergelt.Core.WindowSystem.Commands.WindowCloseCommand	500
Mergelt.Core.WindowSystem.Commands.WindowHideCommand	507
Mergelt.Core.WindowSystem.Commands.WindowOpenCommand	511
Mergelt.Game.UI.InventoryPanel.InventoryPanelItemPair	360
Mergelt.Game.Services.InventoryServiceModel	366
Mergelt.Game.Windows.Inventory.InventoryWindowItemPair	371
IPointerDownHandler	
Mergelt.Game.Field.FieldCellComponent	207
IPointerUpHandler	
Mergelt.Game.Field.FieldCellComponent	207
Mergelt.Core.FieldElements.IProducedByParameters	
Mergelt.Core.FieldElements.ProducedByParameters	447
Mergelt.Core.FieldElements.IProduceParameters	
Mergelt.Core.FieldElements.ProduceParameters	448
Mergelt.Core.Services.IResourcesLoaderService	
Mergelt.Game.Services.ResourcesLoaderService	454
Mergelt.Core.Saves.ISavable	
Mergelt.Core.ElementsStock.IElementsStockData	320
Mergelt.Game.ElementsStock.ElementsStockData	170
Mergelt.Core.Evolutions.IEvolutionsProgressData	325
Mergelt.Game.Evolutions.EvolutionsProgressData	200
Mergelt.Core.Inventory.IInventoryData	345
Mergelt.Game.Factories.Inventory.InventoryData	355
Mergelt.Core.User.IUserData	378
Mergelt.Game.User.UserData	478
Mergelt.Game.Field.FieldData	212
Mergelt.Editor.EvolutionsEditor.ISaveNode	
Mergelt.Editor.EvolutionsEditor.BaseNode	64
Mergelt.Editor.EvolutionsEditor.IElementNode	317
Mergelt.Editor.EvolutionsEditor.ElementNode	149

Mergelt.Editor.EvolutionsEditor.IEvolutionNode	322
Mergelt.Editor.EvolutionsEditor.EvolutionNode	182
ISerializationCallbackReceiver	
Mergelt.Core.Helpers.SlicedFilledImage	466
Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy	377
Mergelt.Game.Services.Saves.Strategies.BaseSerializeStrategy	66
Mergelt.Game.Services.Saves.Strategies.BinaryFormatterStrategy	70
Mergelt.Game.Services.Saves.Strategies.JsonSerializeStrategy	390
Mergelt.SimpleDI.ReservedInterfaces.IUpdatable	378
Mergelt.Game.Effects.EffectsManager	121
Mergelt.Game.Managers.HintsManager	299
Mergelt.Game.Services.GeneratorService	282
Mergelt.Game.Services.Saves.GameSaveEveryIntervalService	263
Mergelt.Game.HUD.IUserListener	380
Mergelt.Game.HUD.UserListenerComponent	486
Mergelt.Game.HUD.HudPanelBase	301
Mergelt.Core.Services.IUserProgressService	381
Mergelt.Game.Services.UserProgressService	487
Mergelt.Core.Services.IUserService	381
Mergelt.Game.Services.UserService	489
Mergelt.Core.MVP.View	382
Mergelt.Core.FieldElements.IFieldElementView	332
Mergelt.Game.Field.Elements.FieldElementView	237
Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorView	222
Mergelt.Core.MVP.View	493
Mergelt.Core.WindowSystem.Windows.WindowBase	495
Mergelt.Game.Windows.ElementInfo.ElementInfoWindow	145
Mergelt.Game.Windows.Inventory.InventoryWindow	367
Mergelt.Game.Windows.UserProgress.UserInfoWindow	482
Mergelt.Game.Field.Elements.FieldElementView	237
Mergelt.Game.Field.FieldView	254
Mergelt.Core.WindowSystem.Windows.IWindow	383
Mergelt.Core.WindowSystem.Windows.WindowBase	495
Mergelt.Core.Animations.IWindowAnimationController	385
Mergelt.Core.Animations.WindowDefaultAnimationController	504
Mergelt.Core.Animations.WindowMecanimAnimationController	509
Mergelt.Core.Animations.IWindowAnimationListener	385
Mergelt.Core.WindowSystem.Windows.WindowBase	495
Mergelt.Core.WindowSystem.Data.IWindowArgs	386
Mergelt.Core.WindowSystem.Data.WindowArgs	495
Mergelt.Game.Windows.ElementInfo.ElementInfoArgs	136
Mergelt.Core.WindowSystem.Factory.IWindowFactory	386
Mergelt.Core.WindowSystem.Factory.WindowFactory	506
Mergelt.Core.WindowSystem.Data.IWindowOpenParameters	387
Mergelt.Core.WindowSystem.Data.WindowOpenParameters	513
Mergelt.Core.WindowSystem.IWindowSystem	390
Mergelt.Core.WindowSystem.WindowsSystem	520
Mergelt.Core.Configs.Data.LevelElementData	406
Mergelt.Core.Configs.LevelUp.LevelUpParameters	409
Mergelt.Editor.Core.LevelEditor.Commands.LimitedStack< T >	410
Mergelt.Editor.Core.LevelEditor.Commands.LimitedStack< Mergelt.Editor.LevelEditor.Commands.← IActionCommand >	410
MaskableGraphic	
Mergelt.Core.Helpers.SlicedFilledImage	466

MonoBehaviour	
Mergelt.Core.Animations.AnimationControllerBase	54
Mergelt.Core.Animations.WindowMecanimAnimationController	509
Mergelt.Core.Configs.Elements.FieldElementIconComponent	224
Mergelt.Core.Helpers.MainThreadDispatcher	413
Mergelt.Core.Helpers.MonoEventsListener	421
Mergelt.Core.MVP.View	493
Mergelt.Core.Utils.ResponsiveGridLayout	455
Mergelt.Core.WindowSystem.BlackoutComponent	74
Mergelt.Game.ElementsStock.ElementsStockComponent	168
Mergelt.Game.Field.Elements.Generator.GeneratorTimerComponent	293
Mergelt.Game.Field.FieldCellComponent	207
Mergelt.Game.HUD.HudComponent	301
Mergelt.Game.HUD.ListenerComponent	486
Mergelt.Game.Helpers.HudTargetComponent	302
Mergelt.Game.TopPanel.TopPanelComponent	476
Mergelt.Game.UI.InfoPanel.ElementInfo	133
Mergelt.Game.UI.InfoPanel.ElementCommonInfo	122
Mergelt.Game.UI.InfoPanel.ElementOpenInfo	154
Mergelt.Game.UI.InfoPanel.ElementSellInfo	157
Mergelt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo	161
Mergelt.Game.UI.InfoPanel.ElementSpeedUpOpeningInfo	163
Mergelt.Game.UI.InfoPanel.ElementSplitInfo	166
Mergelt.Game.UI.InfoPanel.ElementUnlockInfo	173
Mergelt.Game.UI.InfoPanel.ElementInfoPanelComponent	143
Mergelt.Game.UI.InventoryPanel.InventoryPanelComponent	359
Mergelt.Game.UI.InventoryPanel.InventoryPanelItemComponent	359
Mergelt.Game.Windows.ElementInfo.ElementInfoItemComponent	138
Mergelt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent	140
Mergelt.Game.Windows.ElementInfo.ElementInfoWindowContent	148
Mergelt.Game.Windows.Inventory.InventoryWindowItemComponent	369
Mergelt.Game.Windows.Inventory.InventoryWindowPaidCellComponent	372
Mergelt.Game.Windows.UserProgress.UserProgressComponent	487
Mergelt.SimpleDI.DiRoot	115
Mergelt.Game.GameRoot	261
Node	
Mergelt.Editor.EvolutionsEditor.BaseNode	64
Mergelt.Core.MVP.Presenter< FieldView, FieldModel >	445
Mergelt.Game.Field.FieldPresenter	252
Mergelt.Core.MVP.Presenter< TWindow, TModel >	445
Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >	514
Mergelt.Core.FieldElements.SavedConfigParameters	457
Mergelt.Core.FieldElements.SavedGeneratorOpenParameters	458
Mergelt.Core.FieldElements.SavedGeneratorParameters	458
Mergelt.Core.FieldElements.SavedInfoParameters	460
Mergelt.Core.Schemes.SchemeData	461
ScriptableObject	
Mergelt.Core.Configs.Effects.EffectConfig	118
Mergelt.Game.Effects.Configs.JumpEffectConfig	394
Mergelt.Game.Effects.Configs.MoveEffectConfig	425
Mergelt.Game.Effects.Configs.MoveToUiEffectConfig	434
Mergelt.Core.Configs.Elements.ElementConfig	126
Mergelt.Core.Configs.GameConfig	256
Mergelt.Core.Configs.Hints.HintsConfig	298
Mergelt.Core.Configs.Icons.GameIconsConfig	260
Mergelt.Core.Configs.Inventory.InventoryConfig	354
Mergelt.Core.Configs.LevelConfig	400

MergeIt.Core.Configs.LevelUp.LevelUpConfig	407
MergeIt.Core.Configs.Windows.LayersConfig	399
MergeIt.Core.Schemes.SchemeObject	462
MergeIt.Game.Configs.ConfigsContainer	95
MergeIt.Editor.EvolutionsEditor.SerializedElementNodes	465
StateMachineBehaviour	
MergeIt.Core.Animations.AnimatorStateMachineBehaviour	56
MergeIt.Game.Field.Elements.Animations.FieldElementStateMachineBehaviour	236
MergeIt.Game.Services.UserServiceModel	490
MergeIt.Editor.Configs.CurrencyComponent.UxmlFactory	493
MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView.UxmlFactory	493
GraphView.UxmlTraits	
MergeIt.Editor.EvolutionsEditor.EvolutionsGraphViewTraits	200
UxmlTraits	
MergeIt.Editor.Configs.CurrencyComponentsTraits	108
VisualElement	
MergeIt.Editor.Configs.CurrencyComponent	107
MergeIt.Editor.GeneratedItemUI	273
MergeIt.Core.WindowSystem.Windows.WindowPresenter< ElementInfoWindow, ElementInfoModel >	514
MergeIt.Game.Windows.ElementInfo.ElementInfoPresenter	143
MergeIt.Core.WindowSystem.Windows.WindowPresenter< InventoryWindow, InventoryModel >	514
MergeIt.Game.Windows.Inventory.InventoryPresenter	361
MergeIt.Core.WindowSystem.Windows.WindowPresenter< UserInfoWindow, UserInfoModel >	514
MergeIt.Game.Windows.UserProgress.UserInfoPresenter	480
GraphView	
MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView	194

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Mergelt.Editor.LevelEditor.Commands.ActionCommandBase	51
Mergelt.Editor.LevelEditor.Commands.ActionCommandManager	52
Mergelt.Game.Messages.ActivateHintsMessage	53
Mergelt.Core.Animations.AnimationControllerBase	54
Mergelt.Core.Animations.AnimationListener	56
Mergelt.Core.Animations.AnimatorStateMachineBehaviour	56
Mergelt.Editor.LevelEditor.Commands.ApplyCellCommand	58
Mergelt.Game.Effects.Controllers.BaseEffect	59
Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter	62
Mergelt.Editor.EvolutionsEditor.BaseNode	64
Mergelt.Game.Services.Saves.Strategies.BaseSerializeStrategy	66
Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer	67
Mergelt.Game.Services.Saves.Strategies.BinaryFormatterStrategy	70
Mergelt.Core.Utils.Bindable< T >	72
Mergelt.Core.WindowSystem.BlackoutComponent	74
Mergelt.Editor.LevelEditor.Commands.ChangeCellCommand	75
Mergelt.Game.Commands.CheckEvolutionsProgressCommand	77
Mergelt.Game.Messages.CheckGeneratorMessage	78
Mergelt.Game.Commands.CheckOpeningGeneratorsCommand	79
Mergelt.Editor.LevelEditor.Commands.ClearCellCommand	80
Mergelt.Game.Messages.ClickElementMessage	82
Mergelt.Core.WindowSystem.Messages.CloseWindowMessage	83
Mergelt.Core.Commands.Command	83
Mergelt.Core.Commands.CommandManager	85
Mergelt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer	88
Mergelt.Core.FieldElements.ConfigParameters	90
Mergelt.Game.Converters.ConfigProcessor	91
Mergelt.Game.Configs.ConfigsContainer	95
Mergelt.Editor.Helpers.ConfigsPostprocessor	97
Mergelt.Game.Configs.Services.ConfigsService	97
Mergelt.Game.Configs.Services.ConfigsServiceModel	101
Mergelt.Core.Configs.Elements.ConsumableSettings	105
Mergelt.Game.Messages.CreateElementMessage	106
Mergelt.Editor.Configs.CurrencyComponent	107
Mergelt.Editor.Configs.CurrencyComponentsTraits	108

Mergelt.Game.Services.CurrencyService	109
Mergelt.Core.Configs.Elements.CurrencySettings	111
Mergelt.SimpleDI.DiContainer	112
Mergelt.SimpleDI.DiHandler	114
Mergelt.SimpleDI.DiRoot	115
Mergelt.Core.Schemes.Data.EdgeData	116
Mergelt.Game.Commands.EffectCommand	117
Mergelt.Core.Configs.Effects.EffectConfig	118
Mergelt.Game.Effects.EffectsFactory	119
Mergelt.Game.Effects.EffectsManager	121
Mergelt.Game.Messages.ElementActionMessage	122
Mergelt.Game.UI.InfoPanel.ElementCommonInfo	122
Mergelt.Core.Configs.Elements.ElementCommonSettings	124
Mergelt.Core.Configs.Elements.ElementConfig	126
Mergelt.Editor.Configs.ItemsDrawers.ElementConfigInspector	128
Mergelt.Core.Configs.Elements.ElementGeneratorSettings	130
Mergelt.Game.UI.InfoPanel.ElementInfo	133
Mergelt.Game.Windows.ElementInfo.ElementInfoArgs	136
Mergelt.Game.Factories.ElementInfo.ElementInfoFactory	137
Mergelt.Game.Windows.ElementInfo.ElementInfoItemComponent	138
Mergelt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent	140
Mergelt.Game.Windows.ElementInfo.ElementInfoModel	142
Mergelt.Game.UI.InfoPanel.ElementInfoPanelComponent	143
Mergelt.Game.Windows.ElementInfo.ElementInfoPresenter	143
Mergelt.Game.Windows.ElementInfo.ElementInfoWindow	145
Mergelt.Game.Windows.ElementInfo.ElementInfoWindowContent	148
Mergelt.Editor.EvolutionsEditor.ElementNode	149
Mergelt.Core.Schemes.Data.ElementNodeData	152
Mergelt.Game.UI.InfoPanel.ElementOpenInfo	154
Mergelt.Game.Configs.ElementPrefab	156
Mergelt.Game.UI.InfoPanel.ElementSellInfo	157
Mergelt.Game.Services.ElementService	159
Mergelt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo	161
Mergelt.Game.UI.InfoPanel.ElementSpeedUpOpeningInfo	163
Mergelt.Game.UI.InfoPanel.ElementSplitInfo	166
Mergelt.Game.ElementsStock.ElementsStockComponent	168
Mergelt.Game.ElementsStock.ElementsStockData	170
Mergelt.Game.Services.ElementsStockService	170
Mergelt.Game.Services.ElementsStockServiceModel	173
Mergelt.Game.UI.InfoPanel.ElementUnlockInfo	173
Mergelt.Game.Messages.EndDragElementMessage	175
Mergelt.Game.HUD.EnergyComponent	176
Mergelt.Game.Messages.EnergyRestoredMessage	178
Mergelt.Game.Services.EnergyService	179
Mergelt.Core.Configs.Data.EvolutionData	180
Mergelt.Editor.EvolutionsEditor.EvolutionNode	182
Mergelt.Core.Schemes.Data.EvolutionNodeData	185
Mergelt.Core.Evolutions.EvolutionProgressData	187
Mergelt.Editor.EvolutionsEditor.EvolutionsEditorWindow	188
Mergelt.Editor.LevelEditor.EvolutionSelection	191
Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView	194
Mergelt.Editor.EvolutionsEditor.EvolutionsGraphViewTraits	200
Mergelt.Game.Evolutions.EvolutionsProgressData	200
Mergelt.Game.Services.EvolutionsService	201
Mergelt.Game.Services.EvolutionsServiceModel	204
Mergelt.Game.Field.Actions.FieldActionProcessorBase	205
Mergelt.Game.Field.FieldCellComponent	207
Mergelt.Game.Field.Actions.FieldConsumableProcessor	210

Mergelt.Game.Field.FieldData	212
Mergelt.Core.FieldElements.FieldElement	213
Mergelt.Game.Field.Elements.Animations.FieldElementAnimationController	215
Mergelt.Core.FieldElements.FieldElementData	217
Mergelt.Game.Factories.FieldElement.FieldElementFactory	218
Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorModel	219
Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorPresenter	220
Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorView	222
Mergelt.Core.Configs.Elements.FieldElementIconComponent	224
Mergelt.Game.Field.Elements.FieldElementModel	226
Mergelt.Game.Field.Elements.FieldElementPresenter	227
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >	229
Mergelt.Game.Field.Elements.Animations.FieldElementStateMachineBehaviour	236
Mergelt.Game.Field.Elements.FieldElementView	237
Mergelt.Game.Factories.FieldElement.FieldElementVisualFactory	240
Mergelt.Game.Factories.Field.FieldFactory	242
Mergelt.Game.Messages.FieldFullMessage	244
Mergelt.Game.Field.Actions.FieldGenerationProcessor	244
Mergelt.Game.Field.FieldLogic	246
Mergelt.Game.Field.FieldLogicModel	247
Mergelt.Game.Field.Actions.FieldMergeProcessor	249
Mergelt.Game.Field.FieldModel	251
Mergelt.Game.Field.FieldPresenter	252
Mergelt.Game.Field.FieldView	254
Mergelt.Core.Configs.GameConfig	256
Mergelt.Game.Services.GameFieldActionsService	257
Mergelt.Game.Services.GameFieldService	258
Mergelt.Core.Configs.Icons.GameIconsConfig	260
Mergelt.Game.Services.Saves.GameLoadService	261
Mergelt.Game.GameRoot	261
Mergelt.Game.Services.Saves.GameSaveEveryIntervalService	263
Mergelt.Game.Services.Saves.GameSaveOnEveryActionService	265
Mergelt.Game.Services.Saves.GameSaveOnQuitService	266
Mergelt.Game.Services.GameService	268
Mergelt.Game.Services.GameServiceModel	269
Mergelt.Core.Configs.Elements.GeneratableElement	270
Mergelt.Core.FieldElements.GeneratableFieldElement	271
Mergelt.Editor.GeneratedItemsCreator	271
Mergelt.Editor.GeneratedItemUI	273
Mergelt.Game.Messages.GeneratorBusyMessage	275
Mergelt.Game.Messages.GeneratorOpenedMessage	276
Mergelt.Core.FieldElements.GeneratorOpenParameters	277
Mergelt.Game.Messages.GeneratorOpenStartMessage	278
Mergelt.Core.FieldElements.GeneratorParameters	279
Mergelt.Game.Messages.GeneratorRestoredMessage	281
Mergelt.Game.Services.GeneratorService	282
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer	285
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsEvolutionDrawer	288
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsInspectorDrawer	291
Mergelt.Game.Field.Elements.Generator.GeneratorTimerComponent	293
Mergelt.Core.FieldElements.GridPoint	293
Mergelt.Game.HUD.HardCurrencyComponent	297
Mergelt.Core.Configs.Hints.HintsConfig	298
Mergelt.Game.Managers.HintsManager	299
Mergelt.Game.HUD.HudComponent	301
Mergelt.Game.HUD.HudPanelBase	301
Mergelt.Game.Helpers.HudTargetComponent	302
Mergelt.Editor.LevelEditor.Commands.IActionCommand	303

Mergelt.Editor.LevelEditor.Commands.IActionCommandManager	303
Mergelt.Core.Animations.IAnimationController	304
Mergelt.Core.Animations.IAnimationListener	305
Mergelt.Core.Commands.ICommand	305
Mergelt.Core.Commands.ICommandManager	306
Mergelt.Game.Factories(Icons).IconFactory	307
Mergelt.Core.FieldElements.IConfigParameters	308
Mergelt.Game.Converters.IConfigProcessor	309
Mergelt.Core.Services.IConfigsService	310
Mergelt.Core.Configs(Icons).IconParameters	311
Mergelt.Core.Services.ICurrencyService	312
Mergelt.Game.Effects.Controllers.IEffect	313
Mergelt.Game.Effects.Parameters.IEffectParameters	313
Mergelt.Game.Effects.IEffectsFactory	314
Mergelt.Game.Effects.IEffectsManager	315
Mergelt.Game.Effects.Controllers.IEffectWithPresenter	315
Mergelt.Game.Factories.ElementInfo.IElementInfoFactory	316
Mergelt.Editor.EvolutionsEditor.IElementNode	317
Mergelt.Core.Schemes.Data.IElementNodeData	318
Mergelt.Core.Services.IElementService	319
Mergelt.Core.ElementsStock.IElementsStockComponent	320
Mergelt.Core.ElementsStock.IElementsStockData	320
Mergelt.Core.Services.IElementsStockService	321
Mergelt.Core.Services.IEnergyService	321
Mergelt.Editor.EvolutionsEditor.IEvolutionNode	322
Mergelt.Core.Schemes.Data.IEvolutionNodeData	323
Mergelt.Core.Evolutions.IEvolutionsProgressData	325
Mergelt.Core.Services.IEvolutionsService	326
Mergelt.Game.Field.Actions.IFieldActionProcessor	326
Mergelt.Core.FieldElements.IFieldElement	327
Mergelt.Game.Factories.FieldElement.IFieldElementFactory	328
Mergelt.Core.FieldElements.IFieldElementModel	329
Mergelt.Core.FieldElements.IFieldElementPresenter	330
Mergelt.Core.FieldElements.IFieldElementView	332
Mergelt.Game.Factories.FieldElement.IFieldElementVisualFactory	334
Mergelt.Game.Factories.Field.IFieldFactory	334
Mergelt.Game.Field.IFieldLogic	335
Mergelt.Core.Services.IGameFieldActionsService	335
Mergelt.Core.Services.IGameFieldService	335
Mergelt.Core.Services.IGameLoadService	336
Mergelt.Core.Services.IGameSaveService	337
Mergelt.Core.Services.IGameService	337
Mergelt.Core.FieldElements.IGeneratorOpenParameters	337
Mergelt.Core.FieldElements.IGeneratorParameters	339
Mergelt.Core.Services.IGeneratorsService	341
Mergelt.Game.Managers.IHintsManager	341
Mergelt.Game.HUD.IHudPanel	342
Mergelt.Game.Factories(Icons).IconFactory	342
Mergelt.Core.Services.IInfoPanelService	343
Mergelt.Core.FieldElements.IInfoParameters	343
Mergelt.SimpleDI.ReservedInterfaces.IInitializable	345
Mergelt.Core.Inventory.IInventoryData	345
Mergelt.Game.Factories.Inventory.IInventoryFactory	346
Mergelt.Core.Services.IInventoryService	347
Mergelt.Core.Messages.IMessage	348
Mergelt.Core.Messages.IMessageBus	348
Mergelt.Core.MVP.IModel	350
Mergelt.Core.Helpers.IMonoApplicationQuitHandler	350

Mergelt.Core.Helpers.IMonoUpdateHandler	351
Mergelt.Game.Services.InfoPanelService	351
Mergelt.Core.FieldElements.InfoParameters	352
Mergelt.SimpleDI.IntroduceAttribute	354
Mergelt.Core.Configs.Inventory.InventoryConfig	354
Mergelt.Game.Factories.Inventory.InventoryData	355
Mergelt.Game.Factories.Inventory.InventoryFactory	356
Mergelt.Game.Windows.Inventory.InventoryModel	358
Mergelt.Game.UI.InventoryPanel.InventoryPanelComponent	359
Mergelt.Game.UI.InventoryPanel.InventoryPanelItemComponent	359
Mergelt.Game.UI.InventoryPanel.InventoryPanelItemPair	360
Mergelt.Game.Windows.Inventory.InventoryPresenter	361
Mergelt.Game.Services.InventoryService	363
Mergelt.Game.Services.InventoryServiceModel	366
Mergelt.Game.Windows.Inventory.InventoryWindow	367
Mergelt.Game.Windows.Inventory.InventoryWidgetItemComponent	369
Mergelt.Game.Windows.Inventory.InventoryWidgetItemPair	371
Mergelt.Game.Windows.Inventory.InventoryWindowPaidCellComponent	372
Mergelt.Core.MVP.IPresenter	373
Mergelt.Core.FieldElements.IProducedByParameters	374
Mergelt.Core.FieldElements.IProduceParameters	374
Mergelt.Core.Services.IResourcesLoaderService	375
Mergelt.Core.Saves.ISavable	376
Mergelt.Editor.EvolutionsEditor.ISaveNode	376
Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy	377
Mergelt.SimpleDI.ReservedInterfaces.IUpdatable	378
Mergelt.Core.User.IUserData	378
Mergelt.Game.HUD.IUserListener	380
Mergelt.Core.Services.IUserProgressService	381
Mergelt.Core.Services.IUserService	381
Mergelt.Core.MVP.IView	382
Mergelt.Core.WindowSystem.Windows.IWindow	383
Mergelt.Core.Animations.IWindowAnimationController	385
Mergelt.Core.Animations.IWindowAnimationListener	385
Mergelt.Core.WindowSystem.Data.IWindowArgs	386
Mergelt.Core.WindowSystem.Factory.IWindowFactory	386
Mergelt.Core.WindowSystem.Data.IWindowOpenParameters	387
Mergelt.Core.WindowSystem.Windows.IWindowPresenter	388
Mergelt.Core.WindowSystem.IWindowSystem	390
Mergelt.Game.Services.Saves.Strategies.JsonSerializeStrategy	390
Mergelt.Game.Effects.Controllers.JumpEffect	392
Mergelt.Game.Effects.Configs.JumpEffectConfig	394
Mergelt.Game.Effects.Controllers.JumpEffectWithPresenter	395
Mergelt.Core.Configs.Windows.LayerParameters	398
Mergelt.Core.Configs.Windows.LayersConfig	399
Mergelt.Core.Configs.LevelConfig	400
Mergelt.Editor.LevelEditor.LevelConfigEditor	401
Mergelt.Editor.LevelEditor.LevelEditorWindow	402
Mergelt.Core.Configs.Data.LevelElementData	406
Mergelt.Core.Configs.LevelUp.LevelUpConfig	407
Mergelt.Game.Messages.LevelUpdatedMessage	408
Mergelt.Game.Messages.LevelUpMessage	409
Mergelt.Core.Configs.LevelUp.LevelUpParameters	409
Mergelt.Editor.Core.LevelEditor.Commands.LimitedStack< T >	410
Mergelt.Game.Commands.LoadConfigsCommand	412
Mergelt.Game.Messages.LoadedGameMessage	413
Mergelt.Core.Helpers.MainThreadDispatcher	413
Mergelt.Core.Animations.MecanimController	415

Mergelt.Game.Messages.MergeElementsMessage	417
Mergelt.Core.Messages.MessageBus	417
Mergelt.Core.MVP.Model	420
Mergelt.Core.Helpers.MonoEventsListener	421
Mergelt.Game.Effects.Controllers.MoveEffect	423
Mergelt.Game.Effects.Configs.MoveEffectConfig	425
Mergelt.Game.Effects.Parameters.MoveEffectParameters	426
Mergelt.Game.Effects.Controllers.MoveEffectWithPresenter	428
Mergelt.Game.Messages.MoveElementMessage	431
Mergelt.Game.Effects.Controllers.MoveToUiEffect	432
Mergelt.Game.Effects.Configs.MoveToUiEffectConfig	434
Mergelt.Game.Effects.Controllers.MoveToUiEffectWithPresenter	435
Mergelt.Game.Commands.PrepareEnergyCommand	438
Mergelt.Game.Commands.PrepareFieldCommand	439
Mergelt.Game.Messages.PrepareGameMessage	441
Mergelt.Game.Commands.PrepareInventoryCommand	441
Mergelt.Game.Commands.PrepareStockCommand	442
Mergelt.Game.Commands.PrepareUserCommand	444
Mergelt.Core.MVP.Presenter< TView, TModel >	445
Mergelt.Core.FieldElements.ProducedByParameters	447
Mergelt.Core.FieldElements.ProduceParameters	448
Mergelt.Game.HUD.ProgressComponent	449
Mergelt.Game.Messages.RemoveElementMessage	452
Mergelt.Game.Messages.RemoveFromInventoryMessage	452
Mergelt.Game.Messages.ResetHintsMessage	453
Mergelt.Game.Messages.ResetPositionMessage	453
Mergelt.Game.Messages.ResetSelectionMessage	454
Mergelt.Game.Services.ResourcesLoaderService	454
Mergelt.Core.Utils.ResponsiveGridLayout	455
Mergelt.Core.Saves.SavableAttribute	456
Mergelt.Core.FieldElements.SavedConfigParameters	457
Mergelt.Core.FieldElements.SavedGeneratorOpenParameters	458
Mergelt.Core.FieldElements.SavedGeneratorParameters	458
Mergelt.Core.FieldElements.SavedInfoParameters	460
Mergelt.Editor.Helpers.SavesHelper	461
Mergelt.Core.Schemes.SchemeData	461
Mergelt.Core.Schemes.SchemeObject	462
Mergelt.Editor.EvolutionsEditor.SchemeObjectEditor	463
Mergelt.Game.Messages.SelectedElementMessage	464
Mergelt.Editor.EvolutionsEditor.SerializedElementNodes	465
Mergelt.Game.Messages.SkipTimeMessage	465
Mergelt.Core.Helpers.SlicedFilledImage	466
Mergelt.Editor.Helpers.SlicedFilledImageEditor	471
Mergelt.Game.HUD.SoftCurrencyComponent	472
Mergelt.Game.Messages.SplitElementMessage	474
Mergelt.Game.Messages.StartChargingMessage	474
Mergelt.Game.Messages.StartGameMessage	475
Mergelt.Game.Messages.SwapElementsMessage	475
Mergelt.Game.TopPanel.TopPanelComponent	476
Mergelt.Game.Messages.UnlockElementMessage	477
Mergelt.Game.Messages.UpdateStockMessage	477
Mergelt.Game.User.UserData	478
Mergelt.Game.Windows.UserProgress.UserInfoModel	480
Mergelt.Game.Windows.UserProgress.UserInfoPresenter	480
Mergelt.Game.Windows.UserProgress.UserInfoWindow	482
Mergelt.Game.HUD.UserListenerComponent	486
Mergelt.Game.Windows.UserProgress.UserProgressComponent	487
Mergelt.Game.Services.UserProgressService	487

Mergelt.Game.Services.UserService	489
Mergelt.Game.Services.UserServiceModel	490
Mergelt.Editor.Configs.CurrencyComponent.UxmlFactory	493
Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView.UxmlFactory	493
Mergelt.Core.MVP.View	493
Mergelt.Core.WindowSystem.Data.WindowArgs	495
Mergelt.Core.WindowSystem.Windows.WindowBase	495
Mergelt.Core.WindowSystem.Commands.WindowCloseCommand	500
Mergelt.Core.WindowSystem.Windows.WindowCreateInfo	502
Mergelt.Core.Animations.WindowDefaultAnimationController	504
Mergelt.Core.WindowSystem.Factory.WindowFactory	506
Mergelt.Core.WindowSystem.Commands.WindowHideCommand	507
Mergelt.Core.Animations.WindowMecanimAnimationController	509
Mergelt.Core.WindowSystem.Windows.WindowModel	511
Mergelt.Core.WindowSystem.Commands.WindowOpenCommand	511
Mergelt.Core.WindowSystem.Data.WindowOpenParameters	513
Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >	514
Mergelt.Core.WindowSystem.WindowsSystem	520

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Assets/Mergelt/Editor/Core/Constants.cs	533
Assets/Mergelt/Editor/Core/GeneratedItemsCreator.cs	548
Assets/Mergelt/Editor/Core/GeneratedItemUI.cs	550
Assets/Mergelt/Editor/Core/Configs/StylesConstants.cs	532
Assets/Mergelt/Editor/Core/Configs/Components/CurrencyComponent.cs	525
Assets/Mergelt/Editor/Core/Configs/Components/CurrencyComponentsTraits.cs	525
Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/BaseSettingsDrawer.cs	526
Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/CommonSettingsDrawer.cs	527
Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/ElementConfigInspector.cs	528
Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/GeneratorSettingsBaseDrawer.cs	530
Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/GeneratorSettingsEvolutionDrawer.cs	531
Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/GeneratorSettingsInspectorDrawer.cs	532
Assets/Mergelt/Editor/Core/EvolutionsEditor/EvolutionsEditorWindow.cs	540
Assets/Mergelt/Editor/Core/EvolutionsEditor/SchemeObjectEditor.cs	547
Assets/Mergelt/Editor/Core/EvolutionsEditor/Components/EvolutionsGraphView.cs	533
Assets/Mergelt/Editor/Core/EvolutionsEditor/Components/EvolutionsGraphViewTraits.cs	539
Assets/Mergelt/Editor/Core/EvolutionsEditor/ItemsNodes/BaseNode.cs	542
Assets/Mergelt/Editor/Core/EvolutionsEditor/ItemsNodes/ElementNode.cs	543
Assets/Mergelt/Editor/Core/EvolutionsEditor/ItemsNodes/EvolutionNode.cs	545
Assets/Mergelt/Editor/Core/EvolutionsEditor/ItemsNodes/IElementNode.cs	546
Assets/Mergelt/Editor/Core/EvolutionsEditor/ItemsNodes/IEvolutionNode.cs	547
Assets/Mergelt/Editor/Core/EvolutionsEditor/ItemsNodes/ISaveNode.cs	547
Assets/Mergelt/Editor/Core/EvolutionsEditor/ItemsNodes/CopyPaste/SerializedElementNodes.cs	543
Assets/Mergelt/Editor/Core/LevelEditor/EvolutionSelection.cs	555
Assets/Mergelt/Editor/Core/LevelEditor/LevelConfigEditor.cs	556
Assets/Mergelt/Editor/Core/LevelEditor/LevelEditorWindow.cs	557
Assets/Mergelt/Editor/Core/LevelEditor/Commands/ActionCommandBase.cs	552
Assets/Mergelt/Editor/Core/LevelEditor/Commands/ActionCommandManager.cs	552
Assets/Mergelt/Editor/Core/LevelEditor/Commands/ApplyCellCommand.cs	553
Assets/Mergelt/Editor/Core/LevelEditor/Commands/ChangeCellCommand.cs	553
Assets/Mergelt/Editor/Core/LevelEditor/Commands/ClearCellCommand.cs	554
Assets/Mergelt/Editor/Core/LevelEditor/Commands/IActionCommand.cs	554
Assets/Mergelt/Editor/Core/LevelEditor/Commands/IActionCommandManager.cs	554
Assets/Mergelt/Editor/Core/LevelEditor/Commands/LimitedStack.cs	555
Assets/Mergelt/Editor/Helpers/ConfigsPostprocessor.cs	567

Assets/Mergelt/Editor/Helpers/SavesHelper.cs	596
Assets/Mergelt/Editor/Helpers/SlicedFilledImageEditor.cs	568
Assets/Mergelt/Source/Core/Animations/AnimationControllerBase.cs	569
Assets/Mergelt/Source/Core/Animations/AnimationListener.cs	569
Assets/Mergelt/Source/Core/Animations/AnimationWindowStates.cs	569
Assets/Mergelt/Source/Core/Animations/AnimatorStateMachineBehaviour.cs	569
Assets/Mergelt/Source/Core/Animations/IAnimationController.cs	570
Assets/Mergelt/Source/Core/Animations/IAnimationListener.cs	570
Assets/Mergelt/Source/Core/Animations/IWindowAnimationController.cs	571
Assets/Mergelt/Source/Core/Animations/IWindowAnimationListener.cs	571
Assets/Mergelt/Source/Core/Animations/MecanimController.cs	571
Assets/Mergelt/Source/Core/Animations/WindowDefaultAnimationController.cs	571
Assets/Mergelt/Source/Core/Animations/WindowMecanimAnimationController.cs	572
Assets/Mergelt/Source/Core/Commands/Command.cs	573
Assets/Mergelt/Source/Core/Commands/CommandManager.cs	573
Assets/Mergelt/Source/Core/Commands/ICommand.cs	575
Assets/Mergelt/Source/Core/Commands/ICommandManager.cs	575
Assets/Mergelt/Source/Core/Configs/ConfigsConstants.cs	575
Assets/Mergelt/Source/Core/Configs/GameConfig.cs	582
Assets/Mergelt/Source/Core/Configs/LevelConfig.cs	583
Assets/Mergelt/Source/Core/Configs/Data/EvolutionData.cs	575
Assets/Mergelt/Source/Core/Configs/Data/LevelElementData.cs	576
Assets/Mergelt/Source/Core/Configs/Effects/EffectConfig.cs	577
Assets/Mergelt/Source/Core/Configs/Elements/ElementConfig.cs	577
Assets/Mergelt/Source/Core/Configs/Elements/FieldElementIconComponent.cs	578
Assets/Mergelt/Source/Core/Configs/Elements/Settings/ConsumableSettings.cs	578
Assets/Mergelt/Source/Core/Configs/Elements/Settings/CurrencySettings.cs	579
Assets/Mergelt/Source/Core/Configs/Elements/Settings/ElementCommonSettings.cs	579
Assets/Mergelt/Source/Core/Configs/Elements/Settings/ElementGeneratorSettings.cs	580
Assets/Mergelt/Source/Core/Configs/Elements/Settings/GeneratableElement.cs	582
Assets/Mergelt/Source/Core/Configs/Hints/HintsConfig.cs	582
Assets/Mergelt/Source/Core/Configs/Icons/GameliconsConfig.cs	582
Assets/Mergelt/Source/Core/Configs/Icons/IconParameters.cs	583
Assets/Mergelt/Source/Core/Configs/Inventory/InventoryConfig.cs	583
Assets/Mergelt/Source/Core/Configs/LevelUp/LevelUpConfig.cs	584
Assets/Mergelt/Source/Core/Configs/LevelUp/LevelUpParameters.cs	584
Assets/Mergelt/Source/Core/Configs/Types/ConsumableOperationType.cs	584
Assets/Mergelt/Source/Core/Configs/Types/ConsumableTargetType.cs	584
Assets/Mergelt/Source/Core/Configs/Types/CurrencyType.cs	585
Assets/Mergelt/Source/Core/Configs/Types/ElementType.cs	585
Assets/Mergelt/Source/Core/Configs/Windows/LayerParameters.cs	585
Assets/Mergelt/Source/Core/Configs/Windows/LayersConfig.cs	586
Assets/Mergelt/Source/Core/Configs/Windows/SortingLayers.cs	615
Assets/Mergelt/Source/Core/ElementsStock/IElementsStockComponent.cs	586
Assets/Mergelt/Source/Core/ElementsStock/IElementsStockData.cs	586
Assets/Mergelt/Source/Core/Evolutions/EvolutionProgressData.cs	586
Assets/Mergelt/Source/Core/Evolutions/IEvolutionsProgressData.cs	587
Assets/Mergelt/Source/Core/FieldElements/FieldElement.cs	587
Assets/Mergelt/Source/Core/FieldElements/FieldElementData.cs	587
Assets/Mergelt/Source/Core/FieldElements/GeneratableFieldElement.cs	588
Assets/Mergelt/Source/Core/FieldElements/GridPoint.cs	588
Assets/Mergelt/Source/Core/FieldElements/IFieldElement.cs	589
Assets/Mergelt/Source/Core/FieldElements/IFieldElementModel.cs	589
Assets/Mergelt/Source/Core/FieldElements/IFieldElementPresenter.cs	589
Assets/Mergelt/Source/Core/FieldElements/IFieldElementView.cs	590
Assets/Mergelt/Source/Core/FieldElements/Animations/FieldElementState.cs	587
Assets/Mergelt/Source/Core/FieldElements/Parameters/ConfigParameters.cs	590
Assets/Mergelt/Source/Core/FieldElements/Parameters/GeneratorOpenParameters.cs	590

Assets/Mergelt/Source/Core/FieldElements/Parameters/GeneratorParameters.cs	591
Assets/Mergelt/Source/Core/FieldElements/Parameters/IConfigParameters.cs	591
Assets/Mergelt/Source/Core/FieldElements/Parameters/IGeneratorOpenParameters.cs	591
Assets/Mergelt/Source/Core/FieldElements/Parameters/IGeneratorParameters.cs	591
Assets/Mergelt/Source/Core/FieldElements/Parameters/IInfoParameters.cs	592
Assets/Mergelt/Source/Core/FieldElements/Parameters/InfoParameters.cs	592
Assets/Mergelt/Source/Core/FieldElements/Parameters/IProducedByParameters.cs	592
Assets/Mergelt/Source/Core/FieldElements/Parameters/IProduceParameters.cs	592
Assets/Mergelt/Source/Core/FieldElements/Parameters/ProducedByParameters.cs	593
Assets/Mergelt/Source/Core/FieldElements/Parameters/ProduceParameters.cs	593
Assets/Mergelt/Source/Core/FieldElements/Parameters/Serializable/SavedConfigParameters.cs	593
Assets/Mergelt/Source/Core/FieldElements/Parameters/Serializable/SavedGeneratorOpenParameters.cs	593
Assets/Mergelt/Source/Core/FieldElements/Parameters/Serializable/SavedGeneratorParameters.cs	594
Assets/Mergelt/Source/Core/FieldElements/Parameters/Serializable/SavedInfoParameters.cs	594
Assets/Mergelt/Source/Core/Helpers/CurvesHelper.cs	594
Assets/Mergelt/Source/Core/Helpers/ListExtensions.cs	595
Assets/Mergelt/Source/Core/Helpers/NumericExtensions.cs	595
Assets/Mergelt/Source/Core/Helpers/RectTransformExtensions.cs	596
Assets/Mergelt/Source/Core/Helpers/SavesHelper.cs	597
Assets/Mergelt/Source/Core/Helpers/TimeExtensions.cs	597
Assets/Mergelt/Source/Core/Inventory/IInventoryData.cs	598
Assets/Mergelt/Source/Core/Messages/IMessage.cs	599
Assets/Mergelt/Source/Core/Messages/IMessageBus.cs	599
Assets/Mergelt/Source/Core/Messages/MessageBus.cs	599
Assets/Mergelt/Source/Core/MVP/IModel.cs	600
Assets/Mergelt/Source/Core/MVP/IPresenter.cs	601
Assets/Mergelt/Source/Core/MVP/IView.cs	601
Assets/Mergelt/Source/Core/MVP/Model.cs	601
Assets/Mergelt/Source/Core/MVP/Presenter.cs	601
Assets/Mergelt/Source/Core/MVP/View.cs	602
Assets/Mergelt/Source/Core/Saves/GameSaveType.cs	602
Assets/Mergelt/Source/Core/Saves/ISavable.cs	602
Assets/Mergelt/Source/Core/Saves/SavableAttribute.cs	603
Assets/Mergelt/Source/Core/Schemes/SchemeData.cs	605
Assets/Mergelt/Source/Core/Schemes/SchemeObject.cs	605
Assets/Mergelt/Source/Core/Schemes/Data/EdgeData.cs	603
Assets/Mergelt/Source/Core/Schemes/Data/ElementNodeData.cs	603
Assets/Mergelt/Source/Core/Schemes/Data/EvolutionNodeData.cs	604
Assets/Mergelt/Source/Core/Schemes/Data/IElementNodeData.cs	605
Assets/Mergelt/Source/Core/Schemes/Data/IEvolutionNodeData.cs	605
Assets/Mergelt/Source/Core/Services/IConfigsService.cs	606
Assets/Mergelt/Source/Core/Services/ICurrencyService.cs	606
Assets/Mergelt/Source/Core/Services/IElementService.cs	606
Assets/Mergelt/Source/Core/Services/IElementsStockService.cs	607
Assets/Mergelt/Source/Core/Services/IEnergyService.cs	607
Assets/Mergelt/Source/Core/Services/IEvolutionsService.cs	607
Assets/Mergelt/Source/Core/Services/IGameFieldActionsService.cs	607
Assets/Mergelt/Source/Core/Services/IGameFieldService.cs	608
Assets/Mergelt/Source/Core/Services/IGameLoadService.cs	608
Assets/Mergelt/Source/Core/Services/IGameSaveService.cs	608
Assets/Mergelt/Source/Core/Services/IGameService.cs	608
Assets/Mergelt/Source/Core/Services/IGeneratorsService.cs	608
Assets/Mergelt/Source/Core/Services/IInfoPanelService.cs	609
Assets/Mergelt/Source/Core/Services/IInventoryService.cs	609
Assets/Mergelt/Source/Core/Services/IResourcesLoaderService.cs	609
Assets/Mergelt/Source/Core/Services/IUserProgressService.cs	609
Assets/Mergelt/Source/Core/Services/IUserService.cs	609
Assets/Mergelt/Source/Core/User/IUserData.cs	610

Assets/Mergelt/Source/Core/WindowSystem/BlackoutComponent.cs	610
Assets/Mergelt/Source/Core/WindowSystem/IWindowSystem.cs	614
Assets/Mergelt/Source/Core/WindowSystem/SortingLayers.cs	615
Assets/Mergelt/Source/Core/WindowSystem/WindowName.cs	615
Assets/Mergelt/Source/Core/WindowSystem/WindowsStorage.cs	620
Assets/Mergelt/Source/Core/WindowSystem/WindowsSystem.cs	621
Assets/Mergelt/Source/Core/WindowSystem/Commands/WindowCloseCommand.cs	610
Assets/Mergelt/Source/Core/WindowSystem/Commands/WindowHideCommand.cs	611
Assets/Mergelt/Source/Core/WindowSystem/Commands/WindowOpenCommand.cs	612
Assets/Mergelt/Source/Core/WindowSystem/Data/IWindowArgs.cs	612
Assets/Mergelt/Source/Core/WindowSystem/Data/IWindowOpenParameters.cs	612
Assets/Mergelt/Source/Core/WindowSystem/Data/WindowArgs.cs	613
Assets/Mergelt/Source/Core/WindowSystem/Data/WindowOpenParameters.cs	613
Assets/Mergelt/Source/Core/WindowSystem/Factory/IWindowFactory.cs	613
Assets/Mergelt/Source/Core/WindowSystem/Factory/WindowFactory.cs	613
Assets/Mergelt/Source/Core/WindowSystem/Messages/CloseWindowMessage.cs	614
Assets/Mergelt/Source/Core/WindowSystem/Windows/IWindow.cs	615
Assets/Mergelt/Source/Core/WindowSystem/Windows/IWindowPresenter.cs	616
Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowBase.cs	616
Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowCreateInfo.cs	617
Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowModel.cs	618
Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowPresenter.cs	618
Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowState.cs	620
Assets/Mergelt/Source/Game/GameRoot.cs	680
Assets/Mergelt/Source/Game/Commands/CheckEvolutionsProgressCommand.cs	624
Assets/Mergelt/Source/Game/Commands/CheckOpeningGeneratorsCommand.cs	625
Assets/Mergelt/Source/Game/Commands/EffectCommand.cs	625
Assets/Mergelt/Source/Game/Commands/LoadConfigsCommand.cs	625
Assets/Mergelt/Source/Game/Commands/PrepareEnergyCommand.cs	626
Assets/Mergelt/Source/Game/Commands/PrepareFieldCommand.cs	626
Assets/Mergelt/Source/Game/Commands/PrepareInventoryCommand.cs	627
Assets/Mergelt/Source/Game/Commands/PrepareStockCommand.cs	628
Assets/Mergelt/Source/Game/Commands/PrepareUserCommand.cs	628
Assets/Mergelt/Source/Game/Configs/ConfigsContainer.cs	629
Assets/Mergelt/Source/Game/Configs/ElementPrefab.cs	630
Assets/Mergelt/Source/Game/Configs/Services/ConfigsService.cs	630
Assets/Mergelt/Source/Game/Configs/Services/ConfigsServiceModel.cs	632
Assets/Mergelt/Source/Game/Converters/ConfigProcessor.cs	633
Assets/Mergelt/Source/Game/Converters/IConfigProcessor.cs	637
Assets/Mergelt/Source/Game/Effects/EffectName.cs	644
Assets/Mergelt/Source/Game/Effects/EffectsFactory.cs	644
Assets/Mergelt/Source/Game/Effects/EffectsManager.cs	645
Assets/Mergelt/Source/Game/Effects/IEffectsFactory.cs	645
Assets/Mergelt/Source/Game/Effects/IEffectsManager.cs	646
Assets/Mergelt/Source/Game/Effects/Configs/JumpEffectConfig.cs	637
Assets/Mergelt/Source/Game/Effects/Configs/MoveEffectConfig.cs	637
Assets/Mergelt/Source/Game/Effects/Configs/MoveToUiEffectConfig.cs	637
Assets/Mergelt/Source/Game/Effects/Controllers/BaseEffect.cs	638
Assets/Mergelt/Source/Game/Effects/Controllers/BaseEffectWithPresenter.cs	638
Assets/Mergelt/Source/Game/Effects/Controllers/IEffect.cs	639
Assets/Mergelt/Source/Game/Effects/Controllers/IEffectWithPresenter.cs	639
Assets/Mergelt/Source/Game/Effects/Controllers/JumpEffect.cs	639
Assets/Mergelt/Source/Game/Effects/Controllers/JumpEffectWithPresenter.cs	640
Assets/Mergelt/Source/Game/Effects/Controllers/MoveEffect.cs	641
Assets/Mergelt/Source/Game/Effects/Controllers/MoveEffectWithPresenter.cs	642
Assets/Mergelt/Source/Game/Effects/Controllers/MoveToUiEffect.cs	643
Assets/Mergelt/Source/Game/Effects/Controllers/MoveToUiEffectWithPresenter.cs	643
Assets/Mergelt/Source/Game/Effects/Parameters/IEffectParameters.cs	646

Assets/Mergelt/Source/Game/Effects/Parameters/ MoveEffectParameters.cs	646
Assets/Mergelt/Source/Game/ElementsStock/ ElementsStockComponent.cs	647
Assets/Mergelt/Source/Game/ElementsStock/ ElementsStockData.cs	648
Assets/Mergelt/Source/Game/Enums/ HudPanelType.cs	649
Assets/Mergelt/Source/Game/Evolutions/ EvolutionsProgressData.cs	649
Assets/Mergelt/Source/Game/Factories/ ElementInfoFactory.cs	649
Assets/Mergelt/Source/Game/Factories/ ElementInfoType.cs	650
Assets/Mergelt/Source/Game/Factories/ IElementInfoFactory.cs	650
Assets/Mergelt/Source/Game/Factories/Field/ FieldFactory.cs	650
Assets/Mergelt/Source/Game/Factories/Field/ IFieldFactory.cs	651
Assets/Mergelt/Source/Game/Factories/FieldElement/ FieldElementFactory.cs	651
Assets/Mergelt/Source/Game/Factories/FieldElement/ FieldElementVisualFactory.cs	653
Assets/Mergelt/Source/Game/Factories/FieldElement/ IFieldElementFactory.cs	654
Assets/Mergelt/Source/Game/Factories/FieldElement/ IFieldElementVisualFactory.cs	654
Assets/Mergelt/Source/Game/Factories/Icons/ IconFactory.cs	654
Assets/Mergelt/Source/Game/Factories/Icons/ IIIconFactory.cs	655
Assets/Mergelt/Source/Game/Factories/Inventory/ IInventoryFactory.cs	655
Assets/Mergelt/Source/Game/Factories/Inventory/ InventoryData.cs	656
Assets/Mergelt/Source/Game/Factories/Inventory/ InventoryFactory.cs	656
Assets/Mergelt/Source/Game/Field/ FieldCellComponent.cs	672
Assets/Mergelt/Source/Game/Field/ FieldData.cs	674
Assets/Mergelt/Source/Game/Field/ FieldLogic.cs	675
Assets/Mergelt/Source/Game/Field/ FieldLogicModel.cs	676
Assets/Mergelt/Source/Game/Field/ FieldModel.cs	676
Assets/Mergelt/Source/Game/Field/ FieldPresenter.cs	676
Assets/Mergelt/Source/Game/Field/ FieldView.cs	680
Assets/Mergelt/Source/Game/Field/ IFieldLogic.cs	680
Assets/Mergelt/Source/Game/Field/Actions/ FieldActionProcessorBase.cs	657
Assets/Mergelt/Source/Game/Field/Actions/ FieldConsumableProcessor.cs	658
Assets/Mergelt/Source/Game/Field/Actions/ FieldGenerationProcessor.cs	659
Assets/Mergelt/Source/Game/Field/Actions/ FieldMergeProcessor.cs	661
Assets/Mergelt/Source/Game/Field/Actions/ IFieldActionProcessor.cs	664
Assets/Mergelt/Source/Game/Field/Elements/ FieldElementModel.cs	665
Assets/Mergelt/Source/Game/Field/Elements/ FieldElementPresenter.cs	666
Assets/Mergelt/Source/Game/Field/Elements/ FieldElementPresenterBase.cs	666
Assets/Mergelt/Source/Game/Field/Elements/ FieldElementView.cs	668
Assets/Mergelt/Source/Game/Field/Elements/Animations/ FieldElementAnimationController.cs	664
Assets/Mergelt/Source/Game/Field/Elements/Animations/ FieldElementStateMachineBehaviour.cs	665
Assets/Mergelt/Source/Game/Field/Elements/Generator/ FieldElementGeneratorModel.cs	669
Assets/Mergelt/Source/Game/Field/Elements/Generator/ FieldElementGeneratorPresenter.cs	670
Assets/Mergelt/Source/Game/Field/Elements/Generator/ FieldElementGeneratorView.cs	671
Assets/Mergelt/Source/Game/Field/Elements/Generator/ GeneratorTimerComponent.cs	671
Assets/Mergelt/Source/Game/Helpers/ FieldElementExtensions.cs	681
Assets/Mergelt/Source/Game/Helpers/ FieldGeneratorElementExtensions.cs	682
Assets/Mergelt/Source/Game/Helpers/ HudTargetComponent.cs	686
Assets/Mergelt/Source/Game/Helpers/ HudTargets.cs	686
Assets/Mergelt/Source/Game/HUD/ EnergyComponent.cs	686
Assets/Mergelt/Source/Game/HUD/ HardCurrencyComponent.cs	688
Assets/Mergelt/Source/Game/HUD/ HudComponent.cs	688
Assets/Mergelt/Source/Game/HUD/ HudPanelBase.cs	689
Assets/Mergelt/Source/Game/HUD/ IHudPanel.cs	689
Assets/Mergelt/Source/Game/HUD/ IUserListener.cs	689
Assets/Mergelt/Source/Game/HUD/ ProgressComponent.cs	690
Assets/Mergelt/Source/Game/HUD/ SoftCurrencyComponent.cs	691
Assets/Mergelt/Source/Game/HUD/ UserListenerComponent.cs	692
Assets/Mergelt/Source/Game/Managers/ HintsManager.cs	692
Assets/Mergelt/Source/Game/Managers/ IHintsManager.cs	694
Assets/Mergelt/Source/Game/Messages/ ActivateHintsMessage.cs	694

Assets/Mergelt/Source/Game/Messages/CheckGeneratorMessage.cs	694
Assets/Mergelt/Source/Game/Messages/ClickElementMessage.cs	695
Assets/Mergelt/Source/Game/Messages/CreateElementMessage.cs	695
Assets/Mergelt/Source/Game/Messages/ElementActionMessage.cs	695
Assets/Mergelt/Source/Game/Messages/EndDragElementMessage.cs	695
Assets/Mergelt/Source/Game/Messages/EnergyRestoredMessage.cs	696
Assets/Mergelt/Source/Game/Messages/FieldFullMessage.cs	696
Assets/Mergelt/Source/Game/Messages/GeneratorBusyMessage.cs	696
Assets/Mergelt/Source/Game/Messages/GeneratorOpenedMessage.cs	696
Assets/Mergelt/Source/Game/Messages/GeneratorOpenStartMessage.cs	696
Assets/Mergelt/Source/Game/Messages/GeneratorRestoredMessage.cs	697
Assets/Mergelt/Source/Game/Messages/LevelUpdatedMessage.cs	697
Assets/Mergelt/Source/Game/Messages/LevelUpMessage.cs	697
Assets/Mergelt/Source/Game/Messages/LoadedGameMessage.cs	697
Assets/Mergelt/Source/Game/Messages/MergeElementsMessage.cs	697
Assets/Mergelt/Source/Game/Messages/MoveElementMessage.cs	698
Assets/Mergelt/Source/Game/Messages/PrepareGameMessage.cs	698
Assets/Mergelt/Source/Game/Messages/RemoveElementMessage.cs	698
Assets/Mergelt/Source/Game/Messages/RemoveFromInventoryMessage.cs	698
Assets/Mergelt/Source/Game/Messages/ResetHintsMessage.cs	698
Assets/Mergelt/Source/Game/Messages/ResetPositionMessage.cs	699
Assets/Mergelt/Source/Game/Messages/ResetSelectionMessage.cs	699
Assets/Mergelt/Source/Game/Messages/SelectedElementMessage.cs	699
Assets/Mergelt/Source/Game/Messages/SkipTimeMessage.cs	699
Assets/Mergelt/Source/Game/Messages/SplitElementMessage.cs	699
Assets/Mergelt/Source/Game/Messages/StartChargingMessage.cs	700
Assets/Mergelt/Source/Game/Messages/StartGameMessage.cs	700
Assets/Mergelt/Source/Game/Messages/SwapElementsMessage.cs	700
Assets/Mergelt/Source/Game/Messages/UnlockElementMessage.cs	700
Assets/Mergelt/Source/Game/Messages/UpdateStockMessage.cs	700
Assets/Mergelt/Source/Game/Services/CurrencyService.cs	701
Assets/Mergelt/Source/Game/Services/ElementService.cs	702
Assets/Mergelt/Source/Game/Services/ElementsStockService.cs	703
Assets/Mergelt/Source/Game/Services/ElementsStockServiceModel.cs	704
Assets/Mergelt/Source/Game/Services/EnergyService.cs	704
Assets/Mergelt/Source/Game/Services/EvolutionsService.cs	706
Assets/Mergelt/Source/Game/Services/EvolutionsServiceModel.cs	708
Assets/Mergelt/Source/Game/Services/GameFieldActionsService.cs	708
Assets/Mergelt/Source/Game/Services/GameFieldService.cs	709
Assets/Mergelt/Source/Game/Services/GameService.cs	710
Assets/Mergelt/Source/Game/Services/GameServiceModel.cs	711
Assets/Mergelt/Source/Game/Services/GeneratorService.cs	711
Assets/Mergelt/Source/Game/Services/InfoPanelService.cs	714
Assets/Mergelt/Source/Game/Services/InventoryService.cs	715
Assets/Mergelt/Source/Game/Services/InventoryServiceModel.cs	717
Assets/Mergelt/Source/Game/Services/ResourcesLoaderService.cs	717
Assets/Mergelt/Source/Game/Services/UserProgressService.cs	725
Assets/Mergelt/Source/Game/Services/UserService.cs	727
Assets/Mergelt/Source/Game/Services/UserServiceModel.cs	727
Assets/Mergelt/Source/Game/Services/Saves/GameLoadService.cs	717
Assets/Mergelt/Source/Game/Services/Saves/GameSaveEveryIntervalService.cs	718
Assets/Mergelt/Source/Game/Services/Saves/GameSaveOnEveryActionService.cs	720
Assets/Mergelt/Source/Game/Services/Saves/GameSaveOnQuitService.cs	722
Assets/Mergelt/Source/Game/Services/Saves/Strategies/BaseSerializeStrategy.cs	723
Assets/Mergelt/Source/Game/Services/Saves/Strategies/BinaryFormatterStrategy.cs	724
Assets/Mergelt/Source/Game/Services/Saves/Strategies/ISerializeStrategy.cs	724
Assets/Mergelt/Source/Game/Services/Saves/Strategies/JsonSerializeStrategy.cs	725
Assets/Mergelt/Source/Game/TopPanel/TopPanelComponent.cs	728

Assets/Mergelt/Source/Game/UI/InfoPanel/ElementActionType.cs	728
Assets/Mergelt/Source/Game/UI/InfoPanel/ElementCommonInfo.cs	729
Assets/Mergelt/Source/Game/UI/InfoPanel/ElementInfo.cs	729
Assets/Mergelt/Source/Game/UI/InfoPanel/ElementInfoPanelComponent.cs	730
Assets/Mergelt/Source/Game/UI/InfoPanel/ElementOpenInfo.cs	732
Assets/Mergelt/Source/Game/UI/InfoPanel/ElementSellInfo.cs	733
Assets/Mergelt/Source/Game/UI/InfoPanel/ElementSpeedUpChargingInfo.cs	734
Assets/Mergelt/Source/Game/UI/InfoPanel/ElementSpeedUpOpeningInfo.cs	735
Assets/Mergelt/Source/Game/UI/InfoPanel/ElementSplitInfo.cs	736
Assets/Mergelt/Source/Game/UI/InfoPanel/ElementUnlockInfo.cs	737
Assets/Mergelt/Source/Game/UI/InventoryPanel/InventoryPanelComponent.cs	738
Assets/Mergelt/Source/Game/UI/InventoryPanel/InventoryPanelItemComponent.cs	740
Assets/Mergelt/Source/Game/UI/InventoryPanel/InventoryPanelItemPair.cs	740
Assets/Mergelt/Source/Game/User/UserData.cs	741
Assets/Mergelt/Source/Game/Windows/ElementInfo/ElementInfoArgs.cs	741
Assets/Mergelt/Source/Game/Windows/ElementInfo/ElementInfoItemComponent.cs	741
Assets/Mergelt/Source/Game/Windows/ElementInfo/ElementInfoItemsPanelComponent.cs	743
Assets/Mergelt/Source/Game/Windows/ElementInfo/ElementInfoModel.cs	743
Assets/Mergelt/Source/Game/Windows/ElementInfo/ElementInfoPresenter.cs	744
Assets/Mergelt/Source/Game/Windows/ElementInfo/ElementInfoWindow.cs	746
Assets/Mergelt/Source/Game/Windows/ElementInfo/ElementInfoWindowContent.cs	747
Assets/Mergelt/Source/Game/Windows/Inventory/InventoryModel.cs	747
Assets/Mergelt/Source/Game/Windows/Inventory/InventoryPresenter.cs	747
Assets/Mergelt/Source/Game/Windows/Inventory/InventoryWindow.cs	750
Assets/Mergelt/Source/Game/Windows/Inventory/InventoryWindowItemComponent.cs	750
Assets/Mergelt/Source/Game/Windows/Inventory/InventoryWindowItemPair.cs	751
Assets/Mergelt/Source/Game/Windows/Inventory/InventoryWindowPaidCellComponent.cs	751
Assets/Mergelt/Source/Game/Windows/UserProgress/UserInfoModel.cs	752
Assets/Mergelt/Source/Game/Windows/UserProgress/UserInfoPresenter.cs	752
Assets/Mergelt/Source/Game/Windows/UserProgress/UserInfoWindow.cs	754
Assets/Mergelt/Source/Game/Windows/UserProgress/UserProgressComponent.cs	755
Assets/Mergelt/Source/SimpleDI/DiContainer.cs	755
Assets/Mergelt/Source/SimpleDI/DiHandler.cs	757
Assets/Mergelt/Source/SimpleDI/DiRoot.cs	758
Assets/Mergelt/Source/SimpleDI/IntroduceAttribute.cs	759
Assets/Mergelt/Source/SimpleDI/ReservedInterfaces/IInitializable.cs	759
Assets/Mergelt/Source/SimpleDI/ReservedInterfaces/IUpdatable.cs	759
Assets/Mergelt/Source/Utils/Bindable.cs	760
Assets/Mergelt/Source/Utils/IMonoApplicationQuitHandler.cs	761
Assets/Mergelt/Source/Utils/IMonoUpdateHandler.cs	761
Assets/Mergelt/Source/Utils/MainThreadDispatcher.cs	761
Assets/Mergelt/Source/Utils/MonoEventsListener.cs	762
Assets/Mergelt/Source/Utils/ResponsiveGridLayout.cs	763
Assets/Mergelt/Source/Utils/SlicedFilledImage.cs	764

Chapter 5

Namespace Documentation

5.1 Mergelt Namespace Reference

5.2 Mergelt.Core Namespace Reference

5.3 Mergelt.Core.Animations Namespace Reference

Classes

- class [AnimationControllerBase](#)
- class [AnimationListener](#)
- class [AnimationWindowStates](#)
- class [AnimatorStateMachineBehaviour](#)
- interface [IAnimationController](#)
- interface [IAnimationListener](#)
- interface [IWindowAnimationController](#)
- interface [IWindowAnimationListener](#)
- class [MecanimController](#)
- class [WindowDefaultAnimationController](#)
- class [WindowMecanimAnimationController](#)

5.4 Mergelt.Core.Commands Namespace Reference

Classes

- class [Command](#)
- class [CommandManager](#)
- interface [ICommand](#)
- interface [ICommandManager](#)

5.5 Mergelt.Core.Configs Namespace Reference

Classes

- class [ConfigsConstants](#)
- class [GameConfig](#)
- class [LevelConfig](#)

5.6 Mergelt.Core.Configs.Data Namespace Reference

Classes

- class [EvolutionData](#)
- class [LevelElementData](#)

5.7 Mergelt.Core.Configs.Effects Namespace Reference

Classes

- class [EffectConfig](#)

5.8 Mergelt.Core.Configs.Elements Namespace Reference

Classes

- struct [ConsumableSettings](#)
- struct [CurrencySettings](#)
- class [ElementCommonSettings](#)
- class [ElementConfig](#)
- class [ElementGeneratorSettings](#)
- class [FieldElementIconComponent](#)
- class [GeneratableElement](#)

5.9 Mergelt.Core.Configs.Hints Namespace Reference

Classes

- class [HintsConfig](#)

5.10 Mergelt.Core.Configs.Icons Namespace Reference

Classes

- class [GameIconsConfig](#)
- class [IconParameters](#)

5.11 Mergelt.Core.Configs.Inventory Namespace Reference

Classes

- class [InventoryConfig](#)

5.12 Mergelt.Core.Configs.LevelUp Namespace Reference

Classes

- class [LevelUpConfig](#)
- class [LevelUpParameters](#)

5.13 Mergelt.Core.Configs.Types Namespace Reference

Enumerations

- enum [ConsumableOperationType](#) { [Multiply](#) , [Divide](#) , [Add](#) , [Subtract](#) }
- enum [ConsumableTargetType](#) {
 [SoftCurrency](#) , [HardCurrency](#) , [Time](#) , [Energy](#) ,
 [Experience](#) , [Splitter](#) }
- enum [CurrencyType](#) { [Soft](#) , [Hard](#) , [Splitter](#) , [Energy](#) }
- enum [ElementType](#) { [Regular](#) , [Generator](#) }

5.13.1 Enumeration Type Documentation

5.13.1.1 ConsumableOperationType

```
enum MergeIt.Core.Configs.Types.ConsumableOperationType
```

Definition at line 5 of file [ConsumableOperationType.cs](#).

```
00006  {
00007      Multiply,
00008      Divide,
00009      Add,
00010      Subtract,
00011  }
```

5.13.1.2 ConsumableTargetType

```
enum MergeIt.Core.Configs.Types.ConsumableTargetType
```

Definition at line 5 of file [ConsumableTargetType.cs](#).

```
00006  {
00007      SoftCurrency,
00008      HardCurrency,
00009      Time,
00010      Energy,
00011      Experience,
00012      Splitter
00013  }
```

5.13.1.3 CurrencyType

```
enum MergeIt.Core.Configs.Types.CurrencyType
```

Definition at line 5 of file [CurrencyType.cs](#).

```
00006  {
00007      Soft,
00008      Hard,
00009      Splitter,
00010      Energy
00011 }
```

5.13.1.4 ElementType

```
enum MergeIt.Core.Configs.Types.ElementType
```

Definition at line 5 of file [ElementType.cs](#).

```
00006  {
00007      Regular,
00008      Generator
00009 }
```

5.14 Mergelt.Core.Configs.Windows Namespace Reference

Classes

- class [LayerParameters](#)
- class [LayersConfig](#)
- class [SortingLayers](#)

5.15 Mergelt.Core.ElementsStock Namespace Reference

Classes

- interface [IElementsStockComponent](#)
- interface [IElementsStockData](#)

5.16 Mergelt.Core.Evolutions Namespace Reference

Classes

- class [EvolutionProgressData](#)
- interface [IEvolutionsProgressData](#)

5.17 MergeIt.Core.FieldElements Namespace Reference

Classes

- class ConfigParameters
- class FieldElement
- class FieldElementData
- class GeneratableFieldElement
- class GeneratorOpenParameters
- class GeneratorParameters
- struct GridPoint
- interface IConfigParameters
- interface IFieldElement
- interface IFieldElementModel
- interface IFieldElementPresenter
- interface IFieldElementView
- interface IGeneratorOpenParameters
- interface IGeneratorParameters
- interface IInfoParameters
- class InfoParameters
- interface IProducedByParameters
- interface IProduceParameters
- class ProducedByParameters
- class ProduceParameters
- class SavedConfigParameters
- class SavedGeneratorOpenParameters
- class SavedGeneratorParameters
- class SavedInfoParameters

Enumerations

- enum FieldElementState { Idle , Hint }

5.17.1 Enumeration Type Documentation

5.17.1.1 FieldElementState

```
enum MergeIt.Core.FieldElements.FieldElementState
```

Definition at line 5 of file FieldElementState.cs.

```
00006      {
00007          Idle,
00008          Hint
00009      }
```

5.18 Mergelt.Core.Helpers Namespace Reference

Classes

- class **CurvesHelper**
- interface [IMonoApplicationQuitHandler](#)
- interface [IMonoUpdateHandler](#)
- class **ListExtensions**
- class [MainThreadDispatcher](#)
- class [MonoEventsListener](#)
- class **NumericExtensions**
- class **RectTransformExtensions**
- class **SavesHelper**
- class [SlicedFilledImage](#)
- class **TimeExtensions**

5.19 Mergelt.Core.Inventory Namespace Reference

Classes

- interface [IInventoryData](#)

5.20 Mergelt.Core.Messages Namespace Reference

Classes

- interface [IMessage](#)
- interface [IMessageBus](#)
- class [MessageBus](#)

5.21 Mergelt.Core.MVP Namespace Reference

Classes

- interface [IModel](#)
- interface [IPresenter](#)
- interface [IView](#)
- class [Model](#)
- class [Presenter](#)
- class [View](#)

5.22 Mergelt.Core.Saves Namespace Reference

Classes

- interface [ISavable](#)
- class [SavableAttribute](#)

Enumerations

- enum **GameSaveType** {
 Field = 1 << 0 , **Inventory** = 1 << 1 , **User** = 1 << 2 , **Stock** = 1 << 3 ,
 EvolutionsProgress = 1 << 4 , **All** = **Field** | **Inventory** | **User** | **Stock** | **EvolutionsProgress** }

5.22.1 Enumeration Type Documentation

5.22.1.1 GameSaveType

```
enum MergeIt.Core.Saves.GameSaveType
```

Definition at line 8 of file [GameSaveType.cs](#).

```
00009     {  
00010         Field = 1 « 0,  
00011         Inventory = 1 « 1,  
00012         User = 1 « 2,  
00013         Stock = 1 « 3,  
00014         EvolutionsProgress = 1 « 4,  
00015         All = Field | Inventory | User | Stock | EvolutionsProgress  
00016     }
```

5.23 Mergelt.Core.Schemes Namespace Reference

Classes

- class [SchemeData](#)
- class [SchemeObject](#)

5.24 Mergelt.Core.Schemes.Data Namespace Reference

Classes

- class [EdgeData](#)
- class [ElementNodeData](#)
- class [EvolutionNodeData](#)
- interface [IElementNodeData](#)
- interface [IEvolutionNodeData](#)

5.25 Mergelt.Core.Services Namespace Reference

Classes

- interface [IConfigsService](#)
- interface [ICurrencyService](#)
- interface [IElementService](#)
- interface [IElementsStockService](#)
- interface [IEnergyService](#)
- interface [IEvolutionsService](#)
- interface [IGameFieldActionsService](#)

- interface [IGameFieldService](#)
- interface [IGameLoadService](#)
- interface [IGameSaveService](#)
- interface [IGameService](#)
- interface [IGeneratorsService](#)
- interface [IInfoPanelService](#)
- interface [IInventoryService](#)
- interface [IResourcesLoaderService](#)
- interface [IUserProgressService](#)
- interface [IUserService](#)

5.26 Mergelt.Core.User Namespace Reference

Classes

- interface [IUserData](#)

5.27 Mergelt.Core.Utils Namespace Reference

Classes

- class [Bindable](#)
- class [ResponsiveGridLayout](#)

5.28 Mergelt.Core.WindowSystem Namespace Reference

Classes

- class [BlackoutComponent](#)
- interface [IWindowSystem](#)
- class [SortingLayers](#)
- class [WindowName](#)
- class [WindowsStorage](#)
- class [WindowsSystem](#)

5.29 Mergelt.Core.WindowSystem.Commands Namespace Reference

Classes

- class [WindowCloseCommand](#)
- class [WindowHideCommand](#)
- class [WindowOpenCommand](#)

5.30 Mergelt.Core.WindowSystem.Data Namespace Reference

Classes

- interface [IWindowArgs](#)
- interface [IWindowOpenParameters](#)
- class [WindowArgs](#)
- class [WindowOpenParameters](#)

5.31 Mergelt.Core.WindowSystem.Factory Namespace Reference

Classes

- interface [IWindowFactory](#)
- class [WindowFactory](#)

5.32 Mergelt.Core.WindowSystem.Messages Namespace Reference

Classes

- class [CloseWindowMessage](#)

5.33 Mergelt.Core.WindowSystem.Windows Namespace Reference

Classes

- interface [IWindow](#)
- interface [IWindowPresenter](#)
- class [WindowBase](#)
- struct [WindowCreateInfo](#)
- class [WindowModel](#)
- class [WindowPresenter](#)

Enumerations

- enum [WindowState](#) {
 [None](#) , [Opening](#) , [Opened](#) , [Hiding](#) ,
 [Hidden](#) , [Closing](#) , [Closed](#) }

5.33.1 Enumeration Type Documentation

5.33.1.1 [WindowState](#)

```
enum MergeIt.Core.WindowSystem.Windows.WindowState
```

Definition at line 5 of file [WindowState.cs](#).

```
00006      {
00007          None,
00008          Opening,
00009          Opened,
00010          Hiding,
00011          Hidden,
00012          Closing,
00013          Closed,
00014      }
```

5.34 Mergelt.Editor Namespace Reference

Classes

- class **Constants**
- class [GeneratedItemsCreator](#)
- class [GeneratedItemUI](#)

5.35 Mergelt.Editor.Configs Namespace Reference

Classes

- class [CurrencyComponent](#)
- class [CurrencyComponentsTraits](#)
- class **StylesConstants**

5.36 Mergelt.Editor.Configs.ItemsDrawers Namespace Reference

Classes

- class [BaseSettingsDrawer](#)
- class [CommonSettingsDrawer](#)
- class [ElementConfigInspector](#)
- class [GeneratorSettingsBaseDrawer](#)
- class [GeneratorSettingsEvolutionDrawer](#)
- class [GeneratorSettingsInspectorDrawer](#)

5.37 Mergelt.Editor.Core Namespace Reference

5.38 Mergelt.Editor.Core.LevelEditor Namespace Reference

5.39 Mergelt.Editor.Core.LevelEditor.Commands Namespace Reference

Classes

- class [LimitedStack](#)

5.40 Mergelt.Editor.EvolutionsEditor Namespace Reference

Classes

- class [BaseNode](#)
- class [ElementNode](#)
- class [EvolutionNode](#)
- class [EvolutionsEditorWindow](#)
- class [EvolutionsGraphView](#)
- class [EvolutionsGraphViewTraits](#)
- interface [IElementNode](#)
- interface [IEvolutionNode](#)
- interface [ISaveNode](#)
- class [SchemeObjectEditor](#)
- class [SerializedElementNodes](#)

5.41 Mergelt.Editor.Helpers Namespace Reference

Classes

- class [ConfigsPostprocessor](#)
- class [SavesHelper](#)
- class [SlicedFilledImageEditor](#)

5.42 Mergelt.Editor.LevelEditor Namespace Reference

Classes

- class [EvolutionSelection](#)
- class [LevelConfigEditor](#)
- class [LevelEditorWindow](#)

5.43 Mergelt.Editor.LevelEditor.Commands Namespace Reference

Classes

- class [ActionCommandBase](#)
- class [ActionCommandManager](#)
- class [ApplyCellCommand](#)
- class [ChangeCellCommand](#)
- class [ClearCellCommand](#)
- interface [IActionCommand](#)
- interface [IActionCommandManager](#)

5.44 Mergelt.Game Namespace Reference

Classes

- class [GameRoot](#)

5.45 Mergelt.Game.Commands Namespace Reference

Classes

- class [CheckEvolutionsProgressCommand](#)
- class [CheckOpeningGeneratorsCommand](#)
- class [EffectCommand](#)
- class [LoadConfigsCommand](#)
- class [PrepareEnergyCommand](#)
- class [PrepareFieldCommand](#)
- class [PrepareInventoryCommand](#)
- class [PrepareStockCommand](#)
- class [PrepareUserCommand](#)

5.46 Mergelt.Game.Configs Namespace Reference

Classes

- class [ConfigsContainer](#)
- class [ElementPrefab](#)

5.47 Mergelt.Game.Configs.Services Namespace Reference

Classes

- class [ConfigsService](#)
- class [ConfigsServiceModel](#)

5.48 Mergelt.Game.Converters Namespace Reference

Classes

- class [ConfigProcessor](#)
- interface [IConfigProcessor](#)

5.49 Mergelt.Game.Effects Namespace Reference

Classes

- class [EffectName](#)
- class [EffectsFactory](#)
- class [EffectsManager](#)
- interface [IEffectsFactory](#)
- interface [IEffectsManager](#)

5.50 Mergelt.Game.Effects.Configs Namespace Reference

Classes

- class [JumpEffectConfig](#)
- class [MoveEffectConfig](#)
- class [MoveToUiEffectConfig](#)

5.51 Mergelt.Game.Effects.Controllers Namespace Reference

Classes

- class [BaseEffect](#)
- class [BaseEffectWithPresenter](#)
- interface [IEffect](#)
- interface [IEffectWithPresenter](#)
- class [JumpEffect](#)
- class [JumpEffectWithPresenter](#)
- class [MoveEffect](#)
- class [MoveEffectWithPresenter](#)
- class [MoveToUiEffect](#)
- class [MoveToUiEffectWithPresenter](#)

5.52 Mergelt.Game.Effects.Parameters Namespace Reference

Classes

- interface [IEffectParameters](#)
- class [MoveEffectParameters](#)

5.53 Mergelt.Game.ElementsStock Namespace Reference

Classes

- class [ElementsStockComponent](#)
- class [ElementsStockData](#)

5.54 Mergelt.Game.Enums Namespace Reference

Enumerations

- enum HudPanelType {
 None , UserProgress , SoftCurrency , HardCurrency ,
 Energy , ElementStock }

5.54.1 Enumeration Type Documentation

5.54.1.1 HudPanelType

```
enum MergeIt.Game.Enums.HudPanelType
```

Definition at line 5 of file [HudPanelType.cs](#).

```
00006    {  
00007        None,  
00008        UserProgress,  
00009        SoftCurrency,  
00010        HardCurrency,  
00011        Energy,  
00012        ElementStock  
00013    }
```

5.55 Mergelt.Game.Evolutions Namespace Reference

Classes

- class [EvolutionsProgressData](#)

5.56 Mergelt.Game.Factories Namespace Reference

5.57 Mergelt.Game.Factories.ElementInfo Namespace Reference

Classes

- class [ElementInfoFactory](#)
- interface [IElementInfoFactory](#)

Enumerations

- enum ElementInfoType { [InfoWindow](#) , [UserProgressWindow](#) }

5.57.1 Enumeration Type Documentation

5.57.1.1 ElementInfoType

```
enum MergeIt.Game.Factories.ElementInfo.ElementInfoType
```

Definition at line 5 of file [ElementInfoType.cs](#).

```
00006     {  
00007         InfoWindow,  
00008         UserProgressWindow  
00009     }
```

5.58 Mergelt.Game.Factories.Field Namespace Reference

Classes

- class [FieldFactory](#)
- interface [IFieldFactory](#)

5.59 Mergelt.Game.Factories.FieldElement Namespace Reference

Classes

- class [FieldElementFactory](#)
- class [FieldElementVisualFactory](#)
- interface [IFieldElementFactory](#)
- interface [IFieldElementVisualFactory](#)

5.60 Mergelt.Game.Factories.Icons Namespace Reference

Classes

- class [IconFactory](#)
- interface [IIconFactory](#)

5.61 Mergelt.Game.Factories.Inventory Namespace Reference

Classes

- interface [IInventoryFactory](#)
- class [InventoryData](#)
- class [InventoryFactory](#)

5.62 Mergelt.Game.Field Namespace Reference

Classes

- class [FieldCellComponent](#)
- class [FieldData](#)
- class [FieldLogic](#)
- class [FieldLogicModel](#)
- class [FieldModel](#)
- class [FieldPresenter](#)
- class [FieldView](#)
- interface [IFieldLogic](#)

5.63 Mergelt.Game.Field.Actions Namespace Reference

Classes

- class [FieldActionProcessorBase](#)
- class [FieldConsumableProcessor](#)
- class [FieldGenerationProcessor](#)
- class [FieldMergeProcessor](#)
- interface [IFieldActionProcessor](#)

5.64 Mergelt.Game.Field.Elements Namespace Reference

Classes

- class [FieldElementModel](#)
- class [FieldElementPresenter](#)
- class [FieldElementPresenterBase](#)
- class [FieldElementView](#)

5.65 Mergelt.Game.Field.Elements.Animations Namespace Reference

Classes

- class [FieldElementAnimationController](#)
- class [FieldElementStateMachineBehaviour](#)

5.66 Mergelt.Game.Field.Elements.Generator Namespace Reference

Classes

- class [FieldElementGeneratorModel](#)
- class [FieldElementGeneratorPresenter](#)
- class [FieldElementGeneratorView](#)
- class [GeneratorTimerComponent](#)

5.67 Mergelt.Game.Helpers Namespace Reference

Classes

- class [FieldElementExtensions](#)
- class [FieldGeneratorElementExtensions](#)
- class [HudTargetComponent](#)
- class [HudTargets](#)

5.68 Mergelt.Game.HUD Namespace Reference

Classes

- class [EnergyComponent](#)
- class [HardCurrencyComponent](#)
- class [HudComponent](#)
- class [HudPanelBase](#)
- interface [IHudPanel](#)
- interface [IUserListener](#)
- class [ProgressComponent](#)
- class [SoftCurrencyComponent](#)
- class [UserListenerComponent](#)

5.69 Mergelt.Game.Managers Namespace Reference

Classes

- class [HintsManager](#)
- interface [IHintsManager](#)

5.70 Mergelt.Game.Messages Namespace Reference

Classes

- class [ActivateHintsMessage](#)
- class [CheckGeneratorMessage](#)
- class [ClickElementMessage](#)
- class [CreateElementMessage](#)
- class [ElementActionMessage](#)
- class [EndDragElementMessage](#)
- class [EnergyRestoredMessage](#)
- class [FieldFullMessage](#)
- class [GeneratorBusyMessage](#)
- class [GeneratorOpenedMessage](#)
- class [GeneratorOpenStartMessage](#)
- class [GeneratorRestoredMessage](#)
- class [LevelUpdatedMessage](#)
- class [LevelUpMessage](#)

- class [LoadedGameMessage](#)
- class [MergeElementsMessage](#)
- class [MoveElementMessage](#)
- class [PrepareGameMessage](#)
- class [RemoveElementMessage](#)
- class [RemoveFromInventoryMessage](#)
- class [ResetHintsMessage](#)
- class [ResetPositionMessage](#)
- class [ResetSelectionMessage](#)
- class [SelectedElementMessage](#)
- class [SkipTimeMessage](#)
- class [SplitElementMessage](#)
- class [StartChargingMessage](#)
- class [StartGameMessage](#)
- class [SwapElementsMessage](#)
- class [UnlockElementMessage](#)
- class [UpdateStockMessage](#)

5.71 Mergelt.Game.Services Namespace Reference

Classes

- class [CurrencyService](#)
- class [ElementService](#)
- class [ElementsStockService](#)
- class [ElementsStockServiceModel](#)
- class [EnergyService](#)
- class [EvolutionsService](#)
- class [EvolutionsServiceModel](#)
- class [GameFieldActionsService](#)
- class [GameFieldService](#)
- class [GameService](#)
- class [GameServiceModel](#)
- class [GeneratorService](#)
- class [InfoPanelService](#)
- class [InventoryService](#)
- class [InventoryServiceModel](#)
- class [ResourcesLoaderService](#)
- class [UserProgressService](#)
- class [UserService](#)
- class [UserServiceModel](#)

5.72 Mergelt.Game.Services.Saves Namespace Reference

Classes

- class [GameLoadService](#)
- class [GameSaveEveryIntervalService](#)
- class [GameSaveOnEveryActionService](#)
- class [GameSaveOnQuitService](#)

5.73 Mergelt.Game.Services.Saves.Strategies Namespace Reference

Classes

- class [BaseSerializeStrategy](#)
- class [BinaryFormatterStrategy](#)
- interface [ISerializeStrategy](#)
- class [JsonSerializeStrategy](#)

5.74 Mergelt.Game.TopPanel Namespace Reference

Classes

- class [TopPanelComponent](#)

5.75 Mergelt.Game.UI Namespace Reference

5.76 Mergelt.Game.UI.InfoPanel Namespace Reference

Classes

- class [ElementCommonInfo](#)
- class [ElementInfo](#)
- class [ElementInfoPanelComponent](#)
- class [ElementOpenInfo](#)
- class [ElementSellInfo](#)
- class [ElementSpeedUpChargingInfo](#)
- class [ElementSpeedUpOpeningInfo](#)
- class [ElementSplitInfo](#)
- class [ElementUnlockInfo](#)

Enumerations

- enum **ElementActionType** {
 None = 0 , **Common** = 1 << 0 , **SkipCharging** = 1 << 1 , **SkipOpening** = 1 << 2 ,
 Sell = 1 << 3 , **Split** = 1 << 4 , **Open** = 1 << 5 , **Unlock** = 1 << 6 }

5.76.1 Enumeration Type Documentation

5.76.1.1 ElementActionType

```
enum MergeIt.Game.UI.InfoPanel.ElementActionType
```

Definition at line 8 of file [Element ActionType.cs](#).

```
00009  {
00010      None = 0,
00011      Common = 1 << 0,
00012      SkipCharging = 1 << 1,
00013      SkipOpening = 1 << 2,
00014      Sell = 1 << 3,
00015      Split = 1 << 4,
00016      Open = 1 << 5,
00017      Unlock = 1 << 6
00018 }
```

5.77 Mergelt.Game.UI.InventoryPanel Namespace Reference

Classes

- class [InventoryPanelComponent](#)
- class [InventoryPanelItemComponent](#)
- class [InventoryPanelItemPair](#)

5.78 Mergelt.Game.User Namespace Reference

Classes

- class [UserData](#)

5.79 Mergelt.Game.Windows Namespace Reference

5.80 Mergelt.Game.Windows.ElementInfo Namespace Reference

Classes

- class [ElementInfoArgs](#)
- class [ElementInfoItemComponent](#)
- class [ElementInfoItemsPanelComponent](#)
- class [ElementInfoModel](#)
- class [ElementInfoPresenter](#)
- class [ElementInfoWindow](#)
- class [ElementInfoWindowContent](#)

5.81 Mergelt.Game.Windows.Inventory Namespace Reference

Classes

- class [InventoryModel](#)
- class [InventoryPresenter](#)
- class [InventoryWindow](#)
- class [InventoryWindowItemComponent](#)
- class [InventoryWindowItemPair](#)
- class [InventoryWindowPaidCellComponent](#)

5.82 Mergelt.Game.Windows.UserProgress Namespace Reference

Classes

- class [UserInfoModel](#)
- class [UserInfoPresenter](#)
- class [UserInfoWindow](#)
- class [UserProgressComponent](#)

5.83 Mergelt.SimpleDI Namespace Reference

Classes

- class [DiContainer](#)
- class [DiHandler](#)
- class [DiRoot](#)
- class [IntroduceAttribute](#)

5.84 Mergelt.SimpleDI.ReservedInterfaces Namespace Reference

Classes

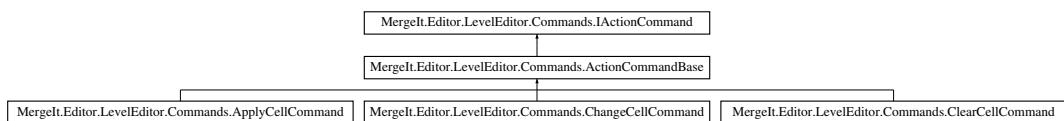
- interface [IInitializable](#)
- interface [IUpdatable](#)

Chapter 6

Class Documentation

6.1 Mergelt.Editor.LevelEditor.Commands.ActionCommandBase Class Reference

Inheritance diagram for Mergelt.Editor.LevelEditor.Commands.ActionCommandBase:



Public Member Functions

- [ActionCommandBase \(LevelEditorWindow window\)](#)
- void [Execute \(\)](#)
- void [Undo \(\)](#)

Protected Attributes

- [LevelEditorWindow Window](#)

6.1.1 Detailed Description

Definition at line 5 of file [ActionButtonBase.cs](#).

6.1.2 Constructor & Destructor Documentation

6.1.2.1 ActionCommandBase()

```
Mergelt.Editor.LevelEditor.Commands.ActionCommandBase.ActionCommandBase (
    LevelEditorWindow window ) [inline]
```

Definition at line 9 of file [ActionButtonBase.cs](#).

```
00010      {
00011          Window = window;
00012      }
```

6.1.3 Member Function Documentation

6.1.3.1 Execute()

```
void MergeIt.Editor.LevelEditor.Commands.ActionCommandBase.Execute ( ) [abstract]
```

Implements [MergeIt.Editor.LevelEditor.Commands.IActionCommand](#).

6.1.3.2 Undo()

```
void MergeIt.Editor.LevelEditor.Commands.ActionCommandBase.Undo ( ) [abstract]
```

Implements [MergeIt.Editor.LevelEditor.Commands.IActionCommand](#).

6.1.4 Member Data Documentation

6.1.4.1 Window

```
LevelEditorWindow MergeIt.Editor.LevelEditor.Commands.ActionCommandBase.Window [protected]
```

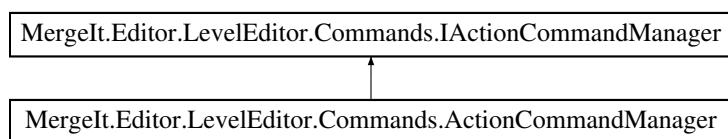
Definition at line 7 of file [ActionCommandBase.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Editor/Core/LevelEditor/Commands/ActionCommandBase.cs

6.2 MergeIt.Editor.LevelEditor.Commands.ActionCommandManager Class Reference

Inheritance diagram for MergeIt.Editor.LevelEditor.Commands.ActionCommandManager:



Public Member Functions

- void [ExecuteCommand \(IActionCommand command\)](#)
- void [Undo \(\)](#)
- void [Redo \(\)](#)

6.2.1 Detailed Description

Definition at line 7 of file [ActionCommandManager.cs](#).

6.2.2 Member Function Documentation

6.2.2.1 ExecuteCommand()

```
void MergeIt.Editor.LevelEditor.Commands.ActionCommandManager.ExecuteCommand (
    IActionCommand command) [inline]
```

Implements [Mergelt.Editor.LevelEditor.Commands.IActionCommandManager](#).

Definition at line 12 of file [ActionCommandManager.cs](#).

```
00013     {
00014         command.Execute();
00015         _undoStack.Push(command);
00016         _redoStack.Clear();
00017     }
```

6.2.2.2 Redo()

```
void MergeIt.Editor.LevelEditor.Commands.ActionCommandManager.Redo () [inline]
```

Implements [Mergelt.Editor.LevelEditor.Commands.IActionCommandManager](#).

Definition at line 30 of file [ActionCommandManager.cs](#).

```
00031     {
00032         if (_redoStack.Any())
00033         {
00034             var command = _redoStack.Pop();
00035             command.Execute();
00036             _undoStack.Push(command);
00037         }
00038     }
00039 }
```

6.2.2.3 Undo()

```
void MergeIt.Editor.LevelEditor.Commands.ActionCommandManager.Undo () [inline]
```

Implements [Mergelt.Editor.LevelEditor.Commands.IActionCommandManager](#).

Definition at line 19 of file [ActionCommandManager.cs](#).

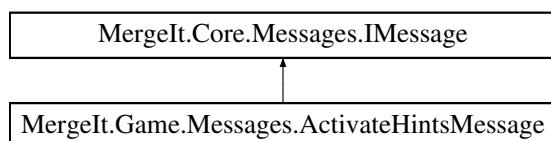
```
00020     {
00021         if (_undoStack.Any())
00022         {
00023             var command = _undoStack.Pop();
00024             command.Undo();
00025             _redoStack.Push(command);
00026         }
00027     }
00028 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/LevelEditor/Commands/ActionCommandManager.cs

6.3 Mergelt.Game.Messages.ActivateHintsMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.ActivateHintsMessage:



Properties

- bool [Active](#) [get, set]

6.3.1 Detailed Description

Definition at line 7 of file [ActivateHintsMessage.cs](#).

6.3.2 Property Documentation

6.3.2.1 Active

```
bool MergeIt.Game.Messages.ActivateHintsMessage.Active [get], [set]
```

Definition at line 9 of file [ActivateHintsMessage.cs](#).

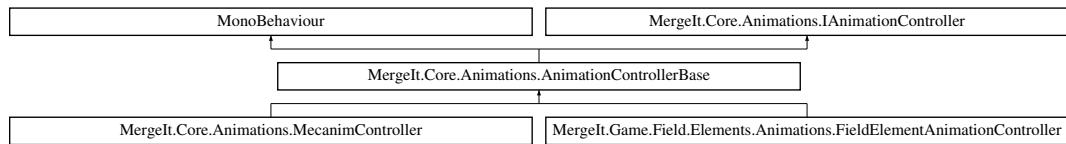
```
00009 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/ActivateHintsMessage.cs

6.4 Mergelt.Core.Animations.AnimationControllerBase Class Reference

Inheritance diagram for Mergelt.Core.Animations.AnimationControllerBase:



Public Member Functions

- void [Initialize \(IAnimationListener listener\)](#)
- virtual void [SetState \(string state\)](#)
- virtual void [SetState \(int state\)](#)
- virtual void [SetState< T > \(T state\)](#)

6.4.1 Detailed Description

Definition at line 8 of file [AnimationControllerBase.cs](#).

6.4.2 Member Function Documentation

6.4.2.1 Initialize()

```
void MergeIt.Core.Animations.AnimationControllerBase.Initialize (
    IAnimationListener listener ) [abstract]
```

Implements [MergeIt.Core.Animations.IAnimationController](#).

6.4.2.2 SetState() [1/2]

```
virtual void MergeIt.Core.Animations.AnimationControllerBase.SetState (
    int state ) [inline], [virtual]
```

Implements [MergeIt.Core.Animations.IAnimationController](#).

Definition at line 17 of file [AnimationControllerBase.cs](#).

```
00018     {
00019         throw new NotImplementedException();
00020     }
```

6.4.2.3 SetState() [2/2]

```
virtual void MergeIt.Core.Animations.AnimationControllerBase.SetState (
    string state ) [inline], [virtual]
```

Implements [MergeIt.Core.Animations.IAnimationController](#).

Definition at line 12 of file [AnimationControllerBase.cs](#).

```
00013     {
00014         throw new NotImplementedException();
00015     }
```

6.4.2.4 SetState< T >()

```
virtual void MergeIt.Core.Animations.AnimationControllerBase.SetState< T > (
    T state ) [inline], [virtual]
```

Implements [MergeIt.Core.Animations.IAnimationController](#).

Type Constraints

T : *Enum*

Definition at line 22 of file [AnimationControllerBase.cs](#).

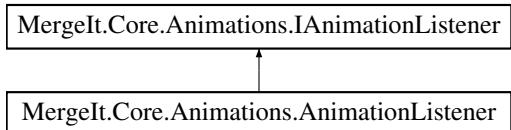
```
00022     {
00023         throw new NotImplementedException();
00024     } : Enum
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Animations/AnimationControllerBase.cs

6.5 Mergelt.Core.Animations.AnimationListener Class Reference

Inheritance diagram for Mergelt.Core.Animations.AnimationListener:



Public Member Functions

- virtual void [ResetAnimationState \(\)](#)

6.5.1 Detailed Description

Definition at line [5](#) of file [AnimationListener.cs](#).

6.5.2 Member Function Documentation

6.5.2.1 [ResetAnimationState\(\)](#)

```
virtual void MergeIt.Core.Animations.AnimationListener.ResetAnimationState ( ) [inline],  
[virtual]
```

Implements [MergeIt.Core.Animations.IAnimationListener](#).

Definition at line [7](#) of file [AnimationListener.cs](#).

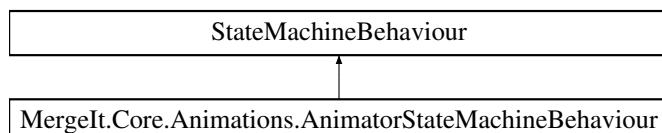
```
00008      {  
00009  
00010      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Animations/AnimationListener.cs

6.6 Mergelt.Core.Animations.AnimatorStateMachineBehaviour Class Reference

Inheritance diagram for Mergelt.Core.Animations.AnimatorStateMachineBehaviour:



Public Member Functions

- override void [OnStateEnter](#) (Animator animator, AnimatorStateInfo stateInfo, int layerIndex)
- override void [OnStateUpdate](#) (Animator animator, AnimatorStateInfo stateInfo, int layerIndex)

6.6.1 Detailed Description

Definition at line 9 of file [AnimatorStateMachineBehaviour.cs](#).

6.6.2 Member Function Documentation

6.6.2.1 OnStateEnter()

```
override void MergeIt.Core.Animations.AnimatorStateMachineBehaviour.OnStateEnter (
    Animator animator,
    AnimatorStateInfo stateInfo,
    int layerIndex) [inline]
```

Definition at line 13 of file [AnimatorStateMachineBehaviour.cs](#).

```
00014     {
00015         base.OnStateEnter(animator, stateInfo, layerIndex);
00016
00017         ActiveAnimators[animator] = true;
00018     }
```

6.6.2.2 OnStateUpdate()

```
override void MergeIt.Core.Animations.AnimatorStateMachineBehaviour.OnStateUpdate (
    Animator animator,
    AnimatorStateInfo stateInfo,
    int layerIndex) [inline]
```

Definition at line 20 of file [AnimatorStateMachineBehaviour.cs](#).

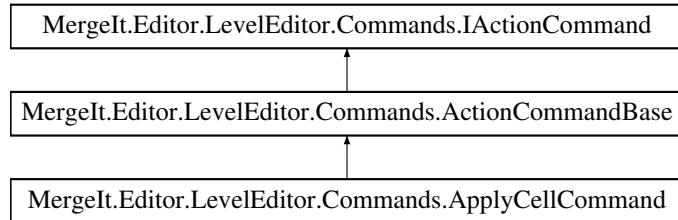
```
00021     {
00022         base.OnStateUpdate(animator, stateInfo, layerIndex);
00023
00024         if (!ActiveAnimators[animator])
00025         {
00026             return;
00027         }
00028
00029         if (stateInfo.normalizedTime >= 1f &&
00030             !animator.IsInTransition(layerIndex))
00031         {
00032             ActiveAnimators[animator] = false;
00033             CheckForEnd(animator, stateInfo.shortNameHash);
00034         }
00035     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Animations/AnimatorStateMachineBehaviour.cs

6.7 Mergelt.Editor.LevelEditor.Commands.ApplyCellCommand Class Reference

Inheritance diagram for Mergelt.Editor.LevelEditor.Commands.ApplyCellCommand:



Public Member Functions

- [ApplyCellCommand](#) (`LevelEditorWindow` window, `LevelElementData` newData)
- override void [Execute](#) ()
- override void [Undo](#) ()

Public Member Functions inherited from

[Mergelt.Editor.LevelEditor.Commands.ActionCommandBase](#)

- [ActionCommandBase](#) (`LevelEditorWindow` window)
- void [Execute](#) ()
- void [Undo](#) ()

Additional Inherited Members

Protected Attributes inherited from

[Mergelt.Editor.LevelEditor.Commands.ActionCommandBase](#)

- `LevelEditorWindow` Window

6.7.1 Detailed Description

Definition at line 7 of file [ApplyCellCommand.cs](#).

6.7.2 Constructor & Destructor Documentation

6.7.2.1 [ApplyCellCommand\(\)](#)

```
MergeIt.Editor.LevelEditor.Commands.ApplyCellCommand.ApplyCellCommand (
    LevelEditorWindow window,
    LevelElementData newData) [inline]
```

Definition at line 10 of file [ApplyCellCommand.cs](#).

```
00011      : base(window)
00012      {
00013          _newData = newData;
00014      }
```

6.7.3 Member Function Documentation

6.7.3.1 Execute()

```
override void MergeIt.Editor.LevelEditor.Commands.ApplyCellCommand.Execute ( ) [inline]
```

Implements [MergeIt.Editor.LevelEditor.Commands.IActionCommand](#).

Definition at line 16 of file [ApplyCellCommand.cs](#).

```
00017      {
00018          Window.ApplyCell(_newData, false);
00019      }
```

6.7.3.2 Undo()

```
override void MergeIt.Editor.LevelEditor.Commands.ApplyCellCommand.Undo ( ) [inline]
```

Implements [MergeIt.Editor.LevelEditor.Commands.IActionCommand](#).

Definition at line 21 of file [ApplyCellCommand.cs](#).

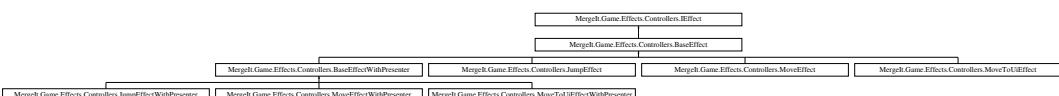
```
00022      {
00023          Window.UndoApplyCell(null, _newData);
00024      }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Editor/Core/LevelEditor/Commands/ApplyCellCommand.cs

6.8 Mergelt.Game.Effects.Controllers.BaseEffect Class Reference

Inheritance diagram for Mergelt.Game.Effects.Controllers.BaseEffect:



Public Member Functions

- virtual void [Setup](#) (Transform target, [IEffectParameters](#) effectParameters=null, Action finishedCallback=null)
- virtual void [Start](#) ()
- void [Update](#) ()

Protected Member Functions

- void [Finish](#) ()
- virtual void [OnStarted](#) ()
- virtual void [OnFinished](#) ()

Protected Attributes

- Transform [Target](#)
- Animator [Animator](#)
- Action [FinishedCallback](#)
- readonly [IConfigsService ConfigsService](#) = DiContainer.Get<[IConfigsService](#)>()

Properties

- bool [Started](#) [get, set]

6.8.1 Detailed Description

Definition at line 11 of file [BaseEffect.cs](#).

6.8.2 Member Function Documentation

6.8.2.1 Finish()

```
void MergeIt.Game.Effects.Controllers.BaseEffect.Finish ( ) [inline], [protected]
```

Definition at line 27 of file [BaseEffect.cs](#).

```
00028     {
00029         Started = false;
00030         FinishedCallback?.Invoke();
00031         FinishedCallback = null;
00032
00033         if (Animator)
00034         {
00035             Animator.enabled = true;
00036         }
00037
00038         OnFinished();
00039     }
```

6.8.2.2 OnFinished()

```
virtual void MergeIt.Game.Effects.Controllers.BaseEffect.OnFinished ( ) [inline], [protected], [virtual]
```

Definition at line 46 of file [BaseEffect.cs](#).

```
00047     {
00048
00049 }
```

6.8.2.3 OnStarted()

```
virtual void MergeIt.Game.Effects.Controllers.BaseEffect.OnStarted ( ) [inline], [protected], [virtual]
```

Definition at line 41 of file [BaseEffect.cs](#).

```
00042     {
00043
00044 }
```

6.8.2.4 Setup()

```
virtual void MergeIt.Game.Effects.Controllers.BaseEffect.Setup (
    Transform target,
    IEffectParameters effectParameters = null,
    Action finishedCallback = null ) [inline], [virtual]
```

Implements [MergeIt.Game.Effects.Controllers.IEffect](#).

Definition at line 21 of file [BaseEffect.cs](#).

```
00022     {
00023         Target = target;
00024         FinishedCallback = finishedCallback;
00025     }
```

6.8.2.5 Start()

```
virtual void MergeIt.Game.Effects.Controllers.BaseEffect.Start () [inline], [virtual]
```

Implements [MergeIt.Game.Effects.Controllers.IEffect](#).

Definition at line 51 of file [BaseEffect.cs](#).

```
00052     {
00053         Started = true;
00054
00055         Target.TryGetComponent (out Animator);
00056
00057         if (Animator)
00058         {
00059             Animator.enabled = false;
00060         }
00061
00062         OnStarted();
00063     }
```

6.8.2.6 Update()

```
void MergeIt.Game.Effects.Controllers.BaseEffect.Update () [abstract]
```

Implements [MergeIt.Game.Effects.Controllers.IEffect](#).

6.8.3 Member Data Documentation

6.8.3.1 Animator

```
Animator MergeIt.Game.Effects.Controllers.BaseEffect.Animator [protected]
```

Definition at line 14 of file [BaseEffect.cs](#).

6.8.3.2 ConfigsService

```
readonly IConfigsService MergeIt.Game.Effects.Controllers.BaseEffect.ConfigsService = Di←
Container.Get<IConfigsService>() [protected]
```

Definition at line 17 of file [BaseEffect.cs](#).

6.8.3.3 FinishedCallback

Action MergeIt.Game.Effects.Controllers.BaseEffect.FinishedCallback [protected]

Definition at line 15 of file [BaseEffect.cs](#).

6.8.3.4 Target

Transform MergeIt.Game.Effects.Controllers.BaseEffect.Target [protected]

Definition at line 13 of file [BaseEffect.cs](#).

6.8.4 Property Documentation

6.8.4.1 Started

bool MergeIt.Game.Effects.Controllers.BaseEffect.Started [get], [set]

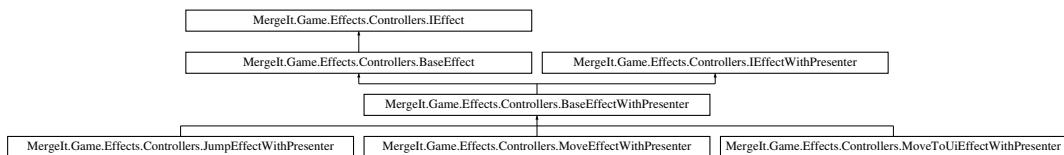
Definition at line 19 of file [BaseEffect.cs](#).
00019 { get; set; }

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Effects/Controllers/BaseEffect.cs

6.9 MergeIt.Game.Effects.Controllers.BaseEffectWithPresenter Class Reference

Inheritance diagram for MergeIt.Game.Effects.Controllers.BaseEffectWithPresenter:



Public Member Functions

- virtual void [Setup](#) ([IFieldElementPresenter](#) presenter, [IEffectParameters](#) effectParameters=null, Action finishedCallback=null)
- [IFieldElementPresenter GetPresenter \(\)](#)

Public Member Functions inherited from [MergeIt.Game.Effects.Controllers.BaseEffect](#)

- virtual void [Setup](#) (Transform target, [IEffectParameters](#) effectParameters=null, Action finishedCallback=null)
- virtual void [Start \(\)](#)
- void [Update \(\)](#)

Protected Member Functions

- virtual void [FinishCallbackHandler \(\)](#)

Protected Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- void [Finish \(\)](#)
- virtual void [OnStarted \(\)](#)
- virtual void [OnFinished \(\)](#)

Protected Attributes

- [IFieldElementPresenter Presenter](#)

Protected Attributes inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- Transform [Target](#)
- Animator [Animator](#)
- Action [FinishedCallback](#)
- readonly [IConfigsService ConfigsService](#) = DiContainer.Get<[IConfigsService](#)>()

Additional Inherited Members

Properties inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- bool [Started](#) [get, set]

6.9.1 Detailed Description

Definition at line 9 of file [BaseEffectWithPresenter.cs](#).

6.9.2 Member Function Documentation

6.9.2.1 [FinishCallbackHandler\(\)](#)

```
virtual void MergeIt.Game.Effects.Controllers.BaseEffectWithPresenter.FinishCallbackHandler (
) [inline], [protected], [virtual]
```

Definition at line 25 of file [BaseEffectWithPresenter.cs](#).

```
00026      {
00027          Presenter.Canvas.sortingOrder = 1;
00028          // Presenter.UpdateInitialPosition();
00029          Presenter.SetBusy(false);
00030
00031          Finish();
00032      }
```

6.9.2.2 GetPresenter()

```
IFieldElementPresenter MergeIt.Game.Effects.Controllers.BaseEffectWithPresenter.GetPresenter (
) [inline]
```

Implements [MergeIt.Game.Effects.Controllers.IEffectWithPresenter](#).

Definition at line 20 of file [BaseEffectWithPresenter.cs](#).

```
00021     {
00022         return Presenter;
00023     }
```

6.9.2.3 Setup()

```
virtual void MergeIt.Game.Effects.Controllers.BaseEffectWithPresenter.Setup (
    IFieldElementPresenter presenter,
    IEFFECTPARAMETERS effectParameters = null,
    Action finishedCallback = null ) [inline], [virtual]
```

Implements [MergeIt.Game.Effects.Controllers.IEffectWithPresenter](#).

Definition at line 13 of file [BaseEffectWithPresenter.cs](#).

```
00014     {
00015         base.Setup(presenter.RectTransform, effectParameters, finishedCallback);
00016         Presenter = presenter;
00018     }
```

6.9.3 Member Data Documentation

6.9.3.1 Presenter

```
IFieldElementPresenter MergeIt.Game.Effects.Controllers.BaseEffectWithPresenter.Presenter
[protected]
```

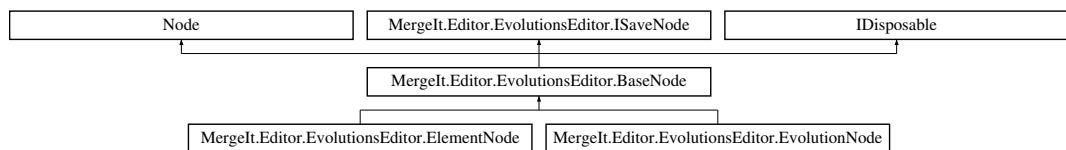
Definition at line 11 of file [BaseEffectWithPresenter.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Effects/Controllers/BaseEffectWithPresenter.cs

6.10 MergeIt.Editor.EvolutionsEditor.BaseNode Class Reference

Inheritance diagram for [MergeIt.Editor.EvolutionsEditor.BaseNode](#):



Public Member Functions

- void [SaveData \(\)](#)
- Port [GetPort \(string portName\)](#)
- virtual void [Dispose \(\)](#)

Properties

- string [Id \[get\]](#)

Properties inherited from [Mergelt.Editor.EvolutionsEditor.ISaveNode](#)

6.10.1 Detailed Description

Definition at line 9 of file [BaseNode.cs](#).

6.10.2 Member Function Documentation

6.10.2.1 Dispose()

```
virtual void MergeIt.Editor.EvolutionsEditor.BaseNode.Dispose ( ) [inline], [virtual]
```

Definition at line 32 of file [BaseNode.cs](#).

```
00033     {  
00034         }
```

6.10.2.2 GetPort()

```
Port MergeIt.Editor.EvolutionsEditor.BaseNode.GetPort (  
    string portName) [inline]
```

Implements [Mergelt.Editor.EvolutionsEditor.ISaveNode](#).

Definition at line 14 of file [BaseNode.cs](#).

```
00015     {  
00016         var port = inputContainer.Children()  
00017             .OfType<Port>()  
00018             .FirstOrDefault(x => x.portName.Equals(portName));  
00019  
00020         if (port != null)  
00021         {  
00022             return port;  
00023         }  
00024  
00025         port = outputContainer.Children()  
00026             .OfType<Port>()  
00027             .FirstOrDefault(x => x.portName.Equals(portName));  
00028  
00029         return port;  
00030     }
```

6.10.2.3 SaveData()

```
void MergeIt.Editor.EvolutionsEditor.BaseNode.SaveData ( ) [abstract]
```

Implements [Mergelt.Editor.EvolutionsEditor.ISaveNode](#).

6.10.3 Property Documentation

6.10.3.1 Id

```
string MergeIt.Editor.EvolutionsEditor.BaseNode.Id [get], [abstract]
```

Implements [MergeIt.Editor.EvolutionsEditor.ISaveNode](#).

Definition at line 11 of file [BaseNode.cs](#).

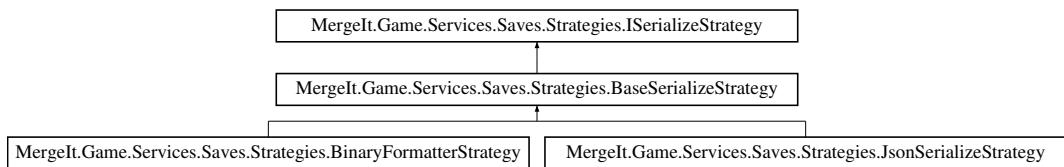
```
00011 { get; }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Editor/Core/EvolutionsEditor/ItemsNodes/BaseNode.cs

6.11 **MergeIt.Game.Services.Saves.Strategies.BaseSerializeStrategy** Class Reference

Inheritance diagram for MergeIt.Game.Services.Saves.Strategies.BaseSerializeStrategy:



Public Member Functions

- UniTask [Save< T >](#) (T data)
- T [Load< T >](#) ()

Properties

- virtual string [SaveDir](#) [get]

Properties inherited from [MergeIt.Game.Services.Saves.Strategies.ISerializeStrategy](#)

6.11.1 Detailed Description

Definition at line 8 of file [BaseSerializeStrategy.cs](#).

6.11.2 Member Function Documentation

6.11.2.1 Load< T >()

`T MergeIt.Game.Services.Saves.Strategies.BaseSerializeStrategy.Load< T > () [abstract]`

Implements [Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy](#).

Type Constraints

T : class

T : ISavable

6.11.2.2 Save< T >()

`UnitTask MergeIt.Game.Services.Saves.Strategies.BaseSerializeStrategy.Save< T > (T data) [abstract]`

Implements [Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy](#).

Type Constraints

T : class

T : ISavable

6.11.3 Property Documentation

6.11.3.1 SaveDir

`virtual string MergeIt.Game.Services.Saves.Strategies.BaseSerializeStrategy.SaveDir [get]`

Implements [Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy](#).

Definition at line 10 of file [BaseSerializeStrategy.cs](#).

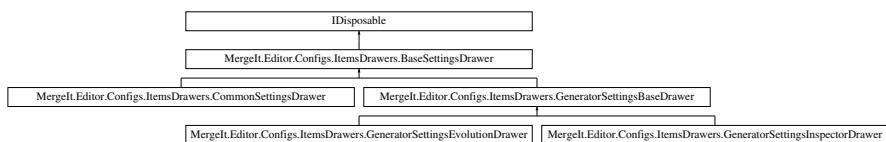
```
00011     {
00012         get => "Saves";
00013     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/Saves/Strategies/BaseSerializeStrategy.cs

6.12 Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer Class Reference

Inheritance diagram for Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer:



Public Member Functions

- `BaseSettingsDrawer` (`VisualElement` parent, `ElementConfig` config)
- `void Draw()`
- `virtual void Hide()`
- `virtual void Show()`
- `virtual void Dispose()`

Protected Member Functions

- `void Expand(bool expand)`
- `void OnExpand(bool expand)`
- `void RotateArrow(bool rotate)`
- `VisualTreeAsset GetUxml()`

Protected Attributes

- `readonly VisualElement Parent`
- `readonly VisualElement Root`
- `readonly ElementConfig Config`
- `Label Arrow`

6.12.1 Detailed Description

Definition at line 9 of file `BaseSettingsDrawer.cs`.

6.12.2 Constructor & Destructor Documentation

6.12.2.1 BaseSettingsDrawer()

```
MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer.BaseSettingsDrawer (
    VisualElement parent,
    ElementConfig config) [inline]
```

Definition at line 16 of file `BaseSettingsDrawer.cs`.

```
00017     {
00018         Config = config;
00019         Root = new VisualElement();
00020         Parent = parent;
00021
00022         VisualTreeAsset visualTreeAsset = GetUxml();
00023         visualTreeAsset.CloneTree(Root);
00024
00025         Parent.Add(Root);
00026     }
```

6.12.3 Member Function Documentation

6.12.3.1 Dispose()

```
virtual void MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer.Dispose() [inline],
[virtual]
```

Definition at line 56 of file `BaseSettingsDrawer.cs`.

```
00057     {
00058 }
```

6.12.3.2 Expand()

```
void MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer.Expand (
    bool expand) [inline], [protected]
```

Definition at line 40 of file [BaseSettingsDrawer.cs](#).

```
00041     {
00042         OnExpand(expand);
00043         RotateArrow(expand);
00045     }
```

6.12.3.3 Hide()

```
virtual void MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer.Hide () [inline], [virtual]
```

Definition at line 30 of file [BaseSettingsDrawer.cs](#).

```
00031     {
00032         Root.style.display = StylesConstants.DisplayNone;
00033     }
```

6.12.3.4 RotateArrow()

```
void MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer.RotateArrow (
    bool rotate) [inline], [protected]
```

Definition at line 49 of file [BaseSettingsDrawer.cs](#).

```
00050     {
00051         Arrow.style.rotate = rotate ? new StyleRotate(new Rotate(90)) : new StyleRotate(new
00052             Rotate(0));
00052     }
```

6.12.3.5 Show()

```
virtual void MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer.Show () [inline], [virtual]
```

Definition at line 35 of file [BaseSettingsDrawer.cs](#).

```
00036     {
00037         Root.style.display = StylesConstants.DisplayFlex;
00038     }
```

6.12.4 Member Data Documentation

6.12.4.1 Arrow

```
Label MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer.Arrow [protected]
```

Definition at line 14 of file [BaseSettingsDrawer.cs](#).

6.12.4.2 Config

```
readonly ElementConfig MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer.Config [protected]
```

Definition at line 13 of file [BaseSettingsDrawer.cs](#).

6.12.4.3 Parent

```
readonly VisualElement MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer.Parent [protected]
```

Definition at line 11 of file [BaseSettingsDrawer.cs](#).

6.12.4.4 Root

```
readonly VisualElement MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer.Root [protected]
```

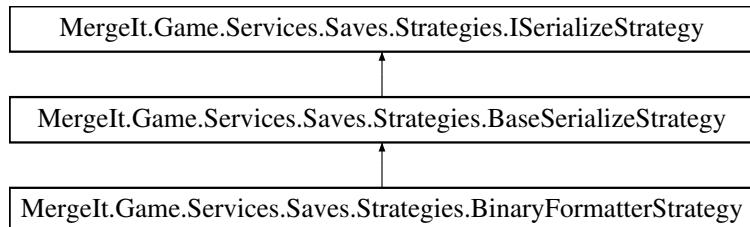
Definition at line 12 of file [BaseSettingsDrawer.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Editor/Core/Configs/ItemsDrawers/BaseSettingsDrawer.cs

6.13 MergeIt.Game.Services.Saves.Strategies.BinaryFormatterStrategy Class Reference

Inheritance diagram for MergeIt.Game.Services.Saves.Strategies.BinaryFormatterStrategy:



Public Member Functions

- async override UniTask [Save< T >](#) (T data)
- override T [Load< T >](#) ()

Public Member Functions inherited from [MergeIt.Game.Services.Saves.Strategies.BaseSerializeStrategy](#)

- UniTask [Save< T >](#) (T data)
- T [Load< T >](#) ()

Additional Inherited Members

Properties inherited from [MergeIt.Game.Services.Saves.Strategies.BaseSerializeStrategy](#)

- virtual string [SaveDir](#) [get]

Properties inherited from [Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy](#)

6.13.1 Detailed Description

Definition at line 12 of file [BinaryFormatterStrategy.cs](#).

6.13.2 Member Function Documentation

6.13.2.1 Load< T >()

```
override T MergeIt.Game.Services.Saves.Strategies.BinaryFormatterStrategy.Load< T > ( ) [inline]
```

Implements [Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy](#).

Definition at line 28 of file [BinaryFormatterStrategy.cs](#).

```
00029     {
00030         string fileName = SavesHelper.GetFileNameWithExtension<T>();
00031         string loadPath = Path.Combine(Application.streamingAssetsPath, SaveDir, fileName);
00032         if (File.Exists(loadPath))
00033         {
00034             FileStream fileStream = File.Open(loadPath, FileMode.Open);
00035
00036             var loadedData = _formatter.Deserialize(fileStream) as T;
00037
00038             fileStream.Close();
00039
00040             return loadedData;
00041         }
00042
00043         Debug.Log($"There is no saved {fileName} found.");
00044
00045         return null;
00046     }
```

6.13.2.2 Save< T >()

```
async override UniTask MergeIt.Game.Services.Saves.Strategies.BinaryFormatterStrategy.Save< T
> (
    T data ) [inline]
```

Implements [Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy](#).

Definition at line 16 of file [BinaryFormatterStrategy.cs](#).

```
00017     {
00018         using var memoryStream = new MemoryStream();
00019
00020         _formatter.Serialize(memoryStream, data);
00021
00022         byte[] serializedData = memoryStream.ToArray();
00023         string fileName = SavesHelper.GetFileNameWithExtension<T>();
00024
00025         await WriteSerializedDataAsync(serializedData, fileName);
00026     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/Saves/Strategies/BinaryFormatterStrategy.cs

6.14 MergeIt.Core.Utils.Bindable< T > Class Template Reference

Public Member Functions

- void [Subscribe](#) (Action< T, T > onChanged, bool immediateCheck=false)
- void [Subscribe](#) (Action< T > onChanged, bool immediateCheck=false)
- void [Unsubscribe](#) (Action< T, T > onChanged)
- void [Unsubscribe](#) (Action< T > onChanged)
- void [SetValueSilently](#) (T value)
- void [Hold](#) ()
- void [Release](#) ()

Properties

- T [Value](#) [get, set]

6.14.1 Detailed Description

Definition at line 8 of file [Bindable.cs](#).

6.14.2 Member Function Documentation

6.14.2.1 Hold()

```
void MergeIt.Core.Utils.Bindable< T >.Hold () [inline]
```

Definition at line 79 of file [Bindable.cs](#).

```
00080     {
00081         _isHeld = true;
00082     }
```

6.14.2.2 Release()

```
void MergeIt.Core.Utils.Bindable< T >.Release () [inline]
```

Definition at line 84 of file [Bindable.cs](#).

```
00085     {
00086         if (!isHeld)
00087         {
00088             return;
00089         }
00090
00091         isHeld = false;
00092
00093         T temp = _holdValue;
00094         _holdValue = default;
00095
00096         if (!temp.Equals(_value))
00097         {
00098             ChangedFromToEvent?.Invoke(temp, _value);
00099             ChangedToEvent?.Invoke(_value);
00100         }
00101     }
```

6.14.2.3 SetValueSilently()

```
void MergeIt.Core.Utils.Bindable< T >.SetValueSilently (
    T value ) [inline]
```

Definition at line 74 of file Bindable.cs.

```
00075     {
00076         _value = value;
00077     }
```

6.14.2.4 Subscribe() [1/2]

```
void MergeIt.Core.Utils.Bindable< T >.Subscribe (
    Action< T > onChanged,
    bool immediateCheck = false ) [inline]
```

Definition at line 54 of file Bindable.cs.

```
00055     {
00056         ChangedToEvent += onChanged;
00057
00058         if (immediateCheck)
00059         {
00060             onChanged?.Invoke(_value);
00061         }
00062     }
```

6.14.2.5 Subscribe() [2/2]

```
void MergeIt.Core.Utils.Bindable< T >.Subscribe (
    Action< T, T > onChanged,
    bool immediateCheck = false ) [inline]
```

Definition at line 44 of file Bindable.cs.

```
00045     {
00046         ChangedFromToEvent += onChanged;
00047
00048         if (immediateCheck)
00049         {
00050             onChanged?.Invoke(default, _value);
00051         }
00052     }
```

6.14.2.6 Unsubscribe() [1/2]

```
void MergeIt.Core.Utils.Bindable< T >.Unsubscribe (
    Action< T > onChanged ) [inline]
```

Definition at line 69 of file Bindable.cs.

```
00070     {
00071         ChangedToEvent -= onChanged;
00072     }
```

6.14.2.7 Unsubscribe() [2/2]

```
void MergeIt.Core.Utils.Bindable< T >.Unsubscribe (
    Action< T, T > onChanged ) [inline]
```

Definition at line 64 of file Bindable.cs.

```
00065     {
00066         ChangedFromToEvent -= onChanged;
00067     }
```

6.14.3 Property Documentation

6.14.3.1 Value

`T MergeIt.Core.Utils.Bindable< T >.Value [get], [set]`

Definition at line 17 of file [Bindable.cs](#).

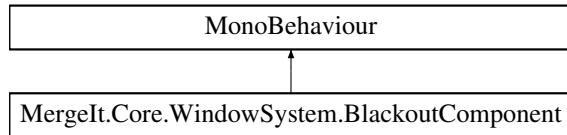
```
00018     {
00019         get => _value;
00020         set
00021     {
00022         if (EqualityComparer<T>.Default.Equals(_value, value))
00023         {
00024             return;
00025         }
00026         if (_isHeld)
00027         {
00028             _holdValue = _value;
00029             _value = value;
00030         }
00031         return;
00032     }
00033 }
00034
00035     T prevValue = _value;
00036
00037     _value = value;
00038
00039     ChangedFromToEvent?.Invoke(prevValue, value);
00040     ChangedToEvent?.Invoke(value);
00041 }
00042 }
```

The documentation for this class was generated from the following file:

- [Assets/MergeIt/Source/Utils/Bindable.cs](#)

6.15 MergeIt.Core.WindowSystem.BlackoutComponent Class Reference

Inheritance diagram for MergeIt.Core.WindowSystem.BlackoutComponent:



Public Member Functions

- void [SetActive](#) (bool active)
- void [SetLayer](#) (string layer=null)

6.15.1 Detailed Description

Definition at line 7 of file [BlackoutComponent.cs](#).

6.15.2 Member Function Documentation

6.15.2.1 SetActive()

```
void MergeIt.Core.WindowSystem.BlackoutComponent.SetActive (
    bool active ) [inline]
```

Definition at line 16 of file [BlackoutComponent.cs](#).

```
00017     {
00018         gameObject.SetActive(active);
00019     }
```

6.15.2.2 SetLayer()

```
void MergeIt.Core.WindowSystem.BlackoutComponent.SetLayer (
    string layer = null ) [inline]
```

Definition at line 21 of file [BlackoutComponent.cs](#).

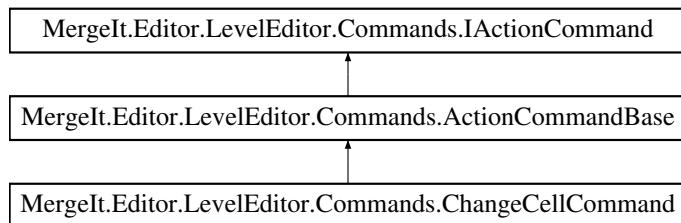
```
00022     {
00023         string layerName = string.IsNullOrEmpty(layer) ? Configs.Windows.SortingLayers.Default :
00024             layer;
00025         _canvas.overrideSorting = true;
00026         _canvas.sortingLayerName = layerName;
00027     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/BlackoutComponent.cs

6.16 Mergelt.Editor.LevelEditor.Commands.ChangeCellCommand Class Reference

Inheritance diagram for Mergelt.Editor.LevelEditor.Commands.ChangeCellCommand:



Public Member Functions

- [ChangeCellCommand](#) ([LevelEditorWindow](#) window, [LevelElementData](#) previousData, [LevelElementData](#) newData)
- [override void Execute \(\)](#)
- [override void Undo \(\)](#)

Public Member Functions inherited from [Mergelt.Editor.LevelEditor.Commands.ActionCommandBase](#)

- [ActionCommandBase \(LevelEditorWindow window\)](#)
- void [Execute \(\)](#)
- void [Undo \(\)](#)

Additional Inherited Members

Protected Attributes inherited from [Mergelt.Editor.LevelEditor.Commands.ActionCommandBase](#)

- [LevelEditorWindow Window](#)

6.16.1 Detailed Description

Definition at line 7 of file [ChangeCellCommand.cs](#).

6.16.2 Constructor & Destructor Documentation

6.16.2.1 [ChangeCellCommand\(\)](#)

```
MergeIt.Editor.LevelEditor.Commands.ChangeCellCommand.ChangeCellCommand (
    LevelEditorWindow window,
    LevelElementData previousData,
    LevelElementData newData ) [inline]
```

Definition at line 12 of file [ChangeCellCommand.cs](#).

```
00013     : base(window)
00014     {
00015         _previousData = previousData.GetClone();
00016         _newData = newData.GetClone();
00017     }
```

6.16.3 Member Function Documentation

6.16.3.1 [Execute\(\)](#)

```
override void MergeIt.Editor.LevelEditor.Commands.ChangeCellCommand.Execute ( ) [inline]
```

Implements [Mergelt.Editor.LevelEditor.Commands.IActionCommand](#).

Definition at line 19 of file [ChangeCellCommand.cs](#).

```
00020     {
00021         Window.ApplyCell(_newData, true);
00022     }
```

6.16.3.2 Undo()

```
override void MergeIt.Editor.LevelEditor.Commands.ChangeCellCommand.Undo ( ) [inline]
```

Implements [Mergelt.Editor.LevelEditor.Commands.IActionCommand](#).

Definition at line 24 of file [ChangeCellCommand.cs](#).

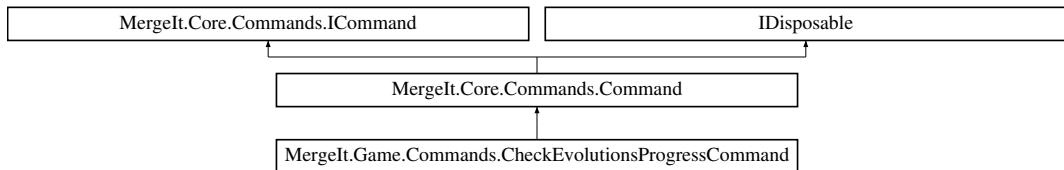
```
00025     {
00026         Window.UndoApplyCell(_previousData, _newData);
00027     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/LevelEditor/Commands/ChangeCellCommand.cs

6.17 Mergelt.Game.Commands.CheckEvolutionsProgressCommand Class Reference

Inheritance diagram for Mergelt.Game.Commands.CheckEvolutionsProgressCommand:



Public Member Functions

- `override void Execute ()`

Public Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- `virtual async UniTask ExecuteAsync ()`
- `void Dispose ()`

Additional Inherited Members

Protected Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- `virtual void Finish ()`
- `virtual void OnDispose ()`

Events inherited from [Mergelt.Core.Commands.Command](#)

- `Action< ICommand > Finished`

Events inherited from [Mergelt.Core.Commands.ICommand](#)

- Action< [ICommand](#) > Finished

6.17.1 Detailed Description

Definition at line 15 of file [CheckEvolutionsProgressCommand.cs](#).

6.17.2 Member Function Documentation

6.17.2.1 Execute()

```
override void MergeIt.Game.Commands.CheckEvolutionsProgressCommand.Execute ( ) [inline],  
[virtual]
```

Reimplemented from [Mergelt.Core.Commands.Command](#).

Definition at line 22 of file [CheckEvolutionsProgressCommand.cs](#).

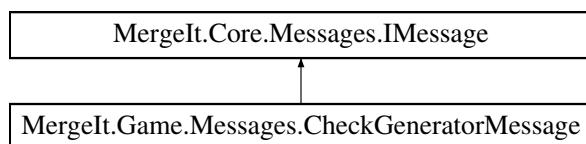
```
00023     {  
00024         var evolutionsProgressData = _gameLoadService.Load<EvolutionsProgressData>();  
00025  
00026         if (evolutionsProgressData == null)  
00027         {  
00028             _evolutionsService.CreateEvolutionsProgress();  
00029         }  
00030         else  
00031         {  
00032             _evolutionsService.SetupEvolutionsProgress(evolutionsProgressData);  
00033         }  
00034  
00035         UpdateProgress();  
00036     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Commands/CheckEvolutionsProgressCommand.cs

6.18 Mergelt.Game.Messages.CheckGeneratorMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.CheckGeneratorMessage:



Properties

- [GridPoint GeneratorPoint](#) [get, set]

6.18.1 Detailed Description

Definition at line 8 of file [CheckGeneratorMessage.cs](#).

6.18.2 Property Documentation

6.18.2.1 GeneratorPoint

`GridPoint` MergeIt.Game.Messages.CheckGeneratorMessage.GeneratorPoint [get], [set]

Definition at line 10 of file [CheckGeneratorMessage.cs](#).

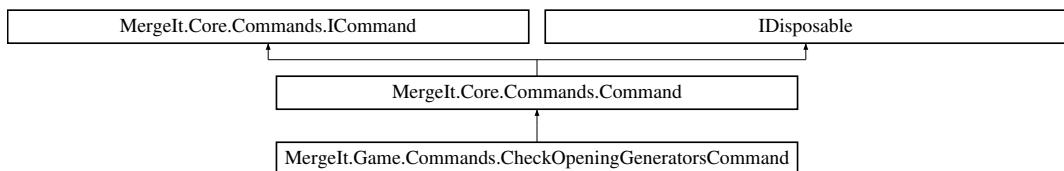
```
00010 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/CheckGeneratorMessage.cs

6.19 Mergelt.Game.Commands.CheckOpeningGeneratorsCommand Class Reference

Inheritance diagram for Mergelt.Game.Commands.CheckOpeningGeneratorsCommand:



Public Member Functions

- override void [Execute \(\)](#)

Public Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual async UniTask [ExecuteAsync \(\)](#)
- void [Dispose \(\)](#)

Additional Inherited Members

Protected Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual void [Finish \(\)](#)
- virtual void [OnDispose \(\)](#)

Events inherited from [Mergelt.Core.Commands.Command](#)

- Action< [ICommand](#) > Finished

Events inherited from [Mergelt.Core.Commands.ICommand](#)

- Action< [ICommand](#) > Finished

6.19.1 Detailed Description

Definition at line 9 of file [CheckOpeningGeneratorsCommand.cs](#).

6.19.2 Member Function Documentation

6.19.2.1 Execute()

```
override void MergeIt.Game.Commands.CheckOpeningGeneratorsCommand.Execute ( ) [inline], [virtual]
```

Reimplemented from [Mergelt.Core.Commands.Command](#).

Definition at line 13 of file [CheckOpeningGeneratorsCommand.cs](#).

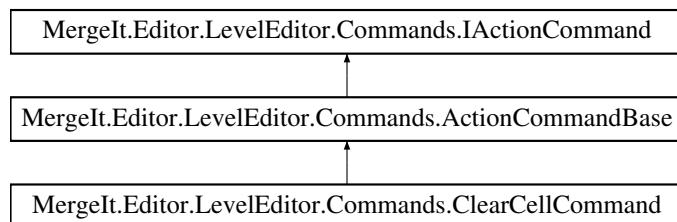
```
00014      {  
00015  
00016      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Commands/CheckOpeningGeneratorsCommand.cs

6.20 Mergelt.Editor.LevelEditor.Commands.ClearCellCommand Class Reference

Inheritance diagram for Mergelt.Editor.LevelEditor.Commands.ClearCellCommand:



Public Member Functions

- [ClearCellCommand](#) ([LevelEditorWindow](#) window, [LevelElementData](#) previousData)
- override void [Undo](#) ()
- override void [Execute](#) ()

Public Member Functions inherited from [MergeIt.Editor.LevelEditor.Commands.ActionCommandBase](#)

- [ActionCommandBase \(LevelEditorWindow window\)](#)
- void [Execute \(\)](#)
- void [Undo \(\)](#)

Additional Inherited Members

Protected Attributes inherited from [MergeIt.Editor.LevelEditor.Commands.ActionCommandBase](#)

- [LevelEditorWindow Window](#)

6.20.1 Detailed Description

Definition at line 7 of file [ClearCellCommand.cs](#).

6.20.2 Constructor & Destructor Documentation

6.20.2.1 ClearCellCommand()

```
MergeIt.Editor.LevelEditor.Commands.ClearCellCommand.ClearCellCommand ( 
    LevelEditorWindow window,
    LevelElementData previousData ) [inline]
```

Definition at line 11 of file [ClearCellCommand.cs](#).

```
00012         : base(window)
00013         {
00014             _previousData = previousData;
00015         }
```

6.20.3 Member Function Documentation

6.20.3.1 Execute()

```
override void MergeIt.Editor.LevelEditor.Commands.ClearCellCommand.Execute ( ) [inline]
```

Implements [MergeIt.Editor.LevelEditor.Commands.IActionCommand](#).

Definition at line 23 of file [ClearCellCommand.cs](#).

```
00024         {
00025             Window.ClearCell(_previousData.Position);
00026         }
```

6.20.3.2 Undo()

```
override void MergeIt.Editor.LevelEditor.Commands.ClearCellCommand.Undo ( ) [inline]
```

Implements [MergeIt.Editor.LevelEditor.Commands.IActionCommand](#).

Definition at line 17 of file [ClearCellCommand.cs](#).

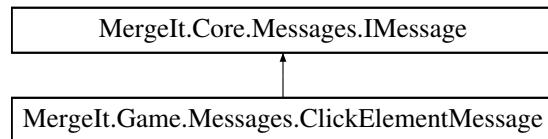
```
00018     {
00019         EvolutionSelection.UpdateCell(_previousData);
00020         Window.UndoClearCell(_previousData);
00021     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Editor/Core/LevelEditor/Commands/ClearCellCommand.cs

6.21 MergeIt.Game.Messages.ClickElementMessage Class Reference

Inheritance diagram for MergeIt.Game.Messages.ClickElementMessage:



Properties

- [FieldCellComponent Cell](#) [get, set]

6.21.1 Detailed Description

Definition at line 8 of file [ClickElementMessage.cs](#).

6.21.2 Property Documentation

6.21.2.1 Cell

```
FieldCellComponent MergeIt.Game.Messages.ClickElementMessage.Cell [get], [set]
```

Definition at line 10 of file [ClickElementMessage.cs](#).

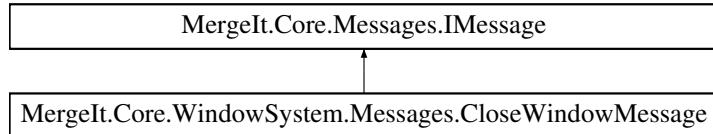
```
00010 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Messages/ClickElementMessage.cs

6.22 Mergelt.Core.WindowSystem.Messages.CloseWindowMessage Class Reference

Inheritance diagram for Mergelt.Core.WindowSystem.Messages.CloseWindowMessage:



Properties

- `IWindowPresenter Presenter [get, set]`

6.22.1 Detailed Description

Definition at line 8 of file [CloseWindowMessage.cs](#).

6.22.2 Property Documentation

6.22.2.1 Presenter

`IWindowPresenter MergeIt.Core.WindowSystem.Messages.CloseWindowMessage.Presenter [get], [set]`

Definition at line 10 of file [CloseWindowMessage.cs](#).

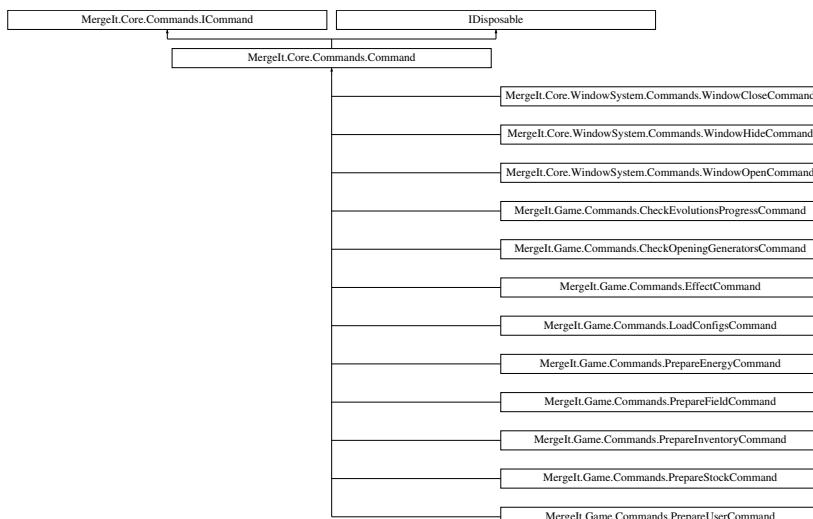
00010 { `get;` `set;` }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/Messages/CloseWindowMessage.cs

6.23 Mergelt.Core.Commands.Command Class Reference

Inheritance diagram for Mergelt.Core.Commands.Command:



Public Member Functions

- virtual void [Execute \(\)](#)
- virtual async UniTask [ExecuteAsync \(\)](#)
- void [Dispose \(\)](#)

Protected Member Functions

- virtual void [Finish \(\)](#)
- virtual void [OnDispose \(\)](#)

Events

- Action< [ICommand >](#) [Finished](#)

Events inherited from [MergeIt.Core.Commands.ICommand](#)

- Action< [ICommand >](#) [Finished](#)

6.23.1 Detailed Description

Definition at line 8 of file [Command.cs](#).

6.23.2 Member Function Documentation

6.23.2.1 Dispose()

```
void MergeIt.Core.Commands.Command.Dispose ( ) [inline]
```

Definition at line 22 of file [Command.cs](#).

```
00023     {  
00024         OnDispose ();  
00025     }
```

6.23.2.2 Execute()

```
virtual void MergeIt.Core.Commands.Command.Execute ( ) [inline], [virtual]
```

Implements [MergeIt.Core.Commands.ICommand](#).

Definition at line 12 of file [Command.cs](#).

```
00013     {  
00014  
00015     }
```

6.23.2.3 ExecuteAsync()

```
virtual async UniTask MergeIt.Core.Commands.Command.ExecuteAsync () [inline], [virtual]
```

Implements [Mergelt.Core.Commands.ICommand](#).

Definition at line 17 of file [Command.cs](#).

```
00018     {
00019         await UniTask.RunOnThreadPool(Execute);
00020     }
```

6.23.2.4 Finish()

```
virtual void MergeIt.Core.Commands.Command.Finish () [inline], [protected], [virtual]
```

Definition at line 27 of file [Command.cs](#).

```
00028     {
00029         Finished?.Invoke(this);
00030     }
```

6.23.2.5 OnDispose()

```
virtual void MergeIt.Core.Commands.Command.OnDispose () [inline], [protected], [virtual]
```

Definition at line 32 of file [Command.cs](#).

```
00033     {
00034
00035     }
```

6.23.3 Event Documentation

6.23.3.1 Finished

```
Action<ICommand> MergeIt.Core.Commands.Command.Finished
```

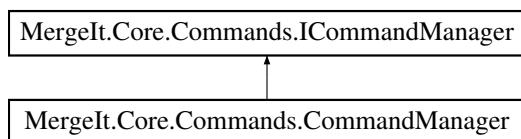
Definition at line 10 of file [Command.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Commands/Command.cs

6.24 Mergelt.Core.Commands.CommandManager Class Reference

Inheritance diagram for Mergelt.Core.Commands.CommandManager:



Public Member Functions

- void [Run](#) ()
- void [RunSimultaneously](#) ()
- async UniTask [RunAsync](#) ()
- void [Add](#) ([ICommand](#) command)

Properties

- bool [Executing](#) [get]

Properties inherited from [Mergelt.Core.Commands.ICommandManager](#)

Events

- Action< [ICommandManager](#) > [Finished](#)

Events inherited from [Mergelt.Core.Commands.ICommandManager](#)

- Action< [ICommandManager](#) > [Finished](#)

6.24.1 Detailed Description

Definition at line 11 of file [CommandManager.cs](#).

6.24.2 Member Function Documentation

6.24.2.1 Add()

```
void MergeIt.Core.Commands.CommandManager.Add (
     ICommand command) [inline]
```

Implements [Mergelt.Core.Commands.ICommandManager](#).

Definition at line 55 of file [CommandManager.cs](#).

```
00056     {
00057         _commands.Enqueue(command);
00058     }
```

6.24.2.2 Run()

```
void MergeIt.Core.Commands.CommandManager.Run () [inline]
```

Implements [Mergelt.Core.Commands.ICommandManager](#).

Definition at line 19 of file [CommandManager.cs](#).

```
00020     {
00021         Executing = true;
00022         RunNext();
00023     }
```

6.24.2.3 RunAsync()

```
async UniTask MergeIt.Core.Commands.CommandManager.RunAsync ( ) [inline]
```

Definition at line 43 of file [CommandManager.cs](#).

```
00044     {
00045         Executing = true;
00046         foreach ( ICommand command in _commands)
00047         {
00048             _currentCommand = command;
00049             await _currentCommand.ExecuteAsync();
00050         }
00051     }
00052     Finish();
00053 }
```

6.24.2.4 RunSimultaneously()

```
void MergeIt.Core.Commands.CommandManager.RunSimultaneously ( ) [inline]
```

Implements [Mergelt.Core.Commands.ICommandManager](#).

Definition at line 25 of file [CommandManager.cs](#).

```
00026     {
00027         Executing = true;
00028         if (_commands.Count > 0)
00029         {
00030             while (_commands.Count > 0)
00031             {
00032                 var currentCommand = _commands.Dequeue();
00033                 currentCommand.Finished += OnSimultaneousCommandFinished;
00034                 currentCommand.Execute();
00035             }
00036         }
00037         else
00038         {
00039             Finish();
00040         }
00041     }
```

6.24.3 Property Documentation

6.24.3.1 Executing

```
bool MergeIt.Core.Commands.CommandManager.Executing [get]
```

Implements [Mergelt.Core.Commands.ICommandManager](#).

Definition at line 14 of file [CommandManager.cs](#).

```
00014 { get; private set; }
```

6.24.4 Event Documentation

6.24.4.1 Finished

```
Action<ICommandManager> MergeIt.Core.Commands.CommandManager.Finished
```

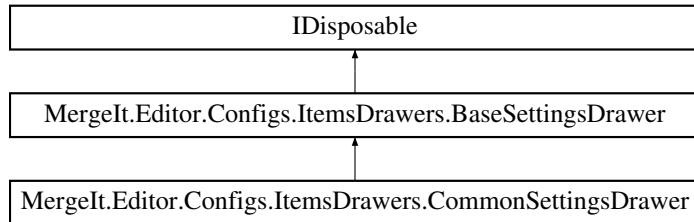
Definition at line 13 of file [CommandManager.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Commands/CommandManager.cs

6.25 MergeIt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer Class Reference

Inheritance diagram for MergeIt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer:



Public Member Functions

- [CommonSettingsDrawer](#) (VisualElement parent, [ElementConfig](#) config)
- override void [Draw](#) ()
- override void [Dispose](#) ()

Public Member Functions inherited from [MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer](#)

- [BaseSettingsDrawer](#) (VisualElement parent, [ElementConfig](#) config)
- void [Draw](#) ()
- virtual void [Hide](#) ()
- virtual void [Show](#) ()

Protected Member Functions

- override void [OnExpand](#) (bool expand)
- override VisualTreeAsset [GetUxml](#) ()

Protected Member Functions inherited from [MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer](#)

- void [Expand](#) (bool expand)
- void [OnExpand](#) (bool expand)
- void [RotateArrow](#) (bool rotate)
- VisualTreeAsset [GetUxml](#) ()

Additional Inherited Members

Protected Attributes inherited from [MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer](#)

- readonly VisualElement [Parent](#)
- readonly VisualElement [Root](#)
- readonly [ElementConfig](#) [Config](#)
- Label [Arrow](#)

6.25.1 Detailed Description

Definition at line 12 of file [CommonSettingsDrawer.cs](#).

6.25.2 Constructor & Destructor Documentation

6.25.2.1 CommonSettingsDrawer()

```
MergeIt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer.CommonSettingsDrawer (
    VisualElement parent,
    ElementConfig config) [inline]
```

Definition at line 21 of file [CommonSettingsDrawer.cs](#).

```
00021      : base(parent, config)
00022      {
00023          _settings = config.CommonSettings;
00024      }
```

6.25.3 Member Function Documentation

6.25.3.1 Dispose()

```
override void MergeIt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer.Dispose () [inline],
[virtual]
```

Reimplemented from [Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer](#).

Definition at line 61 of file [CommonSettingsDrawer.cs](#).

```
00062      {
00063          base.Dispose ();
00064
00065          _consumable.UnregisterValueChangedCallback(OnConsumableChanged);
00066          _expanded.UnregisterValueChangedCallback(OnExpandedChanged);
00067          _icon.UnregisterValueChangedCallback(OnImageChanged);
00068
00069          Parent.Remove(Root);
00070      }
```

6.25.3.2 Draw()

```
override void MergeIt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer.Draw () [inline]
```

Definition at line 26 of file [CommonSettingsDrawer.cs](#).

```
00027      {
00028          _icon = Root.Q<ObjectField>("IconContainer");
00029          _icon.objectType = typeof(FieldElementIconComponent);
00030          _icon.RegisterValueChangedCallback(OnImageChanged);
00031
00032          _commonSettings = Root.Q<Box>("CommonSettings");
00033
00034          var sellParams = Root.Q<CurrencyComponent>("SellParameters");
00035          sellParams.Bind(new SerializedObject(Config));
00036
00037          var splitParams = Root.Q<CurrencyComponent>("SplitParameters");
00038          splitParams.Bind(new SerializedObject(Config));
00039
00040          var unlockParams = Root.Q<CurrencyComponent>("UnlockParameters");
00041          unlockParams.Bind(new SerializedObject(Config));
00042
00043          var consumableContainer = Root.Q<VisualElement>("ConsumableContainer");
00044          consumableContainer.style.display =
00045              _settings.IsConsumable ? StylesConstants.DisplayFlex : StylesConstants.DisplayNone;
```

```

00046     _consumable = Root.Q<Toggle>("IsConsumable");
00047     _consumable.RegisterValueChangedCallback(OnConsumableChanged);
00048
00049     _expanded = Root.Q<Toggle>("CommonSettingsFoldout");
00050     _expanded.value = _settings.Expanded;
00051     _expanded.RegisterValueChangedCallback(OnExpandedChanged);
00052
00053     Arrow = Root.Q<Label>("Arrow");
00054
00055     SetImagePlaceholder(Config.CommonSettings.Icon);
00056
00057     Expand(_settings.Expanded);
00058
00059 }
```

6.25.3.3 GetUxml()

```
override VisualTreeAsset MergeIt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer.GetUxml ( )
[inline], [protected]
```

Definition at line 112 of file [CommonSettingsDrawer.cs](#).

```

00113     {
00114         return
00115             (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.InspectorResourcesRoot,
00116                 "CommonInspector.uxml"));
00116     }
```

6.25.3.4 OnExpand()

```
override void MergeIt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer.OnExpand (
    bool expand) [inline], [protected]
```

Definition at line 83 of file [CommonSettingsDrawer.cs](#).

```

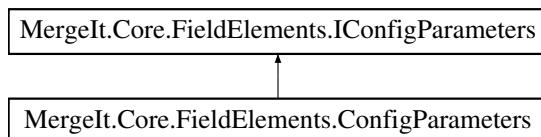
00084     {
00085         _settings.Expanded = expand;
00086         _commonSettings.style.display = expand ? StylesConstants.DisplayFlex :
00087             StylesConstants.DisplayNone;
00087     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Editor/Core/Configs/ItemsDrawers/CommonSettingsDrawer.cs

6.26 MergeIt.Core.FieldElements.ConfigParameters Class Reference

Inheritance diagram for MergeIt.Core.FieldElements.ConfigParameters:



Properties

- [ElementConfig ElementConfig](#) [get, set]
- [EvolutionData EvolutionData](#) [get, set]

Properties inherited from [Mergelt.Core.FieldElements.IConfigParameters](#)

6.26.1 Detailed Description

Definition at line 8 of file [ConfigParameters.cs](#).

6.26.2 Property Documentation

6.26.2.1 ElementConfig

`ElementConfig` MergeIt.Core.FieldElements.ConfigParameters.ElementConfig [get], [set]

Implements [Mergelt.Core.FieldElements.IConfigParameters](#).

Definition at line 10 of file [ConfigParameters.cs](#).

```
00010 { get; set; }
```

6.26.2.2 EvolutionData

`EvolutionData` MergeIt.Core.FieldElements.ConfigParameters.EvolutionData [get], [set]

Implements [Mergelt.Core.FieldElements.IConfigParameters](#).

Definition at line 11 of file [ConfigParameters.cs](#).

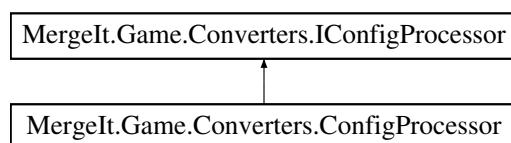
```
00011 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/Parameters/ConfigParameters.cs

6.27 Mergelt.Game.Converters.ConfigProcessor Class Reference

Inheritance diagram for Mergelt.Game.Converters.ConfigProcessor:



Public Member Functions

- `FieldData Convert (LevelConfig levelConfig)`
- `FieldData BuildLevel ()`
- `FieldElementData ConvertToFieldElementData (LevelElementData levelElementData)`
- `FieldElementData ConvertToFieldElementData (IFieldElement fieldElement)`
- `IFieldElement ConvertToFieldElement (FieldElementData data)`

6.27.1 Detailed Description

Definition at line 17 of file [ConfigProcessor.cs](#).

6.27.2 Member Function Documentation

6.27.2.1 BuildLevel()

```
FieldData MergeIt.Game.Converters.ConfigProcessor.BuildLevel ( ) [inline]
```

Implements [MergeIt.Game.Converters.IConfigProcessor](#).

Definition at line 43 of file [ConfigProcessor.cs](#).

```
00044     {
00045         var levelConfig = _configsService.LevelConfig;
00046         var elements = _fieldLogicModel.FieldElements;
00047
00048         var saveFieldData = new FieldData
00049         {
00050             FieldHeight = levelConfig.FieldHeight,
00051             FieldWidth = levelConfig.FieldWidth,
00052             SavedElementsData = new List<FieldElementData>()
00053         };
00054
00055         foreach (IFieldElement fieldElement in elements.Values)
00056         {
00057             var savedElementData = ConvertToFieldElementData(fieldElement);
00058             saveFieldData.SavedElementsData.Add(savedElementData);
00059         }
00060
00061         return saveFieldData;
00062     }
```

6.27.2.2 Convert()

```
FieldData MergeIt.Game.Converters.ConfigProcessor.Convert (
    LevelConfig levelConfig ) [inline]
```

Implements [MergeIt.Game.Converters.IConfigProcessor](#).

Definition at line 25 of file [ConfigProcessor.cs](#).

```
00026     {
00027         var saveFieldData = new FieldData
00028         {
00029             FieldHeight = levelConfig.FieldHeight,
00030             FieldWidth = levelConfig.FieldWidth,
00031             SavedElementsData = new List<FieldElementData>()
00032         };
00033
00034         foreach (LevelElementData levelElementData in levelConfig.FieldElementsData)
00035         {
00036             FieldElementData savedElementData = ConvertToFieldElementData(levelElementData);
00037             saveFieldData.SavedElementsData.Add(savedElementData);
00038         }
00039
00040         return saveFieldData;
00041     }
```

6.27.2.3 ConvertToFieldElement()

```
IFieldElement MergeIt.Game.Converters.ConfigProcessor.ConvertToFieldElement (
    FieldElementData data) [inline]
```

Implements [Mergelt.Game.Converters.IConfigProcessor](#).

Definition at line 157 of file [ConfigProcessor.cs](#).

```
00158     {
00159         ElementConfig elementConfig = _configsService.GetConfig(data.ConfigParameters.ElementId);
00160
00161         IFIELDELEMENT fieldElement = new FieldElement();
00162
00163         fieldElement.ConfigParameters = CreateConfigParameters(data.ConfigParameters);
00164         fieldElement.InfoParameters = CreateInfoParameters(data.InfoParameters, elementConfig);
00165         fieldElement.ProducedByParameters = CreateProducedByParameters(elementConfig);
00166
00167         switch (elementConfig.Type)
00168         {
00169             case ElementType.Generator:
00170                 ElementGeneratorSettings generatorSettings = elementConfig.GeneratorSettings;
00171                 fieldElement.GeneratorParameters =
00172                     CreateGeneratorParameters(data.GeneratorParameters);
00173                 fieldElement.ProduceParameters = CreateProduceParameters(generatorSettings);
00174
00175                 if (data.GeneratorOpenParameters != null &&
00176                     generatorSettings.NeedOpen)
00177                 {
00178                     fieldElement.GeneratorOpenParameters =
00179                         CreateGeneratorOpenParameters(data.GeneratorOpenParameters);
00180
00181                     fieldElement.UpdateGenerator();
00182                     break;
00183
00184             }
00185         }
00186
00187         return fieldElement;
00188     }
```

6.27.2.4 ConvertToFieldElementData() [1/2]

```
FieldElementData MergeIt.Game.Converters.ConfigProcessor.ConvertToFieldElementData (
    IFIELDELEMENT fieldElement) [inline]
```

Implements [Mergelt.Game.Converters.IConfigProcessor](#).

Definition at line 116 of file [ConfigProcessor.cs](#).

```
00117     {
00118         var fieldElementData = new FieldElementData
00119         {
00120             ConfigParameters = new SavedConfigParameters
00121             {
00122                 ElementId = fieldElement.ConfigParameters.ElementConfig.Id,
00123                 EvolutionId = fieldElement.ConfigParameters.EvolutionData.Id
00124             },
00125
00126             InfoParameters = new SavedInfoParameters
00127             {
00128                 LogicPosition = fieldElement.InfoParameters.LogicPosition,
00129                 IsBlocked = fieldElement.InfoParameters.IsBlocked
00130             };
00131
00132             if (fieldElement.GeneratorParameters != null)
00133             {
00134                 var generatorParameters = new SavedGeneratorParameters
00135                 {
00136                     AvailableToDelete = fieldElement.GeneratorParameters.AvailableToDelete,
00137                     StartChargingTime = fieldElement.GeneratorParameters.StartChargingTime,
00138                     DroppedElements = fieldElement.GeneratorParameters.DroppedElements,
00139                     ChargedCount = fieldElement.GeneratorParameters.ChargedCount,
00140                 };
00141
00142                 fieldElementData.GeneratorParameters = generatorParameters;
00143
00144                 if (fieldElement.GeneratorOpenParameters != null)
```

```

00146             {
00147                 fieldElementData.GeneratorOpenParameters = new SavedGeneratorOpenParameters
00148                 {
00149                     StartOpeningTime = fieldElement.GeneratorOpenParameters.StartOpeningTime
00150                 };
00151             }
00152         }
00153     }
00154     return fieldElementData;
00155 }
```

6.27.2.5 ConvertToFieldElementData() [2/2]

```
FieldElementData MergeIt.Game.Converters.ConfigProcessor.ConvertToFieldElementData (
    LevelElementData levelElementData ) [inline]
```

Implements [Mergelt.Game.Converters.IConfigProcessor](#).

Definition at line 64 of file [ConfigProcessor.cs](#).

```

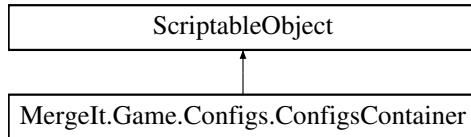
00065         {
00066             ElementConfig elementConfig = levelElementData.Element;
00067             FieldElementData fieldElementData = null;
00068
00069             if (elementConfig != null)
00070             {
00071                 fieldElementData = new FieldElementData
00072                 {
00073                     InfoParameters = new SavedInfoParameters
00074                     {
00075                         LogicPosition = levelElementData.Position,
00076                         IsBlocked = levelElementData.IsBlocked
00077                     },
00078                     ConfigParameters = new SavedConfigParameters
00079                     {
00080                         ElementId = levelElementData.Element.Id,
00081                         EvolutionId = levelElementData.EvolutionId
00082                     }
00083                 };
00084
00085                 switch (elementConfig.Type)
00086                 {
00087                     case ElementType.Generator:
00088                         var generatorParameters = new SavedGeneratorParameters
00089                         {
00090                             DroppedElements = 0
00091                         };
00092
00093                         if (elementConfig.GeneratorSettings.Charged)
00094                         {
00095                             generatorParameters.AvailableToDrop =
00096                             elementConfig.GeneratorSettings.MaxDrop;
00097                         }
00098                         else
00099                         {
00100                             generatorParameters.StartChargingTime =
00101                             DateTimeOffset.UtcNow.UnixTimeSeconds();
00102                         }
00103
00104                         fieldElementData.GeneratorParameters = generatorParameters;
00105
00106                         if (elementConfig.GeneratorSettings.NeedOpen)
00107                         {
00108                             fieldElementData.GeneratorOpenParameters = new
00109                             SavedGeneratorOpenParameters();
00110                         }
00111                         break;
00112                     }
00113                 return fieldElementData;
00114 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Converters/ConfigProcessor.cs

6.28 Mergelt.Game.Configs.ConfigsContainer Class Reference

Inheritance diagram for Mergelt.Game.Configs.ConfigsContainer:



Properties

- `ElementPrefab[] Prefabs [get]`
- `GameConfig GameConfig [get]`
- `GamelconsConfig IconsConfig [get]`
- `InventoryConfig InventoryConfig [get]`
- `LevelConfig LevelConfig [get]`
- `LevelUpConfig LevelUpConfig [get]`
- `EffectConfig[] EffectConfigs [get]`
- `HintsConfig HintsConfig [get]`

6.28.1 Detailed Description

Definition at line 14 of file [ConfigsContainer.cs](#).

6.28.2 Property Documentation

6.28.2.1 EffectConfigs

`EffectConfig [] MergeIt.Game.Configs.ConfigsContainer.EffectConfigs [get]`

Definition at line 70 of file [ConfigsContainer.cs](#).

```
00071      {
00072          get => _effectConfigs;
00073      }
```

6.28.2.2 GameConfig

`GameConfig MergeIt.Game.Configs.ConfigsContainer.GameConfig [get]`

Definition at line 45 of file [ConfigsContainer.cs](#).

```
00046      {
00047          get => _gameConfig;
00048      }
```

6.28.2.3 HintsConfig

`HintsConfig MergeIt.Game.Configs.ConfigsContainer.HintsConfig [get]`

Definition at line 75 of file [ConfigsContainer.cs](#).

```
00076      {
00077          get => _hintsConfig;
00078      }
```

6.28.2.4 IconsConfig

```
GameIconsConfig MergeIt.Game.Configs.ConfigsContainer(IconsConfig) [get]
```

Definition at line 50 of file [ConfigsContainer.cs](#).

```
00051      {  
00052          get => _iconsConfig;  
00053      }
```

6.28.2.5 InventoryConfig

```
InventoryConfig MergeIt.Game.Configs.ConfigsContainer.InventoryConfig [get]
```

Definition at line 55 of file [ConfigsContainer.cs](#).

```
00056      {  
00057          get => _inventoryConfig;  
00058      }
```

6.28.2.6 LevelConfig

```
LevelConfig MergeIt.Game.Configs.ConfigsContainer.LevelConfig [get]
```

Definition at line 60 of file [ConfigsContainer.cs](#).

```
00061      {  
00062          get => _levelConfig;  
00063      }
```

6.28.2.7 LevelUpConfig

```
LevelUpConfig MergeIt.Game.Configs.ConfigsContainer.LevelUpConfig [get]
```

Definition at line 65 of file [ConfigsContainer.cs](#).

```
00066      {  
00067          get => _levelUpConfig;  
00068      }
```

6.28.2.8 Prefabs

```
ElementPrefab [] MergeIt.Game.Configs.ConfigsContainer.Prefabs [get]
```

Definition at line 40 of file [ConfigsContainer.cs](#).

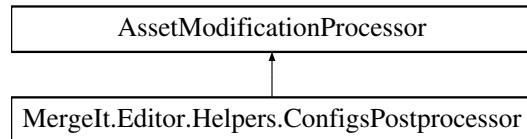
```
00041      {  
00042          get => _prefabs;  
00043      }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Configs/ConfigsContainer.cs

6.29 Mergelt.Editor.Helpers.ConfigsPostprocessor Class Reference

Inheritance diagram for Mergelt.Editor.Helpers.ConfigsPostprocessor:



6.29.1 Detailed Description

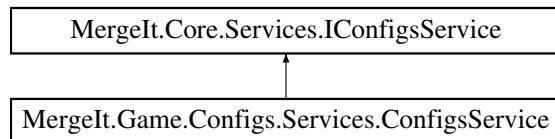
Definition at line 12 of file [ConfigsPostprocessor.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Helpers/ConfigsPostprocessor.cs

6.30 Mergelt.Game.Configs.Services.ConfigsService Class Reference

Inheritance diagram for Mergelt.Game.Configs.Services.ConfigsService:



Public Member Functions

- async UniTask [Load \(\)](#)
- Sprite [GetCurrencyIcon \(CurrencyType type\)](#)
- [EvolutionData GetEvolutionData \(string evolutionId\)](#)
- string [GetEvolutionIdByElement \(ElementConfig element\)](#)
- [EvolutionData GetEvolutionByElement \(ElementConfig element\)](#)
- T [GetEffectConfig< T > \(string type\)](#)
- [LevelUpParameters GetLevelUpData \(int currentLevel\)](#)
- [IFieldElementView GetElementPrefab \(ElementType type\)](#)
- IEnumerable< ElementConfig > [GetConfigs \(Func< ElementConfig, bool > predicate=null\)](#)
- [ElementConfig GetConfig \(string id\)](#)

Properties

- [LevelConfig LevelConfig \[get\]](#)
- [InventoryConfig InventoryConfig \[get\]](#)
- [GameConfig GameConfig \[get\]](#)
- [HintsConfig HintsConfig \[get\]](#)

Properties inherited from [Mergelt.Core.Services.IConfigsService](#)

6.30.1 Detailed Description

Definition at line 22 of file [ConfigsService.cs](#).

6.30.2 Member Function Documentation

6.30.2.1 GetConfig()

```
ElementConfig MergeIt.Game.Configs.Services.ConfigsService.GetConfig (
    string id) [inline]
```

Implements [Mergelt.Core.Services.IConfigsService](#).

Definition at line 123 of file [ConfigsService.cs](#).

```
00124     {
00125         return _model.ElementConfigs.FirstOrDefault(x => x.Id == id);
00126     }
```

6.30.2.2 GetConfigs()

```
IEnumerable< ElementConfig > MergeIt.Game.Configs.Services.ConfigsService.GetConfigs (
    Func< ElementConfig, bool > predicate = null) [inline]
```

Implements [Mergelt.Core.Services.IConfigsService](#).

Definition at line 113 of file [ConfigsService.cs](#).

```
00114     {
00115         if (predicate != null)
00116         {
00117             return _model.ElementConfigs.Where(predicate);
00118         }
00119
00120         return _model.ElementConfigs;
00121     }
```

6.30.2.3 GetCurrencyIcon()

```
Sprite MergeIt.Game.Configs.Services.ConfigsService.GetCurrencyIcon (
    CurrencyType type) [inline]
```

Implements [Mergelt.Core.Services.IConfigsService](#).

Definition at line 72 of file [ConfigsService.cs](#).

```
00073     {
00074         Sprite icon = _model(IconsConfig.InfoPanelCurrencyIcons.FirstOrDefault(x => x.CurrencyType
00075 == type)?.CurrencyIcon;
00076
00077         return icon;
00078     }
```

6.30.2.4 GetEffectConfig< T >()

```
T MergeIt.Game.Configs.Services.ConfigsService.GetEffectConfig< T > (
    string type ) [inline]
```

Implements [MergeIt.Core.Services.IConfigsService](#).

Type Constraints

T : EffectConfig

Definition at line 94 of file [ConfigsService.cs](#).

```
00094             : EffectConfig
00095         {
00096             _model.EffectConfigs.TryGetValue(type, out EffectConfig effectConfig);
00097             return effectConfig as T;
00098         }
00099 }
```

6.30.2.5 GetElementPrefab()

```
IFieldElementView MergeIt.Game.Configs.Services.ConfigsService.GetElementPrefab (
    ElementType type ) [inline]
```

Implements [MergeIt.Core.Services.IConfigsService](#).

Definition at line 106 of file [ConfigsService.cs](#).

```
00107             :
00108         _model.Prefabs.TryGetValue(type, out FieldElementView view);
00109         return view;
00110     }
00111 }
```

6.30.2.6 GetEvolutionByElement()

```
EvolutionData MergeIt.Game.Configs.Services.ConfigsService.GetEvolutionByElement (
    ElementConfig element ) [inline]
```

Implements [MergeIt.Core.Services.IConfigsService](#).

Definition at line 89 of file [ConfigsService.cs](#).

```
00090             :
00091             return _model.LevelConfig.EvolutionsScheme.Evolution.FirstOrDefault(x =>
00092                 x.Chain.Contains(element));
00093 }
```

6.30.2.7 GetEvolutionData()

```
EvolutionData MergeIt.Game.Configs.Services.ConfigsService.GetEvolutionData (
    string evolutionId ) [inline]
```

Implements [MergeIt.Core.Services.IConfigsService](#).

Definition at line 79 of file [ConfigsService.cs](#).

```
00080             :
00081             return _model.LevelConfig.EvolutionsScheme.Evolution.FirstOrDefault(x => x.Id ==
00082                 evolutionId);
00083 }
```

6.30.2.8 GetEvolutionIdByElement()

```
string MergeIt.Game.Configs.Services.ConfigsService.GetEvolutionIdByElement (
    ElementConfig element) [inline]
```

Implements [MergIt.Core.Services.IConfigsService](#).

Definition at line 84 of file [ConfigsService.cs](#).

```
00085     {
00086         return GetEvolutionByElement(element)?.Id;
00087     }
```

6.30.2.9 GetLevelUpData()

```
LevelUpParameters MergeIt.Game.Configs.Services.ConfigsService.GetLevelUpData (
    int currentLevel) [inline]
```

Implements [MergIt.Core.Services.IConfigsService](#).

Definition at line 101 of file [ConfigsService.cs](#).

```
00102     {
00103         return _model.LevelUpConfig.LevelUp[currentLevel - 1];
00104     }
```

6.30.2.10 Load()

```
async UniTask MergeIt.Game.Configs.Services.ConfigsService.Load () [inline]
```

Implements [MergIt.Core.Services.IConfigsService](#).

Definition at line 48 of file [ConfigsService.cs](#).

```
00049     {
00050         var requestResult = await
00051             Resources.LoadAsync<ConfigsContainer>(ConfigsConstants.ConfigsPath);
00052         _configsContainer = requestResult as ConfigsContainer;
00053
00054         if (_configsContainer)
00055         {
00056             _model.StoreElementsPrefabs(_configsContainer.Prefabs);
00057             _model.StoreGameConfig(_configsContainer.GameConfig);
00058             _model.StoreIconsConfig(_configsContainer(IconsConfig));
00059             _model.StoreInventoryConfig(_configsContainer.InventoryConfig);
00060             _model.StoreLevelConfig(_configsContainer.LevelConfig);
00061             _model.StoreLevelUpConfig(_configsContainer.LevelUpConfig);
00062             _model.StoreEffectsConfigs(_configsContainer.EffectConfigs);
00063             _model.StoreHintsConfig(_configsContainer.HintsConfig);
00064         }
00065         else
00066         {
00067             Debug.Log("Something went wrong while loading configs container.");
00068         }
00069     }
```

6.30.3 Property Documentation

6.30.3.1 GameConfig

```
GameConfig MergeIt.Game.Configs.Services.ConfigsService.GameConfig [get]
```

Implements [MergIt.Core.Services.IConfigsService](#).

Definition at line 38 of file [ConfigsService.cs](#).

```
00039     {
00040         get => _model.GameConfig;
00041     }
```

6.30.3.2 HintsConfig

`HintsConfig MergeIt.Game.Configs.Services.ConfigsService.HintsConfig [get]`

Implements [Mergelt.Core.Services.IConfigsService](#).

Definition at line 43 of file [ConfigsService.cs](#).

```
00044      {  
00045          get => _model.HintsConfig;  
00046      }
```

6.30.3.3 InventoryConfig

`InventoryConfig MergeIt.Game.Configs.Services.ConfigsService.InventoryConfig [get]`

Implements [Mergelt.Core.Services.IConfigsService](#).

Definition at line 33 of file [ConfigsService.cs](#).

```
00034      {  
00035          get => _model.InventoryConfig;  
00036      }
```

6.30.3.4 LevelConfig

`LevelConfig MergeIt.Game.Configs.Services.ConfigsService.LevelConfig [get]`

Implements [Mergelt.Core.Services.IConfigsService](#).

Definition at line 28 of file [ConfigsService.cs](#).

```
00029      {  
00030          get => _model.LevelConfig;  
00031      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Configs/Services/ConfigsService.cs

6.31 Mergelt.Game.Configs.Services.ConfigsServiceModel Class Reference

Public Member Functions

- void [StoreElementsPrefabs](#) ([ElementPrefab](#)[] prefabs)
- void [StoreGameConfig](#) ([GameConfig](#) config)
- void [StoreIconsConfig](#) ([GameIconsConfig](#) config)
- void [StoreInventoryConfig](#) ([InventoryConfig](#) config)
- void [StoreLevelConfig](#) ([LevelConfig](#) config)
- void [StoreLevelUpConfig](#) ([LevelUpConfig](#) config)
- void [StoreEffectsConfigs](#) ([EffectConfig](#)[] configs)
- void [StoreHintsConfig](#) ([HintsConfig](#) hintsConfig)

Properties

- GameConfig GameConfig [get]
- GameIconsConfig IconsConfig [get]
- InventoryConfig InventoryConfig [get]
- LevelConfig LevelConfig [get]
- LevelUpConfig LevelUpConfig [get]
- Dictionary< string, EffectConfig > EffectConfigs [get]
- List< ElementConfig > ElementConfigs [get]
- Dictionary< ElementType, FieldElementView > Prefabs [get]
- HintsConfig HintsConfig [get]

6.31.1 Detailed Description

Definition at line 18 of file [ConfigsServiceModel.cs](#).

6.31.2 Member Function Documentation

6.31.2.1 StoreEffectsConfigs()

```
void MergeIt.Game.Configs.Services.ConfigsServiceModel.StoreEffectsConfigs (
    EffectConfig[] configs) [inline]
```

Definition at line 107 of file [ConfigsServiceModel.cs](#).

```
00108     {
00109         _effectConfigs = configs.ToDictionary(config => config.Name, config => config);
00110     }
```

6.31.2.2 StoreElementsPrefabs()

```
void MergeIt.Game.Configs.Services.ConfigsServiceModel.StoreElementsPrefabs (
    ElementPrefab[] prefabs) [inline]
```

Definition at line 75 of file [ConfigsServiceModel.cs](#).

```
00076     {
00077         _elementsPrefabs = prefabs.ToDictionary(key => key.Type, value => value.Prefab);
00078     }
```

6.31.2.3 StoreGameConfig()

```
void MergeIt.Game.Configs.Services.ConfigsServiceModel.StoreGameConfig (
    GameConfig config) [inline]
```

Definition at line 80 of file [ConfigsServiceModel.cs](#).

```
00081     {
00082         _gameConfig = config;
00083     }
```

6.31.2.4 StoreHintsConfig()

```
void MergeIt.Game.Configs.Services.ConfigsServiceModel.StoreHintsConfig (
    HintsConfig hintsConfig) [inline]
```

Definition at line 127 of file [ConfigsServiceModel.cs](#).

```
00128     {
00129         _hintsConfig = hintsConfig;
00130     }
```

6.31.2.5 StoreIconsConfig()

```
void MergeIt.Game.Configs.Services.ConfigsServiceModel.StoreIconsConfig (
    GameIconsConfig config) [inline]
```

Definition at line 85 of file [ConfigsServiceModel.cs](#).

```
00086     {
00087         _iconsConfig = config;
00088     }
```

6.31.2.6 StoreInventoryConfig()

```
void MergeIt.Game.Configs.Services.ConfigsServiceModel.StoreInventoryConfig (
    InventoryConfig config) [inline]
```

Definition at line 90 of file [ConfigsServiceModel.cs](#).

```
00091     {
00092         _inventoryConfig = config;
00093     }
```

6.31.2.7 StoreLevelConfig()

```
void MergeIt.Game.Configs.Services.ConfigsServiceModel.StoreLevelConfig (
    LevelConfig config) [inline]
```

Definition at line 95 of file [ConfigsServiceModel.cs](#).

```
00096     {
00097         _levelConfig = config;
00098         ExtractElementsConfigs(_levelConfig);
00099     }
```

6.31.2.8 StoreLevelUpConfig()

```
void MergeIt.Game.Configs.Services.ConfigsServiceModel.StoreLevelUpConfig (
    LevelUpConfig config) [inline]
```

Definition at line 102 of file [ConfigsServiceModel.cs](#).

```
00103     {
00104         _levelUpConfig = config;
00105     }
```

6.31.3 Property Documentation

6.31.3.1 EffectConfigs

```
Dictionary<string, EffectConfig> MergeIt.Game.Configs.Services.ConfigsServiceModel.EffectConfigs [get]
```

Definition at line 55 of file [ConfigsServiceModel.cs](#).

```
00056     {
00057         get => _effectConfigs;
00058     }
```

6.31.3.2 ElementConfigs

```
List<ElementConfig> MergeIt.Game.Configs.Services.ConfigsServiceModel.ElementConfigs [get]
```

Definition at line 60 of file [ConfigsServiceModel.cs](#).

```
00061     {
00062         get => _elementConfigs;
00063     }
```

6.31.3.3 GameConfig

```
GameConfig MergeIt.Game.Configs.Services.ConfigsServiceModel.GameConfig [get]
```

Definition at line 30 of file [ConfigsServiceModel.cs](#).

```
00031     {
00032         get => _gameConfig;
00033     }
```

6.31.3.4 HintsConfig

```
HintsConfig MergeIt.Game.Configs.Services.ConfigsServiceModel.HintsConfig [get]
```

Definition at line 70 of file [ConfigsServiceModel.cs](#).

```
00071     {
00072         get => _hintsConfig;
00073     }
```

6.31.3.5 IconsConfig

```
GameIconsConfig MergeIt.Game.Configs.Services.ConfigsServiceModel.IconsConfig [get]
```

Definition at line 35 of file [ConfigsServiceModel.cs](#).

```
00036     {
00037         get => _iconsConfig;
00038     }
```

6.31.3.6 InventoryConfig

```
InventoryConfig MergeIt.Game.Configs.Services.ConfigsServiceModel.InventoryConfig [get]
```

Definition at line 40 of file [ConfigsServiceModel.cs](#).

```
00041     {
00042         get => _inventoryConfig;
00043     }
```

6.31.3.7 LevelConfig

```
LevelConfig MergeIt.Game.Configs.Services.ConfigsServiceModel.LevelConfig [get]
```

Definition at line 45 of file [ConfigsServiceModel.cs](#).

```
00046      {
00047          get => _levelConfig;
00048      }
```

6.31.3.8 LevelUpConfig

```
LevelUpConfig MergeIt.Game.Configs.Services.ConfigsServiceModel.LevelUpConfig [get]
```

Definition at line 50 of file [ConfigsServiceModel.cs](#).

```
00051      {
00052          get => _levelUpConfig;
00053      }
```

6.31.3.9 Prefabs

```
Dictionary<ElementType, FieldElementView> MergeIt.Game.Configs.Services.ConfigsServiceModel.Prefabs [get]
```

Definition at line 65 of file [ConfigsServiceModel.cs](#).

```
00066      {
00067          get => _elementsPrefabs;
00068      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Configs/Services/ConfigsServiceModel.cs

6.32 Mergelt.Core.Configs.Elements.ConsumableSettings Struct Reference

Properties

- ConsumableTargetType [Target](#) [get]
- ConsumableOperationType [Operation](#) [get]
- int [Value](#) [get]

6.32.1 Detailed Description

Definition at line 10 of file [ConsumableSettings.cs](#).

6.32.2 Property Documentation

6.32.2.1 Operation

```
ConsumableOperationType MergeIt.Core.Configs.Elements.ConsumableSettings.Operation [get]
```

Definition at line 21 of file [ConsumableSettings.cs](#).

```
00022      {
00023          get => _operation;
00024      }
```

6.32.2.2 Target

ConsumableTargetType MergeIt.Core.Configs.Elements.ConsumableSettings.Target [get]

Definition at line 16 of file [ConsumableSettings.cs](#).

```
00017     {
00018         get => _target;
00019     }
```

6.32.2.3 Value

int MergeIt.Core.Configs.Elements.ConsumableSettings.Value [get]

Definition at line 26 of file [ConsumableSettings.cs](#).

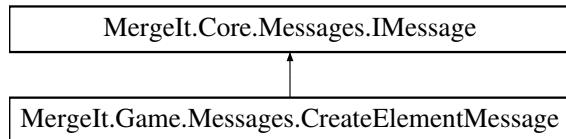
```
00027     {
00028         get => _value;
00029     }
```

The documentation for this struct was generated from the following file:

- Assets/MergeIt/Source/Core/Configs/Elements/Settings/ConsumableSettings.cs

6.33 MergeIt.Game.Messages.CreateElementMessage Class Reference

Inheritance diagram for MergeIt.Game.Messages.CreateElementMessage:



Properties

- [IFieldElement NewElement \[get, set\]](#)
- [Vector3? FromPosition \[get, set\]](#)
- [GridPoint ToPoint \[get, set\]](#)

6.33.1 Detailed Description

Definition at line 9 of file [CreateElementMessage.cs](#).

6.33.2 Property Documentation

6.33.2.1 FromPosition

`Vector3? MergeIt.Game.Messages.CreateElementMessage.FromPosition [get], [set]`

Definition at line 12 of file [CreateElementMessage.cs](#).

```
00012 { get; set; }
```

6.33.2.2 NewElement

```
IFieldElement MergeIt.Game.Messages.CreateElementMessage.NewElement [get], [set]
```

Definition at line 11 of file [CreateElementMessage.cs](#).
00011 { get; set; }

6.33.2.3 ToPoint

```
GridPoint MergeIt.Game.Messages.CreateElementMessageToPoint [get], [set]
```

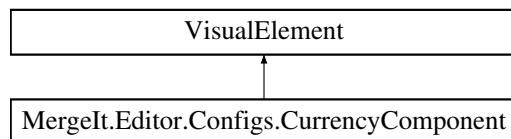
Definition at line 13 of file [CreateElementMessage.cs](#).
00013 { get; set; }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/CreateElementMessage.cs

6.34 Mergelt.Editor.Configs.CurrencyComponent Class Reference

Inheritance diagram for Mergelt.Editor.Configs.CurrencyComponent:



Classes

- class [UxmlFactory](#)

Properties

- string `Title` [get, set]
- string `TypeBinding` [get, set]
- string `AmountBinding` [get, set]

6.34.1 Detailed Description

Definition at line 10 of file [CurrencyComponent.cs](#).

6.34.2 Constructor & Destructor Documentation

6.34.2.1 CurrencyComponent()

```
MergeIt.Editor.Configs.CurrencyComponent.CurrencyComponent () [inline]
```

Definition at line 18 of file [CurrencyComponent.cs](#).

```
00019     {
00020         RegisterCallback<AttachToPanelEvent>(OnAttachedToPanel);
00021     }
```

6.34.3 Property Documentation

6.34.3.1 AmountBinding

```
string MergeIt.Editor.Configs.CurrencyComponent.AmountBinding [get], [set]
```

Definition at line 16 of file [CurrencyComponent.cs](#).

```
00016 { get; set; }
```

6.34.3.2 Title

```
string MergeIt.Editor.Configs.CurrencyComponent.Title [get], [set]
```

Definition at line 14 of file [CurrencyComponent.cs](#).

```
00014 { get; set; }
```

6.34.3.3 TypeBinding

```
string MergeIt.Editor.Configs.CurrencyComponent.TypeBinding [get], [set]
```

Definition at line 15 of file [CurrencyComponent.cs](#).

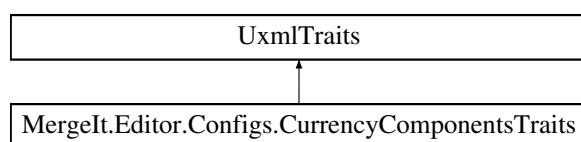
```
00015 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Editor/Core/Configs/Components/CurrencyComponent.cs

6.35 MergeIt.Editor.Configs.CurrencyComponentsTraits Class Reference

Inheritance diagram for MergeIt.Editor.Configs.CurrencyComponentsTraits:



Public Member Functions

- override void [Init](#) (VisualElement ve, IXmlAttributes bag, CreationContext cc)

6.35.1 Detailed Description

Definition at line 7 of file [CurrencyComponentsTraits.cs](#).

6.35.2 Member Function Documentation

6.35.2.1 Init()

```
override void MergeIt.Editor.Configs.CurrencyComponentsTraits.Init (
    VisualElement ve,
    IXmlAttributes bag,
    CreationContext cc ) [inline]
```

Definition at line 21 of file [CurrencyComponentsTraits.cs](#).

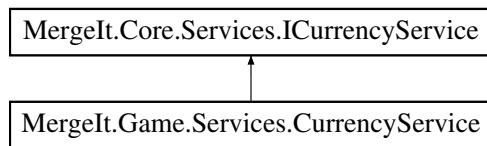
```
00022     {
00023         base.Init(ve, bag, cc);
00024
00025         if (ve is CurrencyComponent currencyParameters)
00026         {
00027             currencyParameters.name = _nameAttribute.GetValueFromBag(bag, cc);
00028             currencyParameters.Title = _titleAttribute.GetValueFromBag(bag, cc);
00029             currencyParameters.TypeBinding = _currencyTypeBindingAttribute.GetValueFromBag(bag,
00030                                         cc);
00031             currencyParameters.AmountBinding =
00032             _currencyAmountBindingAttribute.GetValueFromBag(bag, cc);
00031         }
00032     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/Configs/Components/CurrencyComponentsTraits.cs

6.36 Mergelt.Game.Services.CurrencyService Class Reference

Inheritance diagram for Mergelt.Game.Services.CurrencyService:



Public Member Functions

- bool [TryPay](#) ([CurrencySettings](#) currencySettings)
- void [Sell](#) ([CurrencySettings](#) currencySettings)

6.36.1 Detailed Description

Definition at line 10 of file [CurrencyService.cs](#).

6.36.2 Member Function Documentation

6.36.2.1 Sell()

```
void MergeIt.Game.Services.CurrencyService.Sell (
    CurrencySettings currencySettings ) [inline]
```

Implements [Mergelt.Core.Services.ICurrencyService](#).

Definition at line 60 of file [CurrencyService.cs](#).

```
00061     {
00062         switch (currencySettings.Currency)
00063     {
00064         case CurrencyType.Soft:
00065             _userServiceModel.SoftCurrency.Value += currencySettings.Amount;
00066             break;
00067
00068         case CurrencyType.Hard:
00069             _userServiceModel.HardCurrency.Value += currencySettings.Amount;
00070             break;
00071
00072         case CurrencyType.Splitter:
00073             _userServiceModel.Splitters.Value += currencySettings.Amount;
00074             break;
00075
00076         case CurrencyType.Energy:
00077             _userServiceModel.Energy.Value += currencySettings.Amount;
00078             break;
00079     }
00080 }
```

6.36.2.2 TryPay()

```
bool MergeIt.Game.Services.CurrencyService.TryPay (
    CurrencySettings currencySettings ) [inline]
```

Implements [Mergelt.Core.Services.ICurrencyService](#).

Definition at line 15 of file [CurrencyService.cs](#).

```
00016     {
00017         bool result = false;
00018         switch (currencySettings.Currency)
00019     {
00020         case CurrencyType.Soft:
00021             if (_userServiceModel.SoftCurrency.Value > currencySettings.Amount)
00022             {
00023                 _userServiceModel.SoftCurrency.Value -= currencySettings.Amount;
00024                 result = true;
00025             }
00026
00027             break;
00028
00029         case CurrencyType.Hard:
00030             if (_userServiceModel.HardCurrency.Value > currencySettings.Amount)
00031             {
00032                 _userServiceModel.HardCurrency.Value -= currencySettings.Amount;
00033                 result = true;
00034             }
00035
00036             break;
00037
00038         case CurrencyType.Splitter:
00039             if (_userServiceModel.Splitters.Value > currencySettings.Amount)
00040             {
00041                 _userServiceModel.Splitters.Value -= currencySettings.Amount;
```

```

00042             result = true;
00043         }
00044
00045         break;
00046
00047     case CurrencyType.Energy:
00048         if (_userServiceModel.Energy.Value > currencySettings.Amount)
00049     {
00050         _userServiceModel.Energy.Value -= currencySettings.Amount;
00051         result = true;
00052     }
00053
00054     break;
00055 }
00056
00057 return result;
00058 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/CurrencyService.cs

6.37 Mergelt.Core.Configs.Elements.CurrencySettings Struct Reference

Properties

- CurrencyType [Currency](#) [get]
- int [Amount](#) [get]

6.37.1 Detailed Description

Definition at line 10 of file [CurrencySettings.cs](#).

6.37.2 Property Documentation

6.37.2.1 Amount

```
int MergeIt.Core.Configs.Elements.CurrencySettings.Amount [get]
```

Definition at line 20 of file [CurrencySettings.cs](#).

```
00021     {
00022         get => _amount;
00023     }
```

6.37.2.2 Currency

```
CurrencyType MergeIt.Core.Configs.Elements.CurrencySettings.Currency [get]
```

Definition at line 15 of file [CurrencySettings.cs](#).

```
00016     {
00017         get => _currencyType;
00018     }
```

The documentation for this struct was generated from the following file:

- Assets/Mergelt/Source/Core/Configs/Elements/Settings/CurrencySettings.cs

6.38 MergeIt.SimpleDI.DiContainer Class Reference

Static Public Member Functions

- static `DiHandler Register< TInterface, TType >` (string key="")
- static `DiHandler Register< TType >` (string key="")
- static `DiHandler RegisterInterfacesFor< TType >` (string key="")
- static `TInterface Get< TInterface >` (string key="")
- static void `Drop< TInterface >` (string key="")
- static void `Clear ()`

6.38.1 Detailed Description

Definition at line 10 of file [DiContainer.cs](#).

6.38.2 Member Function Documentation

6.38.2.1 Clear()

```
static void MergeIt.SimpleDI.DiContainer.Clear () [inline], [static]
```

Definition at line 94 of file [DiContainer.cs](#).

```
00095         {
00096             foreach (var typesHandler in TypesHandlers)
00097             {
00098                 typesHandler.Value.Clear();
00099             }
00100
00101             TypesHandlers.Clear();
00102
00103             UpdatableObjects.Clear();
00104         }
```

6.38.2.2 Drop< TInterface >()

```
static void MergeIt.SimpleDI.DiContainer.Drop< TInterface > (
    string key = "") [inline], [static]
```

Type Constraints

TInterface : class

Definition at line 80 of file [DiContainer.cs](#).

```
00081         : class
00082     {
00083         Type type = typeof(TInterface);
00084
00085         if (!TypesHandlers.TryGetValue(type, out Dictionary<string, DiHandler>
dependencyHandlers))
00086         {
00087             Debug.LogWarning($"Binding for {type} was not found");
00088             return;
00089         }
00090
00091         dependencyHandlers.Remove(key);
00092     }
```

6.38.2.3 Get< TInterface >()

```
static TInterface MergeIt.SimpleDI.DiContainer.Get< TInterface > (
    string key = "" ) [inline], [static]
```

Type Constraints

TInterface : class

Definition at line 74 of file [DiContainer.cs](#).

```
00075         : class
00076     {
00077         return Get(typeof(TInterface), key) as TInterface;
00078     }
```

6.38.2.4 Register< TInterface, TType >()

```
static DiHandler MergeIt.SimpleDI.DiContainer.Register< TInterface, TType > (
    string key = "" ) [inline], [static]
```

Type Constraints

TInterface : class

TType : TInterface

Definition at line 31 of file [DiContainer.cs](#).

```
00032         : class
00033     where TType : TInterface
00034     {
00035         Type type = typeof(TInterface);
00036         var diHandler = new DiHandler();
00037         diHandler.AddDependency<TType>();
00038
00039         return Bind<TType>(type, diHandler, key);
00040     }
```

6.38.2.5 Register< TType >()

```
static DiHandler MergeIt.SimpleDI.DiContainer.Register< TType > (
    string key = "" ) [inline], [static]
```

Type Constraints

TType : class

Definition at line 42 of file [DiContainer.cs](#).

```
00043         : class
00044     {
00045         Type type = typeof(TType);
00046         var diHandler = new DiHandler();
00047         diHandler.AddDependency<TType>();
00048
00049         return Bind<TType>(type, diHandler, key);
00050     }
```

6.38.2.6 RegisterInterfacesFor< TType >()

```
static DiHandler MergeIt.SimpleDI.DiContainer.RegisterInterfacesFor< TType > (
    string key = "" ) [inline], [static]
```

Type Constraints

TType : class

Definition at line 52 of file [DiContainer.cs](#).

```
00053             : class
00054         {
00055             Type type = typeof(TType);
00056             Type[] interfaces = type.GetInterfaces();
00057
00058             var diHandler = new DiHandler();
00059             diHandler.AddDependency<TType>();
00060
00061             for (int i = 0; i < interfaces.Length; i++)
00062             {
00063                 Type interfaceType = interfaces[i];
00064
00065                 if (!ReservedInterfaces.Contains(interfaceType))
00066                 {
00067                     Bind<TType>(interfaceType, diHandler, key);
00068                 }
00069             }
00070
00071             return diHandler;
00072 }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/SimpleDI/DiContainer.cs

6.39 MergeIt.SimpleDI.DiHandler Class Reference

Public Member Functions

- void [AddDependency< TType >\(\)](#)
- [DiHandler AsSingleton \(\)](#)
- [DiHandler AsSingleton< TType >\(TType instance\)](#)

6.39.1 Detailed Description

Definition at line 10 of file [DiHandler.cs](#).

6.39.2 Member Function Documentation

6.39.2.1 AddDependency< TType >()

```
void MergeIt.SimpleDI.DiHandler.AddDependency< TType > ( ) [inline]
```

Definition at line 79 of file [DiHandler.cs](#).

```
00080         {
00081             _type = typeof(TType);
00082         }
```

6.39.2.2 AsSingleton()

```
DiHandler MergeIt.SimpleDI.DiHandler.AsSingleton ( ) [inline]
```

Definition at line 84 of file [DiHandler.cs](#).

```
00085     {
00086         _isSingleton = true;
00087         GetOrCreateInstance ();
00089         return this;
00091     }
```

6.39.2.3 AsSingleton< TType >()

```
DiHandler MergeIt.SimpleDI.DiHandler.AsSingleton< TType > (
    TType instance ) [inline]
```

Type Constraints

TType : class

Definition at line 93 of file [DiHandler.cs](#).

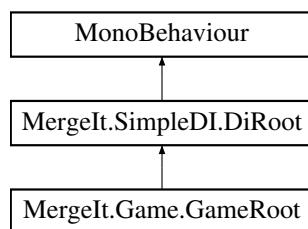
```
00093 : class
00094     {
00095         _isSingleton = true;
00096         _instance = instance;
00097         _needSetup = false;
00098         return this;
00100     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/SimpleDI/DiHandler.cs

6.40 Mergelt.SimpleDI.DiRoot Class Reference

Inheritance diagram for Mergelt.SimpleDI.DiRoot:



Protected Member Functions

- void **OnInstall** ()
- virtual void **Run** ()

6.40.1 Detailed Description

Definition at line 8 of file [DiRoot.cs](#).

6.40.2 Member Function Documentation

6.40.2.1 Run()

```
virtual void MergeIt.SimpleDI.DiRoot.Run ( ) [inline], [protected], [virtual]
```

Definition at line 12 of file [DiRoot.cs](#).

```
00013     {  
00014     }  
00015 }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/SimpleDI/DiRoot.cs

6.41 Mergelt.Core.Schemes.Data.EdgeData Class Reference

Public Attributes

- string [OutputNodeID](#)
- string [InputNodeID](#)
- string [OutputPortName](#)
- string [InputPortName](#)

6.41.1 Detailed Description

Definition at line 8 of file [EdgeData.cs](#).

6.41.2 Member Data Documentation

6.41.2.1 InputNodeID

```
string MergeIt.Core.Schemes.Data.EdgeData.InputNodeID
```

Definition at line 11 of file [EdgeData.cs](#).

6.41.2.2 InputPortName

```
string MergeIt.Core.Schemes.Data.EdgeData.InputPortName
```

Definition at line 13 of file [EdgeData.cs](#).

6.41.2.3 OutputNodeID

`string MergeIt.Core.Schemes.Data.EdgeData.OutputNodeID`

Definition at line 10 of file [EdgeData.cs](#).

6.41.2.4 OutputPortName

`string MergeIt.Core.Schemes.Data.EdgeData.OutputPortName`

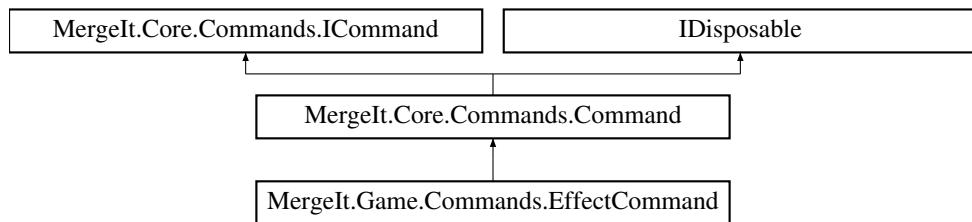
Definition at line 12 of file [EdgeData.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Schemes/Data/EdgeData.cs

6.42 Mergelt.Game.Commands.EffectCommand Class Reference

Inheritance diagram for Mergelt.Game.Commands.EffectCommand:



Public Member Functions

- `EffectCommand (IEffect effect)`
- override void `Execute ()`

Public Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual async UniTask `ExecuteAsync ()`
- void `Dispose ()`

Additional Inherited Members

Protected Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual void `Finish ()`
- virtual void `OnDispose ()`

Events inherited from [Mergelt.Core.Commands.Command](#)

- Action< [ICommand](#) > Finished

Events inherited from [Mergelt.Core.Commands.ICommand](#)

- Action< [ICommand](#) > Finished

6.42.1 Detailed Description

Definition at line 8 of file [EffectCommand.cs](#).

6.42.2 Constructor & Destructor Documentation

6.42.2.1 EffectCommand()

```
MergeIt.Game.Commands.EffectCommand.EffectCommand (
    IEffect effect ) [inline]
```

Definition at line 12 of file [EffectCommand.cs](#).

```
00013     {
00014         _effect = effect;
00015     }
```

6.42.3 Member Function Documentation

6.42.3.1 Execute()

```
override void MergeIt.Game.Commands.EffectCommand.Execute ( ) [inline], [virtual]
```

Reimplemented from [Mergelt.Core.Commands.Command](#).

Definition at line 17 of file [EffectCommand.cs](#).

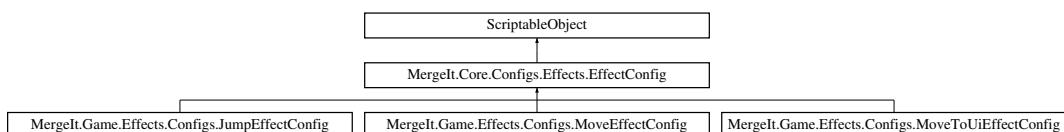
```
00018     {
00019         base.Execute();
00020     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Commands/EffectCommand.cs

6.43 Mergelt.Core.Configs.Effects.EffectConfig Class Reference

Inheritance diagram for Mergelt.Core.Configs.Effects.EffectConfig:



Public Attributes

- string [Name](#)
- float [Duration](#)

6.43.1 Detailed Description

Definition at line [7](#) of file [EffectConfig.cs](#).

6.43.2 Member Data Documentation

6.43.2.1 Duration

```
float MergeIt.Core.Configs.Effects.EffectConfig.Duration
```

Definition at line [10](#) of file [EffectConfig.cs](#).

6.43.2.2 Name

```
string MergeIt.Core.Configs.Effects.EffectConfig.Name
```

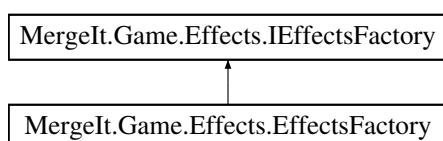
Definition at line [9](#) of file [EffectConfig.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Configs/Effects/EffectConfig.cs

6.44 MergeIt.Game.Effects.EffectsFactory Class Reference

Inheritance diagram for MergeIt.Game.Effects.EffectsFactory:



Public Member Functions

- void [CreateEffect< T >](#) ([IFieldElementPresenter](#) target, [IEffectParameters](#) effectParameters=null, Action finishedCallback=null)
- void [CreateEffect< T >](#) (RectTransform target, [IEffectParameters](#) effectParameters=null, Action finishedCallback=null)

6.44.1 Detailed Description

Definition at line 12 of file [EffectsFactory.cs](#).

6.44.2 Member Function Documentation

6.44.2.1 CreateEffect< T >() [1/2]

```
void MergeIt.Game.Effects.EffectsFactory.CreateEffect< T > (
    IFieldElementPresenter target,
    IEFFECTPARAMETERS effectParameters = null,
    Action finishedCallback = null ) [inline]
```

Implements [Mergelt.Game.Effects.IEffectsFactory](#).

Type Constraints

- T : IEFFECTWITHPRESENTER**
- T : IEFFECT**
- T : new()**

Definition at line 17 of file [EffectsFactory.cs](#).

```
00019      : IEFFECTWITHPRESENTER, IEFFECT, new()
00020      {
00021          var effectController = new T();
00022          effectController.Setup(target, effectParameters, finishedCallback);
00023          _effectsManager.AddEffect(effectController);
00024      }
```

6.44.2.2 CreateEffect< T >() [2/2]

```
void MergeIt.Game.Effects.EffectsFactory.CreateEffect< T > (
    RectTransform target,
    IEFFECTPARAMETERS effectParameters = null,
    Action finishedCallback = null ) [inline]
```

Implements [Mergelt.Game.Effects.IEffectsFactory](#).

Type Constraints

- T : IEFFECT**
- T : new()**

Definition at line 26 of file [EffectsFactory.cs](#).

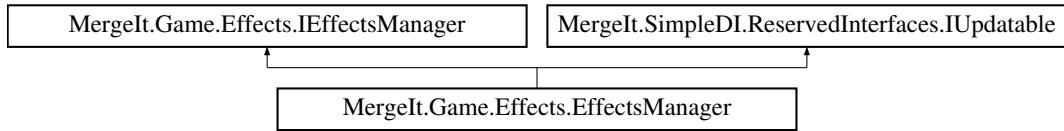
```
00028      : IEFFECT, new()
00029      {
00030          var effectController = new T();
00031          effectController.Setup(target, effectParameters, finishedCallback);
00032          _effectsManager.AddEffect(effectController);
00033      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Effects/EffectsFactory.cs

6.45 Mergelt.Game.Effects.EffectsManager Class Reference

Inheritance diagram for Mergelt.Game.Effects.EffectsManager:



Public Member Functions

- void [AddEffect \(IEffect effectController\)](#)
- void [Update \(\)](#)

6.45.1 Detailed Description

Definition at line [12](#) of file [EffectsManager.cs](#).

6.45.2 Member Function Documentation

6.45.2.1 AddEffect()

```
void MergeIt.Game.Effects.EffectsManager.AddEffect (
    IEffect effectController ) [inline]
```

Implements [Mergelt.Game.Effects.IEffectsManager](#).

Definition at line [22](#) of file [EffectsManager.cs](#).

```
00023     {
00024         _effectControllers.Add(effectController);
00025         effectController.Start();
00026     }
```

6.45.2.2 Update()

```
void MergeIt.Game.Effects.EffectsManager.Update ( ) [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IUpdatable](#).

Definition at line [28](#) of file [EffectsManager.cs](#).

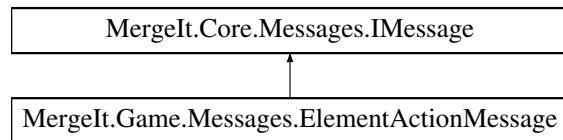
```
00029     {
00030         for (int i = 0; i < _effectControllers.Count; i++)
00031         {
00032             IEffect effect = _effectControllers[i];
00033             effect.Update();
00034         }
00035     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Effects/EffectsManager.cs

6.46 Mergelt.Game.Messages.ElementActionMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.ElementActionMessage:



Properties

- `IFieldElement Element [get, set]`
- `ElementActionType ActionType [get, set]`

6.46.1 Detailed Description

Definition at line 9 of file [ElementActionMessage.cs](#).

6.46.2 Property Documentation

6.46.2.1 ActionType

`ElementActionType MergeIt.Game.Messages.ElementActionMessage.ActionType [get], [set]`

Definition at line 12 of file [ElementActionMessage.cs](#).

```
00012 { get; set; }
```

6.46.2.2 Element

`IFieldElement MergeIt.Game.Messages.ElementActionMessage.Element [get], [set]`

Definition at line 11 of file [ElementActionMessage.cs](#).

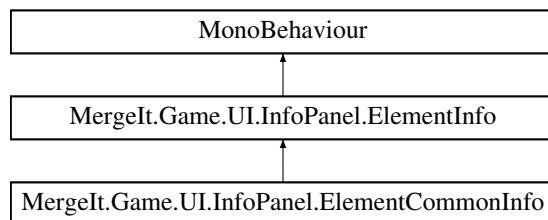
```
00011 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/ElementActionMessage.cs

6.47 Mergelt.Game.UI.InfoPanel.ElementCommonInfo Class Reference

Inheritance diagram for Mergelt.Game.UI.InfoPanel.ElementCommonInfo:



Protected Member Functions

- override bool [OnTrySetup \(\)](#)

Protected Member Functions inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- bool [OnTrySetup \(\)](#)
- void [Clear \(\)](#)
- virtual void [OnStart \(\)](#)
- virtual void [OnClear \(\)](#)
- virtual void [Destroy \(\)](#)
- void [ActionButtonClick \(\)](#)
- virtual void [OnActionButtonClick \(\)](#)
- void [SendActionMessage \(\)](#)

Properties

- override Element ActionType [ActionType](#) [get]

Properties inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- Element ActionType [ActionType](#) [get]

Additional Inherited Members

Public Member Functions inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- void [TrySetup \(IFieldElement fieldElement\)](#)

Protected Attributes inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- IConfigsService [ConfigsService](#)
- IMessageBus [MessageBus](#)
- IFieldElement [SelectedElement](#)

6.47.1 Detailed Description

Definition at line 9 of file [ElementCommonInfo.cs](#).

6.47.2 Member Function Documentation

6.47.2.1 OnTrySetup()

```
override bool MergeIt.Game.UI.InfoPanel.ElementCommonInfo.OnTrySetup () [inline], [protected]
```

Definition at line 22 of file [ElementCommonInfo.cs](#).

```
00023     {
00024         gameObject.SetActive(true);
00025
00026         IInfoParameters infoParameters = SelectedElement.InfoParameters;
00027
00028         _nameLabel.text = infoParameters.Name;
00029         _descriptionLabel.text = infoParameters.Description;
00030
00031         return true;
00032     }
```

6.47.3 Property Documentation

6.47.3.1 ActionType

```
override ElementActionType MergeIt.Game.UI.InfoPanel.ElementCommonInfo.ActionType [get]
```

Definition at line 17 of file [ElementCommonInfo.cs](#).

```
00018     {
00019         get => ElementActionType.Common;
00020     }
```

The documentation for this class was generated from the following file:

- [Assets/MergeIt/Source/Game/UI/InfoPanel/ElementCommonInfo.cs](#)

6.48 MergeIt.Core.Configs.Elements.ElementCommonSettings Class Reference

Properties

- bool [Expanded](#) [get, set]
- string [Description](#) [get]
- [CurrencySettings SplitCostSettings](#) [get]
- [CurrencySettings SellCostSettings](#) [get]
- [CurrencySettings UnlockCostSettings](#) [get]
- [ConsumableSettings ConsumableSettings](#) [get]
- bool [IsConsumable](#) [get]
- string [Name](#) [get]
- [FieldElementIconComponent Icon](#) [get]

6.48.1 Detailed Description

Definition at line 9 of file [ElementCommonSettings.cs](#).

6.48.2 Property Documentation

6.48.2.1 ConsumableSettings

```
ConsumableSettings MergeIt.Core.Configs.Elements.ElementCommonSettings.ConsumableSettings
[get]
```

Definition at line 49 of file [ElementCommonSettings.cs](#).

```
00050     {
00051         get => _consumableSettings;
00052     }
```

6.48.2.2 Description

```
string MergeIt.Core.Configs.Elements.ElementCommonSettings.Description [get]
```

Definition at line 29 of file [ElementCommonSettings.cs](#).

```
00030     {  
00031         get => _description;  
00032     }
```

6.48.2.3 Expanded

```
bool MergeIt.Core.Configs.Elements.ElementCommonSettings.Expanded [get], [set]
```

Definition at line 23 of file [ElementCommonSettings.cs](#).

```
00024     {  
00025         get => _expanded;  
00026         set => _expanded = value;  
00027     }
```

6.48.2.4 Icon

```
FieldElementIconComponent MergeIt.Core.Configs.Elements.ElementCommonSettings.Icon [get]
```

Definition at line 64 of file [ElementCommonSettings.cs](#).

```
00065     {  
00066         get => _icon;  
00067     }
```

6.48.2.5 IsConsumable

```
bool MergeIt.Core.Configs.Elements.ElementCommonSettings.IsConsumable [get]
```

Definition at line 54 of file [ElementCommonSettings.cs](#).

```
00055     {  
00056         get => _isConsumable;  
00057     }
```

6.48.2.6 Name

```
string MergeIt.Core.Configs.Elements.ElementCommonSettings.Name [get]
```

Definition at line 59 of file [ElementCommonSettings.cs](#).

```
00060     {  
00061         get => _name;  
00062     }
```

6.48.2.7 SellCostSettings

```
CurrencySettings MergeIt.Core.Configs.Elements.ElementCommonSettings.SellCostSettings [get]
```

Definition at line 39 of file [ElementCommonSettings.cs](#).

```
00040     {  
00041         get => _sellCostSettings;  
00042     }
```

6.48.2.8 SplitCostSettings

```
CurrencySettings MergeIt.Core.Configs.Elements.ElementCommonSettings.SplitCostSettings [get]
```

Definition at line 34 of file [ElementCommonSettings.cs](#).

```
00035      {  
00036          get => _splitCostSettings;  
00037      }
```

6.48.2.9 UnlockCostSettings

```
CurrencySettings MergeIt.Core.Configs.Elements.ElementCommonSettings.UnlockCostSettings [get]
```

Definition at line 44 of file [ElementCommonSettings.cs](#).

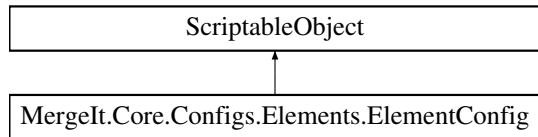
```
00045      {  
00046          get => _unlockCostSettings;  
00047      }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Configs/Elements/Settings/ElementCommonSettings.cs

6.49 MergeIt.Core.Configs.Elements.ElementConfig Class Reference

Inheritance diagram for MergeIt.Core.Configs.Elements.ElementConfig:



Public Member Functions

- [FieldElementIconComponent GetIconComponent \(\)](#)

Protected Member Functions

- void [GenerateGuid \(\)](#)

Properties

- string [Id \[get\]](#)
- ElementType [Type \[get\]](#)
- [ElementCommonSettings CommonSettings \[get, set\]](#)
- [ElementGeneratorSettings GeneratorSettings \[get, set\]](#)

6.49.1 Detailed Description

Definition at line 10 of file [ElementConfig.cs](#).

6.49.2 Member Function Documentation

6.49.2.1 GenerateGuid()

```
void MergeIt.Core.Configs.Elements.ElementConfig.GenerateGuid ( ) [inline], [protected]
```

Definition at line 59 of file [ElementConfig.cs](#).

```
00060     {
00061         _id = Guid.NewGuid().ToString();
00062     }
```

6.49.2.2 GetIconComponent()

```
FieldElementIconComponent MergeIt.Core.Configs.Elements.ElementConfig.GetIconComponent ( )
[inline]
```

Definition at line 54 of file [ElementConfig.cs](#).

```
00055     {
00056         return _commonSettings?.Icon;
00057     }
```

6.49.3 Property Documentation

6.49.3.1 CommonSettings

```
ElementCommonSettings MergeIt.Core.Configs.Elements.ElementConfig.CommonSettings [get], [set]
```

Definition at line 42 of file [ElementConfig.cs](#).

```
00043     {
00044         get => _commonSettings;
00045         set => _commonSettings = value;
00046     }
```

6.49.3.2 GeneratorSettings

```
ElementGeneratorSettings MergeIt.Core.Configs.Elements.ElementConfig.GeneratorSettings [get],
[set]
```

Definition at line 48 of file [ElementConfig.cs](#).

```
00049     {
00050         get => _generatorSettings;
00051         set => _generatorSettings = value;
00052     }
```

6.49.3.3 Id

```
string MergeIt.Core.Configs.Elements.ElementConfig.Id [get]
```

Definition at line 24 of file [ElementConfig.cs](#).

```
00025     {
00026         get
00027     {
00028         if (string.IsNullOrEmpty(_id))
00029         {
00030             _id = Guid.NewGuid().ToString();
00031         }
00032
00033         return _id;
00034     }
00035 }
```

6.49.3.4 Type

```
ElementType MergeIt.Core.Configs.Elements.ElementConfig.Type [get]
```

Definition at line 37 of file [ElementConfig.cs](#).

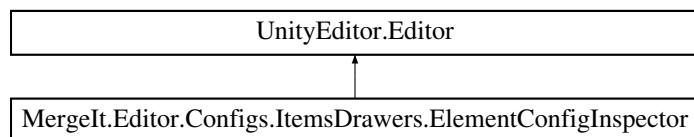
```
00038     {
00039         get => _type;
00040     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Configs/Elements/ElementConfig.cs

6.50 MergeIt.Editor.Configs.ItemsDrawers.ElementConfigInspector Class Reference

Inheritance diagram for MergeIt.Editor.Configs.ItemsDrawers.ElementConfigInspector:



Public Member Functions

- override VisualElement [CreateInspectorGUI \(\)](#)

Properties

- [ElementConfig Target \[get\]](#)
- [VisualElement Root \[get\]](#)

6.50.1 Detailed Description

Definition at line 15 of file [ElementConfigInspector.cs](#).

6.50.2 Member Function Documentation

6.50.2.1 CreateInspectorGUI()

```
override VisualElement MergeIt.Editor.Configs.ItemsDrawers.ElementConfigInspector.Create←
InspectorGUI ( ) [inline]
```

Definition at line 40 of file [ElementConfigInspector.cs](#).

```
00041     {
00042         _root = new VisualElement();
00043
00044         GetUxml().CloneTree(Root);
00045
00046         var typeField = Root.Q<EnumField>("ElementType");
00047         typeField.RegisterValueChangedCallback(OnElementTypeChanged);
00048
00049         var stylesheet = GetStylesheet();
00050         if (!Root.styleSheets.Contains(stylesheet))
00051         {
00052             Root.styleSheets.Add(stylesheet);
00053         }
00054
00055         _commonSettingsDrawer?.Dispose();
00056         _commonSettingsDrawer = null;
00057
00058         _generatorSettingsDrawer?.Clear();
00059         _generatorSettingsDrawer = null;
00060
00061         Refresh(Target.Type);
00062
00063         serializedObject.ApplyModifiedProperties();
00064
00065         return Root;
00066     }
```

6.50.3 Property Documentation

6.50.3.1 Root

```
VisualElement MergeIt.Editor.Configs.ItemsDrawers.ElementConfigInspector.Root [get]
```

Definition at line 35 of file [ElementConfigInspector.cs](#).

```
00036     {
00037         get => _root;
00038     }
```

6.50.3.2 Target

```
ElementConfig MergeIt.Editor.Configs.ItemsDrawers.ElementConfigInspector.Target [get]
```

Definition at line 22 of file [ElementConfigInspector.cs](#).

```
00023     {
00024         get
00025         {
00026             if (_target == null)
00027             {
00028                 _target = target as ElementConfig;
00029             }
00030
00031             return _target;
00032         }
00033     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/ElementConfigInspector.cs

6.51 MergeIt.Core.Configs.Elements.ElementGeneratorSettings Class Reference

Properties

- bool `Expanded` [get, set]
- int `Stack` [get]
- int `MinDrop` [get]
- int `MaxDrop` [get]
- int `ChargeTime` [get]
- int `DropCount` [get]
- `CurrencySettings SkipChargeCostSettings` [get]
- bool `AdSkip` [get]
- bool `Charged` [get]
- int `EnergyCost` [get]
- bool `IsUnlimited` [get]
- bool `NeedOpen` [get]
- int `OpenTime` [get]
- `CurrencySettings SkipOpenCostSettings` [get]
- bool `GenerateBlocked` [get]
- List<`GeneratableElement`> `GenerateItems` [get]

6.51.1 Detailed Description

Definition at line 10 of file `ElementGeneratorSettings.cs`.

6.51.2 Property Documentation

6.51.2.1 AdSkip

```
bool MergeIt.Core.Configs.Elements.ElementGeneratorSettings.AdSkip [get]
```

Definition at line 67 of file `ElementGeneratorSettings.cs`.

```
00068      {
00069          get => _adSkip;
00070      }
```

6.51.2.2 Charged

```
bool MergeIt.Core.Configs.Elements.ElementGeneratorSettings.Charged [get]
```

Definition at line 72 of file `ElementGeneratorSettings.cs`.

```
00073      {
00074          get => _charged;
00075      }
```

6.51.2.3 ChargeTime

```
int MergeIt.Core.Configs.Elements.ElementGeneratorSettings.ChargeTime [get]
```

Definition at line 52 of file `ElementGeneratorSettings.cs`.

```
00053      {
00054          get => _chargeTime;
00055      }
```

6.51.2.4 DropCount

```
int MergeIt.Core.Configs.Elements.ElementGeneratorSettings.DropCount [get]
```

Definition at line 57 of file [ElementGeneratorSettings.cs](#).

```
00058     {  
00059         get => _dropCount;  
00060     }
```

6.51.2.5 EnergyCost

```
int MergeIt.Core.Configs.Elements.ElementGeneratorSettings.EnergyCost [get]
```

Definition at line 77 of file [ElementGeneratorSettings.cs](#).

```
00078     {  
00079         get => _energyCost;  
00080     }
```

6.51.2.6 Expanded

```
bool MergeIt.Core.Configs.Elements.ElementGeneratorSettings.Expanded [get], [set]
```

Definition at line 31 of file [ElementGeneratorSettings.cs](#).

```
00032     {  
00033         get => _expanded;  
00034         set => _expanded = value;  
00035     }
```

6.51.2.7 GenerateBlocked

```
bool MergeIt.Core.Configs.Elements.ElementGeneratorSettings.GenerateBlocked [get]
```

Definition at line 102 of file [ElementGeneratorSettings.cs](#).

```
00103     {  
00104         get => _generateBlocked;  
00105     }
```

6.51.2.8 GenerateItems

```
List<GeneratableElement> MergeIt.Core.Configs.Elements.ElementGeneratorSettings.GenerateItems  
[get]
```

Definition at line 107 of file [ElementGeneratorSettings.cs](#).

```
00108     {  
00109         get => _generateItems;  
00110     }
```

6.51.2.9 IsUnlimited

```
bool MergeIt.Core.Configs.Elements.ElementGeneratorSettings.IsUnlimited [get]
```

Definition at line 82 of file [ElementGeneratorSettings.cs](#).

```
00083     {  
00084         get => _isUnlimited;  
00085     }
```

6.51.2.10 MaxDrop

```
int MergeIt.Core.Configs.Elements.ElementGeneratorSettings.MaxDrop [get]
```

Definition at line 47 of file [ElementGeneratorSettings.cs](#).

```
00048     {
00049         get => _maxDrop;
00050     }
```

6.51.2.11 MinDrop

```
int MergeIt.Core.Configs.Elements.ElementGeneratorSettings.MinDrop [get]
```

Definition at line 42 of file [ElementGeneratorSettings.cs](#).

```
00043     {
00044         get => _minDrop;
00045     }
```

6.51.2.12 NeedOpen

```
bool MergeIt.Core.Configs.Elements.ElementGeneratorSettings.NeedOpen [get]
```

Definition at line 87 of file [ElementGeneratorSettings.cs](#).

```
00088     {
00089         get => _needOpen;
00090     }
```

6.51.2.13 OpenTime

```
int MergeIt.Core.Configs.Elements.ElementGeneratorSettings.OpenTime [get]
```

Definition at line 92 of file [ElementGeneratorSettings.cs](#).

```
00093     {
00094         get => _openTime;
00095     }
```

6.51.2.14 SkipChargeCostSettings

```
CurrencySettings MergeIt.Core.Configs.Elements.ElementGeneratorSettings.SkipChargeCostSettings [get]
```

Definition at line 62 of file [ElementGeneratorSettings.cs](#).

```
00063     {
00064         get => _skipChargeCostSettings;
00065     }
```

6.51.2.15 SkipOpenCostSettings

```
CurrencySettings MergeIt.Core.Configs.Elements.ElementGeneratorSettings.SkipOpenCostSettings [get]
```

Definition at line 97 of file [ElementGeneratorSettings.cs](#).

```
00098     {
00099         get => _skipOpenCostSettings;
00100     }
```

6.51.2.16 Stack

```
int MergeIt.Core.Configs.Elements.ElementGeneratorSettings.Stack [get]
```

Definition at line 37 of file [ElementGeneratorSettings.cs](#).

```
00038     {
00039         get => _stack;
00040     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Configs/Elements/Settings/ElementGeneratorSettings.cs

6.52 Mergelt.Game.UI.InfoPanel.ElementInfo Class Reference

Inheritance diagram for Mergelt.Game.UI.InfoPanel.ElementInfo:



Public Member Functions

- void [TrySetup \(IFieldElement fieldElement\)](#)

Protected Member Functions

- bool [OnTrySetup \(\)](#)
- void [Clear \(\)](#)
- virtual void [OnStart \(\)](#)
- virtual void [OnClear \(\)](#)
- virtual void [Destroy \(\)](#)
- void [ActionButtonClick \(\)](#)
- virtual void [OnActionButtonClick \(\)](#)
- void [SendActionMessage \(\)](#)

Protected Attributes

- [IConfigsService ConfigsService](#)
- [IMessageBus MessageBus](#)
- [IFieldElement SelectedElement](#)

Properties

- Element ActionType [ActionType \[get\]](#)

6.52.1 Detailed Description

Definition at line 12 of file [ElementInfo.cs](#).

6.52.2 Member Function Documentation

6.52.2.1 ActionButtonClick()

```
void MergeIt.Game.UI.InfoPanel.ElementInfo.ActionButtonClick () [inline], [protected]
```

Definition at line 72 of file [ElementInfo.cs](#).

```
00073     {
00074         OnActionButtonClick ();
00075     }
```

6.52.2.2 Clear()

```
void MergeIt.Game.UI.InfoPanel.ElementInfo.Clear () [inline], [protected]
```

Definition at line 51 of file [ElementInfo.cs](#).

```
00052     {
00053         SelectedElement = null;
00054
00055         OnClear ();
00056     }
```

6.52.2.3 Destroy()

```
virtual void MergeIt.Game.UI.InfoPanel.ElementInfo.Destroy () [inline], [protected], [virtual]
```

Definition at line 67 of file [ElementInfo.cs](#).

```
00068     {
00069
00070     }
```

6.52.2.4 OnActionButtonClick()

```
virtual void MergeIt.Game.UI.InfoPanel.ElementInfo.OnActionButtonClick () [inline], [protected], [virtual]
```

Definition at line 77 of file [ElementInfo.cs](#).

```
00078     {
00079         SendActionMessage ();
00080     }
```

6.52.2.5 OnClear()

```
virtual void MergeIt.Game.UI.InfoPanel.ElementInfo.OnClear () [inline], [protected], [virtual]
```

Definition at line 63 of file [ElementInfo.cs](#).

```
00064     {
00065     }
```

6.52.2.6 OnStart()

```
virtual void MergeIt.Game.UI.InfoPanel.ElementInfo.OnStart () [inline], [protected], [virtual]
```

Definition at line 58 of file [ElementInfo.cs](#).

```
00059     {
00060
00061     }
```

6.52.2.7 SendActionMessage()

```
void MergeIt.Game.UI.InfoPanel.ElementInfo.SendActionMessage ( ) [inline], [protected]
```

Definition at line 82 of file [ElementInfo.cs](#).

```
00083     {
00084         var message = new ElementActionMessage
00085         {
00086             Element = SelectedElement,
00087             ActionType = ActionType
00088         };
00089         MessageBus.Fire(message);
00090     }
```

6.52.2.8 TrySetup()

```
void MergeIt.Game.UI.InfoPanel.ElementInfo.TrySetup (
    IFieldElement fieldElement ) [inline]
```

Definition at line 37 of file [ElementInfo.cs](#).

```
00038     {
00039         ConfigsService = DiContainer.Get<IConfigsService>();
00040         SelectedElement = fieldElement;
00041
00042         if (!OnTrySetup())
00043         {
00044             SelectedElement = null;
00045             gameObject.SetActive(false);
00046         }
00047     }
```

6.52.3 Member Data Documentation

6.52.3.1 ConfigsService

```
IConfigsService MergeIt.Game.UI.InfoPanel.ElementInfo.ConfigsService [protected]
```

Definition at line 14 of file [ElementInfo.cs](#).

6.52.3.2 MessageBus

```
IMessageBus MergeIt.Game.UI.InfoPanel.ElementInfo.MessageBus [protected]
```

Definition at line 15 of file [ElementInfo.cs](#).

6.52.3.3 SelectedElement

```
IFieldElement MergeIt.Game.UI.InfoPanel.ElementInfo.SelectedElement [protected]
```

Definition at line 17 of file [ElementInfo.cs](#).

6.52.4 Property Documentation

6.52.4.1 ActionType

```
ElementActionType MergeIt.Game.UI.InfoPanel.ElementInfo.ActionType [get], [abstract]
```

Definition at line 18 of file [ElementInfo.cs](#).

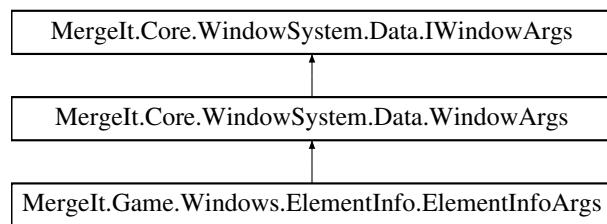
```
00018 { get; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/UI/InfoPanel/ElementInfo.cs

6.53 Mergelt.Game.Windows.ElementInfo.ElementInfoArgs Class Reference

Inheritance diagram for Mergelt.Game.Windows.ElementInfo.ElementInfoArgs:



Properties

- [ElementConfig ElementConfig](#) [get, set]

6.53.1 Detailed Description

Definition at line 8 of file [ElementInfoArgs.cs](#).

6.53.2 Property Documentation

6.53.2.1 ElementConfig

```
ElementConfig MergeIt.Game.Windows.ElementInfo.ElementInfoArgs.ElementConfig [get], [set]
```

Definition at line 10 of file [ElementInfoArgs.cs](#).

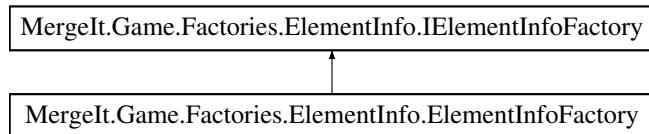
```
00010 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Windows/ElementInfo/ElementInfoArgs.cs

6.54 MergeIt.Game.Factories.ElementInfo.ElementInfoFactory Class Reference

Inheritance diagram for MergeIt.Game.Factories.ElementInfo.ElementInfoFactory:



Public Member Functions

- `ElementInfoItemComponent CreateElementWindowItem (ElementConfig elementConfig, ElementInfoType infoType=ElementInfoType.InfoWindow, bool isLocked=false)`
- `ElementInfoItemComponent CreateUnknownElementWindowItem (ElementInfoType infoType=ElementInfoType.InfoWindow)`

6.54.1 Detailed Description

Definition at line 10 of file [ElementInfoFactory.cs](#).

6.54.2 Member Function Documentation

6.54.2.1 CreateElementWindowItem()

```

ElementInfoItemComponent MergeIt.Game.Factories.ElementInfo.ElementInfoFactory.CreateElement←
WindowItem (
    ElementConfig elementConfig,
    ElementInfoType infoType = ElementInfoType::InfoWindow,
    bool isLocked = false ) [inline]
  
```

Implements [MergeIt.Game.Factories.ElementInfo.IElementInfoFactory](#).

Definition at line 18 of file [ElementInfoFactory.cs](#).

```

00020     {
00021         if (PrefabsLocations.TryGetValue(infoType, out string itemInfoPath))
00022     {
00023         GameObject itemPanelObject = Resources.Load<GameObject>(itemInfoPath);
00024
00025         if (itemPanelObject)
00026         {
00027             var panelItemObject = Object.Instantiate(itemPanelObject);
00028             if (panelItemObject.TryGetComponent(out ElementInfoItemComponent resultComponent))
00029             {
00030                 FieldElementIconComponent iconPrototype = elementConfig.GetIconComponent();
00031
00032                 var icon = Object.Instantiate(iconPrototype, panelItemObject.transform);
00033                 var iconRectTransform = icon.GetComponent<RectTransform>();
00034
00035                 iconPrototype.SetBlocked(isLocked);
00036
00037                 resultComponent.SetIcon(iconRectTransform, isLocked);
00038             }
00039
00040             return resultComponent;
00041         }
00042     }
00043
00044     return null;
00045 }
  
```

6.54.2.2 CreateUnknownElementWindowItem()

```
ElementInfoItemComponent MergeIt.Game.Factories.ElementInfo.ElementInfoFactory.CreateUnknownElementWindowItem (
    ElementInfoType infoType = ElementInfoType::InfoWindow ) [inline]
```

Implements [MergeIt.Game.Factories.ElementInfo.IElementInfoFactory](#).

Definition at line 47 of file [ElementInfoFactory.cs](#).

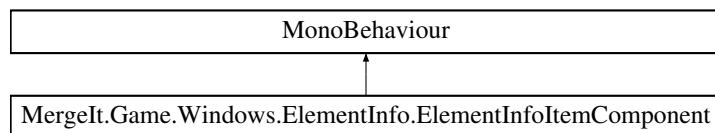
```
00048     {
00049         if (PrefabsLocations.TryGetValue(infoType, out string itemInfoPath))
00050     {
00051         GameObject itemPanelObject = Resources.Load<GameObject>(itemInfoPath);
00052
00053         if (itemPanelObject)
00054         {
00055             var panelItemObject = Object.Instantiate(itemPanelObject);
00056             if (panelItemObject.TryGetComponent(out ElementInfoItemComponent resultComponent))
00057             {
00058                 resultComponent.SetUnknown();
00059             }
00060
00061             return resultComponent;
00062         }
00063     }
00064
00065     return null;
00066 }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Factories/ElementInfo/ElementInfoFactory.cs

6.55 MergeIt.Game.Windows.ElementInfo.ElementInfoItemComponent Class Reference

Inheritance diagram for MergeIt.Game.Windows.ElementInfo.ElementInfoItemComponent:



Public Member Functions

- void [Awake \(\)](#)
- void [OnDestroy \(\)](#)
- void [Setup \(bool isGenerator, bool isSelected, int number=-1\)](#)
- void [SetIcon \(Transform iconTransform, bool isLocked=false\)](#)
- void [SetUnknown \(\)](#)

Public Attributes

- UnityAction< [ElementInfoItemComponent](#) > [ClickEvent](#) = delegate { }

Properties

- RectTransform [RectTransform](#) [get]

6.55.1 Detailed Description

Definition at line 10 of file [ElementInfoItemComponent.cs](#).

6.55.2 Member Function Documentation

6.55.2.1 Awake()

```
void MergeIt.Game.Windows.ElementInfo.ElementInfoItemComponent.Awake () [inline]
```

Definition at line 47 of file [ElementInfoItemComponent.cs](#).

```
00048     {  
00049         RectTransform = GetComponent<RectTransform>();  
00050         _button.onClick.AddListener(OnClick);  
00051     }
```

6.55.2.2 OnDestroy()

```
void MergeIt.Game.Windows.ElementInfo.ElementInfoItemComponent.OnDestroy () [inline]
```

Definition at line 53 of file [ElementInfoItemComponent.cs](#).

```
00054     {  
00055         ClickEvent = null;  
00056         _button.onClick.RemoveListener(OnClick);  
00057     }
```

6.55.2.3 SetIcon()

```
void MergeIt.Game.Windows.ElementInfo.ElementInfoItemComponent.SetIcon (  
    Transform iconTransform,  
    bool isLocked = false ) [inline]
```

Definition at line 75 of file [ElementInfoItemComponent.cs](#).

```
00076     {  
00077         iconTransform.SetParent(_iconContainer);  
00078         iconTransform.localScale = Vector3.one;  
00079  
00080         if (_lockerObject)  
00081         {  
00082             _lockerObject.SetActive(isLocked);  
00083         }  
00084     }
```

6.55.2.4 SetUnknown()

```
void MergeIt.Game.Windows.ElementInfo.ElementInfoItemComponent.SetUnknown () [inline]
```

Definition at line 91 of file [ElementInfoItemComponent.cs](#).

```
00092     {  
00093         _unknownContainer.SetActive(true);  
00094     }
```

6.55.2.5 Setup()

```
void MergeIt.Game.Windows.ElementInfo.ElementInfoItemComponent.Setup (
    bool isGenerator,
    bool isSelected,
    int number = -1 ) [inline]
```

Definition at line 59 of file [ElementInfoItemComponent.cs](#).

```
00060     {
00061         _backgroundImage.color = isSelected ? _selectedColor : Color.white;
00062         _generatorIcon.SetActive(isGenerator);
00063
00064         if (number > -1 && _numberText)
00065         {
00066             _numberContainer.SetActive(true);
00067             _numberText.text = number.ToString();
00068         }
00069         else
00070         {
00071             _numberContainer.SetActive(false);
00072         }
00073     }
```

6.55.3 Member Data Documentation

6.55.3.1 ClickEvent

```
UnityAction<ElementInfoItemComponent> MergeIt.Game.Windows.ElementInfo.ElementInfoItemComponent.ClickEvent = delegate { }
```

Definition at line 43 of file [ElementInfoItemComponent.cs](#).

```
00043 { };
```

6.55.4 Property Documentation

6.55.4.1 RectTransform

```
RectTransform MergeIt.Game.Windows.ElementInfo.ElementInfoItemComponent.RectTransform [get]
```

Definition at line 45 of file [ElementInfoItemComponent.cs](#).

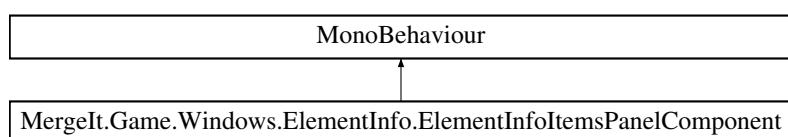
```
00045 { get; private set; }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Windows/ElementInfo/ElementInfoItemComponent.cs

6.56 MergeIt.Game.Windows.ElementInfo.ElementInfoItemsPanel Component Class Reference

Inheritance diagram for MergeIt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent:



Public Member Functions

- void [SetSpacing](#) (float spacing)
- void [SetTitle](#) (string titleText)
- void [SetItem](#) (Transform item)

Properties

- GridLayoutGroup [ItemsGrid](#) [get]
- RectTransform [RectTransform](#) [get]
- TMP_Text [PanelTitle](#) [get]

6.56.1 Detailed Description

Definition at line 9 of file [ElementInfoItemsPanelComponent.cs](#).

6.56.2 Member Function Documentation

6.56.2.1 SetItem()

```
void MergeIt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent.SetItem (
    Transform item) [inline]
```

Definition at line 58 of file [ElementInfoItemsPanelComponent.cs](#).

```
00059     {
00060         item.SetParent(_itemsGrid.transform);
00061         item.localScale = Vector3.one;
00062     }
```

6.56.2.2 SetSpacing()

```
void MergeIt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent.SetSpacing (
    float spacing) [inline]
```

Definition at line 48 of file [ElementInfoItemsPanelComponent.cs](#).

```
00049     {
00050         _layoutGroup.spacing = spacing;
00051     }
```

6.56.2.3 SetTitle()

```
void MergeIt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponentSetTitle (
    string titleText) [inline]
```

Definition at line 53 of file [ElementInfoItemsPanelComponent.cs](#).

```
00054     {
00055         _panelTitle.text = titleText;
00056     }
```

6.56.3 Property Documentation

6.56.3.1 ItemsGrid

```
GridLayoutGroup MergeIt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent.ItemsGrid
[get]
```

Definition at line 20 of file [ElementInfoItemsPanelComponent.cs](#).

```
00021      {
00022          get => _itemsGrid;
00023      }
```

6.56.3.2 PanelTitle

```
TMP_Text MergeIt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent.PanelTitle [get]
```

Definition at line 38 of file [ElementInfoItemsPanelComponent.cs](#).

```
00039      {
00040          get => _panelTitle;
00041      }
```

6.56.3.3 RectTransform

```
RectTransform MergeIt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent.RectTransform
[get]
```

Definition at line 25 of file [ElementInfoItemsPanelComponent.cs](#).

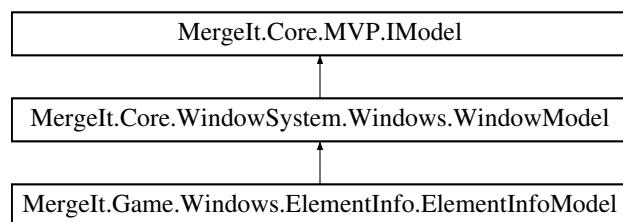
```
00026      {
00027          get
00028          {
00029              if (!_rectTransform)
00030              {
00031                  _rectTransform = GetComponent<RectTransform>();
00032              }
00033
00034              return _rectTransform;
00035          }
00036      }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Windows/ElementInfo/ElementInfoItemsPanelComponent.cs

6.57 MergeIt.Game.Windows.ElementInfo.ElementInfoModel Class Reference

Inheritance diagram for MergeIt.Game.Windows.ElementInfo.ElementInfoModel:



6.57.1 Detailed Description

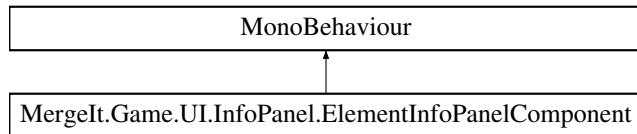
Definition at line 7 of file [ElementInfoModel.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Windows/ElementInfo/ElementInfoModel.cs

6.58 Mergelt.Game.UI.InfoPanel.ElementInfoPanelComponent Class Reference

Inheritance diagram for Mergelt.Game.UI.InfoPanel.ElementInfoPanelComponent:



6.58.1 Detailed Description

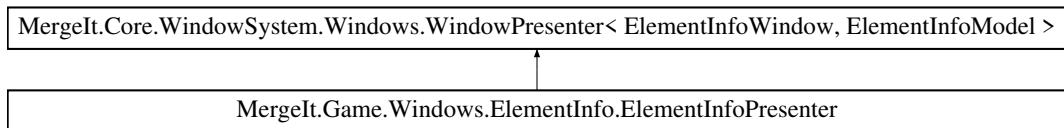
Definition at line 18 of file [ElementInfoPanelComponent.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/UI/InfoPanel/ElementInfoPanelComponent.cs

6.59 Mergelt.Game.Windows.ElementInfo.ElementInfoPresenter Class Reference

Inheritance diagram for Mergelt.Game.Windows.ElementInfo.ElementInfoPresenter:



Protected Member Functions

- override void [OnInitialize \(IWindowArgs args=null\)](#)
- override void [OnDestroyWindow \(\)](#)

Protected Member Functions inherited from

[Mergelt.Core.WindowSystem.Windows.WindowPresenter< ElementInfoWindow, ElementInfoModel >](#)

- virtual void [OnInitialize \(IWindowArgs args=null\)](#)
- override void [OnDispose \(\)](#)
- virtual void [OnWindowStartShowing \(\)](#)
- virtual void [OnWindowShown \(\)](#)
- virtual void [OnWindowStartClosing \(\)](#)
- virtual void [OnWindowClosed \(\)](#)

Additional Inherited Members

Public Member Functions inherited from

[Mergelt.Core.WindowSystem.Windows.WindowPresenter< ElementInfoWindow, ElementInfoModel >](#)

- void [Initialize \(IView view, string layer, IWindowArgs windowArgs=null\)](#)
- void [Show \(\)](#)
- void [Hide \(\)](#)
- void [Close \(\)](#)
- void [SetWindowActive \(bool active\)](#)
- void [SetWindowLayer \(\)](#)
- void [DestroyWindow \(\)](#)

Protected Attributes inherited from

[Mergelt.Core.WindowSystem.Windows.WindowPresenter< ElementInfoWindow, ElementInfoModel >](#)

- [IMessageBus MessageBus](#)
- [IWindowSystem WindowSystem](#)

Properties inherited from

[Mergelt.Core.WindowSystem.Windows.WindowPresenter< ElementInfoWindow, ElementInfoModel >](#)

- string [Layer \[get\]](#)
- [WindowState State \[get\]](#)

6.59.1 Detailed Description

Definition at line 19 of file [ElementInfoPresenter.cs](#).

6.59.2 Member Function Documentation

6.59.2.1 OnDestroyWindow()

```
override void MergeIt.Game.Windows.ElementInfo.ElementInfoPresenter.OnDestroyWindow ( ) [inline],  
[protected], [virtual]
```

Reimplemented from [Mergelt.Core.WindowSystem.Windows.WindowPresenter< ElementInfoWindow, ElementInfoModel >](#).

Definition at line 63 of file [ElementInfoPresenter.cs](#).

```
00064      {  
00065          base.OnDestroyWindow();  
00066  
00067          foreach (var elementConfig in _elementConfigs)  
00068          {  
00069              if (elementConfig.Key)  
00070              {  
00071                  elementConfig.Key.ClickEvent -= OnClickElement;  
00072              }  
00073          }  
00074          _elementConfigs.Clear();  
00075      }
```

6.59.2.2 OnInitialize()

```
override void MergeIt.Game.Windows.ElementInfo.ElementInfoPresenter.OnInitialize (
    IWindowArgs args = null ) [inline], [protected]
```

Definition at line 31 of file [ElementInfoPresenter.cs](#).

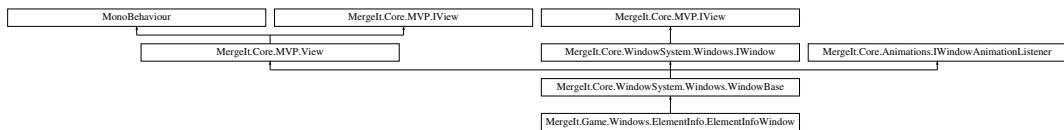
```
00032         {
00033             base.OnInitialize(args);
00034
00035             _elementInfoFactory = DiContainer.Get<IElementInfoFactory>();
00036             _configsService = DiContainer.Get<IConfigsService>();
00037             _evolutionsService = DiContainer.Get<IEvolutionsService>();
00038
00039             _elementInfoArgs = args as ElementInfoArgs;
00040
00041             if (_elementInfoArgs != null)
00042             {
00043                 _selectedConfig = _elementInfoArgs.ElementConfig;
00044                 SchemeObject evolutionConfig = _configsService.LevelConfig.EvolutionsScheme;
00045                 EvolutionData evolutionData = evolutionConfig.Evolution.FirstOrDefault(x =>
00046                     x.Chain.Contains(_selectedConfig));
00047
00048                 View.WindowTitleText.text = evolutionData?.Name;
00049
00050                 var elementChain = _evolutionsService.GetEvolutionChain(evolutionData);
00051                 var generatorsChain = _evolutionsService.GetGeneratedBy(_selectedConfig);
00052                 var generates = _evolutionsService.GetGenerates(_selectedConfig);
00053
00054                 bool lockedElements = _selectedConfig.Type == ElementType.Generator &&
00055                     _selectedConfig.GeneratorSettings.GenerateBlocked;
00056
00057                 SetupPanel(View.EvolutionPanel, elementChain, _prevSetUpPanel,
00058                     evolutionData.Description, evolutionData, true);
00059                 SetupPanel(View.GeneratesPanel, generates, _prevSetUpPanel, "Produce:", lockedItems:
00060                     lockedElements);
00061                 SetupPanel(View.CreatedFromPanel, generatorsChain, _prevSetUpPanel, "Created from:");
00062
00063                 CalculateScrollSize(View.EvolutionPanel.RectTransform,
00064                     View.GeneratesPanel.RectTransform, View.CreatedFromPanel.RectTransform);
00065             }
00066         }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Windows/ElementInfo/ElementInfoPresenter.cs

6.60 Mergelt.Game.Windows.ElementInfo.ElementInfoWindow Class Reference

Inheritance diagram for Mergelt.Game.Windows.ElementInfo.ElementInfoWindow:



Properties

- TMP_Text **WindowTitleText** [get]
- float **SpacingBetweenPanels** [get]
- float **SpacingInsidePanels** [get]
- ElementInfoItemsPanelComponent EvolutionPanel** [get]
- ElementInfoItemsPanelComponent GeneratesPanel** [get]
- ElementInfoItemsPanelComponent CreatedFromPanel** [get]
- RectTransform **ScrollContent** [get]

Properties inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- [Canvas](#) `Canvas` [get]
- [IWindowAnimationController](#) `AnimationController` [get]

Properties inherited from [Mergelt.Core.MVP.View](#)

- [GameObject](#) `GameObject` [get]

Properties inherited from [Mergelt.Core.MVP.IView](#)

Additional Inherited Members

Public Member Functions inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- virtual void [Show](#) ()
- virtual void [Close](#) ()
- virtual void [SetLayer](#) (string layer)
- virtual void [OnOpenStarted](#) ()
- virtual void [OnOpenFinished](#) ()
- virtual void [OnCloseStarted](#) ()
- virtual void [OnCloseFinished](#) ()

Public Member Functions inherited from [Mergelt.Core.MVP.View](#)

- virtual void [Initialize](#) ()

Protected Member Functions inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- override void [OnDestroy](#) ()

Protected Attributes inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- [Button](#) `CloseButton`
- [Button](#) `ClickOutsideArea`

Events inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- Action [InitiateCloseEvent](#)
- Action [ShowStartEvent](#)
- Action [ShowEndEvent](#)
- Action [CloseStartEvent](#)
- Action [CloseEndEvent](#)

Events inherited from [Mergelt.Core.MVP.View](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

Events inherited from [Mergelt.Core.MVP.IView](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

Events inherited from [Mergelt.Core.WindowSystem.Windows.IWindow](#)

- Action [InitiateCloseEvent](#)
- Action [ShowStartEvent](#)
- Action [ShowEndEvent](#)
- Action [CloseStartEvent](#)
- Action [CloseEndEvent](#)

6.60.1 Detailed Description

Definition at line 9 of file [ElementInfoWindow.cs](#).

6.60.2 Property Documentation

6.60.2.1 CreatedFromPanel

```
ElementInfoItemsPanelComponent MergeIt.Game.Windows.ElementInfo.ElementInfoWindow.CreatedFromPanel [get]
```

Definition at line 57 of file [ElementInfoWindow.cs](#).

```
00058      {  
00059          get => _createdFromPanel;  
00060      }
```

6.60.2.2 EvolutionPanel

```
ElementInfoItemsPanelComponent MergeIt.Game.Windows.ElementInfo.ElementInfoWindow.EvolutionPanel [get]
```

Definition at line 47 of file [ElementInfoWindow.cs](#).

```
00048      {  
00049          get => _evolutionPanel;  
00050      }
```

6.60.2.3 GeneratesPanel

```
ElementInfoItemsPanelComponent MergeIt.Game.Windows.ElementInfo.ElementInfoWindow.GeneratesPanel [get]
```

Definition at line 52 of file [ElementInfoWindow.cs](#).

```
00053      {  
00054          get => _generatesPanel;  
00055      }
```

6.60.2.4 ScrollContent

```
RectTransform MergeIt.Game.Windows.ElementInfo.ElementInfoWindow.ScrollContent [get]
```

Definition at line 62 of file [ElementInfoWindow.cs](#).

```
00063     {
00064         get => _scrollContent;
00065     }
```

6.60.2.5 SpacingBetweenPanels

```
float MergeIt.Game.Windows.ElementInfo.ElementInfoWindow.SpacingBetweenPanels [get]
```

Definition at line 37 of file [ElementInfoWindow.cs](#).

```
00038     {
00039         get => _spacingBetweenPanels;
00040     }
```

6.60.2.6 SpacingInsidePanels

```
float MergeIt.Game.Windows.ElementInfo.ElementInfoWindow.SpacingInsidePanels [get]
```

Definition at line 42 of file [ElementInfoWindow.cs](#).

```
00043     {
00044         get => _spacingInsidePanels;
00045     }
```

6.60.2.7 WindowTitleText

```
TMP_Text MergeIt.Game.Windows.ElementInfo.ElementInfoWindow.WindowTitleText [get]
```

Definition at line 32 of file [ElementInfoWindow.cs](#).

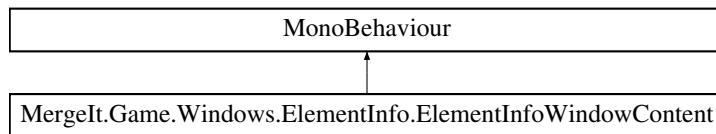
```
00033     {
00034         get => _windowTitleText;
00035     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Windows/ElementInfo/ElementInfoWindow.cs

6.61 MergeIt.Game.Windows.ElementInfo.ElementInfoWindowContent Class Reference

Inheritance diagram for MergeIt.Game.Windows.ElementInfo.ElementInfoWindowContent:



6.61.1 Detailed Description

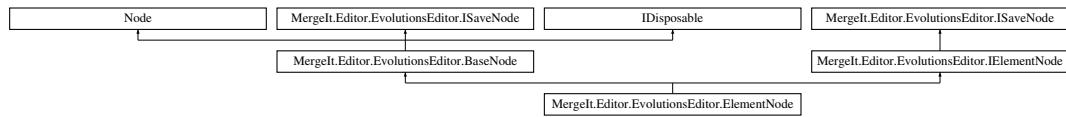
Definition at line 7 of file [ElementInfoWindowContent.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Windows/ElementInfo/ElementInfoWindowContent.cs

6.62 Mergelt.Editor.EvolutionsEditor.ElementNode Class Reference

Inheritance diagram for Mergelt.Editor.EvolutionsEditor.ElementNode:



Public Member Functions

- [ElementNode](#) ([ElementNodeData](#) data, Vector2 positionOffset, [EvolutionsGraphView](#) graphView)
- override void [SaveData](#) ()
- override void [BuildContextualMenu](#) (ContextualMenuPopulateEvent evt)
- override void [Dispose](#) ()
- bool [Validate](#) ()

Public Member Functions inherited from [Mergelt.Editor.EvolutionsEditor.BaseNode](#)

- void [SaveData](#) ()
- Port [GetPort](#) (string portName)

Properties

- override string [Id](#) [get]
- [IElementNodeData](#) [Data](#) [get]
- [ElementConfig](#) [Config](#) [get]

Properties inherited from [Mergelt.Editor.EvolutionsEditor.BaseNode](#)

- string [Id](#) [get]

Properties inherited from [Mergelt.Editor.EvolutionsEditor.ISaveNode](#)

Properties inherited from [Mergelt.Editor.EvolutionsEditor.IElementNode](#)

6.62.1 Detailed Description

Definition at line 19 of file [ElementNode.cs](#).

6.62.2 Constructor & Destructor Documentation

6.62.2.1 ElementNode()

```
MergeIt.Editor.EvolutionsEditor.ElementNode.ElementNode (
    ElementNodeData data,
    Vector2 positionOffset,
    EvolutionsGraphView graphView ) [inline]
```

Definition at line 41 of file [ElementNode.cs](#).

```
00042     {
00043         _graphView = graphView;
00044         Data = data;
00045         _config = data.ElementConfig;
00046
00047         if (_config)
00048         {
00049             var style1 =
00050                 (StyleSheet)EditorGUIUtility.Load(Path.Combine(Constants.EvolutionsEditorResourcesRoot,
00051                                         "GeneratorStyle.uss"));
00052
00053             var titleLabel = titleContainer.Q<Label>("title-label");
00054             titleLabel.bindingPath = "_commonSettings._name";
00055             Port evolvedFromPort = InstantiatePort(Orientation.Horizontal, Direction.Input,
00056                                         Port.Capacity.Single,
00057                                         typeof(string));
00058
00059             evolvedFromPort.portName = Constants.EvolvedFromPort;
00060             inputContainer.Add(evolvedFromPort);
00061
00062             Port evolvePort = InstantiatePort(Orientation.Horizontal, Direction.Output,
00063                                         Port.Capacity.Single,
00064                                         typeof(string));
00065             evolvePort.portName = Constants.EvolvePort;
00066             outputContainer.Add(evolvePort);
00067
00068             _root = new VisualElement();
00069
00070             mainContainer.AddToClassList(StylesConstants.NodeStyles[ElementType.Regular]);
00071             mainContainer.AddToClassList(StylesConstants.NodeStyles[ElementType.Generator]);
00072
00073             var typeEnum = new EnumField(_config.Type);
00074             typeEnum.bindingPath = "_type";
00075             typeEnum.RegisterValueChangedCallback(OnElementTypeChanged);
00076
00077             _root.Add(typeEnum);
00078
00079             Refresh(_config.Type);
00080
00081             mainContainer.Add(_root);
00082
00083             Data.Position = new Rect(
00084                 data.Position.x + positionOffset.x,
00085                 data.Position.y + positionOffset.y,
00086                 data.Position.width,
00087                 data.Position.height);
00088
00089             mainContainer.Bind(new SerializedObject(_config));
00090
00091             _configName = _config.name;
00092             _root.styleSheets.Add(style1);
00093         }
00094         else
00095         {
00096             mainContainer.AddToClassList(StylesConstants.InvalidNode);
00097             title = "Config was missing";
00098         }
00099
00100         SetPosition(Data.Position);
00101     }
```

6.62.3 Member Function Documentation

6.62.3.1 BuildContextualMenu()

```
override void MergeIt.Editor.EvolutionsEditor.ElementNode.BuildContextualMenu (
    ContextualMenuPopulateEvent evt ) [inline]
```

Definition at line 105 of file [ElementNode.cs](#).

```
00106         {
00107             base.BuildContextualMenu(evt);
00108
00109             evt.menu.AppendAction("Select in Project View", SelectAction);
00110             evt.menu.AppendSeparator();
00111     }
```

6.62.3.2 Dispose()

```
override void MergeIt.Editor.EvolutionsEditor.ElementNode.Dispose () [inline], [virtual]
```

Reimplemented from [Mergelt.Editor.EvolutionsEditor.BaseNode](#).

Definition at line 113 of file [ElementNode.cs](#).

```
00114         {
00115             base.Dispose();
00116
00117             _generatorSettingsDrawer?.Dispose();
00118             _commonSettingsDrawer?.Dispose();
00119
00120             mainContainer.Unbind();
00121     }
```

6.62.3.3 SaveData()

```
override void MergeIt.Editor.EvolutionsEditor.ElementNode.SaveData () [inline]
```

Implements [Mergelt.Editor.EvolutionsEditor.ISaveNode](#).

Definition at line 100 of file [ElementNode.cs](#).

```
00101         {
00102             Data.Position = GetPosition ();
00103     }
```

6.62.3.4 Validate()

```
bool MergeIt.Editor.EvolutionsEditor.ElementNode.Validate () [inline]
```

Implements [Mergelt.Editor.EvolutionsEditor.IElementNode](#).

Definition at line 123 of file [ElementNode.cs](#).

```
00124         {
00125             if (Data.ElementConfig)
00126             {
00127                 return true;
00128             }
00129
00130             title = $"Lost config: {_configName}";
00131             mainContainer.AddToClassList(StylesConstants.InvalidNode);
00132             mainContainer.EnableInclassList(StylesConstants.NodeStyles[ElementType.Regular], false);
00133             mainContainer.EnableInclassList(StylesConstants.NodeStyles[ElementType.Generator], false);
00134
00135             mainContainer.SetEnabled(false);
00136             return false;
00137     }
```

6.62.4 Property Documentation

6.62.4.1 Config

`ElementConfig` MergeIt.Editor.EvolutionsEditor.ElementNode.Config [get]

Implements [Mergelt.Editor.EvolutionsEditor.IElementNode](#).

Definition at line 36 of file [ElementNode.cs](#).

```
00037      {  
00038          get => _config;  
00039      }
```

6.62.4.2 Data

`IElementNodeData` MergeIt.Editor.EvolutionsEditor.ElementNode.Data [get]

Implements [Mergelt.Editor.EvolutionsEditor.IElementNode](#).

Definition at line 34 of file [ElementNode.cs](#).

```
00034 { get; private set; }
```

6.62.4.3 Id

`override string` MergeIt.Editor.EvolutionsEditor.ElementNode.Id [get]

Implements [Mergelt.Editor.EvolutionsEditor.ISaveNode](#).

Definition at line 29 of file [ElementNode.cs](#).

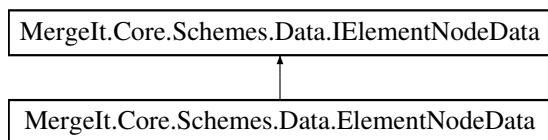
```
00030      {  
00031          get => Data?.Id;  
00032      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/EvolutionsEditor/ItemsNodes/ElementNode.cs

6.63 Mergelt.Core.Schemes.Data.ElementNodeData Class Reference

Inheritance diagram for Mergelt.Core.Schemes.Data.ElementNodeData:



Public Member Functions

- T `Copy< T >()`

Properties

- string `Id` [get, set]
- `ElementConfig ElementConfig` [get, set]
- `Rect Position` [get, set]

Properties inherited from [Mergelt.Core.Schemes.Data.IElementNodeData](#)

6.63.1 Detailed Description

Definition at line 10 of file [ElementNodeData.cs](#).

6.63.2 Member Function Documentation

6.63.2.1 `Copy< T >()`

`T MergeIt.Core.Schemes.Data.ElementNodeData.Copy< T > ()` [inline]

Implements [Mergelt.Core.Schemes.Data.IElementNodeData](#).

Type Constraints

`T : IElementNodeData`

`T : new()`

Definition at line 34 of file [ElementNodeData.cs](#).

```
00034             : IElementNodeData, new()
00035     {
00036         var copiedData = new T
00037         {
00038             Position = Position,
00039             ElementConfig = ElementConfig
00040         };
00041         return copiedData;
00042     }
00043 }
```

6.63.3 Property Documentation

6.63.3.1 `ElementConfig`

`ElementConfig MergeIt.Core.Schemes.Data.ElementNodeData.ElementConfig` [get], [set]

Implements [Mergelt.Core.Schemes.Data.IElementNodeData](#).

Definition at line 22 of file [ElementNodeData.cs](#).

```
00023     {
00024         get => _elementConfig;
00025         set => _elementConfig = value;
00026     }
```

6.63.3.2 Id

string MergeIt.Core.Schemes.Data.ElementNodeData.Id [get], [set]

Implements [MergeIt.Core.Schemes.Data.IElementNodeData](#).

Definition at line 16 of file [ElementNodeData.cs](#).

```
00017     {
00018         get => _id;
00019         set => _id = value;
00020     }
```

6.63.3.3 Position

Rect MergeIt.Core.Schemes.Data.ElementNodeData.Position [get], [set]

Implements [MergeIt.Core.Schemes.Data.IElementNodeData](#).

Definition at line 28 of file [ElementNodeData.cs](#).

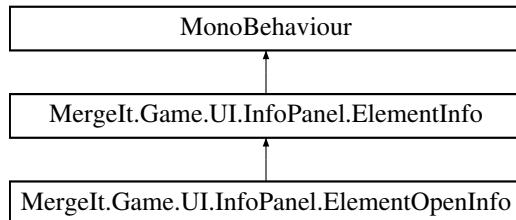
```
00029     {
00030         get => _position;
00031         set => _position = value;
00032     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Schemes/Data/ElementNodeData.cs

6.64 MergeIt.Game.UI.InfoPanel.ElementOpenInfo Class Reference

Inheritance diagram for MergeIt.Game.UI.InfoPanel.ElementOpenInfo:



Protected Member Functions

- override void [OnStart](#) ()
- override bool [OnTrySetup](#) ()
- override void [Destroy](#) ()

Protected Member Functions inherited from [MergeIt.Game.UI.InfoPanel.ElementInfo](#)

- bool [OnTrySetup](#) ()
- void [Clear](#) ()
- virtual void [OnClear](#) ()
- void [ActionButtonClick](#) ()
- virtual void [OnActionButtonClick](#) ()
- void [SendActionMessage](#) ()

Properties

- override ElementActionType [ActionType](#) [get]

Properties inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- ElementActionType [ActionType](#) [get]

Additional Inherited Members

Public Member Functions inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- void [TrySetup](#) ([IFieldElement](#) fieldElement)

Protected Attributes inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- [IConfigsService](#) ConfigsService
- [IMessageBus](#) MessageBus
- [IFieldElement](#) SelectedElement

6.64.1 Detailed Description

Definition at line 11 of file [ElementOpenInfo.cs](#).

6.64.2 Member Function Documentation

6.64.2.1 Destroy()

```
override void MergeIt.Game.UI.InfoPanel.ElementOpenInfo.Destroy () [inline], [protected],  
[virtual]
```

Reimplemented from [Mergelt.Game.UI.InfoPanel.ElementInfo](#).

Definition at line 49 of file [ElementOpenInfo.cs](#).

```
00050      {  
00051          base.Destroy();  
00052  
00053          _openButton.onClick.RemoveListener(ActionButtonClick);  
00054      }
```

6.64.2.2 OnStart()

```
override void MergeIt.Game.UI.InfoPanel.ElementOpenInfo.OnStart () [inline], [protected],  
[virtual]
```

Reimplemented from [Mergelt.Game.UI.InfoPanel.ElementInfo](#).

Definition at line 21 of file [ElementOpenInfo.cs](#).

```
00022      {  
00023          base.OnStart();  
00024  
00025          _openButton.onClick.AddListener(ActionButtonClick);  
00026      }
```

6.64.2.3 OnTrySetup()

```
override bool MergeIt.Game.UI.InfoPanel.ElementOpenInfo.OnTrySetup () [inline], [protected]
```

Definition at line 28 of file [ElementOpenInfo.cs](#).

```
00029     {
00030         if (SelectedElement.InfoParameters.Type == ElementType.Generator &&
00031             !SelectedElement.InfoParameters.IsBlocked)
00032         {
00033             if (SelectedElement.GeneratorOpenParameters is {IsOpening: false})
00034             {
00035                 int openTime =
00036                     SelectedElement.ConfigParameters.ElementConfig.GeneratorSettings.OpenTime;
00037
00038                 gameObject.SetActive(true);
00039
00040                 _openTime.text = openTime.FormatTime();
00041
00042                 return true;
00043             }
00044         }
00045
00046         return false;
00047     }
```

6.64.3 Property Documentation

6.64.3.1 ActionType

```
override ElementActionType MergeIt.Game.UI.InfoPanel.ElementOpenInfo.ActionType [get]
```

Definition at line 19 of file [ElementOpenInfo.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/UI/InfoPanel/ElementOpenInfo.cs

6.65 MergeIt.Game.Configs.ElementPrefab Class Reference

Properties

- ElementType [Type](#) [get]
- [FieldElementView](#) [Prefab](#) [get]

6.65.1 Detailed Description

Definition at line 11 of file [ElementPrefab.cs](#).

6.65.2 Property Documentation

6.65.2.1 Prefab

```
FieldElementView MergeIt.Game.Configs.ElementPrefab.Prefab [get]
```

Definition at line 24 of file [ElementPrefab.cs](#).

```
00025     {
00026         get => _prefab;
00027     }
```

6.65.2.2 Type

```
ElementType MergeIt.Game.Configs.ElementPrefab.Type [get]
```

Definition at line 19 of file [ElementPrefab.cs](#).

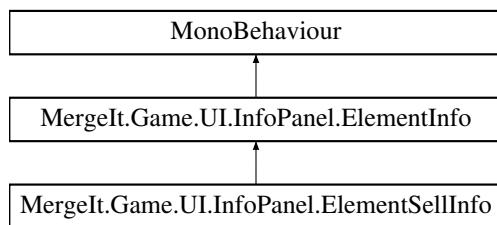
```
00020     {
00021         get => _type;
00022     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Configs/ElementPrefab.cs

6.66 Mergelt.Game.UI.InfoPanel.ElementSellInfo Class Reference

Inheritance diagram for Mergelt.Game.UI.InfoPanel.ElementSellInfo:



Protected Member Functions

- override void [OnStart \(\)](#)
- override bool [OnTrySetup \(\)](#)
- override void [Destroy \(\)](#)

Protected Member Functions inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- bool [OnTrySetup \(\)](#)
- void [Clear \(\)](#)
- virtual void [OnClear \(\)](#)
- void [ActionButtonClick \(\)](#)
- virtual void [OnActionButtonClick \(\)](#)
- void [SendActionMessage \(\)](#)

Properties

- override Element ActionType [ActionType](#) [get]

Properties inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- Element ActionType [ActionType](#) [get]

Additional Inherited Members

Public Member Functions inherited from [MergeIt.Game.UI.InfoPanel.ElementInfo](#)

- void [TrySetup](#) ([IFieldElement](#) fieldElement)

Protected Attributes inherited from [MergeIt.Game.UI.InfoPanel.ElementInfo](#)

- [IConfigsService](#) ConfigsService
- [IMessageBus](#) MessageBus
- [IFieldElement](#) SelectedElement

6.66.1 Detailed Description

Definition at line 11 of file [ElementSellInfo.cs](#).

6.66.2 Member Function Documentation

6.66.2.1 Destroy()

```
override void MergeIt.Game.UI.InfoPanel.ElementSellInfo.Destroy () [inline], [protected], [virtual]
```

Reimplemented from [MergeIt.Game.UI.InfoPanel.ElementInfo](#).

Definition at line 60 of file [ElementSellInfo.cs](#).

```
00061     {
00062         base.Destroy ();
00063
00064         _sellButton.onClick.RemoveListener(ActionButtonClick);
00065     }
```

6.66.2.2 OnStart()

```
override void MergeIt.Game.UI.InfoPanel.ElementSellInfo.OnStart () [inline], [protected], [virtual]
```

Reimplemented from [MergeIt.Game.UI.InfoPanel.ElementInfo](#).

Definition at line 27 of file [ElementSellInfo.cs](#).

```
00028     {
00029         base.OnStart ();
00030
00031         _sellButton.onClick.AddListener(ActionButtonClick);
00032     }
```

6.66.2.3 OnTrySetup()

```
override bool MergeIt.Game.UI.InfoPanel.ElementSellInfo.OnTrySetup () [inline], [protected]
```

Definition at line 34 of file [ElementSellInfo.cs](#).

```
00035     {
00036         CurrencySettings sellCostSettings =
00037             SelectedElement.ConfigParameters.ElementConfig.CommonSettings.SellCostSettings;
00038
00039         if (sellCostSettings.Amount > 0 && !SelectedElement.InfoParameters.IsBlocked)
00040         {
00041             if (SelectedElement.InfoParameters.Type == ElementType.Generator)
00042             {
00043                 if (SelectedElement.GeneratorParameters.AvailableToDrop == 0)
00044                 {
00045                     return false;
00046                 }
00047             }
00048
00049             gameObject.SetActive(true);
00050
00051             _costTypeIcon.sprite = ConfigsService.GetCurrencyIcon(sellCostSettings.Currency);
00052             _costLabel.text = sellCostSettings.Amount.ToString();
00053
00054             return true;
00055         }
00056
00057         return false;
00058     }
```

6.66.3 Property Documentation

6.66.3.1 ActionType

```
override ElementActionType MergeIt.Game.UI.InfoPanel.ElementSellInfo.ActionType [get]
```

Definition at line 22 of file [ElementSellInfo.cs](#).

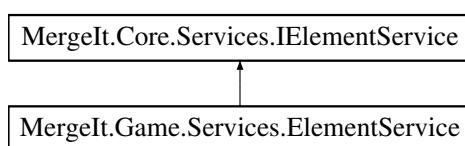
```
00023     {
00024         get => ElementActionType.Sell;
00025     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/UI/InfoPanel/ElementSellInfo.cs

6.67 Mergelt.Game.Services.ElementService Class Reference

Inheritance diagram for Mergelt.Game.Services.ElementService:



Public Member Functions

- void [TrySell \(IFieldElement fieldElement\)](#)
- void [TryUnlock \(IFieldElement fieldElement\)](#)
- void [TrySplit \(IFieldElement element\)](#)

6.67.1 Detailed Description

Definition at line 16 of file [ElementService.cs](#).

6.67.2 Member Function Documentation

6.67.2.1 TrySell()

```
void MergeIt.Game.Services.ElementService.TrySell (
    IFieldElement fieldElement) [inline]
```

Implements [MergeIt.Core.Services.IElementService](#).

Definition at line 39 of file [ElementService.cs](#).

```
00040     {
00041         CurrencySettings costSettings =
00042             fieldElement.ConfigParameters.ElementConfig.CommonSettings.SellCostSettings;
00043         _currencyService.Sell(costSettings);
00044         GridPoint point = fieldElement.InfoParameters.LogicPosition;
00045         var removeMessage = new RemoveElementMessage {RemoveAtPoint = point};
00046         _messageBus.Fire(removeMessage);
00047         _messageBus.Fire<ResetSelectionMessage>();
00048         _saveService.Save(GameSaveType.Field);
00049     }
00050 }
```

6.67.2.2 TrySplit()

```
void MergeIt.Game.Services.ElementService.TrySplit (
    IFieldElement element) [inline]
```

Implements [MergeIt.Core.Services.IElementService](#).

Definition at line 69 of file [ElementService.cs](#).

```
00070     {
00071         var freeCellPoint = _gameFieldService.GetFreeCell();
00072         if (freeCellPoint == null)
00073         {
00074             return;
00075         }
00076         GridPoint secondPoint = freeCellPoint.Value;
00077         var splitResult = TrySplit(element, secondPoint);
00078         if (splitResult != null)
00079         {
00080             CurrencySettings costSettings =
00081                 element.ConfigParameters.ElementConfig.CommonSettings.SplitCostSettings;
00082             if (_currencyService.TryPay(costSettings))
00083             {
00084                 SendSplitMessage(splitResult.Value);
00085                 _saveService.Save(GameSaveType.Field);
00086                 _messageBus.Fire<ResetSelectionMessage>();
00087             }
00088         }
00089     }
00090 }
```

6.67.2.3 TryUnlock()

```
void MergeIt.Game.Services.ElementService.TryUnlock (
    IFIELDELEMENT fieldElement ) [inline]
```

Implements [Mergelt.Core.Services.IElementService](#).

Definition at line 54 of file [ElementService.cs](#).

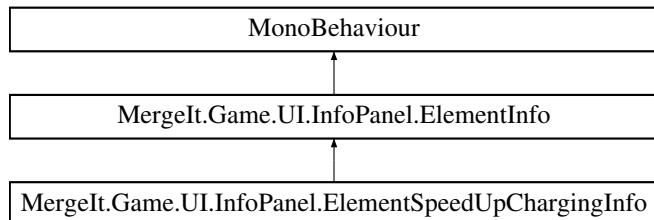
```
00055     {
00056         CURRENCYSETTINGS costSettings =
00057             fieldElement.ConfigParameters.ElementConfig.CommonSettings.UnlockCostSettings;
00058         if (_currencyService.TryPay(costSettings))
00059         {
00060             fieldElement.InfoParameters.IsBlocked = false;
00061             _saveService.Save(GameSaveType.Field);
00062             var unlockMessage = new UnlockElementMessage { Element = fieldElement };
00063             _messageBus.Fire(unlockMessage);
00064         }
00065     }
00066 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/ElementService.cs

6.68 Mergelt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo Class Reference

Inheritance diagram for Mergelt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo:



Protected Member Functions

- override void [OnStart](#) ()
- override bool [OnTrySetup](#) ()
- override void [Destroy](#) ()

Protected Member Functions inherited from [MergeIt.Game.UI.InfoPanel.ElementInfo](#)

- bool [OnTrySetup](#) ()
- void [Clear](#) ()
- virtual void [OnClear](#) ()
- void [ActionButtonClick](#) ()
- virtual void [OnActionButtonClick](#) ()
- void [SendActionMessage](#) ()

Properties

- override Element ActionType **ActionType** [get]

Properties inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- Element ActionType **ActionType** [get]

Additional Inherited Members

Public Member Functions inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- void [TrySetup](#) (IFieldElement fieldElement)

Protected Attributes inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- IConfigsService **ConfigsService**
- IMessageBus **MessageBus**
- IFieldElement **SelectedElement**

6.68.1 Detailed Description

Definition at line 11 of file [ElementSpeedUpChargingInfo.cs](#).

6.68.2 Member Function Documentation

6.68.2.1 Destroy()

```
override void MergeIt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo.Destroy ( ) [inline],  
[protected], [virtual]
```

Reimplemented from [Mergelt.Game.UI.InfoPanel.ElementInfo](#).

Definition at line 58 of file [ElementSpeedUpChargingInfo.cs](#).

```
00059      {  
00060          base.Destroy();  
00061  
00062          _skipChargeButton.onClick.AddListener(OnActionButtonClick);  
00063      }
```

6.68.2.2 OnStart()

```
override void MergeIt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo.OnStart ( ) [inline],  
[protected], [virtual]
```

Reimplemented from [Mergelt.Game.UI.InfoPanel.ElementInfo](#).

Definition at line 27 of file [ElementSpeedUpChargingInfo.cs](#).

```
00028      {  
00029          base.OnStart();  
00030  
00031          _skipChargeButton.onClick.AddListener(OnActionButtonClick);  
00032      }
```

6.68.2.3 OnTrySetup()

```
override bool MergeIt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo.OnTrySetup () [inline],  
[protected]
```

Definition at line 34 of file [ElementSpeedUpChargingInfo.cs](#).

```
00035         {  
00036             if (SelectedElement.InfoParameters.Type == ElementType.Generator &&  
00037                 !SelectedElement.InfoParameters.IsBlocked)  
00038             {  
00039                 CurrencySettings skipChargeCostSettings =  
00040                     SelectedElement.ConfigParameters.ElementConfig.GeneratorSettings.SkipChargeCostSettings;  
00041  
00042                     bool generatorCharging = SelectedElement.GeneratorParameters.AvailableToDrop == 0;  
00043  
00044                     if (generatorCharging && skipChargeCostSettings.Amount > 0)  
00045                     {  
00046                         gameObject.SetActive(true);  
00047  
00048                         _costTypeIcon.sprite =  
ConfigsService.GetCurrencyIcon(skipChargeCostSettings.Currency);  
00049                         _costLabel.text = skipChargeCostSettings.Amount.ToString();  
00050  
00051                     }  
00052                 }  
00053             }  
00054  
00055             return true;  
00056         }  
00057     }
```

6.68.3 Property Documentation

6.68.3.1 ActionType

```
override ElementActionType MergeIt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo.ActionType  
[get]
```

Definition at line 22 of file [ElementSpeedUpChargingInfo.cs](#).

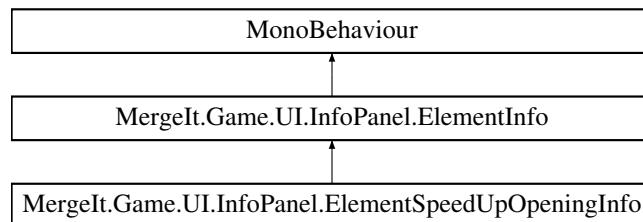
```
00023         {  
00024             get => ElementActionType.SkipCharging;  
00025         }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/UI/InfoPanel/ElementSpeedUpChargingInfo.cs

6.69 MergeIt.Game.UI.InfoPanel.ElementSpeedUpOpeningInfo Class Reference

Inheritance diagram for MergeIt.Game.UI.InfoPanel.ElementSpeedUpOpeningInfo:



Protected Member Functions

- override void [OnStart \(\)](#)
- override bool [OnTrySetup \(\)](#)
- override void [OnClear \(\)](#)

Protected Member Functions inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- bool [OnTrySetup \(\)](#)
- void [Clear \(\)](#)
- virtual void [Destroy \(\)](#)
- void [ActionButtonClick \(\)](#)
- virtual void [OnActionButtonClick \(\)](#)
- void [SendActionMessage \(\)](#)

Properties

- override Element ActionType [ActionType](#) [get]

Properties inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- Element ActionType [ActionType](#) [get]

Additional Inherited Members

Public Member Functions inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- void [TrySetup \(IFieldElement fieldElement\)](#)

Protected Attributes inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- IConfigsService [ConfigsService](#)
- IMessageBus [MessageBus](#)
- IFieldElement [SelectedElement](#)

6.69.1 Detailed Description

Definition at line 14 of file [ElementSpeedUpOpeningInfo.cs](#).

6.69.2 Member Function Documentation

6.69.2.1 OnClear()

```
override void MergeIt.Game.UI.InfoPanel.ElementSpeedUpOpeningInfo.OnClear ( ) [inline], [protected], [virtual]
```

Reimplemented from [Mergelt.Game.UI.InfoPanel.ElementInfo](#).

Definition at line 78 of file [ElementSpeedUpOpeningInfo.cs](#).

```
00079      {
00080          base.OnClear();
00081
00082          _remainingTime?.Unsubscribe(OnRemainingTimeChanged);
00083          _remainingTime = null;
00084      }
```

6.69.2.2 OnStart()

```
override void MergeIt.Game.UI.InfoPanel.ElementSpeedUpOpeningInfo.OnStart () [inline], [protected],  
[virtual]
```

Reimplemented from [Mergelt.Game.UI.InfoPanel.ElementInfo](#).

Definition at line 35 of file [ElementSpeedUpOpeningInfo.cs](#).

```
00036     {  
00037         base.OnStart ();  
00038         _skipOpeningButton.onClick.AddListener (OnActionButtonClick);  
00039     }
```

6.69.2.3 OnTrySetup()

```
override bool MergeIt.Game.UI.InfoPanel.ElementSpeedUpOpeningInfo.OnTrySetup () [inline],  
[protected]
```

Definition at line 41 of file [ElementSpeedUpOpeningInfo.cs](#).

```
00042     {  
00043         if (SelectedElement.InfoParameters.Type == ElementType.Generator &&  
00044             !SelectedElement.InfoParameters.IsBlocked)  
00045         {  
00046             CurrencySettings skipOpenCostSettings =  
00047                 SelectedElement.ConfigParameters.ElementConfig.GeneratorSettings.SkipOpenCostSettings;  
00048  
00049             IGeneratorOpenParameters openParameters = SelectedElement.GeneratorOpenParameters;  
00050  
00051             if (openParameters is { IsOpening: true } &&  
00052                 openParameters.RemainingTime.Value > 0f)  
00053             {  
00054                 gameObject.SetActive(true);  
00055  
00056                 _remainingTime?.Unsubscribe(OnRemainingTimeChanged);  
00057                 _remainingTime = openParameters.RemainingTime;  
00058                 _remainingTime.Subscribe(OnRemainingTimeChanged);  
00059                 _openingTimerLabel.text = _remainingTime.Value.FormatTime();  
00060                 _costTypeIcon.sprite =  
00061                     ConfigsService.GetCurrencyIcon(skipOpenCostSettings.Currency);  
00062  
00063                 if (skipOpenCostSettings.Amount == 0)  
00064                 {  
00065                     _skipOpeningButton.gameObject.SetActive(false);  
00066                 }  
00067                 else  
00068                 {  
00069                     _costLabel.text = skipOpenCostSettings.Amount.ToString();  
00070                 }  
00071             }  
00072         }  
00073     }  
00074  
00075     return false;  
00076 }
```

6.69.3 Property Documentation

6.69.3.1 ActionType

```
override ElementActionType MergeIt.Game.UI.InfoPanel.ElementSpeedUpOpeningInfo.ActionType  
[get]
```

Definition at line 30 of file [ElementSpeedUpOpeningInfo.cs](#).

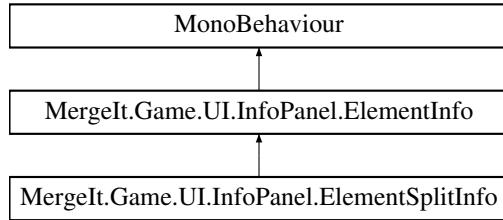
```
00031     {  
00032         get => ElementActionType.SkipOpening;  
00033     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/UI/InfoPanel/ElementSpeedUpOpeningInfo.cs

6.70 MergeIt.Game.UI.InfoPanel.ElementSplitInfo Class Reference

Inheritance diagram for MergeIt.Game.UI.InfoPanel.ElementSplitInfo:



Protected Member Functions

- override void [OnStart \(\)](#)
- override bool [OnTrySetup \(\)](#)
- override void [Destroy \(\)](#)

Protected Member Functions inherited from [MergeIt.Game.UI.InfoPanel.ElementInfo](#)

- bool [OnTrySetup \(\)](#)
- void [Clear \(\)](#)
- virtual void [OnClear \(\)](#)
- void [ActionButtonClick \(\)](#)
- virtual void [OnActionButtonClick \(\)](#)
- void [SendActionMessage \(\)](#)

Properties

- override Element ActionType [ActionType \[get\]](#)

Properties inherited from [MergeIt.Game.UI.InfoPanel.ElementInfo](#)

- Element ActionType [ActionType \[get\]](#)

Additional Inherited Members

Public Member Functions inherited from [MergeIt.Game.UI.InfoPanel.ElementInfo](#)

- void [TrySetup \(IFieldElement fieldElement\)](#)

Protected Attributes inherited from [MergeIt.Game.UI.InfoPanel.ElementInfo](#)

- IConfigsService [ConfigsService](#)
- IMessageBus [MessageBus](#)
- IFieldElement [SelectedElement](#)

6.70.1 Detailed Description

Definition at line 14 of file [ElementSplitInfo.cs](#).

6.70.2 Member Function Documentation

6.70.2.1 Destroy()

```
override void MergeIt.Game.UI.InfoPanel.ElementSplitInfo.Destroy () [inline], [protected],  
[virtual]
```

Reimplemented from [Mergelt.Game.UI.InfoPanel.ElementInfo](#).

Definition at line 61 of file [ElementSplitInfo.cs](#).

```
00062     {  
00063         base.Destroy();  
00064  
00065         _splitButton.onClick.RemoveListener(ActionButtonClick);  
00066         _userServiceModel?.Splitters.Unsubscribe(OnSplittersValueChanged);  
00067     }
```

6.70.2.2 OnStart()

```
override void MergeIt.Game.UI.InfoPanel.ElementSplitInfo.OnStart () [inline], [protected],  
[virtual]
```

Reimplemented from [Mergelt.Game.UI.InfoPanel.ElementInfo](#).

Definition at line 32 of file [ElementSplitInfo.cs](#).

```
00033     {  
00034         base.OnStart();  
00035  
00036         TrySetupUser();  
00037  
00038         _splitButton.onClick.AddListener(ActionButtonClick);  
00039     }
```

6.70.2.3 OnTrySetup()

```
override bool MergeIt.Game.UI.InfoPanel.ElementSplitInfo.OnTrySetup () [inline], [protected]
```

Definition at line 41 of file [ElementSplitInfo.cs](#).

```
00042     {  
00043         TrySetupUser();  
00044  
00045         ElementCommonSettings commonSettings =  
SelectedElement.ConfigParameters.ElementConfig.CommonSettings;  
00046  
00047         if (!SelectedElement.InfoParameters.IsBlocked &&  
SelectedElement.GetPreviousInEvolution() &&  
commonSettings.SplitCostSettings.Amount > 0)  
00048         {  
00049             gameObject.SetActive(true);  
00050  
00051             _costLabel.text = commonSettings.SplitCostSettings.Amount.ToString();  
00052  
00053             return true;  
00054         }  
00055  
00056         return false;  
00057     }
```

6.70.3 Property Documentation

6.70.3.1 ActionType

```
override ElementActionType MergeIt.Game.UI.InfoPanel.ElementSplitInfo.ActionType [get]
```

Definition at line 27 of file [ElementSplitInfo.cs](#).

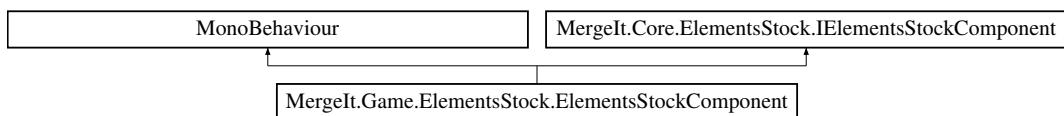
```
00028     {
00029         get => ElementActionType.Split;
00030     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/UI/InfoPanel/ElementSplitInfo.cs

6.71 MergeIt.Game.ElementsStock.ElementsStockComponent Class Reference

Inheritance diagram for MergeIt.Game.ElementsStock.ElementsStockComponent:



Public Member Functions

- void [Initialize\(\)](#)
- void [OnDestroy\(\)](#)
- void [SetupElement\(ElementConfig elementConfig\)](#)
- void [PopElement\(\)](#)

6.71.1 Detailed Description

Definition at line 15 of file [ElementsStockComponent.cs](#).

6.71.2 Member Function Documentation

6.71.2.1 Initialize()

```
void MergeIt.Game.ElementsStock.ElementsStockComponent.Initialize () [inline]
```

Definition at line 30 of file [ElementsStockComponent.cs](#).

```
00031     {
00032         _elementButton.onClick.AddListener(OnElementClick);
00033
00034         _messageBus = DiContainer.Get<IMessageBus>();
00035         _fieldService = DiContainer.Get<IGameFieldService>();
00036         _stockService = DiContainer.Get<IElementsStockService>();
00037
00038         _messageBus.AddListener<UpdateStockMessage>(UpdateStockMessageHandler);
00039
00040         SetupElement(_stockService.GetNext ());
00041     }
```

6.71.2.2 OnDestroy()

```
void MergeIt.Game.ElementsStock.ElementsStockComponent.OnDestroy () [inline]
```

Definition at line 43 of file [ElementsStockComponent.cs](#).

```
00044     {
00045         _elementButton.onClick.RemoveListener(OnElementClick);
00046         _messageBus.RemoveListener<UpdateStockMessage>(UpdateStockMessageHandler);
00047     }
```

6.71.2.3 PopElement()

```
void MergeIt.Game.ElementsStock.ElementsStockComponent.PopElement () [inline]
```

Implements [Mergelt.Core.ElementsStock.IElementsStockComponent](#).

Definition at line 64 of file [ElementsStockComponent.cs](#).

```
00065     {
00066         if (_currentElement != null)
00067         {
00068             GridPoint? pointContainer = _fieldService.GetFreeCell();
00069
00070             if (pointContainer != null)
00071             {
00072                 GridPoint point = pointContainer.Value;
00073
00074                 IFIELDELEMENT newElement = _fieldService.CreateNewElement(_currentElement, point);
00075
00076                 var message = new CreateElementMessage
00077                 {
00078                     NewElement = newElement,
00079                     FromPosition = _elementContainer.position,
00080                     ToPoint = point,
00081                 };
00082                 _messageBus.Fire(message);
00083
00084                 if (_iconPrefab)
00085                 {
00086                     Destroy(_iconPrefab);
00087                 }
00088
00089                 if (_stockService.Remove())
00090                 {
00091                     SetupElement(_stockService.GetNext());
00092                 }
00093                 else
00094                 {
00095                     Hide();
00096                 }
00097             }
00098         }
00099     }
```

6.71.2.4 SetupElement()

```
void MergeIt.Game.ElementsStock.ElementsStockComponent.SetupElement (
    ElementConfig elementConfig) [inline]
```

Implements [Mergelt.Core.ElementsStock.IElementsStockComponent](#).

Definition at line 49 of file [ElementsStockComponent.cs](#).

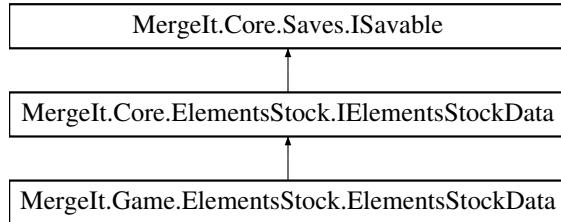
```
00050     {
00051         if (elementConfig != null)
00052         {
00053             gameObject.SetActive(true);
00054             _currentElement = elementConfig;
00055
00056             UpdateView();
00057         }
00058         else
00059         {
00060             Hide();
00061         }
00062     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/ElementsStock/ElementsStockComponent.cs

6.72 Mergelt.Game.ElementsStock.ElementsStockData Class Reference

Inheritance diagram for Mergelt.Game.ElementsStock.ElementsStockData:



Properties

- string[] Elements [get, set]

Properties inherited from [Mergelt.Core.ElementsStock.IElementsStockData](#)

6.72.1 Detailed Description

Definition at line 11 of file [ElementsStockData.cs](#).

6.72.2 Property Documentation

6.72.2.1 Elements

```
string [] MergeIt.Game.ElementsStock.ElementsStockData.Elements [get], [set]
```

Implements [Mergelt.Core.ElementsStock.IElementsStockData](#).

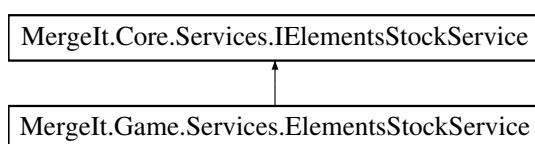
Definition at line 14 of file [ElementsStockData.cs](#).
00014 { get; set; }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/ElementsStock/ElementsStockData.cs

6.73 Mergelt.Game.Services.ElementsStockService Class Reference

Inheritance diagram for Mergelt.Game.Services.ElementsStockService:



Public Member Functions

- void [CreateStock \(\)](#)
- void [SetupStock \(IElementsStockData stockData\)](#)
- void [Add \(ElementConfig elementConfig\)](#)
- bool [Remove \(\)](#)
- [ElementConfig GetNext \(\)](#)
- [ElementConfig GetCurrent \(\)](#)
- [IElementsStockData GetData \(\)](#)

6.73.1 Detailed Description

Definition at line 14 of file [ElementsStockService.cs](#).

6.73.2 Member Function Documentation

6.73.2.1 Add()

```
void MergeIt.Game.Services.ElementsStockService.Add (
    ElementConfig elementConfig ) [inline]
```

Implements [Mergelt.Core.Services.IElementsStockService](#).

Definition at line 43 of file [ElementsStockService.cs](#).

```
00044     {
00045         bool stockIsEmpty = !_serviceModel.Elements.Any();
00046         _serviceModel.Elements.Enqueue(elementConfig);
00047
00048         if (stockIsEmpty)
00049         {
00050             _currentElement = elementConfig;
00051         }
00052
00053         _saveService.Save(GameSaveType.Stock);
00054     }
```

6.73.2.2 CreateStock()

```
void MergeIt.Game.Services.ElementsStockService.CreateStock ( ) [inline]
```

Implements [Mergelt.Core.Services.IElementsStockService](#).

Definition at line 30 of file [ElementsStockService.cs](#).

```
00031     {
00032         var stockData = new ElementsStockData();
00033         SetupStock(stockData);
00034
00035         _saveService.Save(GameSaveType.Stock);
00036     }
```

6.73.2.3 GetCurrent()

```
ElementConfig MergeIt.Game.Services.ElementsStockService.GetCurrent ( ) [inline]
```

Implements [Mergelt.Core.Services.IElementsStockService](#).

Definition at line 68 of file [ElementsStockService.cs](#).

```
00069     {
00070         return _currentElement;
00071     }
```

6.73.2.4 GetData()

```
IElementsStockData MergeIt.Game.Services.ElementsStockService.GetData () [inline]
```

Implements [Mergelt.Core.Services.IElementsStockService](#).

Definition at line 73 of file [ElementsStockService.cs](#).

```
00074     {
00075         var data = new ElementsStockData();
00076
00077         if (_serviceModel.Elements.Count > 0)
00078         {
00079             data.Elements = _serviceModel.Elements.Select(x => x.Id).ToArray();
00080         }
00081
00082         return data;
00083     }
```

6.73.2.5 GetNext()

```
ElementConfig MergeIt.Game.Services.ElementsStockService.GetNext () [inline]
```

Implements [Mergelt.Core.Services.IElementsStockService](#).

Definition at line 61 of file [ElementsStockService.cs](#).

```
00062     {
00063         _serviceModel.Elements.TryPeek(out ElementConfig config);
00064         _currentElement = config;
00065         return _currentElement;
00066     }
```

6.73.2.6 Remove()

```
bool MergeIt.Game.Services.ElementsStockService.Remove () [inline]
```

Implements [Mergelt.Core.Services.IElementsStockService](#).

Definition at line 56 of file [ElementsStockService.cs](#).

```
00057     {
00058         return _serviceModel.Elements.TryDequeue(out_);
00059     }
```

6.73.2.7 SetupStock()

```
void MergeIt.Game.Services.ElementsStockService.SetupStock (
    IElementsStockData stockData) [inline]
```

Implements [Mergelt.Core.Services.IElementsStockService](#).

Definition at line 38 of file [ElementsStockService.cs](#).

```
00039     {
00040         SetModel(stockData);
00041     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/ElementsStockService.cs

6.74 Mergelt.Game.Services.ElementsStockServiceModel Class Reference

Properties

- Queue<[ElementConfig](#)> [Elements](#) = new() [get]

6.74.1 Detailed Description

Definition at line 8 of file [ElementsStockServiceModel.cs](#).

6.74.2 Property Documentation

6.74.2.1 Elements

Queue<[ElementConfig](#)> MergeIt.Game.Services.ElementsStockServiceModel.Elements = new() [get]

Definition at line 10 of file [ElementsStockServiceModel.cs](#).

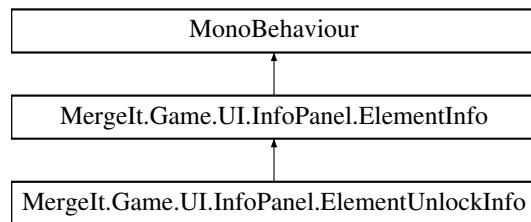
00010 { get; } = new();

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/ElementsStockServiceModel.cs

6.75 Mergelt.Game.UI.InfoPanel.ElementUnlockInfo Class Reference

Inheritance diagram for Mergelt.Game.UI.InfoPanel.ElementUnlockInfo:



Protected Member Functions

- override void [OnStart](#) ()
- override bool [OnTrySetup](#) ()
- override void [Destroy](#) ()

Protected Member Functions inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- bool [OnTrySetup \(\)](#)
- void [Clear \(\)](#)
- virtual void [OnClear \(\)](#)
- void [ActionButtonClick \(\)](#)
- virtual void [OnActionButtonClick \(\)](#)
- void [SendActionMessage \(\)](#)

Properties

- override Element ActionType [ActionType](#) [get]

Properties inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- Element ActionType [ActionType](#) [get]

Additional Inherited Members

Public Member Functions inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- void [TrySetup \(IFieldElement fieldElement\)](#)

Protected Attributes inherited from [Mergelt.Game.UI.InfoPanel.ElementInfo](#)

- IConfigsService [ConfigsService](#)
- IMessageBus [MessageBus](#)
- IFieldElement [SelectedElement](#)

6.75.1 Detailed Description

Definition at line 10 of file [ElementUnlockInfo.cs](#).

6.75.2 Member Function Documentation

6.75.2.1 [Destroy\(\)](#)

```
override void MergeIt.Game.UI.InfoPanel.ElementUnlockInfo.Destroy ( ) [inline], [protected], [virtual]
```

Reimplemented from [Mergelt.Game.UI.InfoPanel.ElementInfo](#).

Definition at line 51 of file [ElementUnlockInfo.cs](#).

```
00052      {
00053          base.Destroy ();
00054
00055          _unlockButton.onClick.RemoveListener(ActionButtonClick);
00056      }
```

6.75.2.2 OnStart()

```
override void MergeIt.Game.UI.InfoPanel.ElementUnlockInfo.OnStart () [inline], [protected], [virtual]
```

Reimplemented from [Mergelt.Game.UI.InfoPanel.ElementInfo](#).

Definition at line 26 of file [ElementUnlockInfo.cs](#).

```
00027     {
00028         base.OnStart ();
00029         _unlockButton.onClick.AddListener(ActionButtonClick);
00030     }
00031 }
```

6.75.2.3 OnTrySetup()

```
override bool MergeIt.Game.UI.InfoPanel.ElementUnlockInfo.OnTrySetup () [inline], [protected]
```

Definition at line 33 of file [ElementUnlockInfo.cs](#).

```
00034     {
00035         CurrencySettings unlockCostSettings =
00036             SelectedElement.ConfigParameters.ElementConfig.CommonSettings.UnlockCostSettings;
00037
00038         if (unlockCostSettings.Amount > 0 && SelectedElement.InfoParameters.IsBlocked)
00039         {
00040             gameObject.SetActive(true);
00041
00042             costTypeIcon.sprite = ConfigsService.GetCurrencyIcon(unlockCostSettings.Currency);
00043             costLabel.text = unlockCostSettings.Amount.ToString();
00044
00045             return true;
00046         }
00047
00048         return false;
00049     }
```

6.75.3 Property Documentation

6.75.3.1 ActionType

```
override ElementActionType MergeIt.Game.UI.InfoPanel.ElementUnlockInfo.ActionType [get]
```

Definition at line 21 of file [ElementUnlockInfo.cs](#).

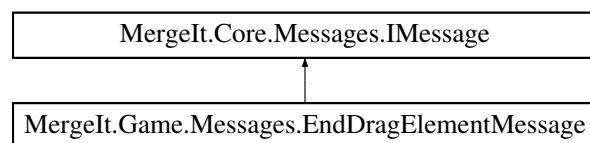
```
00022     {
00023         get => ElementActionType.Unlock;
00024     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/UI/InfoPanel/ElementUnlockInfo.cs

6.76 Mergelt.Game.Messages.EndDragElementMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.EndDragElementMessage:



Properties

- `GridPoint FromPoint [get, set]`
- `GameObject ToGameObject [get, set]`
- `Vector2 Position [get, set]`

6.76.1 Detailed Description

Definition at line 9 of file [EndDragElementMessage.cs](#).

6.76.2 Property Documentation

6.76.2.1 FromPoint

`GridPoint MergeIt.Game.Messages.EndDragElementMessage.FromPoint [get], [set]`

Definition at line 11 of file [EndDragElementMessage.cs](#).

```
00011 { get; set; }
```

6.76.2.2 Position

`Vector2 MergeIt.Game.Messages.EndDragElementMessage.Position [get], [set]`

Definition at line 13 of file [EndDragElementMessage.cs](#).

```
00013 { get; set; }
```

6.76.2.3 ToGameObject

`GameObject MergeIt.Game.Messages.EndDragElementMessage.ToGameObject [get], [set]`

Definition at line 12 of file [EndDragElementMessage.cs](#).

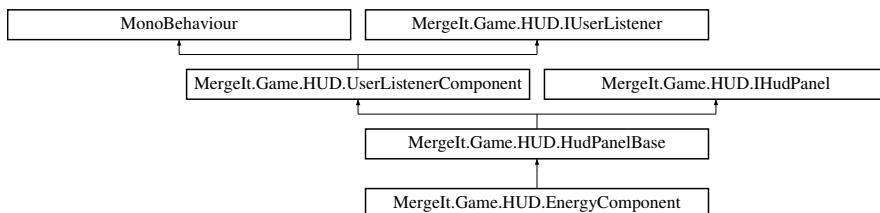
```
00012 { get; set; }
```

The documentation for this class was generated from the following file:

- `Assets/MergeIt/Source/Game/Messages/EndDragElementMessage.cs`

6.77 MergeIt.Game.HUD.EnergyComponent Class Reference

Inheritance diagram for MergeIt.Game.HUD.EnergyComponent:



Protected Member Functions

- override void [Start \(\)](#)
- override void [OnApplyModel \(UserServiceModel userServiceModel\)](#)

Protected Member Functions inherited from [Mergelt.Game.HUD.UserListenerComponent](#)

- void [OnApplyModel \(UserServiceModel userServiceModel\)](#)

Properties

- override HudPanelType [Type \[get\]](#)

Properties inherited from [Mergelt.Game.HUD.HudPanelBase](#)

- virtual HudPanelType [Type \[get\]](#)

Properties inherited from [Mergelt.Game.HUD.UserListenerComponent](#)

- [UserServiceModel UserServiceModel \[get\]](#)

Properties inherited from [Mergelt.Game.HUD.IHudPanel](#)

Additional Inherited Members

Public Member Functions inherited from [Mergelt.Game.HUD.UserListenerComponent](#)

- void [ApplyModel \(UserServiceModel userServiceModel\)](#)

6.77.1 Detailed Description

Definition at line 15 of file [EnergyComponent.cs](#).

6.77.2 Member Function Documentation

6.77.2.1 [OnApplyModel\(\)](#)

```
override void MergeIt.Game.HUD.EnergyComponent.OnApplyModel (
    UserServiceModel userServiceModel ) [inline], [protected]
```

Definition at line 73 of file [EnergyComponent.cs](#).

```
00074     {
00075         UserServiceModel.Energy.Subscribe(OnEnergyChanged, true);
00076     }
```

6.77.2.2 Start()

```
override void MergeIt.Game.HUD.EnergyComponent.Start ( ) [inline], [protected], [virtual]
```

Reimplemented from [Mergelt.Game.HUD.HudPanelBase](#).

Definition at line 32 of file [EnergyComponent.cs](#).

```
00033     {
00034         base.Start ();
00035
00036         _messageBus = DiContainer.Get<IMessageBus> ();
00037         _messageBus.AddListener<StartChargingMessage>(OnStartChargingMessageHandler);
00038     }
```

6.77.3 Property Documentation

6.77.3.1 Type

```
override HudPanelType MergeIt.Game.HUD.EnergyComponent.Type [get]
```

Implements [Mergelt.Game.HUD.IHudPanel](#).

Definition at line 23 of file [EnergyComponent.cs](#).

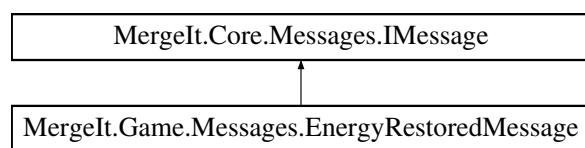
```
00024     {
00025         get => HudPanelType.Energy;
00026     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/HUD/EnergyComponent.cs

6.78 Mergelt.Game.Messages.EnergyRestoredMessage Class Reference

Inheritance diagram for [Mergelt.Game.Messages.EnergyRestoredMessage](#):



6.78.1 Detailed Description

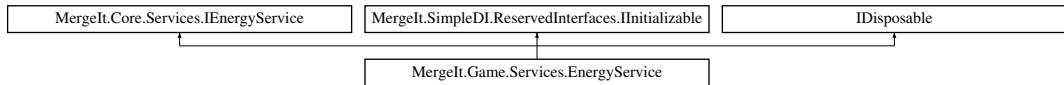
Definition at line 7 of file [EnergyRestoredMessage.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/EnergyRestoredMessage.cs

6.79 Mergelt.Game.Services.EnergyService Class Reference

Inheritance diagram for Mergelt.Game.Services.EnergyService:



Public Member Functions

- void [Dispose \(\)](#)
- void [Initialize \(\)](#)

6.79.1 Detailed Description

Definition at line 15 of file [EnergyService.cs](#).

6.79.2 Member Function Documentation

6.79.2.1 Dispose()

```
void MergeIt.Game.Services.EnergyService.Dispose ( ) [inline]
```

Definition at line 33 of file [EnergyService.cs](#).

```
00034     {
00035         _messageBus.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00036         _messageBus.RemoveListener<EnergyRestoredMessage>(OnEnergyRestoredMessageHandler);
00037         _messageBus.RemoveListener<SkipTimeMessage>(OnSkipTimeMessageHandler);
00038
00039         _userServiceModel.Energy.Unsubscribe(OnEnergyChanged);
00040     }
```

6.79.2.2 Initialize()

```
void MergeIt.Game.Services.EnergyService.Initialize ( ) [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 42 of file [EnergyService.cs](#).

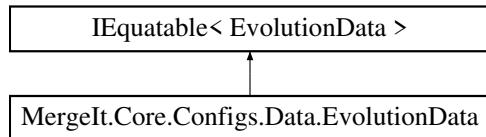
```
00043     {
00044         _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00045         _messageBus.AddListener<EnergyRestoredMessage>(OnEnergyRestoredMessageHandler);
00046         _messageBus.AddListener<SkipTimeMessage>(OnSkipTimeMessageHandler);
00047     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/EnergyService.cs

6.80 MergeIt.Core.Configs.Data.EvolutionData Class Reference

Inheritance diagram for MergeIt.Core.Configs.Data.EvolutionData:



Public Member Functions

- [EvolutionData](#) (string guid, string name, string description, bool discovered)
- bool [Equals](#) ([EvolutionData](#) other)
- override bool [Equals](#) (object obj)
- override int [GetHashCode](#) ()

Properties

- string [Name](#) [get]
- string [Description](#) [get]
- bool [Discovered](#) [get]
- List<[ElementConfig](#)> [Chain](#) [get]
- string [Id](#) [get]

6.80.1 Detailed Description

Definition at line 11 of file [EvolutionData.cs](#).

6.80.2 Constructor & Destructor Documentation

6.80.2.1 EvolutionData()

```
MergeIt.Core.Configs.Data.EvolutionData.EvolutionData (
    string guid,
    string name,
    string description,
    bool discovered ) [inline]
```

Definition at line 20 of file [EvolutionData.cs](#).

```
00021     {
00022         _guid = guid;
00023         _name = name;
00024         _description = description;
00025         _discovered = discovered;
00026         _chain = new List<ElementConfig>();
00027     }
```

6.80.3 Member Function Documentation

6.80.3.1 Equals() [1/2]

```
bool MergeIt.Core.Configs.Data.EvolutionData.Equals (
    EvolutionData other) [inline]
```

Definition at line 54 of file [EvolutionData.cs](#).

```
00055     {
00056         if (ReferenceEquals(null, other)) return false;
00057         if (ReferenceEquals(this, other)) return true;
00058         return _guid == other._guid;
00059     }
```

6.80.3.2 Equals() [2/2]

```
override bool MergeIt.Core.Configs.Data.EvolutionData.Equals (
    object obj) [inline]
```

Definition at line 61 of file [EvolutionData.cs](#).

```
00062     {
00063         if (ReferenceEquals(null, obj)) return false;
00064         if (ReferenceEquals(this, obj)) return true;
00065         if (obj.GetType() != this.GetType()) return false;
00066         return Equals((EvolutionData)obj);
00067     }
```

6.80.3.3 GetHashCode()

```
override int MergeIt.Core.Configs.Data.EvolutionData.GetHashCode () [inline]
```

Definition at line 69 of file [EvolutionData.cs](#).

```
00070     {
00071         return (_guid != null ? _guid.GetHashCode() : 0);
00072     }
```

6.80.4 Property Documentation

6.80.4.1 Chain

```
List<ElementConfig> MergeIt.Core.Configs.Data.EvolutionData.Chain [get]
```

Definition at line 44 of file [EvolutionData.cs](#).

```
00045     {
00046         get => _chain;
00047     }
```

6.80.4.2 Description

```
string MergeIt.Core.Configs.Data.EvolutionData.Description [get]
```

Definition at line 34 of file [EvolutionData.cs](#).

```
00035     {
00036         get => _description;
00037     }
```

6.80.4.3 Discovered

```
bool MergeIt.Core.Configs.Data.EvolutionData.Discovered [get]
```

Definition at line 39 of file [EvolutionData.cs](#).

```
00040      {
00041          get => _discovered;
00042      }
```

6.80.4.4 Id

```
string MergeIt.Core.Configs.Data.EvolutionData.Id [get]
```

Definition at line 49 of file [EvolutionData.cs](#).

```
00050      {
00051          get => _guid;
00052      }
```

6.80.4.5 Name

```
string MergeIt.Core.Configs.Data.EvolutionData.Name [get]
```

Definition at line 29 of file [EvolutionData.cs](#).

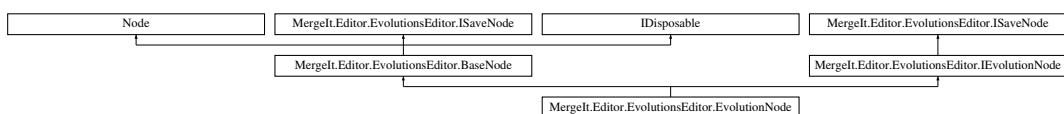
```
00030      {
00031          get => _name;
00032      }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Configs/Data/EvolutionData.cs

6.81 MergeIt.Editor.EvolutionsEditor.EvolutionNode Class Reference

Inheritance diagram for MergeIt.Editor.EvolutionsEditor.EvolutionNode:



Public Member Functions

- [EvolutionNode \(EvolutionNodeData data\)](#)
- override void [SaveData \(\)](#)
- override void [Dispose \(\)](#)

Public Member Functions inherited from [MergeIt.Editor.EvolutionsEditor.BaseNode](#)

- void [SaveData \(\)](#)
- Port [GetPort \(string portName\)](#)

Properties

- string `Name` [get]
- string `Description` [get]
- bool `Discovered` [get]
- override string `Id` [get]
- `IEvolutionNodeData Data` [get]

Properties inherited from [Mergelt.Editor.EvolutionsEditor.BaseNode](#)

- string `Id` [get]

Properties inherited from [Mergelt.Editor.EvolutionsEditor.ISaveNode](#)

Properties inherited from [Mergelt.Editor.EvolutionsEditor.IEvolutionNode](#)

6.81.1 Detailed Description

Definition at line 11 of file [EvolutionNode.cs](#).

6.81.2 Constructor & Destructor Documentation

6.81.2.1 EvolutionNode()

```
MergeIt.Editor.EvolutionsEditor.EvolutionNode.EvolutionNode (
    EvolutionNodeData data ) [inline]
```

Definition at line 29 of file [EvolutionNode.cs](#).

```
00030     {
00031         Data = data;
00032
00033         var visualTreeAsset =
00034     (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.EvolutionsEditorResourcesRoot,
00035     "EvolutionNode.uxml"));
00036
00036         mainContainer.AddToClassList("nodeBorderEvolution");
00037
00038         var root = visualTreeAsset.CloneTree();
00039         _nameField = root.Q<TextField>("Name");
00040         _descField = root.Q<TextField>("Description");
00041         _discoveredToggle = root.Q<Toggle>("Discovered");
00042
00043         title = Data.Name;
00044         _nameField.SetValueWithoutNotify(Data.Name);
00045         _descField.SetValueWithoutNotify(Data.Description);
00046         _discoveredToggle.SetValueWithoutNotify(Data.Discovered);
00047
00048         _nameField.RegisterValueChangedCallback(OnEvoNameTextChanged);
00049         _descField.RegisterValueChangedCallback(OnEvoDescriptionTextChanged);
00050         _discoveredToggle.RegisterValueChangedCallback(OnDiscoveredSelected);
00051
00052         mainContainer.Add(root);
00053
00054         topContainer.Remove(inputContainer);
00055         _evolutionRootPort = InstantiatePort(Orientation.Horizontal, Direction.Output,
00055         Port.Capacity.Single, typeof(string));
00056         _evolutionRootPort.portName = Constants.EvolutionRootPort;
00057         outputContainer.Add(_evolutionRootPort);
00058     }
```

6.81.3 Member Function Documentation

6.81.3.1 Dispose()

```
override void MergeIt.Editor.EvolutionsEditor.EvolutionNode.Dispose () [inline], [virtual]
```

Reimplemented from [MergeIt.Editor.EvolutionsEditor.BaseNode](#).

Definition at line 82 of file [EvolutionNode.cs](#).

```
00083     {
00084         base.Dispose ();
00085
00086         _nameField.UnregisterValueChangedCallback (OnEvoNameTextChanged);
00087         _descField.UnregisterValueChangedCallback (OnEvoDescriptionTextChanged);
00088         _discoveredToggle.UnregisterValueChangedCallback (OnDiscoveredSelected);
00089     }
```

6.81.3.2 SaveData()

```
override void MergeIt.Editor.EvolutionsEditor.EvolutionNode.SaveData () [inline]
```

Implements [MergeIt.Editor.EvolutionsEditor.ISaveNode](#).

Definition at line 77 of file [EvolutionNode.cs](#).

```
00078     {
00079         Data.Position = GetPosition ();
00080     }
```

6.81.4 Property Documentation

6.81.4.1 Data

```
IEvolutionNodeData MergeIt.Editor.EvolutionsEditor.EvolutionNode.Data [get]
```

Implements [MergeIt.Editor.EvolutionsEditor.IEvolutionNode](#).

Definition at line 22 of file [EvolutionNode.cs](#).

```
00022 { get; }
```

6.81.4.2 Description

```
string MergeIt.Editor.EvolutionsEditor.EvolutionNode.Description [get]
```

Implements [MergeIt.Editor.EvolutionsEditor.IEvolutionNode](#).

Definition at line 14 of file [EvolutionNode.cs](#).

```
00014 { get; }
```

6.81.4.3 Discovered

```
bool MergeIt.Editor.EvolutionsEditor.EvolutionNode.Discovered [get]
```

Implements [MergeIt.Editor.EvolutionsEditor.IEvolutionNode](#).

Definition at line 15 of file [EvolutionNode.cs](#).

```
00015 { get; }
```

6.81.4.4 Id

```
override string MergeIt.Editor.EvolutionsEditor.EvolutionNode.Id [get]
```

Implements [Mergelt.Editor.EvolutionsEditor.ISaveNode](#).

Definition at line 17 of file [EvolutionNode.cs](#).

```
00018     {
00019         get => Data?.Id;
00020     }
```

6.81.4.5 Name

```
string MergeIt.Editor.EvolutionsEditor.EvolutionNode.Name [get]
```

Implements [Mergelt.Editor.EvolutionsEditor.IEvolutionNode](#).

Definition at line 13 of file [EvolutionNode.cs](#).

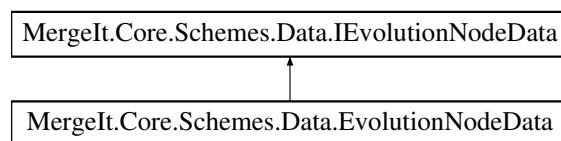
```
00013 { get; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/EvolutionsEditor/ItemsNodes/EvolutionNode.cs

6.82 Mergelt.Core.Schemes.Data.EvolutionNodeData Class Reference

Inheritance diagram for Mergelt.Core.Schemes.Data.EvolutionNodeData:



Public Member Functions

- T [Copy< T >\(\)](#)

Properties

- string [Id](#) [get, set]
- bool [Discovered](#) [get, set]
- string [Name](#) [get, set]
- string [Description](#) [get, set]
- Rect [Position](#) [get, set]

Properties inherited from [Mergelt.Core.Schemes.Data.IEvolutionNodeData](#)

6.82.1 Detailed Description

Definition at line 9 of file [EvolutionNodeData.cs](#).

6.82.2 Member Function Documentation

6.82.2.1 `Copy< T >()`

```
T MergeIt.Core.Schemes.Data.EvolutionNodeData.Copy< T > ( ) [inline]
```

Implements [Mergelt.Core.Schemes.Data.IEvolutionNodeData](#).

Type Constraints

T : IEvolutionNodeData

T : new()

Definition at line 55 of file [EvolutionNodeData.cs](#).

```
00055                               : IEvolutionNodeData, new()
00056     {
00057         var copiedData = new T
00058         {
00059             Position = Position,
00060             Discovered = Discovered,
00061             Name = Name,
00062             Description = Description
00063         };
00064
00065         return copiedData;
00066     }
```

6.82.3 Property Documentation

6.82.3.1 Description

```
string MergeIt.Core.Schemes.Data.EvolutionNodeData.Description [get], [set]
```

Implements [Mergelt.Core.Schemes.Data.IEvolutionNodeData](#).

Definition at line 43 of file [EvolutionNodeData.cs](#).

```
00044     {
00045         get => _description;
00046         set => _description = value;
00047     }
```

6.82.3.2 Discovered

```
bool MergeIt.Core.Schemes.Data.EvolutionNodeData.Discovered [get], [set]
```

Implements [Mergelt.Core.Schemes.Data.IEvolutionNodeData](#).

Definition at line 31 of file [EvolutionNodeData.cs](#).

```
00032     {
00033         get => _discovered;
00034         set => _discovered = value;
00035     }
```

6.82.3.3 Id

```
string MergeIt.Core.Schemes.Data.EvolutionNodeData.Id [get], [set]
```

Implements [Mergelt.Core.Schemes.Data.IEvolutionNodeData](#).

Definition at line 17 of file [EvolutionNodeData.cs](#).

```
00018     {
00019         get
00020     {
00021         if (string.IsNullOrEmpty(_id))
00022         {
00023             _id = Guid.NewGuid().ToString();
00024         }
00025
00026         return _id;
00027     }
00028     set => _id = value;
00029 }
```

6.82.3.4 Name

```
string MergeIt.Core.Schemes.Data.EvolutionNodeData.Name [get], [set]
```

Implements [Mergelt.Core.Schemes.Data.IEvolutionNodeData](#).

Definition at line 37 of file [EvolutionNodeData.cs](#).

```
00038     {
00039         get => _name;
00040         set => _name = value;
00041     }
```

6.82.3.5 Position

```
Rect MergeIt.Core.Schemes.Data.EvolutionNodeData.Position [get], [set]
```

Implements [Mergelt.Core.Schemes.Data.IEvolutionNodeData](#).

Definition at line 49 of file [EvolutionNodeData.cs](#).

```
00050     {
00051         get => _position;
00052         set => _position = value;
00053     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Schemes/Data/EvolutionNodeData.cs

6.83 Mergelt.Core.Evolutions.EvolutionProgressData Class Reference

Properties

- string [EvolutionId](#) [get, set]
- int [Progress](#) [get, set]

6.83.1 Detailed Description

Definition at line 9 of file [EvolutionProgressData.cs](#).

6.83.2 Property Documentation

6.83.2.1 EvolutionId

```
string MergeIt.Core.Evolutions.EvolutionProgressData.EvolutionId [get], [set]
```

Definition at line 12 of file [EvolutionProgressData.cs](#).
00012 { get; set; }

6.83.2.2 Progress

```
int MergeIt.Core.Evolutions.EvolutionProgressData.Progress [get], [set]
```

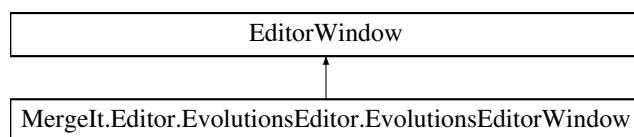
Definition at line 15 of file [EvolutionProgressData.cs](#).
00015 { get; set; }

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Evolutions/EvolutionProgressData.cs

6.84 MergeIt.Editor.EvolutionsEditor.EvolutionsEditorWindow Class Reference

Inheritance diagram for MergeIt.Editor.EvolutionsEditor.EvolutionsEditorWindow:



Public Member Functions

- void [Initialize \(\)](#)
- void [OnEnable \(\)](#)
- void [OnDisable \(\)](#)
- void [OnFocus \(\)](#)
- void [OnDestroy \(\)](#)
- void [Reload \(SchemeObject scheme\)](#)

Static Public Member Functions

- static void [ShowWindow \(\)](#)

Properties

- bool `Initialized` [get]

6.84.1 Detailed Description

Definition at line 14 of file [EvolutionsEditorWindow.cs](#).

6.84.2 Member Function Documentation

6.84.2.1 Initialize()

```
void MergeIt.Editor.EvolutionsEditor.EvolutionsEditorWindow.Initialize () [inline]
```

Definition at line 41 of file [EvolutionsEditorWindow.cs](#).

```
00042     {
00043         if (!initialized)
00044         {
00045             _windowStyle =
00046             (StyleSheet)EditorGUIUtility.Load(Path.Combine(Constants.EvolutionsEditorResourcesRoot,
00047                                         "EvolutionsEditorStyle.uss"));
00048             _windowTree =
00049             (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.EvolutionsEditorResourcesRoot,
00050                                         "EvolutionsEditorWindow.uxml"));
00051
00052             VisualElement root = rootVisualElement;
00053             VisualElement windowRoot = _windowTree.CloneTree();
00054
00055             windowRoot.AddToClassList("root");
00056
00057             var toolbarMenu = windowRoot.Q<ToolbarMenu>("FileMenu");
00058             toolbarMenu.menu.AppendAction("Open...", _ => OnLoadButtonClicked());
00059             toolbarMenu.menu.AppendSeparator();
00060             toolbarMenu.menu.AppendAction("Save", _ => OnSaveButtonClicked(), _ =>
00061             CanSaveAndExport());
00062             toolbarMenu.menu.AppendAction("Save As...", _ => OnSaveAsButtonClicked());
00063
00064             _graphView = windowRoot.Q<EvolutionsGraphView>("EvoGraph");
00065             _graphView.AddManipulator(new ContentDragger());
00066             _graphView.AddManipulator(new SelectionDragger());
00067             _graphView.AddManipulator(new RectangleSelector());
00068             _graphView.AddManipulator(new ClickSelector());
00069
00070             _graphView.SetupZoom(0.1f, 3f);
00071
00072             root.Add(windowRoot);
00073             root.styleSheets.Add(_windowStyle);
00074
00075             initialized = true;
00076         }
00077     }
```

6.84.2.2 OnDestroy()

```
void MergeIt.Editor.EvolutionsEditor.EvolutionsEditorWindow.OnDestroy () [inline]
```

Definition at line 93 of file [EvolutionsEditorWindow.cs](#).

```
00094     {
00095         _graphView.ClearNodes ();
00096     }
```

6.84.2.3 OnDisable()

```
void MergeIt.Editor.EvolutionsEditor.EvolutionsEditorWindow.OnDisable () [inline]
```

Definition at line 83 of file [EvolutionsEditorWindow.cs](#).

```
00084     {
00085         _initialized = false;
00086     }
```

6.84.2.4 OnEnable()

```
void MergeIt.Editor.EvolutionsEditor.EvolutionsEditorWindow.OnEnable () [inline]
```

Definition at line 78 of file [EvolutionsEditorWindow.cs](#).

```
00079     {
00080         Initialize();
00081     }
```

6.84.2.5 OnFocus()

```
void MergeIt.Editor.EvolutionsEditor.EvolutionsEditorWindow.OnFocus () [inline]
```

Definition at line 88 of file [EvolutionsEditorWindow.cs](#).

```
00089     {
00090         _graphView?.Validate();
00091     }
```

6.84.2.6 Reload()

```
void MergeIt.Editor.EvolutionsEditor.EvolutionsEditorWindow.Reload (
    SchemeObject scheme) [inline]
```

Definition at line 98 of file [EvolutionsEditorWindow.cs](#).

```
00099     {
00100         _currentScheme = scheme;
00101         SchemeData data = scheme.SchemeData;
00102         _graphView.Restore(data);
00103         _graphView.ClearSelection();
00104     }
```

6.84.2.7 ShowWindow()

```
static void MergeIt.Editor.EvolutionsEditor.EvolutionsEditorWindow.ShowWindow () [inline],
[static]
```

Definition at line 29 of file [EvolutionsEditorWindow.cs](#).

```
00030     {
00031         var window = GetWindow<EvolutionsEditorWindow>(true, "Evolutions Editor");
00032         if (!window.Initialized)
00033         {
00034             window.Initialize();
00035         }
00036         window.Show();
00037     }
```

6.84.3 Property Documentation

6.84.3.1 Initialized

```
bool MergeIt.Editor.EvolutionsEditor.EvolutionsEditorWindow.Initialized [get]
```

Definition at line 23 of file [EvolutionsEditorWindow.cs](#).

```
00024     {  
00025         get => _initialized;  
00026     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/EvolutionsEditor/EvolutionsEditorWindow.cs

6.85 Mergelt.Editor.LevelEditor.EvolutionSelection Class Reference

Static Public Member Functions

- static void [Init](#) (List< [EvolutionData](#) > evolutions)
- static void [SelectCell](#) (int row, int column, [LevelElementData](#) itemData=null)
- static void [LockCell](#) (bool isLocked)
- static void [UpdateCell](#) ([LevelElementData](#) itemData, [GridPoint](#) newPosition=default, bool copyPosition=true)
- static void [SelectEvolution](#) ([EvolutionData](#) evolutionData)
- static void [SelectElement](#) ([ElementConfig](#) evolutionElement)
- static void [ClearSelection](#) (bool force=false)

Properties

- static [EvolutionData](#) [Evolution](#) [get]
- static [ElementConfig](#) [EvolItem](#) [get]
- static [LevelElementData](#) [ElementData](#) [get]

6.85.1 Detailed Description

Definition at line 12 of file [EvolutionSelection.cs](#).

6.85.2 Member Function Documentation

6.85.2.1 ClearSelection()

```
static void MergeIt.Editor.LevelEditor.EvolutionSelection.ClearSelection (  
    bool force = false) [inline], [static]
```

Definition at line 87 of file [EvolutionSelection.cs](#).

```
00088     {  
00089         if (force)  
00090         {  
00091             _elementData = null;  
00092         }  
00093     }
```

6.85.2.2 Init()

```
static void MergeIt.Editor.LevelEditor.EvolutionSelection.Init (
    List< EvolutionData > evolutions ) [inline], [static]
```

Definition at line 22 of file EvolutionSelection.cs.

```
00023     {
00024         _evolutions = evolutions;
00025
00026         var firstEvo = _evolutions.First();
00027         var firstEvoItem = firstEvo.Chain.First();
00028
00029         _evolution = firstEvo;
00030
00031         _elementData = new LevelElementData
00032         {
00033             EvolutionId = firstEvo.Id,
00034             Element = firstEvoItem,
00035         };
00036     }
```

6.85.2.3 LockCell()

```
static void MergeIt.Editor.LevelEditor.EvolutionSelection.LockCell (
    bool isLocked ) [inline], [static]
```

Definition at line 43 of file EvolutionSelection.cs.

```
00044     {
00045         _elementData.IsBlocked = isLocked;
00046     }
```

6.85.2.4 SelectCell()

```
static void MergeIt.Editor.LevelEditor.EvolutionSelection.SelectCell (
    int row,
    int column,
    LevelElementData itemData = null ) [inline], [static]
```

Definition at line 38 of file EvolutionSelection.cs.

```
00039     {
00040         UpdateCell(itemData, GridPoint.Create(row, column));
00041     }
```

6.85.2.5 SelectElement()

```
static void MergeIt.Editor.LevelEditor.EvolutionSelection.SelectElement (
    ElementConfig evolutionElement ) [inline], [static]
```

Definition at line 73 of file EvolutionSelection.cs.

```
00074     {
00075         var evolution = Evolution;
00076
00077         if (evolution.Chain.Contains(evolutionElement))
00078         {
00079             _elementData.Element = evolutionElement;
00080         }
00081         else
00082         {
00083             Debug.LogWarning($"Evolution doesn't contain item with id: ({evolutionElement})!");
00084         }
00085     }
```

6.85.2.6 SelectEvolution()

```
static void MergeIt.Editor.LevelEditor.EvolutionSelection.SelectEvolution (
    EvolutionData evolutionData) [inline], [static]
```

Definition at line 61 of file [EvolutionSelection.cs](#).

```
00062     {
00063         _evolution = evolutionData;
00064         _elementData.EvolutionId = _evolution.Id;
00065         if (!_evolution.Chain.Contains(_elementData.Element))
00066         {
00067             _elementData.Element = _evolution.Chain.FirstOrDefault();
00068         }
00069     }
00070 }
```

6.85.2.7 UpdateCell()

```
static void MergeIt.Editor.LevelEditor.EvolutionSelection.UpdateCell (
    LevelElementData itemData,
    GridPoint newPosition = default,
    bool copyPosition = true) [inline], [static]
```

Definition at line 48 of file [EvolutionSelection.cs](#).

```
00049     {
00050         if (itemData != null)
00051         {
00052             _elementData.CopyFrom(itemData, copyPosition);
00053             _evolution = _evolutions.FirstOrDefault(x => x.Id == itemData.EvolutionId);
00054         }
00055         else
00056         {
00057             _elementData.Position = newPosition;
00058         }
00059 }
```

6.85.3 Property Documentation

6.85.3.1 ElementData

[LevelElementData](#) MergeIt.Editor.LevelEditor.EvolutionSelection.ElementData [static], [get]

Definition at line 16 of file [EvolutionSelection.cs](#).

6.85.3.2 Evoltem

[ElementConfig](#) MergeIt.Editor.LevelEditor.EvolutionSelection.EvoItem [static], [get]

Definition at line 15 of file [EvolutionSelection.cs](#).

6.85.3.3 Evolution

[EvolutionData](#) MergeIt.Editor.LevelEditor.EvolutionSelection.Evolution [static], [get]

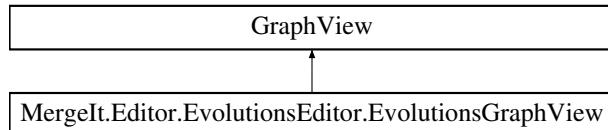
Definition at line 14 of file [EvolutionSelection.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/LevelEditor/EvolutionSelection.cs

6.86 MergIt.Editor.EvolutionsEditor.EvolutionsGraphView Class Reference

Inheritance diagram for MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView:



Classes

- class [UxmlFactory](#)

Public Member Functions

- override void [BuildContextualMenu](#) (ContextMenuPopulateEvent evt)
- void [Validate](#) ()
- override List< Port > [GetCompatiblePorts](#) (Port startPort, NodeAdapter nodeAdapter)
- [SchemeData Save](#) ()
- List< EvolutionData > [BakeEvolutionsData](#) ()
- void [Restore](#) (SchemeData savedGraph)
- override void [HandleEvent](#) (EventBase evt)
- void [ClearNodes](#) ()

Properties

- override bool [canCopySelection](#) [get]
- override bool [canDuplicateSelection](#) [get]

6.86.1 Detailed Description

Definition at line 17 of file [EvolutionsGraphView.cs](#).

6.86.2 Constructor & Destructor Documentation

6.86.2.1 EvolutionsGraphView()

`MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView.EvolutionsGraphView ()` [inline]

Definition at line 47 of file [EvolutionsGraphView.cs](#).

```

00048     {
00049         serializeGraphElements = OnSerializeGraphElements;
00050         canPasteSerializedData = OnCanPasteSerializedData;
00051         unserializeAndPaste = OnUnserializeAndPaste;
00052         deleteSelection = OnDeleteSelection;
00053     }
  
```

6.86.3 Member Function Documentation

6.86.3.1 BakeEvolutionsData()

```
List< EvolutionData > MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView.BakeEvolutionsData (
) [inline]
```

Definition at line 217 of file [EvolutionsGraphView.cs](#).

```
00218     {
00219         var allElements = graphElements.ToList();
00220         var evolutionsNodes = allElements.OfType<IEvolutionNode>();
00221         _evolutionsData = new List<EvolutionData>();
00222
00223         foreach (IEvolutionNode itemNode in evolutionsNodes)
00224         {
00225             Port startPort = itemNode.GetPort(Constants.EvolutionRootPort);
00226             if (startPort.connected)
00227             {
00228                 var evoData = new EvolutionData(itemNode.Data.Id,
00229                     itemNode.Data.Name,
00230                     itemNode.Data.Description,
00231                     itemNode.Data.Discovered);
00232
00233                 IElementNode firstNode = startPort.connections.First().input.node as IElementNode;
00234                 FillChain(evoData.Chain, firstNode);
00235
00236                 _evolutionsData.Add(evoData);
00237             }
00238         }
00239
00240         ExtractConfigs();
00241
00242         return _evolutionsData;
00243     }
```

6.86.3.2 BuildContextualMenu()

```
override void MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView.BuildContextualMenu (
    ContextualMenuPopulateEvent evt) [inline]
```

Definition at line 55 of file [EvolutionsGraphView.cs](#).

```
00056     {
00057         if (evt.target is GraphView)
00058         {
00059             evt.menu.AppendSeparator();
00060             evt.menu.AppendAction("Paste", _ => PasteCallback(),
00061             _ => canPaste ? DropdownMenuAction.Status.Normal :
00062             DropdownMenuAction.Status.Disabled);
00062             evt.menu.AppendAction("Create evolution root", CreateEvolutionRoot);
00063         }
00064
00065         if (evt.target is ISaveNode)
00066         {
00067             evt.menu.AppendSeparator();
00068             evt.menu.AppendAction("Copy", _ => CopySelectionCallback(),
00069             _ => canCopySelection ? DropdownMenuAction.Status.Normal :
00070             DropdownMenuAction.Status.Disabled);
00070             evt.menu.AppendAction("Duplicate", a => DuplicateSelectionCallback(),
00071             a =>
00072                 canDuplicateSelection ? DropdownMenuAction.Status.Normal :
00072                 DropdownMenuAction.Status.Disabled);
00073             evt.menu.AppendAction("Delete", _ => DeleteSelectionCallback(AskUser.DontAskUser),
00074             _ => canDeleteSelection ? DropdownMenuAction.Status.Normal :
00074                 DropdownMenuAction.Status.Disabled);
00075             evt.menu.AppendSeparator();
00076         }
00077     }
```

6.86.3.3 ClearNodes()

```
void MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView.ClearNodes ( ) [inline]
```

Definition at line 535 of file [EvolutionsGraphView.cs](#).

```
00536     {
00537         foreach (Node node in nodes)
00538         {
00539             if (node is BaseNode saveNode)
00540             {
00541                 saveNode.Dispose();
00542             }
00543         }
00544     }
```

6.86.3.4 GetCompatiblePorts()

```
override List< Port > MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView.GetCompatiblePorts (
    Port startPort,
    NodeAdapter nodeAdapter ) [inline]
```

Definition at line 114 of file [EvolutionsGraphView.cs](#).

```
00115     {
00116         var compatibleAnchors = new List<Port>();
00117
00118         foreach (Port candidateAnchor in ports.ToList())
00119         {
00120             if (startPort == candidateAnchor ||
00121                 startPort.node == candidateAnchor.node ||
00122                 startPort.direction == candidateAnchor.direction ||
00123                 startPort.portType != candidateAnchor.portType)
00124             {
00125                 continue;
00126             }
00127
00128             compatibleAnchors.Add(candidateAnchor);
00129         }
00130
00131         return compatibleAnchors;
00132     }
```

6.86.3.5 HandleEvent()

```
override void MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView.HandleEvent (
    EventBase evt ) [inline]
```

Definition at line 473 of file [EvolutionsGraphView.cs](#).

```
00474     {
00475         base.HandleEvent(evt);
00476
00477         if (_currentEvent is DragPerformEvent)
00478         {
00479             _currentEvent = evt;
00480             return;
00481         }
00482
00483         _currentEvent = evt;
00484
00485         switch (evt)
00486         {
00487             case DragUpdatedEvent _:
00488                 var dragDrop = DragAndDrop.objectReferences;
00489                 if (dragDrop.All(x => x is ElementConfig))
00490                 {
00491                     DragAndDrop.visualMode = DragAndDropVisualMode.Copy;
00492                 }
00493
00494                 break;
00495
00496             case DragPerformEvent dragPerformEvent:
00497                 var type = dragPerformEvent.target.GetType();
```

```

00499     var typeName = type.Name;
00500
00501     if (typeName == "ObjectFieldDisplay")
00502     {
00503         return;
00504     }
00505
00506     var mousePosition = GetLocalMousePosition(dragPerformEvent.local.mousePosition);
00507
00508     var rect = new Rect(mousePosition.x, mousePosition.y, 100, 100);
00509     var dragDropObjects = DragAndDrop.objectReferences;
00510     if (dragDropObjects.All(x => x is ElementConfig))
00511     {
00512         List<ElementNodeData> elementNodesData =
00513             dragDropObjects.Where(x => x is ElementConfig).Select(x =>
00514             {
00515                 var config = x as ElementConfig;
00516                 var nodeData = new ElementNodeData
00517                 {
00518                     ElementConfig = config,
00519                     Position = rect
00520                 };
00521
00522                 return nodeData;
00523             }).ToList();
00524
00525         if (elementNodesData.Count > 0)
00526         {
00527             CreateNodes(elementNodesData, Constants.NodeCreationOffset);
00528         }
00529     }
00530
00531     break;
00532 }
00533 }
```

6.86.3.6 Restore()

```
void MergeIT.Editor.EvolutionsEditor.EvolutionsGraphView.Restore (
    SchemeData savedGraph ) [inline]
```

Definition at line 258 of file [EvolutionsGraphView.cs](#).

```

00259     {
00260         _evolutionsData = savedGraph.EvolutionsData;
00261
00262         ExtractConfigs();
00263
00264         DeleteElements(nodes.ToList(), edges.ToList());
00265
00266         List<ElementNodeData> elementsNodesData = savedGraph.ElementsNodesData;
00267         List<EvolutionNodeData> evolutionNodesData = savedGraph.EvolutionsNodesData;
00268         List<EdgeData> edgesData = savedGraph.Edges;
00269
00270         if (elementsNodesData != null)
00271         {
00272             CreateNodes(elementsNodesData);
00273         }
00274
00275         if (evolutionNodesData != null)
00276         {
00277             CreateEvolutionsNodes(evolutionNodesData, CreateEvolutionNode);
00278         }
00279
00280         if (edgesData != null)
00281         {
00282             var allNodes = nodes.ToList().OfType<ISaveNode>();
00283
00284             foreach (EdgeData edgeData in edgesData)
00285             {
00286                 ISaveNode inputNode = null;
00287                 ISaveNode outputNode = null;
00288                 foreach (ISaveNode itemNode in allNodes)
00289                 {
00290                     if (itemNode.Id == edgeData.InputNodeID)
00291                     {
00292                         inputNode = itemNode;
00293                     }
00294
00295                     if (itemNode.Id == edgeData.OutputNodeID)
00296                     {
00297                         outputNode = itemNode;
00298                     }
00299
00300                     if (inputNode != null && outputNode != null)
00301                     {
00302                         edgeData.InputNode = inputNode;
00303                         edgeData.OutputNode = outputNode;
00304                     }
00305                 }
00306             }
00307         }
00308     }
00309 }
```

```

00298             }
00299
00300         if (outputNode != null && inputNode != null)
00301     {
00302         break;
00303     }
00304
00305
00306     if (outputNode != null && inputNode != null)
00307     {
00308         var outputPort = outputNode.GetPort(edgeData.OutputPortName);
00309         var inputPort = inputNode.GetPort(edgeData.InputPortName);
00310
00311         if (outputPort != null && inputPort != null)
00312     {
00313             Edge newEdge = inputPort.ConnectTo(outputPort);
00314             inputPort.Connect(newEdge);
00315             outputPort.Connect(newEdge);
00316
00317             AddElement(newEdge);
00318         }
00319     }
00320 }
00321 }
00322 }
```

6.86.3.7 Save()

SchemeData MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView.Save () [inline]

Definition at line 134 of file [EvolutionsGraphView.cs](#).

```

00135     {
00136         var allElements = graphElements.ToList();
00137
00138         var allEdges = allElements.OfType<Edge>();
00139         var allNodes = allElements.OfType<ISaveNode>();
00140
00141         var elementsToSend = new List<ElementNodeData>();
00142         var evolutionsToSend = new List<EvolutionNodeData>();
00143         var edgesToSend = new List<EdgeData>();
00144
00145         var brokenNodes = new List<ElementNode>();
00146
00147         foreach (ISaveNode saveNode in allNodes)
00148     {
00149         if (saveNode is ElementNode elementNode)
00150     {
00151         if (!elementNode.Validate())
00152         {
00153             brokenNodes.Add(elementNode);
00154         }
00155     }
00156 }
00157
00158         if (brokenNodes.Count > 0)
00159     {
00160         ClearSelection();
00161
00162         foreach (ElementNode brokenNode in brokenNodes)
00163     {
00164             AddToSelection(brokenNode);
00165         }
00166
00167         EditorWindow.focusedWindow.ShowNotification(new GUIContent($"Can't save scheme with
broken node. Broken nodes was selected."));
00168         return null;
00169     }
00170
00171         foreach (ISaveNode saveNode in allNodes)
00172     {
00173         saveNode.SaveData();
00174         switch (saveNode)
00175     {
00176             case IElementNode elementNode:
00177                 elementsToSend.Add(elementNode.Data as ElementNodeData);
00178                 break;
00179
00180             case IEvolutionNode evolutionNode:
00181                 evolutionsToSend.Add(evolutionNode.Data as EvolutionNodeData);
00182                 break;
00183         }
00184     }
```

```

00185
00186     foreach (Edge edge in allEdges)
00187     {
00188         var edgeData = new EdgeData();
00189         edgeData.InputPortName = edge.input.portName;
00190         edgeData.OutputPortName = edge.output.portName;
00191
00192         if (edge.input.node is ISaveNode saveNode)
00193         {
00194             edgeData.InputNodeID = saveNode.Id;
00195         }
00196
00197         if (edge.output.node is ISaveNode outputNode)
00198         {
00199             edgeData.OutputNodeID = outputNode.Id;
00200         }
00201
00202         edgesToSave.Add(edgeData);
00203     }
00204
00205     List<EvolutionData> evolutionsData = BakeEvolutionsData();
00206     var saveData = new SchemeData
00207     {
00208         ElementsNodesData = elementsToSave,
00209         EvolutionsNodesData = evolutionsToSave,
00210         Edges = edgesToSave,
00211         EvolutionsData = evolutionsData
00212     };
00213
00214     return saveData;
00215 }
```

6.86.3.8 Validate()

```
void MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView.Validate () [inline]
```

Definition at line 79 of file [EvolutionsGraphView.cs](#).

```

00080     {
00081         nodes.ForEach(x =>
00082         {
00083             if (x is IELEMENTnode elementNode)
00084             {
00085                 elementNode.Validate();
00086             }
00087         });
00088     }
```

6.86.4 Property Documentation

6.86.4.1 canCopySelection

```
override bool MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView.canCopySelection [get],  
[protected]
```

Definition at line 23 of file [EvolutionsGraphView.cs](#).

```

00024     {
00025         get => selection.TrueForAll(x => x is IEvolutionNode);
00026     }
```

6.86.4.2 canDuplicateSelection

```
override bool MergeIt.Editor.EvolutionsEditor.EvolutionsGraphView.canDuplicateSelection [get],  
[protected]
```

Definition at line 28 of file [EvolutionsGraphView.cs](#).

```

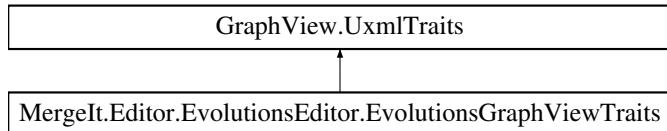
00029     {
00030         get => selection.TrueForAll(x => x is IEvolutionNode);
00031     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/EvolutionsEditor/Components/EvolutionsGraphView.cs

6.87 Mergelt.Editor.EvolutionsEditor.EvolutionsGraphViewTraits Class Reference

Inheritance diagram for Mergelt.Editor.EvolutionsEditor.EvolutionsGraphViewTraits:



Public Member Functions

- override void [Init](#) (VisualElement ve, IUxmlAttributes bag, CreationContext cc)

6.87.1 Detailed Description

Definition at line 8 of file [EvolutionsGraphViewTraits.cs](#).

6.87.2 Member Function Documentation

6.87.2.1 Init()

```
override void MergeIt.Editor.EvolutionsEditor.EvolutionsGraphViewTraits.Init (
    VisualElement ve,
    IUxmlAttributes bag,
    CreationContext cc ) [inline]
```

Definition at line 13 of file [EvolutionsGraphViewTraits.cs](#).

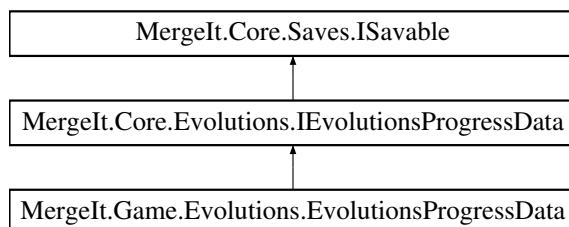
```
00014     {
00015         base.Init(ve, bag, cc);
00016
00017         if (ve is EvolutionsGraphView evolutionsGraphView)
00018         {
00019             evolutionsGraphView.name = _nameAttribute.GetValueFromBag(bag, cc);
00020         }
00021     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/EvolutionsEditor/Components/EvolutionsGraphViewTraits.cs

6.88 Mergelt.Game.Evolutions.EvolutionsProgressData Class Reference

Inheritance diagram for Mergelt.Game.Evolutions.EvolutionsProgressData:



Properties

- List< EvolutionProgressData > EvolutionsProgress = new() [get]

Properties inherited from [Mergelt.Core.Evolutions.IEvolutionsProgressData](#)

6.88.1 Detailed Description

Definition at line 12 of file [EvolutionsProgressData.cs](#).

6.88.2 Property Documentation

6.88.2.1 EvolutionsProgress

```
List<EvolutionProgressData> MergeIt.Game.Evolutions.EvolutionsProgressData.EvolutionsProgress
= new() [get]
```

Implements [Mergelt.Core.Evolutions.IEvolutionsProgressData](#).

Definition at line 15 of file [EvolutionsProgressData.cs](#).

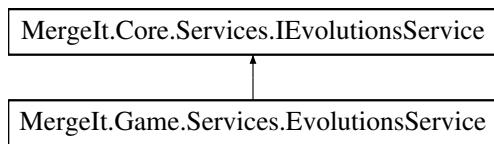
```
00015 { get; } = new();
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Evolutions/EvolutionsProgressData.cs

6.89 Mergelt.Game.Services.EvolutionsService Class Reference

Inheritance diagram for Mergelt.Game.Services.EvolutionsService:



Public Member Functions

- List< ElementConfig > GetEvolutionChain (EvolutionData evolutionData)
- List< ElementConfig > GetGeneratedBy (ElementConfig config)
- List< ElementConfig > GetGenerates (ElementConfig config)
- void UpdateProgress (ElementConfig elementId)
- int GetEvolutionProgress (string id)
- void SetupEvolutionsProgress (IEvolutionsProgressData data)
- IEvolutionsProgressData GetData ()
- void CreateEvolutionsProgress ()

6.89.1 Detailed Description

Definition at line 17 of file [EvolutionsService.cs](#).

6.89.2 Member Function Documentation

6.89.2.1 CreateEvolutionsProgress()

```
void MergeIt.Game.Services.EvolutionsService.CreateEvolutionsProgress ( ) [inline]
```

Implements [Mergelt.Core.Services.IEvolutionsService](#).

Definition at line 135 of file [EvolutionsService.cs](#).

```
00136     {
00137         SchemeObject evolutionsConfig = _configsService.LevelConfig.EvolutionsScheme;
00138         var evolutions = new EvolutionsProgressData();
00139
00140         for (int i = 0; i < evolutionsConfig.Evolution.Count; i++)
00141         {
00142             EvolutionData evolution = evolutionsConfig.Evolution[i];
00143
00144             _serviceModel.EvolutionsProgress[evolution.Id] = evolution.Discovered ?
00145                 evolution.Chain.Count : 1;
00146
00147             SetupEvolutionsProgress(evolutions);
00148
00149         _saveService.Save(GameSaveType.EvolutionsProgress);
00150     }
```

6.89.2.2 GetData()

```
IIEvolutionsProgressData MergeIt.Game.Services.EvolutionsService.GetData ( ) [inline]
```

Implements [Mergelt.Core.Services.IEvolutionsService](#).

Definition at line 117 of file [EvolutionsService.cs](#).

```
00118     {
00119         var evolutions = new EvolutionsProgressData();
00120
00121         foreach (var evoProgressItem in _serviceModel.EvolutionsProgress)
00122         {
00123             var evoProgressData = new EvolutionProgressData
00124             {
00125                 Progress = evoProgressItem.Value,
00126                 EvolutionId = evoProgressItem.Key
00127             };
00128
00129             evolutions.EvolutionsProgress.Add(evoProgressData);
00130         }
00131
00132         return evolutions;
00133     }
```

6.89.2.3 GetEvolutionChain()

```
List< ElementConfig > MergeIt.Game.Services.EvolutionsService.GetEvolutionChain (
    EvolutionData evolutionData ) [inline]
```

Implements [Mergelt.Core.Services.IEvolutionsService](#).

Definition at line 30 of file [EvolutionsService.cs](#).

```
00031     {
00032         var chain = evolutionData?.Chain;
00033
00034         return chain;
00035     }
```

6.89.2.4 GetEvolutionProgress()

```
int MergeIt.Game.Services.EvolutionsService.GetEvolutionProgress (
    string id ) [inline]
```

Implements [MergeIt.Core.Services.IEvolutionsService](#).

Definition at line 96 of file [EvolutionsService.cs](#).

```
00097     {
00098         _serviceModel.EvolutionsProgress.TryGetValue(id, out int progress);
00099         return progress;
00100     }
```

6.89.2.5 GetGeneratedBy()

```
List< ElementConfig > MergeIt.Game.Services.EvolutionsService.GetGeneratedBy (
    ElementConfig config ) [inline]
```

Implements [MergeIt.Core.Services.IEvolutionsService](#).

Definition at line 37 of file [EvolutionsService.cs](#).

```
00038     {
00039         var generators = _configsService.GetConfigs(x =>
00040             {
00041                 ElementGeneratorSettings generatorSettings = x.GeneratorSettings;
00042                 return generatorSettings != null && generatorSettings.GenerateItems.Any(el =>
00043                     el.Element == config);
00044             }).ToList();
00045         return generators;
00046     }
```

6.89.2.6 GetGenerates()

```
List< ElementConfig > MergeIt.Game.Services.EvolutionsService.GetGenerates (
    ElementConfig config ) [inline]
```

Implements [MergeIt.Core.Services.IEvolutionsService](#).

Definition at line 49 of file [EvolutionsService.cs](#).

```
00050     {
00051         List<ElementConfig> generatesElements = null;
00052         if (config.GeneratorSettings != null)
00053         {
00054             generatesElements = new List<ElementConfig>();
00055             var generatedItems = config.GeneratorSettings.GenerateItems;
00056
00057             for (int i = 0; i < generatedItems.Count; i++)
00058             {
00059                 GeneratableElement generatedItem = generatedItems[i];
00060
00061                 ElementConfig elementConfig = generatedItem.Element;
00062
00063                 if (elementConfig != null)
00064                 {
00065                     generatesElements.Add(elementConfig);
00066                 }
00067             }
00068         }
00069
00070         return generatesElements;
00071     }
```

6.89.2.7 SetupEvolutionsProgress()

```
void MergeIt.Game.Services.EvolutionsService.SetupEvolutionsProgress (
    IEvolutionsProgressData data) [inline]
```

Implements [Mergelt.Core.Services.IEvolutionsService](#).

Definition at line 103 of file [EvolutionsService.cs](#).

```
00104     {
00105         var evolutions = data.EvolutionsProgress;
00106
00107         if (evolutions != null)
00108         {
00109             for (int i = 0; i < evolutions.Count; i++)
00110             {
00111                 EvolutionProgressData evolutionData = evolutions[i];
00112                 _serviceModel.EvolutionsProgress[evolutionData.EvolutionId] =
00113                     evolutionData.Progress;
00114             }
00115         }
00116     }
```

6.89.2.8 UpdateProgress()

```
void MergeIt.Game.Services.EvolutionsService.UpdateProgress (
    ElementConfig elementId) [inline]
```

Implements [Mergelt.Core.Services.IEvolutionsService](#).

Definition at line 73 of file [EvolutionsService.cs](#).

```
00074     {
00075         EvolutionData evolution = _configsService.GetEvolutionByElement(elementId);
00076
00077         if (evolution == null || evolution.Discovered)
00078         {
00079             return;
00080         }
00081
00082         string evolutionId = evolution.Id;
00083
00084         if (_serviceModel.EvolutionsProgress.TryGetValue(evolutionId, out int progress))
00085         {
00086             int index = evolution.Chain.IndexOf(elementId);
00087             int order = index + 1;
00088
00089             if (index != -1 && order > progress)
00090             {
00091                 _serviceModel.EvolutionsProgress[evolutionId] = order;
00092             }
00093         }
00094     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/EvolutionsService.cs

6.90 Mergelt.Game.Services.EvolutionsServiceModel Class Reference

Properties

- Dictionary< string, int > [EvolutionsProgress](#) = new() [get]

6.90.1 Detailed Description

Definition at line 7 of file [EvolutionsServiceModel.cs](#).

6.90.2 Property Documentation

6.90.2.1 EvolutionsProgress

```
Dictionary<string, int> MergeIt.Game.Services.EvolutionsServiceModel.EvolutionsProgress =  
new() [get]
```

Definition at line 9 of file [EvolutionsServiceModel.cs](#).

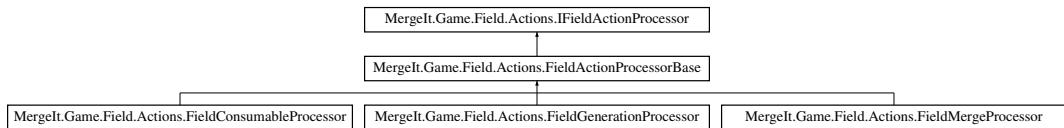
```
00009 { get; } = new();
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/EvolutionsServiceModel.cs

6.91 Mergelt.Game.Field.Actions.FieldActionProcessorBase Class Reference

Inheritance diagram for Mergelt.Game.Field.Actions.FieldActionProcessorBase:



Public Member Functions

- virtual void [ProcessClick \(FieldCellComponent cellComponent\)](#)
- virtual void [ProcessEndDrag \(GridPoint fromPoint, GameObject toGameObject\)](#)

Protected Attributes

- readonly [IMessageBus MessageBus](#) = DiContainer.Get<[IMessageBus](#)>()
- readonly [IGameSaveService SaveService](#) = DiContainer.Get<[IGameSaveService](#)>()
- readonly [FieldLogicModel FieldLogicModel](#) = DiContainer.Get<[FieldLogicModel](#)>()

6.91.1 Detailed Description

Definition at line 11 of file [FieldActionProcessorBase.cs](#).

6.91.2 Member Function Documentation

6.91.2.1 ProcessClick()

```
virtual void MergeIt.Game.Field.Actions.FieldActionProcessorBase.ProcessClick (
    FieldCellComponent cellComponent ) [inline], [virtual]
```

Implements [MergeIt.Game.Field.Actions.IFieldActionProcessor](#).

Definition at line 17 of file [FieldActionProcessorBase.cs](#).

```
00018     {
00019     }
00020 }
```

6.91.2.2 ProcessEndDrag()

```
virtual void MergeIt.Game.Field.Actions.FieldActionProcessorBase.ProcessEndDrag (
    GridPoint fromPoint,
    GameObject toGameObject ) [inline], [virtual]
```

Implements [MergeIt.Game.Field.Actions.IFieldActionProcessor](#).

Definition at line 22 of file [FieldActionProcessorBase.cs](#).

```
00023     {
00024     }
00025 }
```

6.91.3 Member Data Documentation

6.91.3.1 FieldLogicModel

```
readonly FieldLogicModel MergeIt.Game.Field.Actions.FieldActionProcessorBase.FieldLogicModel =
DiContainer.Get<FieldLogicModel>() [protected]
```

Definition at line 15 of file [FieldActionProcessorBase.cs](#).

6.91.3.2 MessageBus

```
readonly IMessageBus MergeIt.Game.Field.Actions.FieldActionProcessorBase.MessageBus = Di←
Container.Get<IMessageBus>() [protected]
```

Definition at line 13 of file [FieldActionProcessorBase.cs](#).

6.91.3.3 SaveService

```
readonly IGameSaveService MergeIt.Game.Field.Actions.FieldActionProcessorBase.SaveService =
DiContainer.Get<IGameSaveService>() [protected]
```

Definition at line 14 of file [FieldActionProcessorBase.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Field/Actions/FieldActionProcessorBase.cs

6.92 Mergelt.Game.Field.FieldCellComponent Class Reference

Inheritance diagram for Mergelt.Game.Field.FieldCellComponent:



Public Member Functions

- void [Initialize](#) (GridPoint gridPoint)
- void [ToggleSelector](#) (bool activate)
- void [AttachObject](#) (IFieldElementPresenter presenter, bool resetPosition=true)
- void [DetachObject](#) ()
- void [OnPointerDown](#) (PointerEventData eventData)
- void [OnPointerUp](#) (PointerEventData eventData)
- void [OnDrag](#) (PointerEventData eventData)
- void [OnBeginDrag](#) (PointerEventData eventData)
- void [OnEndDrag](#) (PointerEventData eventData)

Properties

- IFIELDELEMENTPRESENTER [FieldElementPresenter](#) [get]
- RectTransform [RectTransform](#) [get]
- Transform [ObjectContainer](#) [get]
- GridPoint [Point](#) [get]

6.92.1 Detailed Description

Definition at line 13 of file [FieldCellComponent.cs](#).

6.92.2 Member Function Documentation

6.92.2.1 AttachObject()

```
void MergeIt.Game.Field.FieldCellComponent.AttachObject (
    IFIELDELEMENTPRESENTER presenter,
    bool resetPosition = true ) [inline]
```

Definition at line 91 of file [FieldCellComponent.cs](#).

```
00092         {
00093             if (presenter != null)
00094             {
00095                 _fieldElementPresenter = presenter;
00096                 _fieldElementPresenter.SetParent(ObjectContainer, resetPosition);
00097                 _fieldElementPresenter.SetPoint(Point);
00098             }
00099             else
00100             {
00101                 ToggleSelector(false);
00102             }
00103         }
```

6.92.2.2 DetachObject()

```
void MergeIt.Game.Field.FieldCellComponent.DetachObject ( ) [inline]
```

Definition at line 105 of file [FieldCellComponent.cs](#).

```
00106     {
00107         _fieldElementPresenter = null;
00108     }
```

6.92.2.3 Initialize()

```
void MergeIt.Game.Field.FieldCellComponent.Initialize (
    GridPoint gridPoint ) [inline]
```

Definition at line 67 of file [FieldCellComponent.cs](#).

```
00068     {
00069         _gridPoint = gridPoint;
00070         _messageBus = DiContainer.Get<IMessageBus>();
00071         _gameServiceModel = DiContainer.Get<GameServiceModel>();
00072
00073         _messageBus.AddListener<ResetPositionMessage>(OnResetPositionMessageHandler);
00074     }
```

6.92.2.4 OnBeginDrag()

```
void MergeIt.Game.Field.FieldCellComponent.OnBeginDrag (
    PointerEventData eventData ) [inline]
```

Definition at line 141 of file [FieldCellComponent.cs](#).

```
00142     {
00143         if (_fieldElementPresenter?.IsAvailable == true)
00144         {
00145             _fieldElementPresenter.Select(true);
00146             _fieldElementPresenter.StartDrag();
00147
00148             _messageBus.Fire(new ActivateHintsMessage {Active = false});
00149
00150             SetVisualSelection(false);
00151         }
00152     }
```

6.92.2.5 OnDrag()

```
void MergeIt.Game.Field.FieldCellComponent.OnDrag (
    PointerEventData eventData ) [inline]
```

Definition at line 133 of file [FieldCellComponent.cs](#).

```
00134     {
00135         if (_fieldElementPresenter?.IsAvailable == true)
00136         {
00137             _fieldElementPresenter.Transform.position =
00138                 _gameServiceModel.MainCamera.ScreenToWorldPoint(eventData.position) + _offset;
00139         }
00140     }
```

6.92.2.6 OnEndDrag()

```
void MergeIt.Game.Field.FieldCellComponent.OnEndDrag (
    PointerEventData eventData) [inline]
```

Definition at line 154 of file [FieldCellComponent.cs](#).

```
00155     {
00156         if (_fieldElementPresenter?.IsAvailable == true)
00157     {
00158         _fieldElementPresenter.EndDrag();
00159         SendEndDrag(eventData);
00160
00161         _messageBus.Fire(new ActivateHintsMessage { Active = true });
00162     }
00163 }
```

6.92.2.7 OnPointerDown()

```
void MergeIt.Game.Field.FieldCellComponent.OnPointerDown (
    PointerEventData eventData) [inline]
```

Definition at line 110 of file [FieldCellComponent.cs](#).

```
00111     {
00112         if (_fieldElementPresenter?.IsBusy == false)
00113     {
00114         _offset = _fieldElementPresenter.Transform.position -
00115             _gameServiceModel.MainCamera.ScreenToWorldPoint(eventData.position);
00116         _pointerDownTime = Time.realtimeSinceStartup;
00117     }
00118 }
```

6.92.2.8 OnPointerUp()

```
void MergeIt.Game.Field.FieldCellComponent.OnPointerUp (
    PointerEventData eventData) [inline]
```

Definition at line 119 of file [FieldCellComponent.cs](#).

```
00120     {
00121         if (_fieldElementPresenter?.IsBusy == false)
00122     {
00123         var pointerUpTime = Time.realtimeSinceStartup;
00124         var diff = pointerUpTime - _pointerDownTime;
00125
00126         if (diff <= _pointerClickTime)
00127     {
00128             SendClick();
00129         }
00130     }
00131 }
```

6.92.2.9 ToggleSelector()

```
void MergeIt.Game.Field.FieldCellComponent.ToggleSelector (
    bool activate) [inline]
```

Definition at line 76 of file [FieldCellComponent.cs](#).

```
00077     {
00078         _selector.SetActive(activate);
00079
00080         if (activate)
00081     {
00082             _messageBus.Fire<ResetHintsMessage>();
00083         }
00084
00085         if (!activate && _fieldElementPresenter != null)
00086     {
00087             _fieldElementPresenter.GetModel().Selected = false;
00088         }
00089     }
```

6.92.3 Property Documentation

6.92.3.1 FieldElementPresenter

```
IFieldElementPresenter MergeIt.Game.Field.FieldCellComponent.FieldElementPresenter [get]
```

Definition at line 41 of file [FieldCellComponent.cs](#).

6.92.3.2 ObjectContainer

```
Transform MergeIt.Game.Field.FieldCellComponent.ObjectContainer [get]
```

Definition at line 57 of file [FieldCellComponent.cs](#).

```
00058     {
00059         get => _objectContainer;
00060     }
```

6.92.3.3 Point

```
GridPoint MergeIt.Game.Field.FieldCellComponent.Point [get]
```

Definition at line 62 of file [FieldCellComponent.cs](#).

```
00063     {
00064         get => _gridPoint;
00065     }
```

6.92.3.4 RectTransform

```
RectTransform MergeIt.Game.Field.FieldCellComponent.RectTransform [get]
```

Definition at line 43 of file [FieldCellComponent.cs](#).

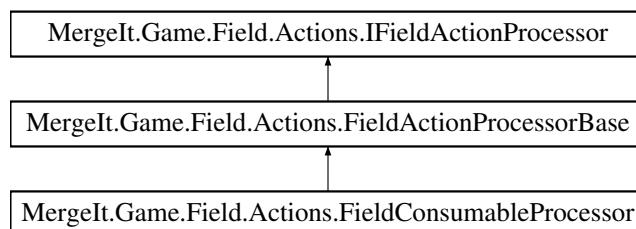
```
00044     {
00045         get
00046     {
00047         if (!_rectTransform)
00048         {
00049             _rectTransform = GetComponent<RectTransform>();
00050         }
00051
00052         return _rectTransform;
00053     }
00054 }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Field/FieldCellComponent.cs

6.93 MergeIt.Game.Field.Actions.FieldConsumableProcessor Class Reference

Inheritance diagram for MergeIt.Game.Field.Actions.FieldConsumableProcessor:



Public Member Functions

- override void [ProcessClick \(FieldCellComponent cellComponent\)](#)

Public Member Functions inherited from [Mergelt.Game.Field.Actions.FieldActionProcessorBase](#)

- virtual void [ProcessEndDrag \(GridPoint fromPoint, GameObject toGameObject\)](#)

Additional Inherited Members

Protected Attributes inherited from [Mergelt.Game.Field.Actions.FieldActionProcessorBase](#)

- readonly [IMessageBus MessageBus](#) = DiContainer.Get<[IMessageBus](#)>()
- readonly [IGameSaveService SaveService](#) = DiContainer.Get<[IGameSaveService](#)>()
- readonly [FieldLogicModel FieldLogicModel](#) = DiContainer.Get<[FieldLogicModel](#)>()

6.93.1 Detailed Description

Definition at line 20 of file [FieldConsumableProcessor.cs](#).

6.93.2 Member Function Documentation

6.93.2.1 ProcessClick()

```
override void MergeIt.Game.Field.Actions.FieldConsumableProcessor.ProcessClick (
    FieldCellComponent cellComponent ) [inline], [virtual]
```

Reimplemented from [Mergelt.Game.Field.Actions.FieldActionProcessorBase](#).

Definition at line 26 of file [FieldConsumableProcessor.cs](#).

```
00027     {
00028         base.ProcessClick(cellComponent);
00029
00030         if (cellComponent.FieldElementPresenter?.GetModel().ClicksCount < 2)
00031         {
00032             return;
00033         }
00034
00035         if (!FieldLogicModel.FieldElements.TryGetValue(cellComponent.Point, out IFieldElement
fieldElement))
00036         {
00037             return;
00038         }
00039
00040         if (fieldElement.InfoParameters.IsBlocked)
00041         {
00042             return;
00043         }
00044
00045         ElementCommonSettings settings =
00046             fieldElement.ConfigParameters.ElementConfig.CommonSettings;
00047
00048         if (!settings.IsConsumable)
00049         {
00050             return;
00051         }
00052         ConsumableSettings consumableSettings = settings.ConsumableSettings;
```

```

00053     Transform transform = cellComponent.FieldElementPresenter.Transform;
00054     MoveEffectParameters parameters = null;
00055
00056     MessageBus.Fire<ResetHintsMessage>();
00057
00058     switch (consumableSettings.Target)
00059     {
00060         case ConsumableTargetType.SoftCurrency:
00061             _userServiceModel.SoftCurrency.ApplyOperation(consumableSettings.Operation,
00062                 consumableSettings.Value, false);
00062             _saveService.Save(GameSaveType.User);
00063             parameters = new MoveEffectParameters(transform.position, Vector3.zero,
00064                 HudPanelType.SoftCurrency);
00064
00065             _effectsFactory.CreateEffect<MoveToUiEffectWithPresenter>(cellComponent.FieldElementPresenter,
00066                 parameters, OnEffectFinished);
00066             break;
00067
00068         case ConsumableTargetType.HardCurrency:
00069             _userServiceModel.HardCurrency.ApplyOperation(consumableSettings.Operation,
00070                 consumableSettings.Value, false);
00071             _saveService.Save(GameSaveType.User);
00072             parameters = new MoveEffectParameters(transform.position, Vector3.zero,
00071                 HudPanelType.HardCurrency);
00072
00073             _effectsFactory.CreateEffect<MoveToUiEffectWithPresenter>(cellComponent.FieldElementPresenter,
00074                 parameters, OnEffectFinished);
00074             break;
00075
00076         case ConsumableTargetType.Time:
00077             RemoveItem(cellComponent);
00078             MessageBus.Fire(new SkipTimeMessage {Seconds = consumableSettings.Value});
00077             break;
00078
00079         case ConsumableTargetType.Energy:
00080             _userServiceModel.Energy.ApplyOperation(consumableSettings.Operation,
00081                 consumableSettings.Value, false);
00082             _saveService.Save(GameSaveType.User);
00083             parameters = new MoveEffectParameters(transform.position, Vector3.zero,
00082                 HudPanelType.Energy);
00083
00084             _effectsFactory.CreateEffect<MoveToUiEffectWithPresenter>(cellComponent.FieldElementPresenter,
00085                 parameters, OnEffectFinished);
00085             break;
00086
00087         case ConsumableTargetType.Experience:
00088             _userServiceModel.Experience.ApplyOperation(consumableSettings.Operation,
00089                 consumableSettings.Value, false);
00089             _saveService.Save(GameSaveType.User);
00090             parameters = new MoveEffectParameters(transform.position, Vector3.zero,
00090                 HudPanelType.UserProgress);
00090
00091             _effectsFactory.CreateEffect<MoveToUiEffectWithPresenter>(cellComponent.FieldElementPresenter,
00092                 parameters, OnEffectFinished);
00092             break;
00093
00094         case ConsumableTargetType.Splitter:
00095             _userServiceModel.Splitters.ApplyOperation(consumableSettings.Operation,
00096                 consumableSettings.Value, false);
00096             _saveService.Save(GameSaveType.User);
00097             parameters = new MoveEffectParameters(transform.position, Vector3.zero,
00097                 HudPanelType.UserProgress);
00097
00098             _effectsFactory.CreateEffect<MoveToUiEffectWithPresenter>(cellComponent.FieldElementPresenter,
00099                 parameters, OnEffectFinished);
00099             break;
00100
00101         void OnEffectFinished()
00102         {
00103             RemoveItem(cellComponent);
00104         }
00105     }

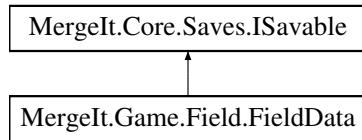
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Actions/FieldConsumableProcessor.cs

6.94 Mergelt.Game.Field.FieldData Class Reference

Inheritance diagram for Mergelt.Game.Field.FieldData:

**Public Attributes**

- int `FieldWidth`
- int `FieldHeight`
- List<`FieldElementData`> `SavedElementsData`

6.94.1 Detailed Description

Definition at line 12 of file [FieldData.cs](#).

6.94.2 Member Data Documentation**6.94.2.1 FieldHeight**

```
int MergeIt.Game.Field.FieldData.FieldHeight
```

Definition at line 17 of file [FieldData.cs](#).

6.94.2.2 FieldWidth

```
int MergeIt.Game.Field.FieldData.FieldWidth
```

Definition at line 15 of file [FieldData.cs](#).

6.94.2.3 SavedElementsData

```
List<FieldElementData> MergeIt.Game.Field.FieldData.SavedElementsData
```

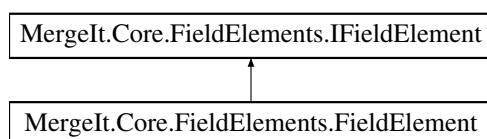
Definition at line 19 of file [FieldData.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/FieldData.cs

6.95 Mergelt.Core.FieldElements.FieldElement Class Reference

Inheritance diagram for Mergelt.Core.FieldElements.FieldElement:



Public Member Functions

- override string [ToString \(\)](#)

Properties

- [IConfigParameters ConfigParameters \[get, set\]](#)
- [IInfoParameters InfoParameters \[get, set\]](#)
- [IGeneratorParameters GeneratorParameters \[get, set\]](#)
- [IGeneratorOpenParameters GeneratorOpenParameters \[get, set\]](#)
- [IProduceParameters ProduceParameters \[get, set\]](#)
- [IProducedByParameters ProducedByParameters \[get, set\]](#)

Properties inherited from [Mergelt.Core.FieldElements.IFieldElement](#)

6.95.1 Detailed Description

Definition at line 8 of file [FieldElement.cs](#).

6.95.2 Member Function Documentation

6.95.2.1 [ToString\(\)](#)

```
override string MergeIt.Core.FieldElements.FieldElement.ToString ( ) [inline]
```

Definition at line 17 of file [FieldElement.cs](#).

```
00018     {
00019         return $"Type: {InfoParameters.Type}, Pos: {InfoParameters.LogicPosition}, Blocked:
00020             {InfoParameters.IsBlocked}";
00020     }
```

6.95.3 Property Documentation

6.95.3.1 [ConfigParameters](#)

[IConfigParameters](#) MergeIt.Core.FieldElements.FieldElement.ConfigParameters [get], [set]

Implements [Mergelt.Core.FieldElements.IFieldElement](#).

Definition at line 10 of file [FieldElement.cs](#).

```
00010 { get; set; }
```

6.95.3.2 [GeneratorOpenParameters](#)

[IGeneratorOpenParameters](#) MergeIt.Core.FieldElements.FieldElement.GeneratorOpenParameters [get], [set]

Implements [Mergelt.Core.FieldElements.IFieldElement](#).

Definition at line 13 of file [FieldElement.cs](#).

```
00013 { get; set; }
```

6.95.3.3 GeneratorParameters

`IGeneratorParameters MergeIt.Core.FieldElements.FieldElement.GeneratorParameters [get], [set]`

Implements [Mergelt.Core.FieldElements.IFieldElement](#).

Definition at line 12 of file [FieldElement.cs](#).

```
00012 { get; set; }
```

6.95.3.4 InfoParameters

`IIInfoParameters MergeIt.Core.FieldElements.FieldElement.InfoParameters [get], [set]`

Implements [Mergelt.Core.FieldElements.IFieldElement](#).

Definition at line 11 of file [FieldElement.cs](#).

```
00011 { get; set; }
```

6.95.3.5 ProducedByParameters

`IProducedByParameters MergeIt.Core.FieldElements.FieldElement.ProducedByParameters [get], [set]`

Implements [Mergelt.Core.FieldElements.IFieldElement](#).

Definition at line 15 of file [FieldElement.cs](#).

```
00015 { get; set; }
```

6.95.3.6 ProduceParameters

`IProduceParameters MergeIt.Core.FieldElements.FieldElement.ProduceParameters [get], [set]`

Implements [Mergelt.Core.FieldElements.IFieldElement](#).

Definition at line 14 of file [FieldElement.cs](#).

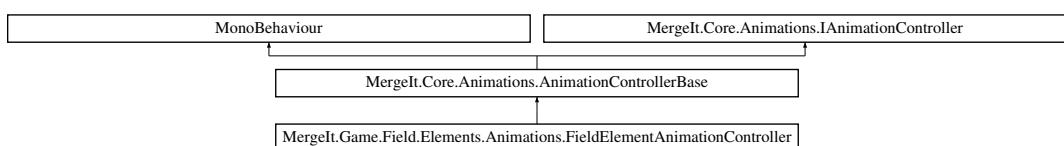
```
00014 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/FieldElement.cs

6.96 Mergelt.Game.Field.Elements.Animations.FieldElementAnimationController Class Reference ←

Inheritance diagram for Mergelt.Game.Field.Elements.Animations.FieldElementAnimationController:



Public Member Functions

- override void [Initialize \(IAnimationListener listener\)](#)
- override void [SetState< T > \(T state\)](#)

Public Member Functions inherited from [MergeIt.Core.Animations.AnimationControllerBase](#)

- void [Initialize \(IAnimationListener listener\)](#)
- virtual void [SetState \(string state\)](#)
- virtual void [SetState \(int state\)](#)

Static Public Member Functions

- static int [GetFieldElementState \(FieldElementState state\)](#)

6.96.1 Detailed Description

Definition at line 11 of file [FieldElementAnimationController.cs](#).

6.96.2 Member Function Documentation

6.96.2.1 GetFieldElementState()

```
static int MergeIt.Game.Field.Elements.Animations.FieldElementAnimationController.GetField←
ElementState (
    FieldElementState state ) [inline], [static]
```

Definition at line 24 of file [FieldElementAnimationController.cs](#).

```
00025     {
00026         StatesHash.TryGetValue(state, out int hash);
00027         return hash;
00028     }
```

6.96.2.2 Initialize()

```
override void MergeIt.Game.Field.Elements.Animations.FieldElementAnimationController.Initialize
(
    IAnimationListener listener ) [inline]
```

Implements [MergeIt.Core.Animations.IAnimationController](#).

Definition at line 36 of file [FieldElementAnimationController.cs](#).

```
00037     {
00038         _listener = listener;
00039     }
```

6.96.2.3 SetState< T >()

```
override void MergeIt.Game.Field.Elements.Animations.FieldElementAnimationController.SetState<
T > (
    T state ) [inline], [virtual]
```

Reimplemented from [Mergelt.Core.Animations.AnimationControllerBase](#).

Definition at line 41 of file [FieldElementAnimationController.cs](#).

```
00042     {
00043         var concreteState = (FieldElementState)(object)state;
00044
00045         if (StatesHash.TryGetValue(concreteState, out int hash))
00046         {
00047             switch (concreteState)
00048             {
00049                 case FieldElementState.Idle:
00050                     _animator.Play(hash);
00051                     break;
00052
00053                 case FieldElementState.Hint:
00054                     _animator.SetTrigger(hash);
00055                     break;
00056             }
00057         }
00058     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Elements/Animations/FieldElementAnimationController.cs

6.97 Mergelt.Core.FieldElements.FieldElementData Class Reference

Public Attributes

- [SavedConfigParameters ConfigParameters](#)
- [SavedInfoParameters InfoParameters](#)
- [SavedGeneratorParameters GeneratorParameters](#)
- [SavedGeneratorOpenParameters GeneratorOpenParameters](#)

6.97.1 Detailed Description

Definition at line 9 of file [FieldElementData.cs](#).

6.97.2 Member Data Documentation

6.97.2.1 ConfigParameters

[SavedConfigParameters](#) [MergeIt.Core.FieldElements.FieldElementData.ConfigParameters](#)

Definition at line 12 of file [FieldElementData.cs](#).

6.97.2.2 GeneratorOpenParameters

`SavedGeneratorOpenParameters MergeIt.Core.FieldElements.FieldElementData.GeneratorOpenParameters`

Definition at line 18 of file [FieldElementData.cs](#).

6.97.2.3 GeneratorParameters

`SavedGeneratorParameters MergeIt.Core.FieldElements.FieldElementData.GeneratorParameters`

Definition at line 16 of file [FieldElementData.cs](#).

6.97.2.4 InfoParameters

`SavedInfoParameters MergeIt.Core.FieldElements.FieldElementData.InfoParameters`

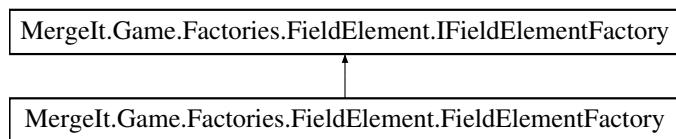
Definition at line 14 of file [FieldElementData.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/FieldElements/FieldElementData.cs

6.98 MergeIt.Game.Factories.FieldElement.FieldElementFactory Class Reference

Inheritance diagram for MergeIt.Game.Factories.FieldElement.FieldElementFactory:



Public Member Functions

- `IFieldElement CreateFieldElement (FieldElementData data)`
- `IFieldElement CreateFieldElement (LevelElementData data)`
- `IFieldElement CreateFieldElement (ElementConfig elementConfig, GridPoint point, bool isBlocked=false)`

6.98.1 Detailed Description

Definition at line 14 of file [FieldElementFactory.cs](#).

6.98.2 Member Function Documentation

6.98.2.1 CreateFieldElement() [1/3]

```
IFieldElement MergeIt.Game.Factories.FieldElement.Factory.CreateFieldElement (
    ElementConfig elementConfig,
    GridPoint point,
    bool isBlocked = false ) [inline]
```

Implements [Mergelt.Game.Factories.FieldElement.IFieldElementFactory](#).

Definition at line 35 of file [FieldElementFactory.cs](#).

```
00036     {
00037         FieldElementData fieldElementData = CreateFieldElementData(elementConfig, point,
00038         isBlocked);
00039         return _configProcessor.ConvertToFieldElement(fieldElementData);
00040     }
```

6.98.2.2 CreateFieldElement() [2/3]

```
IFieldElement MergeIt.Game.Factories.FieldElement.Factory.CreateFieldElement (
    FieldElementData data ) [inline]
```

Implements [Mergelt.Game.Factories.FieldElement.IFieldElementFactory](#).

Definition at line 23 of file [FieldElementFactory.cs](#).

```
00024     {
00025         return _configProcessor.ConvertToFieldElement(data);
00026     }
```

6.98.2.3 CreateFieldElement() [3/3]

```
IFieldElement MergeIt.Game.Factories.FieldElement.Factory.CreateFieldElement (
    LevelElementData data ) [inline]
```

Implements [Mergelt.Game.Factories.FieldElement.IFieldElementFactory](#).

Definition at line 28 of file [FieldElementFactory.cs](#).

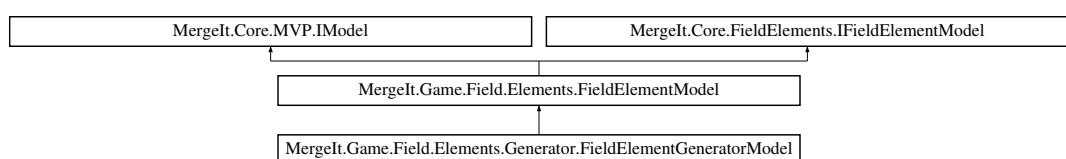
```
00029     {
00030         FieldElementData fieldElementData = _configProcessor.ConvertToFieldElementData(data);
00031         return _configProcessor.ConvertToFieldElement(fieldElementData);
00032     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Factories/FieldElement/FieldElementFactory.cs

6.99 Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorModel Class Reference

Inheritance diagram for Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorModel:



Additional Inherited Members

Properties inherited from [Mergelt.Game.Elements.FieldElementModel](#)

- bool [IsBusy](#) [get, set]
- bool [IsLocked](#) [get, set]
- bool [Selected](#) [get, set]
- int [ClicksCount](#) [get, set]
- [GridPoint Point](#) [get, set]

Properties inherited from [Mergelt.Core.FieldElements.IFieldElementModel](#)

6.99.1 Detailed Description

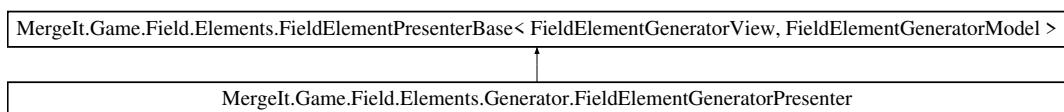
Definition at line 5 of file [FieldElementGeneratorModel.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Elements/Generator/FieldElementGeneratorModel.cs

6.100 [Mergelt.Game.Elements.Generator.FieldElementGenerator](#) ↵ Presenter Class Reference

Inheritance diagram for [Mergelt.Game.Elements.Generator.FieldElementGeneratorPresenter](#):



Public Member Functions

- override void [Update \(IFieldElement fieldElement\)](#)
- override void [Release \(\)](#)

Public Member Functions inherited from

[Mergelt.Game.Elements.FieldElementPresenterBase< FieldElementGeneratorView, FieldElementGeneratorModel >](#)

- virtual void [Update \(IFieldElement fieldElement\)](#)
- virtual void [Activate \(bool isActive\)](#)
- virtual void [Remove \(\)](#)
- virtual void [SetParent \(Transform parent, bool resetPosition=true\)](#)
- virtual void [ResetPosition \(\)](#)
- virtual void [SetPoint \(GridPoint point\)](#)
- virtual void [SetLock \(bool block\)](#)
- virtual void [Select \(bool select\)](#)
- virtual void [StartDrag \(\)](#)
- virtual void [EndDrag \(\)](#)
- virtual void [SetBusy \(bool isBusy\)](#)
- virtual [IFieldElementView GetView \(\)](#)
- virtual [IFieldElementModel GetModel \(\)](#)
- virtual void [SetState \(FieldElementState state\)](#)
- virtual void [ResetAnimationState \(\)](#)

Protected Member Functions

- override void [OnDispose \(\)](#)

Protected Member Functions inherited from

[MergeIt.Game.Field.Elements.FieldElementPresenterBase< FieldElementGeneratorView, FieldElement>](#)

- override void [OnInitialize \(TView view\)](#)

Additional Inherited Members

Protected Attributes inherited from

[MergeIt.Game.Field.Elements.FieldElementPresenterBase< FieldElementGeneratorView, FieldElement>](#)

- [IFieldElement FieldElement](#)

Properties inherited from

[MergeIt.Game.Field.Elements.FieldElementPresenterBase< FieldElementGeneratorView, FieldElement>](#)

- FieldElementState [State \[get\]](#)
- Transform [Transform \[get\]](#)
- bool [IsAvailable \[get\]](#)
- bool [IsBusy \[get\]](#)
- bool [IsLocked \[get\]](#)
- RectTransform [RectTransform \[get\]](#)
- Canvas [Canvas \[get\]](#)

6.100.1 Detailed Description

Definition at line 10 of file [FieldElementGeneratorPresenter.cs](#).

6.100.2 Constructor & Destructor Documentation

6.100.2.1 FieldElementGeneratorPresenter()

```
MergeIt.Game.Field.Elements.Generator.FieldElementGeneratorPresenter.FieldElementGeneratorPresenter () [inline]
```

Definition at line 16 of file [FieldElementGeneratorPresenter.cs](#).

```
00017     {
00018         _messageBus = DiContainer.Get<IMessageBus>();
00019         _messageBus.AddListener<CheckGeneratorMessage>(CheckGeneratorMessageHandler);
00020         _messageBus.AddListener<GeneratorOpenStartMessage>(GeneratorOpenStartMessageHandler);
00021     }
```

6.100.3 Member Function Documentation

6.100.3.1 OnDispose()

```
override void MergeIt.Game.Field.Elements.Generator.FieldElementGeneratorPresenter.OnDispose (
) [inline], [protected]
```

Definition at line 41 of file [FieldElementGeneratorPresenter.cs](#).

```
00042     {
00043         base.OnDispose();
00044
00045         _messageBus.RemoveListener<CheckGeneratorMessage>(CheckGeneratorMessageHandler);
00046         _messageBus.RemoveListener<GeneratorOpenStartMessage>(GeneratorOpenStartMessageHandler);
00047     }
```

6.100.3.2 Release()

```
override void MergeIt.Game.Field.Elements.Generator.FieldElementGeneratorPresenter.Release ( )
[inline], [virtual]
```

Reimplemented from [Mergelt.Game.Field.Elements.FieldElementPresenterBase< FieldElementGeneratorView, FieldElementGeneratorPresenter >](#).

Definition at line 34 of file [FieldElementGeneratorPresenter.cs](#).

```
00035     {
00036         base.Release();
00037
00038         View.HideTimer();
00039     }
```

6.100.3.3 Update()

```
override void MergeIt.Game.Field.Elements.Generator.FieldElementGeneratorPresenter.Update (
    IFIELDELEMENT fieldElement ) [inline]
```

Definition at line 23 of file [FieldElementGeneratorPresenter.cs](#).

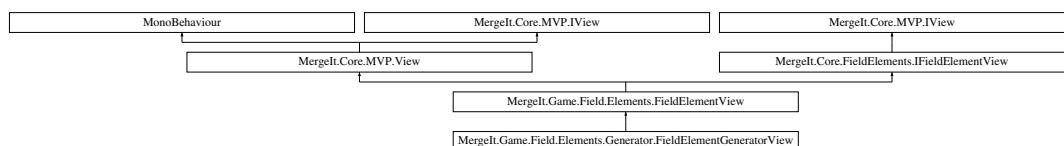
```
00024     {
00025         base.Update(fieldElement);
00026
00027         _generatorParameters = FieldElement.GeneratorParameters;
00028         _generatorOpenParameters = FieldElement.GeneratorOpenParameters;
00029
00030         Check();
00031         CheckOpening();
00032     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Elements/Generator/FieldElementGeneratorPresenter.cs

6.101 Mergelt.Game.Field.Elements.Generator.FieldElementGenerator ↵ View Class Reference

Inheritance diagram for Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorView:



Public Member Functions

- void [SetTimer](#) ([Bindable< float >](#) remainChargeTime, float fullRemainTime)
- void [HideTimer](#) ()

Public Member Functions inherited from [Mergelt.Game.Field.Elements.FieldElementView](#)

- void [Lock](#) (bool isLocked)
- void [ResetState](#) ()

Public Member Functions inherited from [Mergelt.Core.MVP.View](#)

- virtual void [Initialize](#) ()

Additional Inherited Members

Public Attributes inherited from [Mergelt.Game.Field.Elements.FieldElementView](#)

- UnityAction [ResetStateEvent](#)

Protected Member Functions inherited from [Mergelt.Core.MVP.View](#)

- virtual void [OnDestroy](#) ()

Properties inherited from [Mergelt.Game.Field.Elements.FieldElementView](#)

- GraphicRaycaster [GraphicRaycaster](#) [get]
- RectTransform [RectTransform](#) [get]
- Canvas [Canvas](#) [get]
- IAnimationController [AnimationController](#) [get]

Properties inherited from [Mergelt.Core.MVP.View](#)

- GameObject [GameObject](#) [get]

Properties inherited from [Mergelt.Core.MVP.IView](#)

Properties inherited from [Mergelt.Core.FieldElements.IFieldElementView](#)

Events inherited from [Mergelt.Core.MVP.View](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

Events inherited from [Mergelt.Core.MVP.IView](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

6.101.1 Detailed Description

Definition at line 8 of file [FieldElementGeneratorView.cs](#).

6.101.2 Member Function Documentation

6.101.2.1 HideTimer()

```
void MergeIt.Game.Field.Elements.Generator.FieldElementGeneratorView.HideTimer () [inline]
```

Definition at line 18 of file [FieldElementGeneratorView.cs](#).

```
00019     {
00020         _timer.gameObject.SetActive(false);
00021     }
```

6.101.2.2 SetTimer()

```
void MergeIt.Game.Field.Elements.Generator.FieldElementGeneratorView.SetTimer (
    Bindable< float > remainChargeTime,
    float fullRemainTime ) [inline]
```

Definition at line 13 of file [FieldElementGeneratorView.cs](#).

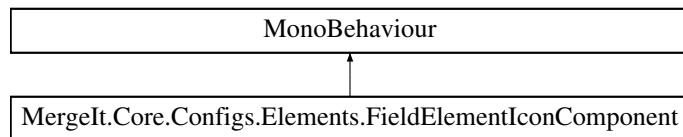
```
00014     {
00015         _timer.StartTimer(remainChargeTime, fullRemainTime);
00016     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Elements/Generator/FieldElementGeneratorView.cs

6.102 Mergelt.Core.Configs.Elements.FieldElementIconComponent Class Reference

Inheritance diagram for Mergelt.Core.Configs.Elements.FieldElementIconComponent:



Public Member Functions

- void [SetBlocked](#) (bool locked)
- Image [GetImage](#) ()

Properties

- RectTransform [RectTransform](#) [get]

6.102.1 Detailed Description

Definition at line 9 of file [FieldElementIconComponent.cs](#).

6.102.2 Member Function Documentation

6.102.2.1 GetImage()

```
Image MergeIt.Core.Configs.Elements.FieldElementIconComponent.GetImage () [inline]
```

Definition at line 54 of file [FieldElementIconComponent.cs](#).

```
00055     {  
00056         return _image;  
00057     }
```

6.102.2.2 SetBlocked()

```
void MergeIt.Core.Configs.Elements.FieldElementIconComponent.SetBlocked (  
    bool locked) [inline]
```

Definition at line 42 of file [FieldElementIconComponent.cs](#).

```
00043     {  
00044         if (locked)  
00045         {  
00046             _image.material = _blockedMaterial;  
00047         }  
00048         else  
00049         {  
00050             _image.material = _defaultMaterial;  
00051         }  
00052     }
```

6.102.3 Property Documentation

6.102.3.1 RectTransform

```
RectTransform MergeIt.Core.Configs.Elements.FieldElementIconComponent.RectTransform [get]
```

Definition at line 22 of file [FieldElementIconComponent.cs](#).

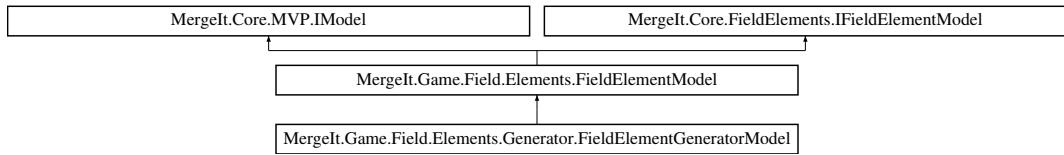
```
00023     {  
00024         get => _rectTransform;  
00025     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Configs/Elements/FieldElementIconComponent.cs

6.103 Mergelt.Game.Field.Elements.FieldElementModel Class Reference

Inheritance diagram for Mergelt.Game.Field.Elements.FieldElementModel:



Properties

- bool `IsBusy` [get, set]
- bool `IsLocked` [get, set]
- bool `Selected` [get, set]
- int `ClicksCount` [get, set]
- `GridPoint Point` [get, set]

Properties inherited from [Mergelt.Core.FieldElements.IFieldElementModel](#)

6.103.1 Detailed Description

Definition at line 13 of file [FieldElementModel.cs](#).

6.103.2 Property Documentation

6.103.2.1 ClicksCount

```
int MergeIt.Game.Field.Elements.FieldElementModel.ClicksCount [get], [set]
```

Implements [Mergelt.Core.FieldElements.IFieldElementModel](#).

Definition at line 46 of file [FieldElementModel.cs](#).

```
00047      {
00048          get => _clicksCount;
00049          set => _clicksCount = value;
00050      }
```

6.103.2.2 IsBusy

```
bool MergeIt.Game.Field.Elements.FieldElementModel.IsBusy [get], [set]
```

Implements [Mergelt.Core.FieldElements.IFieldElementModel](#).

Definition at line 22 of file [FieldElementModel.cs](#).

```
00022 { get; set; }
```

6.103.2.3 IsLocked

```
bool MergeIt.Game.Field.Elements.FieldElementModel.IsLocked [get], [set]
```

Implements [Mergelt.Core.FieldElements.IFieldElementModel](#).

Definition at line 23 of file [FieldElementModel.cs](#).

```
00023 { get; set; }
```

6.103.2.4 Point

```
GridPoint MergeIt.Game.Field.Elements.FieldElementModel.Point [get], [set]
```

Implements [Mergelt.Core.FieldElements.IFieldElementModel](#).

Definition at line 52 of file [FieldElementModel.cs](#).

```
00053 {
00054     get => _point;
00055     set => _point = value;
00056 }
```

6.103.2.5 Selected

```
bool MergeIt.Game.Field.Elements.FieldElementModel.Selected [get], [set]
```

Implements [Mergelt.Core.FieldElements.IFieldElementModel](#).

Definition at line 25 of file [FieldElementModel.cs](#).

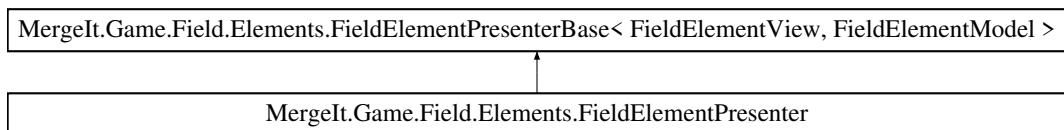
```
00026 {
00027     get => _selected;
00028     set
00029     {
00030         if (value != _selected)
00031         {
00032             _selected = value;
00033
00034             if (_selected)
00035             {
00036                 SendSelected();
00037             }
00038             else
00039             {
00040                 _clicksCount = 0;
00041             }
00042         }
00043     }
00044 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Elements/FieldElementModel.cs

6.104 Mergelt.Game.Field.Elements.FieldElementPresenter Class Reference

Inheritance diagram for MergeIt.Game.Field.Elements.FieldElementPresenter:



Additional Inherited Members

Public Member Functions inherited from

Mergelt.Game.Field.Elements.FieldElementPresenterBase< FieldElementView, FieldElementModel >

- virtual void [Update](#) (IFieldElement fieldElement)
- virtual void [Activate](#) (bool isActive)
- virtual void [Release](#) ()
- virtual void [Remove](#) ()
- virtual void [SetParent](#) (Transform parent, bool resetPosition=true)
- virtual void [ResetPosition](#) ()
- virtual void [SetPoint](#) (GridPoint point)
- virtual void [SetLock](#) (bool block)
- virtual void [Select](#) (bool select)
- virtual void [StartDrag](#) ()
- virtual void [EndDrag](#) ()
- virtual void [SetBusy](#) (bool isBusy)
- virtual IFieldElementView [GetView](#) ()
- virtual IFieldElementModel [GetModel](#) ()
- virtual void [SetState](#) (FieldElementState state)
- virtual void [ResetAnimationState](#) ()

Protected Member Functions inherited from

Mergelt.Game.Field.Elements.FieldElementPresenterBase< FieldElementView, FieldElementModel >

- override void [OnInitialize](#) (TView view)

Protected Attributes inherited from

Mergelt.Game.Field.Elements.FieldElementPresenterBase< FieldElementView, FieldElementModel >

- IFieldElement [FieldElement](#)

Properties inherited from

Mergelt.Game.Field.Elements.FieldElementPresenterBase< FieldElementView, FieldElementModel >

- FieldElementState [State](#) [get]
- Transform [Transform](#) [get]
- bool [IsAvailable](#) [get]
- bool [IsBusy](#) [get]
- bool [IsLocked](#) [get]
- RectTransform [RectTransform](#) [get]
- Canvas [Canvas](#) [get]

6.104.1 Detailed Description

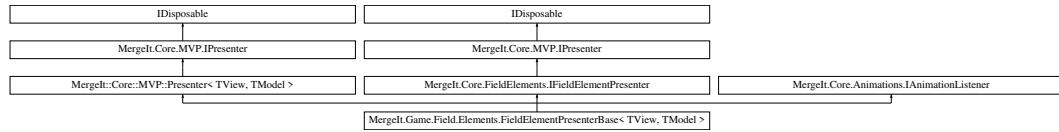
Definition at line 5 of file [FieldElementPresenter.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Elements/FieldElementPresenter.cs

6.105 Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel > Class Template Reference

Inheritance diagram for Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >:



Public Member Functions

- virtual void [Update \(IFieldElement fieldElement\)](#)
- virtual void [Activate \(bool isActive\)](#)
- virtual void [Release \(\)](#)
- virtual void [Remove \(\)](#)
- virtual void [SetParent \(Transform parent, bool resetPosition=true\)](#)
- virtual void [ResetPosition \(\)](#)
- virtual void [SetPoint \(GridPoint point\)](#)
- virtual void [SetLock \(bool block\)](#)
- virtual void [Select \(bool select\)](#)
- virtual void [StartDrag \(\)](#)
- virtual void [EndDrag \(\)](#)
- virtual void [SetBusy \(bool isBusy\)](#)
- virtual [IFieldElementView GetView \(\)](#)
- virtual [IFieldElementModel GetModel \(\)](#)
- virtual void [SetState \(FieldElementState state\)](#)
- virtual void [ResetAnimationState \(\)](#)

Public Member Functions inherited from [MergeIt.Core.MVP.Presenter< TView, TModel >](#)

- void [Initialize \(IView view\)](#)
- void [Dispose \(\)](#)

Protected Member Functions

- override void [OnInitialize \(TView view\)](#)

Protected Member Functions inherited from [MergeIt.Core.MVP.Presenter< TView, TModel >](#)

- virtual void [OnDispose \(\)](#)

Protected Attributes

- [IFieldElement FieldElement](#)

Properties

- FieldElementState **State** [get]
- Transform **Transform** [get]
- bool **IsAvailable** [get]
- bool **IsBusy** [get]
- bool **IsLocked** [get]
- RectTransform **RectTransform** [get]
- Canvas **Canvas** [get]

Properties inherited from [Mergelt.Core.MVP.Presenter< TView, TModel >](#)

- TView **View** [get]
- TModel **Model = new()** [get]

Properties inherited from [Mergelt.Core.FieldElements.IFieldElementPresenter](#)

6.105.1 Detailed Description

Type Constraints

TView : [FieldElementView](#)
TModel : [FieldElementModel](#)
TModel : [new\(\)](#)

Definition at line 12 of file [FieldElementPresenterBase.cs](#).

6.105.2 Member Function Documentation

6.105.2.1 Activate()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.Activate (
    bool isActive) [inline], [virtual]
```

Implements [Mergelt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 93 of file [FieldElementPresenterBase.cs](#).

```
00094     {
00095         Model.Selected = false;
00096         View.GameObject.SetActive(isActive);
00097     }
```

6.105.2.2 EndDrag()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.EndDrag (
) [inline], [virtual]
```

Implements [Mergelt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 152 of file [FieldElementPresenterBase.cs](#).

```
00153     {
00154         View.Canvas.sortingOrder = 1;
00155         View.GraphicRaycaster.enabled = true;
00156     }
```

6.105.2.3 GetModel()

```
virtual IFieldElementModel MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.GetModel ( ) [inline], [virtual]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 168 of file [FieldElementPresenterBase.cs](#).

```
00169     {
00170         return Model;
00171     }
```

6.105.2.4 GetView()

```
virtual IFieldElementView MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.GetView ( ) [inline], [virtual]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 163 of file [FieldElementPresenterBase.cs](#).

```
00164     {
00165         return View;
00166     }
```

6.105.2.5 OnInitialize()

```
override void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.OnInitialize (
    TView view) [inline], [protected], [virtual]
```

Reimplemented from [MergeIt.Core.MVP.Presenter< TView, TModel >](#).

Definition at line 184 of file [FieldElementPresenterBase.cs](#).

```
00185     {
00186         base.OnInitialize(view);
00187         View.ResetStateEvent += OnResetState;
00188         View.AnimationController.Initialize(this);
00189     }
00190 }
```

6.105.2.6 Release()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.Release (
) [inline], [virtual]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 99 of file [FieldElementPresenterBase.cs](#).

```
00100     {
00101
00102     }
```

6.105.2.7 Remove()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.Remove ( )  
[inline], [virtual]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 104 of file [FieldElementPresenterBase.cs](#).

```
00105      {  
00106          Model.Selected = false;  
00107          View.ResetStateEvent -= OnResetState;  
00108  
00109          Object.Destroy(View.GameObject);  
00110      }
```

6.105.2.8 ResetAnimationState()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.Reset←  
AnimationState ( ) [inline], [virtual]
```

Implements [MergeIt.Core.Animations.IAnimationListener](#).

Definition at line 179 of file [FieldElementPresenterBase.cs](#).

```
00180      {  
00181          View.ResetState();  
00182      }
```

6.105.2.9 ResetPosition()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.Reset←  
Position ( ) [inline], [virtual]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 125 of file [FieldElementPresenterBase.cs](#).

```
00126      {  
00127          RectTransform.anchoredPosition = Vector2.zero;  
00128      }
```

6.105.2.10 Select()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.Select (  
    bool select ) [inline], [virtual]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 141 of file [FieldElementPresenterBase.cs](#).

```
00142      {  
00143          Model.Selected = select;  
00144      }
```

6.105.2.11 SetBusy()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.SetBusy (   
    bool isBusy ) [inline], [virtual]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 158 of file [FieldElementPresenterBase.cs](#).

```
00159      {  
00160          Model.IsBusy = isBusy;  
00161      }
```

6.105.2.12 SetLock()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.SetLock ( bool block ) [inline], [virtual]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 135 of file [FieldElementPresenterBase.cs](#).

```
00136     {  
00137         Model.IsLocked = block;  
00138         View.Lock(block);  
00139     }
```

6.105.2.13 SetParent()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.SetParent ( Transform parent, bool resetPosition = true ) [inline], [virtual]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 112 of file [FieldElementPresenterBase.cs](#).

```
00113     {  
00114         Transform.SetParent(parent);  
00115         RectTransform.Stretch();  
00116  
00117         if (resetPosition)  
00118         {  
00119             RectTransform.anchoredPosition = Vector2.zero;  
00120         }  
00121  
00122         Transform.localScale = Vector3.one;  
00123     }
```

6.105.2.14 SetPoint()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.SetPoint ( GridPoint point ) [inline], [virtual]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 130 of file [FieldElementPresenterBase.cs](#).

```
00131     {  
00132         Model.Point = point;  
00133     }
```

6.105.2.15 SetState()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.SetState ( FieldElementState state ) [inline], [virtual]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 173 of file [FieldElementPresenterBase.cs](#).

```
00174     {  
00175         _state = state;  
00176         View.AnimationController.SetState(state);  
00177     }
```

6.105.2.16 StartDrag()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.StartDrag()
( ) [inline], [virtual]
```

Implements [Mergelt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 146 of file [FieldElementPresenterBase.cs](#).

```
00147     {
00148         View.Canvas.sortingOrder = 2;
00149         View.GraphicRaycaster.enabled = false;
00150     }
```

6.105.2.17 Update()

```
virtual void MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.Update (
    IFieldElement fieldElement ) [inline], [virtual]
```

Implements [Mergelt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 81 of file [FieldElementPresenterBase.cs](#).

```
00082     {
00083         FieldElement = fieldElement;
00084         Model.Point = FieldElement.InfoParameters.LogicPosition;
00085         Model.IsLocked = FieldElement.InfoParameters.IsBlocked;
00086
00087         View.GameObject.name = $"[{Model.Point.X}, {Model.Point.Y}]"
        {FieldElement.InfoParameters.Name}";
00088
00089         View.Lock(Model.IsLocked);
00090         View.ResetState();
00091     }
```

6.105.3 Member Data Documentation

6.105.3.1 FieldElement

```
IFieldElement MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.FieldElement
[protected]
```

Definition at line 16 of file [FieldElementPresenterBase.cs](#).

6.105.4 Property Documentation

6.105.4.1 Canvas

```
Canvas MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.Canvas [get]
```

Implements [Mergelt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 68 of file [FieldElementPresenterBase.cs](#).

```
00069     {
00070         get
00071         {
00072             if (!__canvas)
00073             {
00074                 __canvas = View.Canvas;
00075             }
00076
00077             return __canvas;
00078         }
00079     }
```

6.105.4.2 IsAvailable

```
bool MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.IsAvailable [get]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 40 of file [FieldElementPresenterBase.cs](#).

```
00041     {
00042         get => !Model.IsBusy && !Model.IsLocked;
00043     }
```

6.105.4.3 IsBusy

```
bool MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.IsBusy [get]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 45 of file [FieldElementPresenterBase.cs](#).

```
00046     {
00047         get => Model.IsBusy;
00048     }
```

6.105.4.4 IsLocked

```
bool MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.IsLocked [get]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 50 of file [FieldElementPresenterBase.cs](#).

```
00051     {
00052         get => Model.IsLocked;
00053     }
```

6.105.4.5 RectTransform

```
RectTransform MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.RectTransform [get]
```

Implements [MergeIt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 55 of file [FieldElementPresenterBase.cs](#).

```
00056     {
00057         get
00058         {
00059             if (!_rectTransform)
00060             {
00061                 _rectTransform = View.RectTransform;
00062             }
00063             return _rectTransform;
00064         }
00065     }
00066 }
```

6.105.4.6 State

```
FieldElementState MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.State
[get]
```

Implements [Mergelt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 22 of file [FieldElementPresenterBase.cs](#).

```
00023     {
00024         get => _state;
00025     }
```

6.105.4.7 Transform

```
Transform MergeIt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >.Transform
[get]
```

Implements [Mergelt.Core.FieldElements.IFieldElementPresenter](#).

Definition at line 27 of file [FieldElementPresenterBase.cs](#).

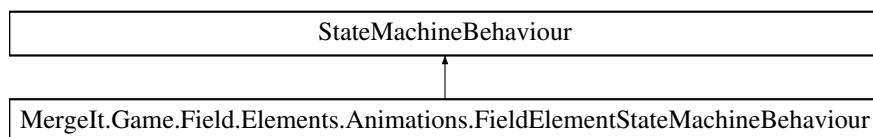
```
00028     {
00029         get
00030         {
00031             if (!_transform)
00032             {
00033                 _transform = View.transform;
00034             }
00035
00036             return _transform;
00037         }
00038     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Elements/FieldElementPresenterBase.cs

6.106 Mergelt.Game.Field.Elements.Animations.FieldElementStateMachineBehaviour Class Reference

Inheritance diagram for Mergelt.Game.Field.Elements.Animations.FieldElementStateMachineBehaviour:



Public Member Functions

- override void [OnStateExit](#) (Animator animator, AnimatorStateInfo stateInfo, int layerIndex)

6.106.1 Detailed Description

Definition at line 9 of file [FieldElementStateMachineBehaviour.cs](#).

6.106.2 Member Function Documentation

6.106.2.1 OnStateExit()

```
override void MergeIt.Game.Field.Elements.Animations.FieldElementStateMachineBehaviour.OnStateExit (
    Animator animator,
    AnimatorStateInfo stateInfo,
    int layerIndex ) [inline]
```

Definition at line 11 of file [FieldElementStateMachineBehaviour.cs](#).

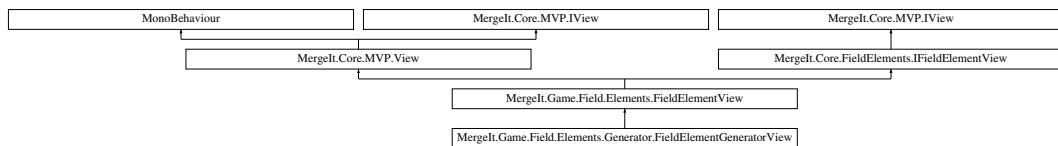
```
00012     {
00013         base.OnStateExit(animator, stateInfo, layerIndex);
00014         var view = animator.gameObject.GetComponent<IFieldElementView>();
00015         int stateHash = stateInfo.shortNameHash;
00016
00017         if (view != null)
00018         {
00019             if (stateHash ==
FieldElementAnimationController.GetFieldElementState(FieldElementState.Hint))
00020                 {
00021                     view.ResetState();
00022                 }
00023         }
00024     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Elements/Animations/FieldElementStateMachineBehaviour.cs

6.107 Mergelt.Game.Field.Elements.FieldElementView Class Reference

Inheritance diagram for Mergelt.Game.Field.Elements.FieldElementView:



Public Member Functions

- void [Lock](#) (bool isLocked)
- void [ResetState](#) ()

Public Member Functions inherited from [MergeIt.Core.MVP.View](#)

- virtual void [Initialize](#) ()

Public Attributes

- UnityAction [ResetStateEvent](#)

Properties

- GraphicRaycaster [GraphicRaycaster](#) [get]
- RectTransform [RectTransform](#) [get]
- Canvas [Canvas](#) [get]
- IAnimationController [AnimationController](#) [get]

Properties inherited from [Mergelt.Core.MVP.View](#)

- GameObject [GameObject](#) [get]

Properties inherited from [Mergelt.Core.MVP.IView](#)

Properties inherited from [Mergelt.Core.FieldElements.IFieldElementView](#)

Additional Inherited Members

Protected Member Functions inherited from [Mergelt.Core.MVP.View](#)

- virtual void [OnDestroy](#) ()

Events inherited from [Mergelt.Core.MVP.View](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

Events inherited from [Mergelt.Core.MVP.IView](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

6.107.1 Detailed Description

Definition at line 13 of file [FieldElementView.cs](#).

6.107.2 Member Function Documentation

6.107.2.1 Lock()

```
void MergeIt.Game.Field.Elements.FieldElementView.Lock (
    bool isLocked ) [inline]
```

Implements [Mergelt.Core.FieldElements.IFieldElementView](#).

Definition at line 66 of file [FieldElementView.cs](#).

```
00067      {
00068          if ( !_iconComponent )
00069          {
00070              _iconComponent = GetComponentInChildren<FieldElementIconComponent>();
00071          }
00072
00073          _iconComponent.SetBlocked(isLocked);
00074          _blocker.SetActive(isLocked);
00075      }
```

6.107.2.2 ResetState()

```
void MergeIt.Game.Field.Elements.FieldElementView.ResetState ( ) [inline]
```

Implements [MergeIt.Core.FieldElements.IFieldElementView](#).

Definition at line 77 of file [FieldElementView.cs](#).

```
00078      {  
00079          ResetStateEvent?.Invoke ();  
00080      }
```

6.107.3 Member Data Documentation

6.107.3.1 ResetStateEvent

```
UnityAction MergeIt.Game.Field.Elements.FieldElementView.ResetStateEvent
```

Definition at line 15 of file [FieldElementView.cs](#).

6.107.4 Property Documentation

6.107.4.1 AnimationController

```
IAnimationController MergeIt.Game.Field.Elements.FieldElementView.AnimationController [get]
```

Definition at line 50 of file [FieldElementView.cs](#).

```
00051      {  
00052          get { return _animationController ??= GetComponent<IAnimationController>(); }  
00053      }
```

6.107.4.2 Canvas

```
Canvas MergeIt.Game.Field.Elements.FieldElementView.Canvas [get]
```

Implements [MergeIt.Core.FieldElements.IFieldElementView](#).

Definition at line 37 of file [FieldElementView.cs](#).

```
00038      {  
00039          get  
00040          {  
00041              if (!_canvas)  
00042              {  
00043                  _canvas = GetComponent<Canvas>();  
00044              }  
00045  
00046              return _canvas;  
00047          }  
00048      }
```

6.107.4.3 GraphicRaycaster

```
GraphicRaycaster MergeIt.Game.Field.Elements.FieldElementView.GraphicRaycaster [get]
```

Implements [MergeIt.Core.FieldElements.IFieldElementView](#).

Definition at line 22 of file [FieldElementView.cs](#).

6.107.4.4 RectTransform

RectTransform MergeIt.Game.Field.Elements.FieldElementView.RectTransform [get]

Implements [Mergelt.Core.FieldElements.IFieldElementView](#).

Definition at line 24 of file [FieldElementView.cs](#).

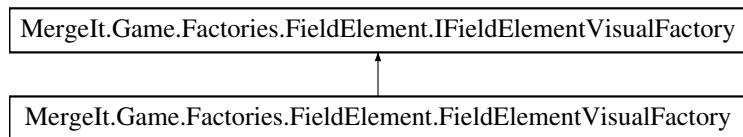
```
00025      {
00026          get
00027          {
00028              if (!_rectTransform)
00029              {
00030                  _rectTransform = GetComponent<RectTransform>();
00031              }
00032
00033              return _rectTransform;
00034          }
00035      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Elements/FieldElementView.cs

6.108 Mergelt.Game.Factories.FieldElement.FieldElementVisualFactory Class Reference

Inheritance diagram for Mergelt.Game.Factories.FieldElement.FieldElementVisualFactory:



Public Member Functions

- [IFieldElementPresenter CreateFieldElement \(IFieldElement element\)](#)

6.108.1 Detailed Description

Definition at line 19 of file [FieldElementVisualFactory.cs](#).

6.108.2 Member Function Documentation

6.108.2.1 CreateFieldElement()

```
IFieldElementPresenter MergeIt.Game.Factories.FieldElement.FieldElementVisualFactory.Create←
FieldElement (
    IFIELDLEMENT element ) [inline]
```

Implements [Mergelt.Game.Factories.FieldElement.IFieldElementVisualFactory](#).

Definition at line 31 of file [FieldElementVisualFactory.cs](#).

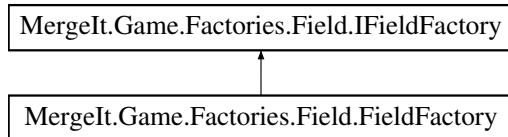
```
00032     {
00033         ElementConfig config = element.ConfigParameters.ElementConfig;
00034         _actualElement = element;
00035
00036         if (!_pools.TryGetValue(config, out IObjectPool<IFieldElementPresenter> pool))
00037         {
00038             pool = new ObjectPool<IFieldElementPresenter>(
00039                 () =>
00040                 {
00041                     ElementType type = _actualElement.InfoParameters.Type;
00042                     IFIELDELEMENTVIEW objectPrototype = _configsService.GetElementPrefab(type);
00043                     GameObject fieldObject = Object.Instantiate(objectPrototype.GameObject);
00044
00045                     _iconFactory.CreateIcon(config, fieldObject.transform);
00046
00047                     var view = fieldObject.GetComponent<IFIELDELEMENTVIEW>();
00048
00049                     IFIELDELEMENTPresenter elementPresenter = null;
00050                     switch (type)
00051                     {
00052                         case ElementType.Regular:
00053                             elementPresenter = new FieldElementPresenter();
00054                             break;
00055
00056                         case ElementType.Generator:
00057                             elementPresenter = new FieldElementGeneratorPresenter();
00058                             break;
00059
00060                         default:
00061                             throw new InvalidEnumArgumentException($"Unknown item type: {type}");
00062                     }
00063
00064                     if (elementPresenter != null)
00065                     {
00066                         elementPresenter.Initialize(view);
00067                         elementPresenter.Update(_actualElement);
00068
00069                         if (view != null)
00070                         {
00071                             view.Initialize();
00072                         }
00073                     }
00074
00075
00076                     return elementPresenter;
00077     },
00078     pooledElement =>
00079     {
00080         pooledElement.Update(_actualElement);
00081         pooledElement.Activate(true);
00082     },
00083     pooledElement =>
00084     {
00085         pooledElement.Release();
00086         pooledElement.Activate(false);
00087     },
00088     pooledElement =>
00089     {
00090         pooledElement.Remove();
00091     },
00092     defaultCapacity: 2,
00093     maxSize: 20);
00094
00095     _pools[config] = pool;
00096 }
00097
00098 IFIELDELEMENTPresenter presenter = SetupElement(pool);
00099
00100 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Factories/FieldElement/FieldElementVisualFactory.cs

6.109 Mergelt.Game.Factories.Field.FieldFactory Class Reference

Inheritance diagram for Mergelt.Game.Factories.Field.FieldFactory:



Public Member Functions

- [FieldPresenter CreateField](#) (Transform parent)
- [FieldCellComponent CreateFieldCell](#) (Transform parent, GridPoint gridPoint)

Static Public Attributes

- const string [FieldGameObjectPath](#) = "Prefabs/Field/FieldContainer"
- const string [FieldCellObjectOddPath](#) = "Prefabs/Field/FieldCellOdd"
- const string [FieldCellObjectEvenPath](#) = "Prefabs/Field/FieldCellEven"

6.109.1 Detailed Description

Definition at line 11 of file [FieldFactory.cs](#).

6.109.2 Member Function Documentation

6.109.2.1 CreateField()

```
FieldPresenter MergeIt.Game.Factories.Field.FieldFactory.CreateField (
    Transform parent ) [inline]
```

Implements [Mergelt.Game.Factories.Field.IFieldFactory](#).

Definition at line 20 of file [FieldFactory.cs](#).

```

00021     {
00022         var fieldViewPrefab = _resourcesLoaderService.GetObject<FieldView>(FieldGameObjectPath);
00023         var fieldViewObject = Object.Instantiate(fieldViewPrefab.gameObject, parent);
00024
00025         fieldViewObject.TryGetComponent(out FieldView fieldView);
00026
00027         if (fieldView)
00028     {
00029             var presenter = new FieldPresenter();
00030             presenter.Initialize(fieldView);
00031
00032             fieldView.Initialize();
00033
00034             return presenter;
00035         }
00036
00037         return null;
00038     }
  
```

6.109.2.2 CreateFieldCell()

```
FieldCellComponent MergeIt.Game.Factories.Field.FieldFactory.CreateFieldCell (
    Transform parent,
    GridPoint gridPoint ) [inline]
```

Implements [MergeIt.Game.Factories.Field.IFieldFactory](#).

Definition at line 40 of file [FieldFactory.cs](#).

```
00041     {
00042         int sum = (gridPoint.X + gridPoint.Y) % 2;
00043         var fieldCellPrefab = sum == 0 ?
00044             _resourcesLoaderService.GetObject<FieldCellComponent>(FieldCellObjectEvenPath) :
00045             _resourcesLoaderService.GetObject<FieldCellComponent>(FieldCellObjectOddPath) ;
00046         var fieldCellObject = Object.Instantiate(fieldCellPrefab.gameObject, parent);
00047         fieldCellObject.TryGetComponent(out FieldCellComponent fieldCell);
00048         if (fieldCell)
00049         {
00050             fieldCell.Initialize(gridPoint);
00051         }
00052         return fieldCell;
00053     }
00054 }
```

6.109.3 Member Data Documentation

6.109.3.1 FieldCellObjectEvenPath

```
const string MergeIt.Game.Factories.Field.FieldFactory.FieldCellObjectEvenPath = "Prefabs/Field/Field←
CellEven" [static]
```

Definition at line 15 of file [FieldFactory.cs](#).

6.109.3.2 FieldCellObjectOddPath

```
const string MergeIt.Game.Factories.Field.FieldFactory.FieldCellObjectOddPath = "Prefabs/Field/Field←
CellOdd" [static]
```

Definition at line 14 of file [FieldFactory.cs](#).

6.109.3.3 FieldGameObjectPath

```
const string MergeIt.Game.Factories.Field.FieldFactory.FieldGameObjectPath = "Prefabs/Field/Field←
Container" [static]
```

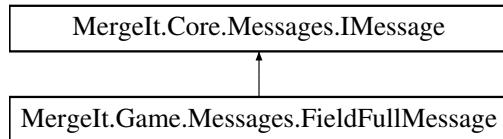
Definition at line 13 of file [FieldFactory.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Factories/Field/FieldFactory.cs

6.110 Mergelt.Game.Messages.FieldFullMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.FieldFullMessage:



6.110.1 Detailed Description

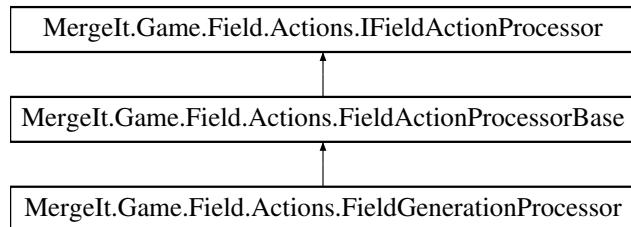
Definition at line 7 of file [FieldFullMessage.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/FieldFullMessage.cs

6.111 Mergelt.Game.Field.Actions.FieldGenerationProcessor Class Reference

Inheritance diagram for Mergelt.Game.Field.Actions.FieldGenerationProcessor:



Public Member Functions

- override void [ProcessClick](#) ([FieldCellComponent](#) cellComponent)

Public Member Functions inherited from [Mergelt.Game.Field.Actions.FieldActionProcessorBase](#)

- virtual void [ProcessEndDrag](#) ([GridPoint](#) fromPoint, [GameObject](#) toGameObject)

Additional Inherited Members

Protected Attributes inherited from [Mergelt.Game.Field.Actions.FieldActionProcessorBase](#)

- readonly [IMessageBus](#) MessageBus = DiContainer.Get<[IMessageBus](#)>()
- readonly [IGameSaveService](#) SaveService = DiContainer.Get<[IGameSaveService](#)>()
- readonly [FieldLogicModel](#) FieldLogicModel = DiContainer.Get<[FieldLogicModel](#)>()

6.111.1 Detailed Description

Definition at line 15 of file [FieldGenerationProcessor.cs](#).

6.111.2 Member Function Documentation

6.111.2.1 ProcessClick()

```
override void MergeIt.Game.Field.Actions.FieldGenerationProcessor.ProcessClick (
    FieldCellComponent cellComponent ) [inline], [virtual]
```

Reimplemented from [MergeIt.Game.Field.Actions.FieldActionProcessorBase](#).

Definition at line 21 of file [FieldGenerationProcessor.cs](#).

```
00022     {
00023         base.ProcessClick(cellComponent);
00024
00025         if (cellComponent.FieldElementPresenter?.GetModel().ClicksCount < 2)
00026         {
00027             return;
00028         }
00029
00030         if (!FieldLogicModel.FieldElements.TryGetValue(cellComponent.Point, out IFieldElement
00031             fieldElement))
00031         {
00032             return;
00033         }
00034
00035         IGeneratorParameters generatorParameters = fieldElement.GeneratorParameters;
00036         if (generatorParameters == null)
00037         {
00038             return;
00039         }
00040
00041         IGeneratorOpenParameters generatorOpenParameters = fieldElement.GeneratorOpenParameters;
00042         if (generatorOpenParameters != null)
00043         {
00044             Debug.Log("Generator should be open or opening.");
00045             return;
00046         }
00047
00048         if (_userServiceModel.Energy.Value == 0)
00049         {
00050             Debug.Log("Not enough energy.");
00051             return;
00052         }
00053
00054         GridPoint? freePoint = _gameFieldService.GetFreeCell();
00055
00056         if (!freePoint.HasValue)
00057         {
00058             SendFieldFull();
00059             return;
00060         }
00061
00062         ElementGeneratorSettings generatorSettings =
00063             fieldElement.ConfigParameters.ElementConfig.GeneratorSettings;
00064
00065         if (generatorParameters.AvailableToDrop > 0)
00066         {
00067             GeneratableElement candidate = fieldElement.GetCandidateToGenerate();
00068
00069             if (candidate == null)
00070             {
00071                 SendGeneratorBusy(fieldElement.InfoParameters.LogicPosition);
00072                 return;
00073             }
00074
00075             IFieldElement generatedElement =
00076                 _gameFieldService.CreateNewElement(candidate.Element, freePoint.Value,
00077                 generatorSettings.GenerateBlocked);
00078
00079             var position = fieldElement.InfoParameters.LogicPosition;
00080
00081             var generateResult = fieldElement.TryGenerate();
00082             if (generateResult.HasValue)
```

```

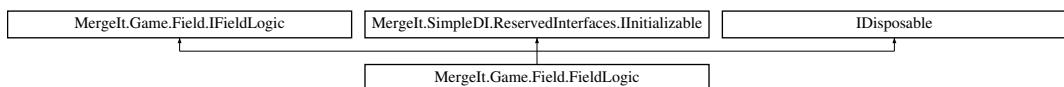
00081         {
00082             if (!generateResult.Value)
00083             {
00084                 SendRemoveElement(position);
00085             }
00086             else
00087             {
00088                 SendCheckGenerator(position);
00089             }
00090         }
00091     else
00092     {
00093         return;
00094     }
00095
00096     _userServiceModel.Energy--;
00097
00098     SendCreateElement(generatedElement, position, freePoint.Value);
00099     SaveService.Save(GameSaveType.Field);
00100 }
00101 else
00102 {
00103     SendGeneratorBusy(fieldElement.InfoParameters.LogicPosition);
00104 }
00105 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Actions/FieldGenerationProcessor.cs

6.112 Mergelt.Game.Field.FieldLogic Class Reference

Inheritance diagram for Mergelt.Game.Field.FieldLogic:



Public Member Functions

- void [Initialize\(\)](#)
- void [Dispose\(\)](#)

6.112.1 Detailed Description

Definition at line 12 of file [FieldLogic.cs](#).

6.112.2 Member Function Documentation

6.112.2.1 Dispose()

```
void MergeIt.Game.Field.FieldLogic.Dispose ( ) [inline]
```

Definition at line 29 of file [FieldLogic.cs](#).

```

00030     {
00031         _messageBus.RemoveListener<CreateElementMessage>(OnCreateElementMessageHandler);
00032         _messageBus.RemoveListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00033         _messageBus.RemoveListener<SwapElementsMessage>(OnSwapElementsMessageHandler);
00034         _messageBus.RemoveListener<SplitElementMessage>(OnSplitElementMessageHandler);
00035         _messageBus.RemoveListener<MergeElementsMessage>(OnMergeElementMessageHandler);
00036     }
```

6.112.2.2 Initialize()

```
void MergeIt.Game.Field.FieldLogic.Initialize ( ) [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 20 of file [FieldLogic.cs](#).

```
00021     {
00022         _messageBus.AddListener<CreateElementMessage>(OnCreateElementMessageHandler);
00023         _messageBus.AddListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00024         _messageBus.AddListener<SwapElementsMessage>(OnSwapElementsMessageHandler);
00025         _messageBus.AddListener<SplitElementMessage>(OnSplitElementMessageHandler);
00026         _messageBus.AddListener<MergeElementsMessage>(OnMergeElementMessageHandler);
00027     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/FieldLogic.cs

6.113 Mergelt.Game.Field.FieldLogicModel Class Reference

Public Member Functions

- [Vector3 GetPosition \(GridPoint fromPoint\)](#)

Properties

- [Dictionary< GridPoint, FieldCellComponent > CellComponents = new\(\) \[get\]](#)
- [Dictionary< GridPoint, IFieldElement > FieldElements = new\(\) \[get\]](#)
- [IFieldElement OpeningGenerator \[get, set\]](#)
- [List< IFieldElement > AllGenerators = new\(\) \[get, set\]](#)
- [int FieldWidth \[get, set\]](#)
- [int FieldHeight \[get, set\]](#)
- [float CellSize \[get, set\]](#)
- [Rect FieldRect \[get, set\]](#)
- [float PositionOffset \[get, set\]](#)
- [float CellSpacing \[get, set\]](#)

6.113.1 Detailed Description

Definition at line 9 of file [FieldLogicModel.cs](#).

6.113.2 Member Function Documentation

6.113.2.1 GetPosition()

```
Vector3 MergeIt.Game.Field.FieldLogicModel.GetPosition (
    GridPoint fromPoint ) [inline]
```

Definition at line 22 of file [FieldLogicModel.cs](#).

```
00023     {
00024         if (CellComponents.TryGetValue(fromPoint, out FieldCellComponent cell))
00025         {
00026             return cell.ObjectContainer.position;
00027         }
00028         return Vector3.zero;
00030     }
```

6.113.3 Property Documentation

6.113.3.1 AllGenerators

```
List<IFieldElement> MergeIt.Game.Field.FieldLogicModel.AllGenerators = new() [get], [set]
```

Definition at line 14 of file [FieldLogicModel.cs](#).

```
00014 { get; set; } = new();
```

6.113.3.2 CellComponents

```
Dictionary<GridPoint, FieldCellComponent> MergeIt.Game.Field.FieldLogicModel.CellComponents = new() [get]
```

Definition at line 11 of file [FieldLogicModel.cs](#).

```
00011 { get; } = new();
```

6.113.3.3 CellSize

```
float MergeIt.Game.Field.FieldLogicModel.CellSize [get], [set]
```

Definition at line 17 of file [FieldLogicModel.cs](#).

```
00017 { get; set; }
```

6.113.3.4 CellSpacing

```
float MergeIt.Game.Field.FieldLogicModel.CellSpacing [get], [set]
```

Definition at line 20 of file [FieldLogicModel.cs](#).

```
00020 { get; set; }
```

6.113.3.5 FieldElements

```
Dictionary<GridPoint, IFIELDElement> MergeIt.Game.Field.FieldLogicModel.FieldElements = new()  
[get]
```

Definition at line 12 of file [FieldLogicModel.cs](#).

```
00012 { get; } = new();
```

6.113.3.6 FieldHeight

```
int MergeIt.Game.Field.FieldLogicModel.FieldHeight [get], [set]
```

Definition at line 16 of file [FieldLogicModel.cs](#).

```
00016 { get; set; }
```

6.113.3.7 FieldRect

```
Rect MergeIt.Game.Field.FieldLogicModel.FieldRect [get], [set]
```

Definition at line 18 of file [FieldLogicModel.cs](#).

```
00018 { get; set; }
```

6.113.3.8 FieldWidth

```
int MergeIt.Game.Field.FieldLogicModel.FieldWidth [get], [set]
```

Definition at line 15 of file [FieldLogicModel.cs](#).

```
00015 { get; set; }
```

6.113.3.9 OpeningGenerator

```
IFieldElement MergeIt.Game.Field.FieldLogicModel.OpeningGenerator [get], [set]
```

Definition at line 13 of file [FieldLogicModel.cs](#).

```
00013 { get; set; }
```

6.113.3.10 PositionOffset

```
float MergeIt.Game.Field.FieldLogicModel.PositionOffset [get], [set]
```

Definition at line 19 of file [FieldLogicModel.cs](#).

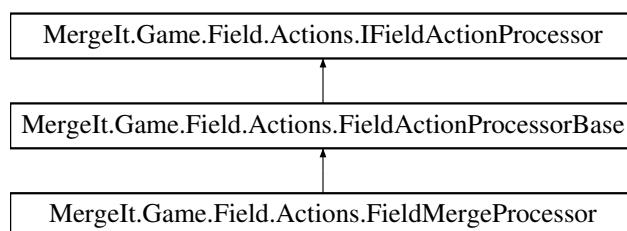
```
00019 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/FieldLogicModel.cs

6.114 Mergelt.Game.Field.Actions.FieldMergeProcessor Class Reference

Inheritance diagram for Mergelt.Game.Field.Actions.FieldMergeProcessor:



Public Member Functions

- override void [ProcessClick \(FieldCellComponent cellComponent\)](#)
- override void [ProcessEndDrag \(GridPoint fromPoint, GameObject toGameObject\)](#)

Additional Inherited Members

Protected Attributes inherited from Mergelt.Game.Field.Actions.FieldActionProcessorBase

- readonly IMessageBus MessageBus = DiContainer.Get<IMessageBus>()
- readonly IGameSaveService SaveService = DiContainer.Get<IGameSaveService>()
- readonly FieldLogicModel FieldLogicModel = DiContainer.Get<FieldLogicModel>()

6.114.1 Detailed Description

Definition at line 16 of file [FieldMergeProcessor.cs](#).

6.114.2 Member Function Documentation

6.114.2.1 ProcessClick()

```
override void MergeIt.Game.Field.Actions.FieldMergeProcessor.ProcessClick (
    FieldCellComponent cellComponent) [inline], [virtual]
```

Reimplemented from [Mergelt.Game.Field.Actions.FieldActionProcessorBase](#).

Definition at line 22 of file [FieldMergeProcessor.cs](#).

```
00023     {
00024         base.ProcessClick(cellComponent);
00025
00026         var presenter = cellComponent.FieldElementPresenter;
00027         if (presenter != null)
00028         {
00029             presenter.GetModel().ClicksCount++;
00030
00031             if (!presenter.GetModel().Selected)
00032             {
00033                 presenter.Select(true);
00034             }
00035         }
00036     }
```

6.114.2.2 ProcessEndDrag()

```
override void MergeIt.Game.Field.Actions.FieldMergeProcessor.ProcessEndDrag (
    GridPoint fromPoint,
    GameObject toGameObject) [inline], [virtual]
```

Reimplemented from [Mergelt.Game.Field.Actions.FieldActionProcessorBase](#).

Definition at line 38 of file [FieldMergeProcessor.cs](#).

```
00039     {
00040         base.ProcessEndDrag(fromPoint, toGameObject);
00041
00042         GridPoint toPoint = GridPoint.Default;
00043
00044         if (toGameObject)
00045         {
00046             if (toGameObject.TryGetComponent(out FieldCellComponent toCell))
00047             {
00048                 toPoint = toCell.Point;
00049
00050                 if (fromPoint.Equals(toPoint))
00051                 {
00052                     SendResetPositionMessage(fromPoint);
```

```

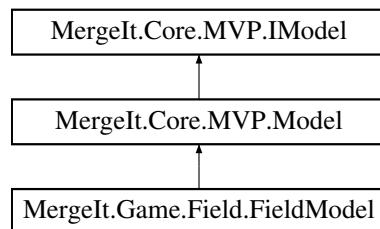
00053                     return;
00054     }
00055 }
00056
00057     IFieldElementPresenter toPresenter = toCell.FieldElementPresenter;
00058
00059     if (toPresenter != null)
00060     {
00061         var mergedResult = TryMerge(fromPoint, toPoint);
00062         if (mergedResult != null)
00063         {
00064             SendRemoveElement(fromPoint);
00065             SendRemoveElement(toPoint);
00066             SendMergeMessage(fromPoint, mergedResult);
00067
00068             SaveService.Save(GameSaveType.Field);
00069         }
00070         else if (!toPresenter.IsLocked)
00071         {
00072             SendSwapMessage(fromPoint, toPoint);
00073             SaveService.Save(GameSaveType.Field);
00074         }
00075         else
00076         {
00077             SendResetPositionMessage(fromPoint);
00078         }
00079     }
00080     else
00081     {
00082         SendSwapMessage(fromPoint, toPoint);
00083         SaveService.Save(GameSaveType.Field);
00084     }
00085 }
00086 else
00087 {
00088     SendResetPositionMessage(fromPoint);
00089 }
00090 }
00091 else
00092 {
00093     SendResetPositionMessage(fromPoint);
00094 }
00095 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Actions/FieldMergeProcessor.cs

6.115 Mergelt.Game.Field.FieldModel Class Reference

Inheritance diagram for Mergelt.Game.Field.FieldModel:



6.115.1 Detailed Description

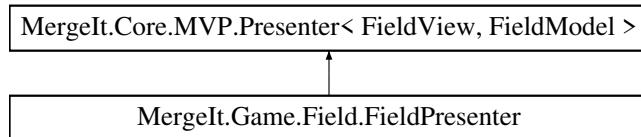
Definition at line 7 of file [FieldModel.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/FieldModel.cs

6.116 Mergelt.Game.Field.FieldPresenter Class Reference

Inheritance diagram for Mergelt.Game.Field.FieldPresenter:



Public Member Functions

- void [Initialize \(\)](#)

Public Member Functions inherited from

[Mergelt.Core.MVP.Presenter< FieldView, FieldModel >](#)

- void [Initialize \(IView view\)](#)
- void [Dispose \(\)](#)

Protected Member Functions

- override void [OnDispose \(\)](#)

Protected Member Functions inherited from

[Mergelt.Core.MVP.Presenter< FieldView, FieldModel >](#)

- virtual void [OnInitialize \(TView view\)](#)

Additional Inherited Members

Properties inherited from [Mergelt.Core.MVP.Presenter< FieldView, FieldModel >](#)

- TView [View](#) [get]
- TModel [Model](#) [get]

6.116.1 Detailed Description

Definition at line 19 of file [FieldPresenter.cs](#).

6.116.2 Constructor & Destructor Documentation

6.116.2.1 FieldPresenter()

```
MergeIt.Game.Field.FieldPresenter () [inline]
```

Definition at line 30 of file [FieldPresenter.cs](#).

```
00031     {
00032         _fieldLogicModel = DiContainer.Get<FieldLogicModel>();
00033         _fieldFactory = DiContainer.Get<IFieldFactory>();
00034         _fieldElementVisualFactory = DiContainer.Get<IFieldElementVisualFactory>();
00035         _messageBus = DiContainer.Get<IMessageBus>();
00036         _saveService = DiContainer.Get<IGameSaveService>();
00037         _effectsFactory = DiContainer.Get<IEffectsFactory>();
00038     }
```

6.116.3 Member Function Documentation

6.116.3.1 Initialize()

```
void MergeIt.Game.Field.FieldPresenter.Initialize () [inline]
```

Definition at line 53 of file [FieldPresenter.cs](#).

```
00054     {
00055         _messageBus.AddListener<SelectedElementMessage>(OnSelectedElementMessageHandler);
00056         _messageBus.AddListener<ResetSelectionMessage>(OnResetSelectionMessageHandler);
00057         _messageBus.AddListener<MergeElementsMessage>(OnMergeElementsMessageHandler);
00058         _messageBus.AddListener<SplitElementMessage>(OnSplitResultMessageHandler);
00059         _messageBus.AddListener<UnlockElementMessage>(OnUnlockElementMessageHandler);
00060         _messageBus.AddListener<ResetPositionMessage>(OnResetPositionsMessageHandler);
00061         _messageBus.AddListener<SwapElementsMessage>(OnSwapElementsMessageHandler);
00062         _messageBus.AddListener<CreateElementMessage>(OnCreateElementMessageHandler);
00063         _messageBus.AddListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00064
00065         _cellComponents = _fieldLogicModel.CellComponents;
00066
00067         var fieldRect = View.CellsContainer.rect;
00068         var fieldRectSize = fieldRect.size;
00069
00070         _fieldLogicModel.FieldRect = fieldRect;
00071
00072         int fieldWidth = _fieldLogicModel.FieldWidth;
00073         int fieldHeight = _fieldLogicModel.FieldHeight;
00074
00075         float sizeByWidth = (fieldRectSize.x - (fieldWidth - 1) * View.CellSpacing) / fieldWidth;
00076         float sizeByHeight = (fieldRectSize.y - (fieldHeight - 1) * View.CellSpacing) /
00077             fieldHeight;
00078         float cellSize = Mathf.Min(sizeByHeight, sizeByWidth);
00079
00080         bool horizontalAlign = Mathf.Approximately(cellSize, sizeByHeight);
00081         float offset = 0f;
00082
00083         if (horizontalAlign)
00084         {
00085             offset = (fieldRectSize.x - ((fieldWidth - 1) * View.CellSpacing + cellSize *
00086                 fieldWidth)) / 2f;
00087         }
00088         else
00089         {
00090             offset = (fieldRectSize.y - ((fieldHeight - 1) * View.CellSpacing + cellSize *
00091                 fieldHeight)) / 2f;
00092         }
00093
00094         _fieldLogicModel.CellSize = cellSize;
00095         _fieldLogicModel.PositionOffset = offset;
00096         _fieldLogicModel.CellSpacing = View.CellSpacing;
00097
00098         for (int i = 0; i < fieldHeight; i++)
00099         {
00100             for (int j = 0; j < fieldWidth; j++)
00101             {
00102                 var point = GridPoint.Create(i, j);
00103                 var cell = _fieldFactory.CreateFieldCell(View.CellsContainer.transform, point);
00104                 cell.RectTransform.localScale = Vector3.one;
```

```

00103         cell.RectTransform.pivot = Vector2.zero;
00104         cell.RectTransform.sizeDelta = new Vector2(cellSize, cellSize);
00105         cell.RectTransform.anchoredPosition = horizontalAlign
00106             ? new Vector2(offset + j * (cellSize + View.CellSpacing), (fieldHeight - 1 -
00107             i) * (cellSize + View.CellSpacing))
00108             : new Vector2(j * (cellSize + View.CellSpacing), offset + i * (cellSize +
00109             View.CellSpacing));
00110
00111         cell.gameObject.name = $"FieldCell [{point.X}, {point.Y}]";
00112
00113         _cellComponents[point] = cell;
00114
00115         if (_fieldLogicModel.FieldElements.TryGetValue(point, out var fieldElement))
00116         {
00117             var fieldElementPresenter =
00118                 _fieldElementVisualFactory.CreateFieldElement(fieldElement);
00119             cell.AttachObject(fieldElementPresenter);
00120         }
00121     }
00122 }
```

6.116.3.2 OnDispose()

override void MergeIt.Game.Field.FieldPresenter.OnDispose () [inline], [protected], [virtual]

Reimplemented from [MergeIt.Core.MVP.Presenter< FieldView, FieldModel >](#).

Definition at line 40 of file [FieldPresenter.cs](#).

```

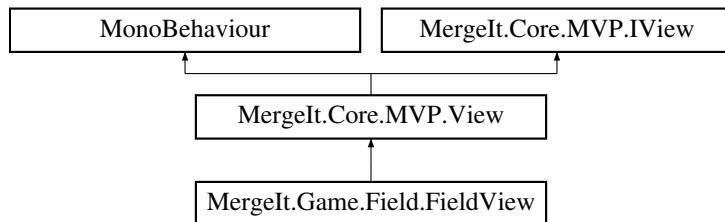
00041     {
00042         _messageBus.RemoveListener<SelectedElementMessage>(OnSelectedElementMessageHandler);
00043         _messageBus.RemoveListener<ResetSelectionMessage>(OnResetSelectionMessageHandler);
00044         _messageBus.RemoveListener<MergeElementsMessage>(OnMergeElementsMessageHandler);
00045         _messageBus.RemoveListener<UnlockElementMessage>(OnUnlockElementMessageHandler);
00046         _messageBus.RemoveListener<SplitElementMessage>(OnSplitResultMessageHandler);
00047         _messageBus.RemoveListener<ResetPositionMessage>(OnResetPositionsMessageHandler);
00048         _messageBus.RemoveListener<SwapElementsMessage>(OnSwapElementsMessageHandler);
00049         _messageBus.RemoveListener<CreateElementMessage>(OnCreateElementMessageHandler);
00050         _messageBus.RemoveListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00051     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Field/FieldPresenter.cs

6.117 MergeIt.Game.Field.FieldView Class Reference

Inheritance diagram for MergeIt.Game.Field.FieldView:



Properties

- RectTransform [CellsContainer](#) [get]
- float [CellSpacing](#) [get]

Properties inherited from [Mergelt.Core.MVP.View](#)

- `GameObject` [GameObject](#) [get]

Properties inherited from [Mergelt.Core.MVP.IView](#)

Additional Inherited Members

Public Member Functions inherited from [Mergelt.Core.MVP.View](#)

- `virtual void` [Initialize](#) ()

Protected Member Functions inherited from [Mergelt.Core.MVP.View](#)

- `virtual void` [OnDestroy](#) ()

Events inherited from [Mergelt.Core.MVP.View](#)

- `Action` [InitializeEvent](#)
- `Action` [DestroyEvent](#)

Events inherited from [Mergelt.Core.MVP.IView](#)

- `Action` [InitializeEvent](#)
- `Action` [DestroyEvent](#)

6.117.1 Detailed Description

Definition at line 8 of file [FieldView.cs](#).

6.117.2 Property Documentation

6.117.2.1 CellsContainer

`RectTransform` [MergeIt.Game.Field.FieldView.CellsContainer](#) [get]

Definition at line 16 of file [FieldView.cs](#).

6.117.2.2 CellSpacing

`float` [MergeIt.Game.Field.FieldView.CellSpacing](#) [get]

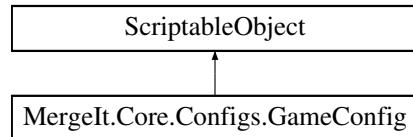
Definition at line 17 of file [FieldView.cs](#).

The documentation for this class was generated from the following file:

- `Assets/Mergelt/Source/Game/Field/FieldView.cs`

6.118 Mergelt.Core.Configs.GameConfig Class Reference

Inheritance diagram for Mergelt.Core.Configs.GameConfig:



Public Attributes

- int [EnergyCap](#)
- int [EnergyRestoreTime](#)
- int [InitialSoftCurrency](#)
- int [InitialHardCurrency](#)
- int [InitialSplittersCount](#)
- string [LevelString](#)
- float [SaveInterval](#) = 10

6.118.1 Detailed Description

Definition at line [8](#) of file [GameConfig.cs](#).

6.118.2 Member Data Documentation

6.118.2.1 EnergyCap

```
int MergeIt.Core.Configs.GameConfig.EnergyCap
```

Definition at line [11](#) of file [GameConfig.cs](#).

6.118.2.2 EnergyRestoreTime

```
int MergeIt.Core.Configs.GameConfig.EnergyRestoreTime
```

Definition at line [14](#) of file [GameConfig.cs](#).

6.118.2.3 InitialHardCurrency

```
int MergeIt.Core.Configs.GameConfig.InitialHardCurrency
```

Definition at line [16](#) of file [GameConfig.cs](#).

6.118.2.4 InitialSoftCurrency

```
int MergeIt.Core.Configs.GameConfig.InitialSoftCurrency
```

Definition at line 15 of file [GameConfig.cs](#).

6.118.2.5 InitialSplittersCount

```
int MergeIt.Core.Configs.GameConfig.InitialSplittersCount
```

Definition at line 17 of file [GameConfig.cs](#).

6.118.2.6 LevelString

```
string MergeIt.Core.Configs.GameConfig.LevelString
```

Definition at line 20 of file [GameConfig.cs](#).

6.118.2.7 SaveInterval

```
float MergeIt.Core.Configs.GameConfig.SaveInterval = 10
```

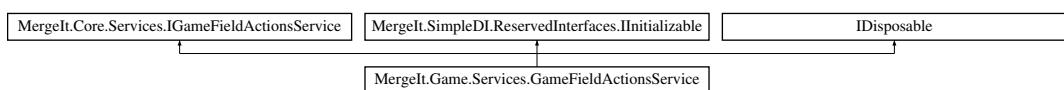
Definition at line 22 of file [GameConfig.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Configs/GameConfig.cs

6.119 Mergelt.Game.Services.GameFieldActionsService Class Reference

Inheritance diagram for Mergelt.Game.Services.GameFieldActionsService:



Public Member Functions

- void [Dispose \(\)](#)
- void [Initialize \(\)](#)

6.119.1 Detailed Description

Definition at line 14 of file [GameFieldActionsService.cs](#).

6.119.2 Member Function Documentation

6.119.2.1 Dispose()

```
void MergeIt.Game.Services.GameFieldActionsService.Dispose () [inline]
```

Definition at line 21 of file [GameFieldActionsService.cs](#).

```
00022     {
00023         _messageBus.RemoveListener<ClickElementMessage>(OnClickElementMessageHandler);
00024         _messageBus.RemoveListener<EndDragElementMessage>(OnEndDragElementMessageHandler);
00025     }
```

6.119.2.2 Initialize()

```
void MergeIt.Game.Services.GameFieldActionsService.Initialize () [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 27 of file [GameFieldActionsService.cs](#).

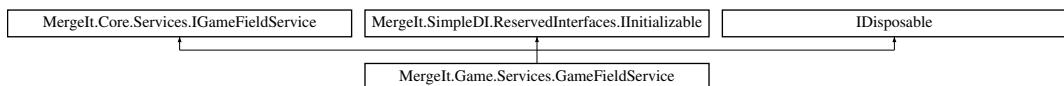
```
00028     {
00029         _messageBus.AddListener<ClickElementMessage>(OnClickElementMessageHandler);
00030         _messageBus.AddListener<EndDragElementMessage>(OnEndDragElementMessageHandler);
00031
00032         _fieldActionProcessors.Add(new FieldMergeProcessor());
00033         _fieldActionProcessors.Add(new FieldGenerationProcessor());
00034         _fieldActionProcessors.Add(new FieldConsumableProcessor());
00035     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/GameFieldActionsService.cs

6.120 Mergelt.Game.Services.GameFieldService Class Reference

Inheritance diagram for Mergelt.Game.Services.GameFieldService:



Public Member Functions

- void [Dispose \(\)](#)
- [GridPoint? GetFreeCell \(\)](#)
- [IFieldElement CreateNewElement \(ElementConfig config, GridPoint point, bool blocked=false\)](#)
- void [Initialize \(\)](#)

6.120.1 Detailed Description

Definition at line 18 of file [GameFieldService.cs](#).

6.120.2 Member Function Documentation

6.120.2.1 CreateNewElement()

```
IFieldElement MergeIt.Game.Services.GameFieldService.CreateNewElement (
    ElementConfig config,
    GridPoint point,
    bool blocked = false ) [inline]
```

Implements [Mergelt.Core.Services.IGameFieldService](#).

Definition at line 69 of file [GameFieldService.cs](#).

```
00070     {
00071         IFieldElement newElement =
00072             _fieldElementFactory.CreateFieldElement(config, point, blocked);
00073         return newElement;
00075     }
```

6.120.2.2 Dispose()

```
void MergeIt.Game.Services.GameFieldService.Dispose () [inline]
```

Definition at line 38 of file [GameFieldService.cs](#).

```
00039     {
00040         _messageBus.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00041     }
```

6.120.2.3 GetFreeCell()

```
GridPoint? MergeIt.Game.Services.GameFieldService.GetFreeCell () [inline]
```

Implements [Mergelt.Core.Services.IGameFieldService](#).

Definition at line 43 of file [GameFieldService.cs](#).

```
00044     {
00045         int fieldHeight = _fieldLogicModel.FieldHeight;
00046         int fieldWidth = _fieldLogicModel.FieldWidth;
00047
00048         var randomHeight = ListExtensions.GenerateShuffledArray(fieldHeight);
00049         var randomWidth = ListExtensions.GenerateShuffledArray(fieldWidth);
00050
00051         for (int i = 0; i < randomHeight.Count; i++)
00052         {
00053             for (int j = 0; j < randomWidth.Count; j++)
00054             {
00055                 int row = randomHeight[i];
00056                 int column = randomWidth[j];
00057
00058                 var point = GridPoint.Create(row, column);
00059                 if (!_fieldLogicModel.FieldElements.ContainsKey(point))
00060                 {
00061                     return point;
00062                 }
00063             }
00064         }
00065
00066         return null;
00067     }
```

6.120.2.4 Initialize()

```
void MergeIt.Game.Services.GameFieldService.Initialize ( ) [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 77 of file [GameFieldService.cs](#).

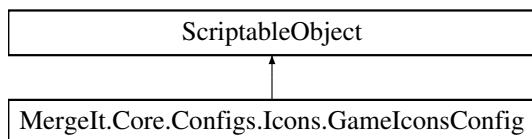
```
00078      {
00079          _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00080      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/GameFieldService.cs

6.121 Mergelt.Core.Configs(Icons).GameIconsConfig Class Reference

Inheritance diagram for Mergelt.Core.Configs(Icons).GameIconsConfig:



Properties

- [IconParameters\[\] InfoPanelCurrencyIcons](#) [get]

6.121.1 Detailed Description

Definition at line 8 of file [GameIconsConfig.cs](#).

6.121.2 Property Documentation

6.121.2.1 InfoPanelCurrencyIcons

```
IconParameters [ ] MergeIt.Core.Configs(Icons).GameIconsConfig.InfoPanelCurrencyIcons [get]
```

Definition at line 13 of file [GameIconsConfig.cs](#).

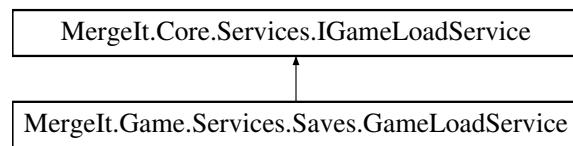
```
00014      {
00015          get => _infoPanelCurrencyIcons;
00016      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Configs/Icons/GameIconsConfig.cs

6.122 Mergelt.Game.Services.Saves.GameLoadService Class Reference

Inheritance diagram for Mergelt.Game.Services.Saves.GameLoadService:



Public Member Functions

- `T Load< T >()`

6.122.1 Detailed Description

Definition at line 11 of file [GameLoadService.cs](#).

6.122.2 Member Function Documentation

6.122.2.1 Load< T >()

`T MergeIt.Game.Services.Saves.GameLoadService.Load< T > ()` [inline]

Implements [Mergelt.Core.Services.IGameLoadService](#).

Type Constraints

T : class
T : ISavable

Definition at line 19 of file [GameLoadService.cs](#).

```

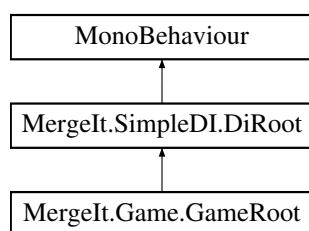
00019                                     : class, ISavable
00020     {
00021         var data = _serializeStrategy.Load<T>();
00022
00023         return data;
00024     }
  
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/Saves/GameLoadService.cs

6.123 Mergelt.Game.GameRoot Class Reference

Inheritance diagram for Mergelt.Game.GameRoot:



Protected Member Functions

- override void [OnInstall \(\)](#)
- override void [Run \(\)](#)

Protected Member Functions inherited from [Mergelt.SimpleDI.DiRoot](#)

- void [OnInstall \(\)](#)

6.123.1 Detailed Description

Definition at line 27 of file [GameRoot.cs](#).

6.123.2 Member Function Documentation

6.123.2.1 [OnInstall\(\)](#)

```
override void MergeIt.Game.GameRoot.OnInstall () [inline], [protected]
```

Definition at line 29 of file [GameRoot.cs](#).

```
00030     {
00031         DiContainer.RegisterInterfacesFor<MessageBus>().AsSingleton();
00032
00033         DiContainer.RegisterInterfacesFor<GameService>().AsSingleton();
00034         DiContainer.RegisterInterfacesFor<GameFieldService>().AsSingleton();
00035         DiContainer.RegisterInterfacesFor<GameFieldActionsService>().AsSingleton();
00036         DiContainer.RegisterInterfacesFor<GameSaveEveryIntervalService>().AsSingleton();
00037         DiContainer.RegisterInterfacesFor<UserService>().AsSingleton();
00038         DiContainer.RegisterInterfacesFor<UserProgressService>().AsSingleton();
00039         DiContainer.RegisterInterfacesFor<InventoryService>().AsSingleton();
00040         DiContainer.RegisterInterfacesFor<EnergyService>().AsSingleton();
00041         DiContainer.RegisterInterfacesFor<GeneratorService>().AsSingleton();
00042         DiContainer.RegisterInterfacesFor<InfoPanelService>().AsSingleton();
00043         DiContainer.RegisterInterfacesFor<CurrencyService>().AsSingleton();
00044         DiContainer.RegisterInterfacesFor<ElementService>().AsSingleton();
00045         DiContainer.RegisterInterfacesFor<ElementsStockService>().AsSingleton();
00046         DiContainer.RegisterInterfacesFor<EvolutionsService>().AsSingleton();
00047
00048         DiContainer.RegisterInterfacesFor<FieldLogic>().AsSingleton();
00049
00050         DiContainer.RegisterInterfacesFor<EffectsFactory>().AsSingleton();
00051         DiContainer.RegisterInterfacesFor<EffectsManager>().AsSingleton();
00052
00053         DiContainer.RegisterInterfacesFor<ConfigsService>().AsSingleton();
00054         DiContainer.RegisterInterfacesFor<ConfigProcessor>().AsSingleton();
00055         DiContainer.RegisterInterfacesFor<GameLoadService>().AsSingleton();
00056         DiContainer.RegisterInterfacesFor<ResourcesLoaderService>().AsSingleton();
00057
00058         DiContainer.RegisterInterfacesFor<FieldElementVisualFactory>().AsSingleton();
00059         DiContainer.RegisterInterfacesFor<FieldElementFactory>().AsSingleton();
00060         DiContainer.RegisterInterfacesFor<FieldFactory>().AsSingleton();
00061         DiContainer.RegisterInterfacesFor<InventoryFactory>().AsSingleton();
00062         DiContainer.RegisterInterfacesFor<ElementInfoFactory>().AsSingleton();
00063         DiContainer.RegisterInterfacesFor<IconFactory>().AsSingleton();
00064
00065         DiContainer.RegisterInterfacesFor<HintsManager>().AsSingleton();
00066
00067         DiContainer.Register<FieldLogicModel>().AsSingleton();
00068         DiContainer.Register<GameServiceModel>().AsSingleton();
00069         DiContainer.Register<UserServiceModel>().AsSingleton();
00070         DiContainer.Register<InventoryServiceModel>().AsSingleton();
00071         DiContainer.Register<ElementsStockServiceModel>().AsSingleton();
00072
00073         DiContainer.RegisterInterfacesFor<WindowsSystem>().AsSingleton();
00074         DiContainer.RegisterInterfacesFor<WindowFactory>().AsSingleton();
00075
00076         DiContainer.RegisterInterfacesFor<JsonSerializeStrategy>().AsSingleton();
00077 }
```

6.123.2.2 Run()

```
override void MergeIt.Game.GameRoot.Run ( ) [inline], [protected], [virtual]
```

Reimplemented from [Mergelt.SimpleDI.DiRoot](#).

Definition at line 79 of file [GameRoot.cs](#).

```
00080      {
00081          RegisterWindows ();
00082
00083          var messageBus = DiContainer.Get<IMessageBus> ();
00084          messageBus.Fire<StartGameMessage> ();
00085      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/GameRoot.cs

6.124 Mergelt.Game.Services.Saves.GameSaveEveryIntervalService Class Reference

Inheritance diagram for Mergelt.Game.Services.Saves.GameSaveEveryIntervalService:



Public Member Functions

- `async UniTask Save (GameSaveType gameSaveType)`
- `void Initialize ()`
- `void Dispose ()`
- `async void OnApplicationQuit ()`
- `async void Update ()`

6.124.1 Detailed Description

Definition at line 27 of file [GameSaveEveryIntervalService.cs](#).

6.124.2 Member Function Documentation

6.124.2.1 Dispose()

```
void MergeIt.Game.Services.Saves.GameSaveEveryIntervalService.Dispose ( ) [inline]
```

Definition at line 77 of file [GameSaveEveryIntervalService.cs](#).

```
00078      {
00079          _messageBus.RemoveListener<LoadedGameMessage> (OnLoadedGameMessageHandler);
00080          MonoEventsListener.Instance.UnsubscribeFromApplicationQuit (this);
00081      }
```

6.124.2.2 Initialize()

```
void MergeIt.Game.Services.Saves.GameSaveEveryIntervalService.Initialize ( ) [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 71 of file [GameSaveEveryIntervalService.cs](#).

```
00072     {
00073         _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00074         MonoEventsListener.Instance.SubscribeOnApplicationQuit(this);
00075     }
```

6.124.2.3 OnApplicationQuit()

```
async void MergeIt.Game.Services.Saves.GameSaveEveryIntervalService.OnApplicationQuit ( ) [inline]
```

Implements [Mergelt.Core.Helpers.IMonoApplicationQuitHandler](#).

Definition at line 83 of file [GameSaveEveryIntervalService.cs](#).

```
00084     {
00085         await TrySave(GameSaveType.All);
00086     }
```

6.124.2.4 Save()

```
async UniTask MergeIt.Game.Services.Saves.GameSaveEveryIntervalService.Save (
    GameSaveType gameSaveType ) [inline]
```

Implements [Mergelt.Core.Services.IGameSaveService](#).

Definition at line 64 of file [GameSaveEveryIntervalService.cs](#).

```
00065     {
00066         _saveQueue.Add(gameSaveType);
00067         await UniTask.Yield();
00068     }
```

6.124.2.5 Update()

```
async void MergeIt.Game.Services.Saves.GameSaveEveryIntervalService.Update ( ) [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IUpdatable](#).

Definition at line 88 of file [GameSaveEveryIntervalService.cs](#).

```
00089     {
00090         if (_canSave)
00091         {
00092             _time += Time.unscaledDeltaTime;
00093             if (_time >= _interval)
00094             {
00095                 _canSave = false;
00096                 await TrySave();
00097                 _time = 0f;
00098                 _canSave = true;
00099             }
00100         }
00101     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/Saves/GameSaveEveryIntervalService.cs

6.125 Mergelt.Game.Services.Saves.GameSaveOnEveryActionService Class Reference

Inheritance diagram for Mergelt.Game.Services.Saves.GameSaveOnEveryActionService:



Public Member Functions

- `async UniTask Save (GameSaveType gameSaveType)`
- `void Initialize ()`
- `void Dispose ()`
- `async void OnApplicationQuit ()`

6.125.1 Detailed Description

Definition at line 22 of file [GameSaveOnEveryActionService.cs](#).

6.125.2 Member Function Documentation

6.125.2.1 Dispose()

```
void MergeIt.Game.Services.Saves.GameSaveOnEveryActionService.Dispose ( ) [inline]
```

Definition at line 83 of file [GameSaveOnEveryActionService.cs](#).

```
00084     {
00085         MonoEventsListener.Instance.UnsubscribeFromApplicationQuit (this);
00086     }
```

6.125.2.2 Initialize()

```
void MergeIt.Game.Services.Saves.GameSaveOnEveryActionService.Initialize ( ) [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 78 of file [GameSaveOnEveryActionService.cs](#).

```
00079     {
00080         MonoEventsListener.Instance.SubscribeOnApplicationQuit (this);
00081     }
```

6.125.2.3 OnApplicationQuit()

```
async void MergeIt.Game.Services.Saves.GameSaveOnEveryActionService.OnApplicationQuit ( ) [inline]
```

Implements [Mergelt.Core.Helpers.IMonoApplicationQuitHandler](#).

Definition at line 88 of file [GameSaveOnEveryActionService.cs](#).

```
00089     {
00090         await Save (GameSaveType.All);
00091     }
```

6.125.2.4 Save()

```
async UniTask MergeIt.Game.Services.Saves.GameSaveOnEveryActionService.Save (
    GameSaveType gameSaveType ) [inline]
```

Implements [Mergelt.Core.Services.IGameSaveService](#).

Definition at line 50 of file [GameSaveOnEveryActionService.cs](#).

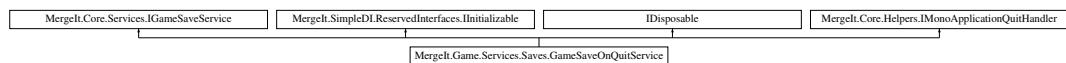
```
00051     {
00052         if ((gameSaveType & GameSaveType.Field) == GameSaveType.Field)
00053     {
00054         await SaveLevel();
00055     }
00056
00057     if ((gameSaveType & GameSaveType.User) == GameSaveType.User)
00058     {
00059         await SaveUser();
00060     }
00061
00062     if ((gameSaveType & GameSaveType.Inventory) == GameSaveType.Inventory)
00063     {
00064         await SaveInventory();
00065     }
00066
00067     if ((gameSaveType & GameSaveType.Stock) == GameSaveType.Stock)
00068     {
00069         await SaveStock();
00070     }
00071
00072     if ((gameSaveType & GameSaveType.EvolutionsProgress) == GameSaveType.EvolutionsProgress)
00073     {
00074         await SaveEvolutionsProgress();
00075     }
00076 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/Saves/GameSaveOnEveryActionService.cs

6.126 Mergelt.Game.Services.Saves.GameSaveOnQuitService Class Reference

Inheritance diagram for Mergelt.Game.Services.Saves.GameSaveOnQuitService:



Public Member Functions

- void [Dispose \(\)](#)
- async UniTask [Save \(GameSaveType gameSaveType\)](#)
- void [Initialize \(\)](#)
- async void [OnApplicationQuit \(\)](#)

6.126.1 Detailed Description

Definition at line 23 of file [GameSaveOnQuitService.cs](#).

6.126.2 Member Function Documentation

6.126.2.1 Dispose()

```
void MergeIt.Game.Services.Saves.GameSaveOnQuitService.Dispose () [inline]
```

Definition at line 49 of file [GameSaveOnQuitService.cs](#).

```
00050     {
00051         MonoEventsListener.Instance.UnsubscribeFromApplicationQuit(this);
00052     }
```

6.126.2.2 Initialize()

```
void MergeIt.Game.Services.Saves.GameSaveOnQuitService.Initialize () [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 74 of file [GameSaveOnQuitService.cs](#).

```
00075     {
00076         MonoEventsListener.Instance.SubscribeOnApplicationQuit(this);
00077     }
```

6.126.2.3 OnApplicationQuit()

```
async void MergeIt.Game.Services.Saves.GameSaveOnQuitService.OnApplicationQuit () [inline]
```

Implements [Mergelt.Core.Helpers.IMonoApplicationQuitHandler](#).

Definition at line 79 of file [GameSaveOnQuitService.cs](#).

```
00080     {
00081         await Save(GameSaveType.All);
00082     }
```

6.126.2.4 Save()

```
async UniTask MergeIt.Game.Services.Saves.GameSaveOnQuitService.Save (
    GameSaveType gameSaveType) [inline]
```

Implements [Mergelt.Core.Services.IGameSaveService](#).

Definition at line 54 of file [GameSaveOnQuitService.cs](#).

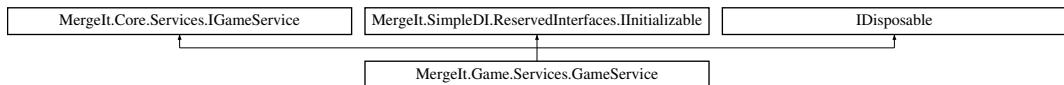
```
00055     {
00056         if (gameSaveType != GameSaveType.All)
00057         {
00058             Debug.Log("This save strategy works only on application quit.");
00059             return;
00060         }
00061     }
00062     await SaveLevel();
00063     await SaveUser();
00064     await SaveInventory();
00065     await SaveStock();
00066     await SaveEvolutionsProgress();
00067 }
00068 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/Saves/GameSaveOnQuitService.cs

6.127 Mergelt.Game.Services.GameService Class Reference

Inheritance diagram for Mergelt.Game.Services.GameService:



Public Member Functions

- void [Dispose \(\)](#)
- void [Initialize \(\)](#)

Public Attributes

- [IMessageBus _messageBus](#)

6.127.1 Detailed Description

Definition at line 16 of file [GameService.cs](#).

6.127.2 Member Function Documentation

6.127.2.1 Dispose()

```
void MergeIt.Game.Services.GameService.Dispose ( ) [inline]
```

Definition at line 39 of file [GameService.cs](#).

```
00040     {
00041         _messageBus.RemoveListener<StartGameMessage>(StartGameMessageHandler);
00042     }
```

6.127.2.2 Initialize()

```
void MergeIt.Game.Services.GameService.Initialize ( ) [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 44 of file [GameService.cs](#).

```
00045     {
00046         _messageBus.AddListener<StartGameMessage>(StartGameMessageHandler);
00047     }
```

6.127.3 Member Data Documentation

6.127.3.1 _messageBus

```
IMessageBus MergeIt.Game.Services.GameService._messageBus
```

Definition at line 34 of file [GameService.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/GameService.cs

6.128 Mergelt.Game.Services.GameServiceModel Class Reference

Properties

- FieldData [LoadedLevel](#) [get, set]
- Camera [MainCamera](#) [get]
- Canvas [MainCanvas](#) [get]

6.128.1 Detailed Description

Definition at line 8 of file [GameServiceModel.cs](#).

6.128.2 Property Documentation

6.128.2.1 LoadedLevel

```
FieldData MergeIt.Game.Services.GameServiceModel.LoadedLevel [get], [set]
```

Definition at line 13 of file [GameServiceModel.cs](#).

```
00013 { get; set; }
```

6.128.2.2 MainCamera

```
Camera MergeIt.Game.Services.GameServiceModel.MainCamera [get]
```

Definition at line 15 of file [GameServiceModel.cs](#).

```
00016     {
00017         get
00018     {
00019         if (!_mainCamera)
00020         {
00021             _mainCamera = Camera.main;
00022         }
00023         return _mainCamera;
00024     }
00025 }
00026 }
```

6.128.2.3 MainCanvas

```
Canvas MergeIt.Game.Services.GameServiceModel.MainCanvas [get]
```

Definition at line 28 of file [GameServiceModel.cs](#).

```
00029         {
00030             get
00031             {
00032                 if (!_mainCanvas)
00033                 {
00034                     _mainCanvas =
00035                         GameObject.FindGameObjectWithTag("MainCanvas").GetComponent<Canvas>();
00036                 }
00037                 return _mainCanvas;
00038             }
00039         }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Services/GameServiceModel.cs

6.129 MergeIt.Core.Configs.Elements.GeneratableElement Class Reference

Public Attributes

- [ElementConfig](#) Element
- int [Possibility](#)

6.129.1 Detailed Description

Definition at line 8 of file [GeneratableElement.cs](#).

6.129.2 Member Data Documentation

6.129.2.1 Element

```
ElementConfig MergeIt.Core.Configs.Elements.GeneratableElement.Element
```

Definition at line 10 of file [GeneratableElement.cs](#).

6.129.2.2 Possibility

```
int MergeIt.Core.Configs.Elements.GeneratableElement.Possibility
```

Definition at line 11 of file [GeneratableElement.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Configs/Elements/Settings/GeneratableElement.cs

6.130 Mergelt.Core.FieldElements.GeneratableFieldElement Class Reference

Properties

- `ElementConfig Config [get, set]`
- `int Possibility [get, set]`

6.130.1 Detailed Description

Definition at line 7 of file [GeneratableFieldElement.cs](#).

6.130.2 Property Documentation

6.130.2.1 Config

`ElementConfig MergeIt.Core.FieldElements.GeneratableFieldElement.Config [get], [set]`

Definition at line 9 of file [GeneratableFieldElement.cs](#).
00009 { `get; set;` }

6.130.2.2 Possibility

`int MergeIt.Core.FieldElements.GeneratableFieldElement.Possibility [get], [set]`

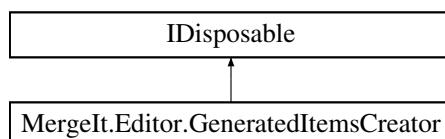
Definition at line 10 of file [GeneratableFieldElement.cs](#).
00010 { `get; set;` }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/GeneratableFieldElement.cs

6.131 Mergelt.Editor.GeneratedItemsCreator Class Reference

Inheritance diagram for Mergelt.Editor.GeneratedItemsCreator:



Public Member Functions

- `GeneratedItemsCreator (VisualElement genItemsControls, VisualElement genItemsContainer, ElementConfig config, List< EvolutionData > evolutions, List< ElementConfig > configs)`
- `void Generate ()`
- `void Clear ()`
- `void Dispose ()`

6.131.1 Detailed Description

Definition at line 15 of file [GeneratedItemsCreator.cs](#).

6.131.2 Constructor & Destructor Documentation

6.131.2.1 GeneratedItemsCreator()

```
MergeIt.Editor.GeneratedItemsCreator.GeneratedItemsCreator (
    VisualElement genItemsControls,
    VisualElement genItemsContainer,
    ElementConfig config,
    List< EvolutionData > evolutions,
    List< ElementConfig > configs ) [inline]
```

Definition at line 30 of file [GeneratedItemsCreator.cs](#).

```
00032     {
00033         _config = config;
00034         _evolutions = evolutions;
00035         _configs = configs;
00036         _itemsContainer = genItemsContainer;
00037         _settings = config.GeneratorSettings;
00038         _generatedItemUis = new List<GeneratedItemUI>();
00039
00040         _addButton = genItemsControls.Q<Button>("CreateGenerateItem");
00041         _addButton.clicked += OnClickAddNewGeneratedItem;
00042
00043         _clearAllButton = genItemsControls.Q<Button>("ClearGenerateItem");
00044         _clearAllButton.clicked += OnClickClearAllGeneratedItems;
00045
00046         _clickAddNewGeneratedItem += CreateNewGeneratedElementUI;
00047         _clickRemoveGeneratedItem += RemoveGeneratedItemUI;
00048         _clickClearAllGeneratedItems += ClearAllGeneratedItems;
00049
00050         if (_configs.Count == 0)
00051     {
00052         var helpBox = new HelpBox("There are no saved evolutions.",
00053             HelpBoxMessageType.Warning);
00054
00055         _itemsContainer.Add(helpBox);
00056
00057         _addButton.style.display = StylesConstants.DisplayNone;
00058         _clearAllButton.style.display = StylesConstants.DisplayNone;
00059     }
00060 }
```

6.131.3 Member Function Documentation

6.131.3.1 Clear()

```
void MergeIt.Editor.GeneratedItemsCreator.Clear () [inline]
```

Definition at line 170 of file [GeneratedItemsCreator.cs](#).

```
00171     {
00172         ClearVisualItems();
00173
00174         _clearAllButton.clicked -= OnClickClearAllGeneratedItems;
00175         _addButton.clicked -= OnClickAddNewGeneratedItem;
00176
00177         _clickAddNewGeneratedItem -= CreateNewGeneratedElementUI;
00178         _clickRemoveGeneratedItem -= RemoveGeneratedItemUI;
00179         _clickClearAllGeneratedItems -= ClearAllGeneratedItems;
00180     }
```

6.131.3.2 Dispose()

```
void MergeIt.Editor.GeneratedItemsCreator.Dispose ( ) [inline]
```

Definition at line 182 of file [GeneratedItemsCreator.cs](#).

```
00183     {
00184         Clear ();
00185     }
```

6.131.3.3 Generate()

```
void MergeIt.Editor.GeneratedItemsCreator.Generate ( ) [inline]
```

Definition at line 62 of file [GeneratedItemsCreator.cs](#).

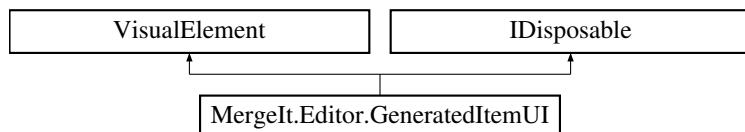
```
00063     {
00064         var generateItems = _settings.GenerateItems;
00065         if (generateItems?.Count > 0)
00066         {
00067             for (int i = 0; i < generateItems.Count; i++)
00068             {
00069                 CreateGeneratedElementUI (i, generateItems[i]);
00070             }
00071         }
00072     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/GeneratedItemsCreator.cs

6.132 Mergelt.Editor.GeneratedItemUI Class Reference

Inheritance diagram for Mergelt.Editor.GeneratedItemUI:



Public Member Functions

- `GeneratedItemUI (GeneratableElement item, ElementConfig currentConfig, int index, List< EvolutionData > evolutions, List< ElementConfig > configs)`
- `void Dispose ()`

Properties

- `Action< int, GeneratableElement > RemoveGeneratedItemEvent [get, set]`
- `int Index [get, set]`

6.132.1 Detailed Description

Definition at line 16 of file [GeneratedItemUI.cs](#).

6.132.2 Constructor & Destructor Documentation

6.132.2.1 GeneratedItemUI()

```
MergeIt.Editor.GeneratedItemUI.GeneratedItemUI (
    GeneratableElement item,
    ElementConfig currentConfig,
    int index,
    List< EvolutionData > evolutions,
    List< ElementConfig > configs ) [inline]
```

Definition at line 41 of file [GeneratedItemUI.cs](#).

```
00046     {
00047         _currentConfig = currentConfig;
00048         _allConfigs = configs;
00049         _evolutions = evolutions;
00050
00051         _target = item;
00052         _index = index;
00053
00054         var template =
00055             (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.InspectorResourcesRoot,
00056                         "GeneratedItemTemplate.uxml"));
00057
00058         style.flexDirection = new StyleEnum<FlexDirection>(FlexDirection.Row);
00059         style.paddingTop = new StyleLength(5);
00060
00061         template.CloneTree(this);
00062
00063         if (item.Element == null && !_allConfigs.Contains(_selectedElement))
00064         {
00065             _selectedElement = _allConfigs.FirstOrDefault(x => x != _currentConfig);
00066             _target.Element = _selectedElement;
00067         }
00068         else
00069         {
00070             _selectedElement = item.Element;
00071         }
00072
00073         var container = this.Q<VisualElement>("ElementsPopup");
00074         if (_selectedElement != null)
00075         {
00076             _elementsConfigPopup = new PopupField<ElementConfig>("", _allConfigs,
00077                         _selectedElement);
00078
00079             _elementsConfigPopup.formatListItemCallback = FormatElementCallback;
00080             _elementsConfigPopup.formatSelectedValueCallback = FormatElementCallback;
00081
00082             _elementsConfigPopup.AddToClassList("genItemList");
00083             _elementsConfigPopup.RegisterValueChangedCallback(OnElementChanged);
00084
00085             container.Add(_elementsConfigPopup);
00086
00087             EditorUtility.SetDirty(_currentConfig);
00088         }
00089         else
00090         {
00091             container.Add(new Label("Add more configs!"));
00092         }
00093
00094         _indexLabel = this.Q<Label>("ItemGenIndex");
00095         _indexLabel.text = $"{_index + 1}. ";
00096
00097         _chanceField = this.Q<IntegerField>("ItemChance");
00098         _removeButton = this.Q<Button>("RemoveButton");
00099
00100         _removeButton.clicked += OnRemoveClicked;
00101
00102         _chanceField.value = _target.Possibility;
00103         _chanceField.RegisterValueChangedCallback(OnChanceChanged);
00104
00105         _changedChanceEvent += ChanceChanged;
00106         _changedElementEvent += ElementChanged;
00107         _changedIndexEvent += IndexChanged;
00108     }
```

6.132.3 Member Function Documentation

6.132.3.1 Dispose()

```
void MergeIt.Editor.GeneratedItemUI.Dispose ( ) [inline]
```

Definition at line 181 of file [GeneratedItemUI.cs](#).

```
00182     {
00183         _chanceField.UnregisterValueChangedCallback(OnChanceChanged);
00184         _removeButton.clicked -= OnRemoveClicked;
00185
00186         _changedChanceEvent -= ChanceChanged;
00187         _changedElementEvent -= ElementChanged;
00188         _changedIndexEvent -= IndexChanged;
00189     }
```

6.132.4 Property Documentation

6.132.4.1 Index

```
int MergeIt.Editor.GeneratedItemUI.Index [get], [set]
```

Definition at line 24 of file [GeneratedItemUI.cs](#).

```
00025     {
00026         set => _changedIndexEvent?.Invoke(_target, value);
00027     }
```

6.132.4.2 RemoveGeneratedItemEvent

```
Action<int, GeneratableElement> MergeIt.Editor.GeneratedItemUI.RemoveGeneratedItemEvent [get], [set]
```

Definition at line 22 of file [GeneratedItemUI.cs](#).

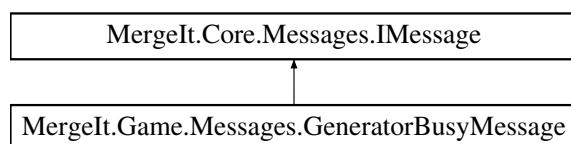
```
00022 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/GeneratedItemUI.cs

6.133 Mergelt.Game.Messages.GeneratorBusyMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.GeneratorBusyMessage:



Properties

- [GridPoint Point](#) [get, set]

6.133.1 Detailed Description

Definition at line 8 of file [GeneratorBusyMessage.cs](#).

6.133.2 Property Documentation

6.133.2.1 Point

`GridPoint MergeIt.Game.Messages.GeneratorBusyMessage.Point [get], [set]`

Definition at line 10 of file [GeneratorBusyMessage.cs](#).

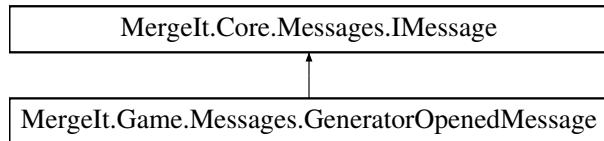
```
00010 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Messages/GeneratorBusyMessage.cs

6.134 MergeIt.Game.Messages.GeneratorOpenedMessage Class Reference

Inheritance diagram for MergeIt.Game.Messages.GeneratorOpenedMessage:



Properties

- `GridPoint GeneratorPoint [get, set]`

6.134.1 Detailed Description

Definition at line 8 of file [GeneratorOpenedMessage.cs](#).

6.134.2 Property Documentation

6.134.2.1 GeneratorPoint

`GridPoint MergeIt.Game.Messages.GeneratorOpenedMessage.GeneratorPoint [get], [set]`

Definition at line 10 of file [GeneratorOpenedMessage.cs](#).

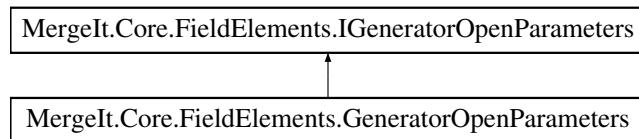
```
00010 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Messages/GeneratorOpenedMessage.cs

6.135 Mergelt.Core.FieldElements.GeneratorOpenParameters Class Reference

Inheritance diagram for Mergelt.Core.FieldElements.GeneratorOpenParameters:



Public Member Functions

- void [CopyFrom \(SavedGeneratorOpenParameters other\)](#)

Properties

- bool [IsOpening \[get\]](#)
- long [StartOpeningTime \[get, set\]](#)
- Bindable< float > [RemainingTime = new\(\) \[get, set\]](#)

Properties inherited from [Mergelt.Core.FieldElements.IGeneratorOpenParameters](#)

6.135.1 Detailed Description

Definition at line 7 of file [GeneratorOpenParameters.cs](#).

6.135.2 Member Function Documentation

6.135.2.1 [CopyFrom\(\)](#)

```
void MergeIt.Core.FieldElements.GeneratorOpenParameters.CopyFrom (
    SavedGeneratorOpenParameters other ) [inline]
```

Implements [Mergelt.Core.FieldElements.IGeneratorOpenParameters](#).

Definition at line 13 of file [GeneratorOpenParameters.cs](#).

```
00014     {
00015         StartOpeningTime = other.StartOpeningTime;
00016     }
```

6.135.3 Property Documentation

6.135.3.1 [IsOpening](#)

```
bool MergeIt.Core.FieldElements.GeneratorOpenParameters.IsOpening [get]
```

Implements [Mergelt.Core.FieldElements.IGeneratorOpenParameters](#).

Definition at line 9 of file [GeneratorOpenParameters.cs](#).

6.135.3.2 RemainingTime

```
Bindable<float> MergeIt.Core.FieldElements.GeneratorOpenParameters.RemainingTime = new()
[get], [set]
```

Implements [Mergelt.Core.FieldElements.IGeneratorOpenParameters](#).

Definition at line 11 of file [GeneratorOpenParameters.cs](#).

```
00011 { get; set; } = new();
```

6.135.3.3 StartOpeningTime

```
long MergeIt.Core.FieldElements.GeneratorOpenParameters.StartOpeningTime [get], [set]
```

Implements [Mergelt.Core.FieldElements.IGeneratorOpenParameters](#).

Definition at line 10 of file [GeneratorOpenParameters.cs](#).

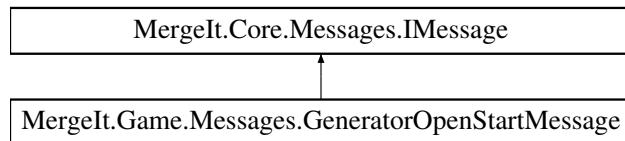
```
00010 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/Parameters/GeneratorOpenParameters.cs

6.136 Mergelt.Game.Messages.GeneratorOpenStartMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.GeneratorOpenStartMessage:



Properties

- [GridPoint GeneratorPoint \[get, set\]](#)

6.136.1 Detailed Description

Definition at line 8 of file [GeneratorOpenStartMessage.cs](#).

6.136.2 Property Documentation

6.136.2.1 GeneratorPoint

`GridPoint` MergeIt.Game.Messages.GeneratorOpenStartMessage.GeneratorPoint [get], [set]

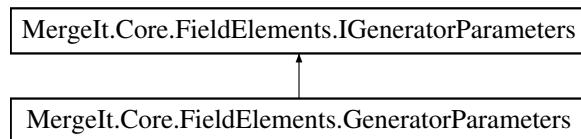
Definition at line 10 of file [GeneratorOpenStartMessage.cs](#).
00010 { get; set; }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/GeneratorOpenStartMessage.cs

6.137 Mergelt.Core.FieldElements.GeneratorParameters Class Reference

Inheritance diagram for Mergelt.Core.FieldElements.GeneratorParameters:



Public Member Functions

- void [CopyFrom](#) (SavedGeneratorParameters other)

Properties

- bool [Charging](#) [get]
- int [AvailableToDrop](#) [get, set]
- int [ChargedCount](#) [get, set]
- long [StartChargingTime](#) [get, set]
- float [RemainChargeTime](#) [get, set]
- float [MinDropFullChargeTime](#) [get, set]
- int [DroppedElements](#) [get, set]
- Bindable< float > [MinDropChargeTime](#) = new() [get, set]

Properties inherited from [Mergelt.Core.FieldElements.IGeneratorParameters](#)

6.137.1 Detailed Description

Definition at line 7 of file [GeneratorParameters.cs](#).

6.137.2 Member Function Documentation

6.137.2.1 CopyFrom()

```
void MergeIt.Core.FieldElements.GeneratorParameters.CopyFrom (
    SavedGeneratorParameters other) [inline]
```

Implements [MergeIt.Core.FieldElements.IGeneratorParameters](#).

Definition at line 20 of file [GeneratorParameters.cs](#).

```
00021     {
00022         AvailableToDrop = other.AvailableToDrop;
00023         StartChargingTime = other.StartChargingTime;
00024         DroppedElements = other.DroppedElements;
00025         ChargedCount = other.ChargedCount;
00026     }
```

6.137.3 Property Documentation

6.137.3.1 AvailableToDrop

```
int MergeIt.Core.FieldElements.GeneratorParameters.AvailableToDrop [get], [set]
```

Implements [MergeIt.Core.FieldElements.IGeneratorParameters](#).

Definition at line 11 of file [GeneratorParameters.cs](#).

```
00011 { get; set; }
```

6.137.3.2 ChargedCount

```
int MergeIt.Core.FieldElements.GeneratorParameters.ChargedCount [get], [set]
```

Implements [MergeIt.Core.FieldElements.IGeneratorParameters](#).

Definition at line 12 of file [GeneratorParameters.cs](#).

```
00012 { get; set; }
```

6.137.3.3 Charging

```
bool MergeIt.Core.FieldElements.GeneratorParameters.Charging [get]
```

Implements [MergeIt.Core.FieldElements.IGeneratorParameters](#).

Definition at line 9 of file [GeneratorParameters.cs](#).

6.137.3.4 DroppedElements

```
int MergeIt.Core.FieldElements.GeneratorParameters.DroppedElements [get], [set]
```

Implements [MergeIt.Core.FieldElements.IGeneratorParameters](#).

Definition at line 16 of file [GeneratorParameters.cs](#).

```
00016 { get; set; }
```

6.137.3.5 MinDropChargeTime

```
Bindable<float> MergeIt.Core.FieldElements.GeneratorParameters.MinDropChargeTime = new()
[get], [set]
```

Implements [Mergelt.Core.FieldElements.IGeneratorParameters](#).

Definition at line 18 of file [GeneratorParameters.cs](#).

```
00018 { get; set; } = new();
```

6.137.3.6 MinDropFullChargeTime

```
float MergeIt.Core.FieldElements.GeneratorParameters.MinDropFullChargeTime [get], [set]
```

Implements [Mergelt.Core.FieldElements.IGeneratorParameters](#).

Definition at line 15 of file [GeneratorParameters.cs](#).

```
00015 { get; set; }
```

6.137.3.7 RemainChargeTime

```
float MergeIt.Core.FieldElements.GeneratorParameters.RemainChargeTime [get], [set]
```

Implements [Mergelt.Core.FieldElements.IGeneratorParameters](#).

Definition at line 14 of file [GeneratorParameters.cs](#).

```
00014 { get; set; }
```

6.137.3.8 StartChargingTime

```
long MergeIt.Core.FieldElements.GeneratorParameters.StartChargingTime [get], [set]
```

Implements [Mergelt.Core.FieldElements.IGeneratorParameters](#).

Definition at line 13 of file [GeneratorParameters.cs](#).

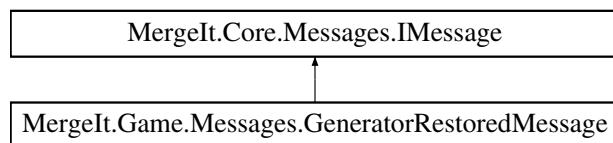
```
00013 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/Parameters/GeneratorParameters.cs

6.138 Mergelt.Game.Messages.GeneratorRestoredMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.GeneratorRestoredMessage:



Properties

- `GridPoint GeneratorPoint [get, set]`

6.138.1 Detailed Description

Definition at line 8 of file [GeneratorRestoredMessage.cs](#).

6.138.2 Property Documentation

6.138.2.1 GeneratorPoint

`GridPoint MergeIt.Game.Messages.GeneratorRestoredMessage.GeneratorPoint [get], [set]`

Definition at line 10 of file [GeneratorRestoredMessage.cs](#).

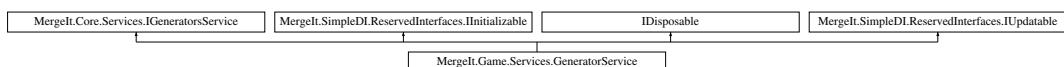
```
00010 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/GeneratorRestoredMessage.cs

6.139 Mergelt.Game.Services.GeneratorService Class Reference

Inheritance diagram for `Mergelt.Game.Services.GeneratorService`:



Public Member Functions

- `void Initialize ()`
- `void Dispose ()`
- `void TryOpen (IFieldElement generator)`
- `void TrySkipOpening (IFieldElement generator)`
- `void TrySkipCharging (IFieldElement generator)`
- `void Update ()`

6.139.1 Detailed Description

Definition at line 19 of file [GeneratorService.cs](#).

6.139.2 Member Function Documentation

6.139.2.1 Dispose()

```
void MergeIt.Game.Services.GeneratorService.Dispose ( ) [inline]
```

Definition at line 43 of file [GeneratorService.cs](#).

```
00044     {
00045         _messageBus.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00046         _messageBus.RemoveListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00047         _messageBus.RemoveListener<MergeElementsMessage>(MergeElementsMessageHandler);
00048         _messageBus.RemoveListener<SplitElementMessage>(SplitElementsMessageHandler);
00049         _messageBus.RemoveListener<CreateElementMessage>(CreateElementMessageHandler);
00050         _messageBus.RemoveListener<SkipTimeMessage>(OnSkipTimeMessageHandler);
00051     }
```

6.139.2.2 Initialize()

```
void MergeIt.Game.Services.GeneratorService.Initialize ( ) [inline]
```

Implements [MergeIt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 33 of file [GeneratorService.cs](#).

```
00034     {
00035         _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00036         _messageBus.AddListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00037         _messageBus.AddListener<MergeElementsMessage>(MergeElementsMessageHandler);
00038         _messageBus.AddListener<SplitElementMessage>(SplitElementsMessageHandler);
00039         _messageBus.AddListener<CreateElementMessage>(CreateElementMessageHandler);
00040         _messageBus.AddListener<SkipTimeMessage>(OnSkipTimeMessageHandler);
00041     }
```

6.139.2.3 TryOpen()

```
void MergeIt.Game.Services.GeneratorService.TryOpen (
    IFIELDELEMENT generator ) [inline]
```

Implements [MergeIt.Core.Services.IGeneratorsService](#).

Definition at line 53 of file [GeneratorService.cs](#).

```
00054     {
00055         if (_fieldLogicModel.OpeningGenerator != null)
00056         {
00057             Debug.Log("Another generator is already opening.");
00058             return;
00059         }
00060
00061         ElementConfig generatorConfig = generator.ConfigParameters.ElementConfig;
00062         generator.GeneratorOpenParameters.StartOpeningTime =
00063             DateTimeOffset.UtcNow.ToUnixTimeSeconds();
00064         generator.GeneratorOpenParameters.RemainingTime.Value =
00065             generatorConfig.GeneratorSettings.OpenTime;
00066         _fieldLogicModel.OpeningGenerator = generator;
00067
00068         _messageBus.Fire(new GeneratorOpenStartMessage
00069         {
00070             GeneratorPoint = generator.InfoParameters.LogicPosition
00071         });
00071     }
```

6.139.2.4 TrySkipCharging()

```
void MergeIt.Game.Services.GeneratorService.TrySkipCharging (
    IFIELDELEMENT generator) [inline]
```

Implements [MergeIt.Core.Services.IGeneratorsService](#).

Definition at line 87 of file [GeneratorService.cs](#).

```
00088     {
00089         ElementGeneratorSettings generatorSettings =
00090             generator.ConfigParameters.ElementConfig.GeneratorSettings;
00091         if (_currencyService.TryPay(generatorSettings.SkipChargeCostSettings))
00092         {
00093             generator.SkipCharging();
00094             _messageBus.Fire(new GeneratorRestoredMessage
00095             {
00096                 GeneratorPoint = generator.InfoParameters.LogicPosition
00097             });
00098             _messageBus.Fire(new CheckGeneratorMessage
00099             {
00100                 GeneratorPoint = generator.InfoParameters.LogicPosition
00101             });
00102         }
00103     }
00104 }
```

6.139.2.5 TrySkipOpening()

```
void MergeIt.Game.Services.GeneratorService.TrySkipOpening (
    IFIELDELEMENT generator) [inline]
```

Implements [MergeIt.Core.Services.IGeneratorsService](#).

Definition at line 73 of file [GeneratorService.cs](#).

```
00074     {
00075         ElementGeneratorSettings generatorSettings =
00076             generator.ConfigParameters.ElementConfig.GeneratorSettings;
00077         if (_currencyService.TryPay(generatorSettings.SkipOpenCostSettings))
00078         {
00079             ClearOpenableGenerator(generator);
00080             _messageBus.Fire(new GeneratorOpenedMessage
00081             {
00082                 GeneratorPoint = generator.InfoParameters.LogicPosition
00083             });
00084         }
00085     }
```

6.139.2.6 Update()

```
void MergeIt.Game.Services.GeneratorService.Update () [inline]
```

Implements [MergeIt.SimpleDI.ReservedInterfaces.IUpdatable](#).

Definition at line 106 of file [GeneratorService.cs](#).

```
00107     {
00108         if (_fieldLogicModel != null)
00109         {
00110             float deltaTime = Time.unscaledDeltaTime;
00111             for (var index = _fieldLogicModel.AllGenerators.Count - 1; index >= 0; index--)
00112             {
00113                 var generator = _fieldLogicModel.AllGenerators[index];
00114                 IGeneratorOpenParameters generatorOpenParameters =
00115                     generator.GeneratorOpenParameters;
00116                 IGeneratorParameters parameters = generator.GeneratorParameters;
00117                 if (parameters.Charging)
00118                 {
00119                     if (parameters.MinDropChargeTime.Value > 0f)
```

```

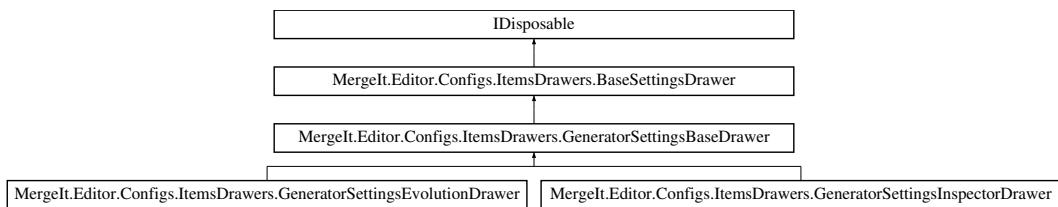
00119             {
00120                 parameters.MinDropChargeTime.Value -= deltaTime;
00121             }
00122 
00123             parameters.RemainChargeTime -= deltaTime;
00124 
00125             if (parameters.RemainChargeTime <= 0f)
00126             {
00127                 generator.ChargeGenerator();
00128             }
00129 
00130             if (parameters.MinDropChargeTime.Value <= 0f)
00131             {
00132                 _messageBus.Fire(new GeneratorRestoredMessage
00133                 {
00134                     GeneratorPoint = generator.InfoParameters.LogicPosition
00135                 });
00136             }
00137         }
00138         else if (generatorOpenParameters is { IsOpening: true })
00139         {
00140             generatorOpenParameters.RemainingTime.Value -= deltaTime;
00141 
00142             if (generatorOpenParameters.RemainingTime.Value <= 0f)
00143             {
00144                 ClearOpenableGenerator(generator);
00145 
00146                 _messageBus.Fire(new GeneratorOpenedMessage
00147                 {
00148                     GeneratorPoint = generator.InfoParameters.LogicPosition
00149                 });
00150             }
00151         }
00152     }
00153 }
00154 }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Services/GeneratorService.cs

6.140 MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer Class Reference

Inheritance diagram for MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer:



Public Member Functions

- [GeneratorSettingsBaseDrawer](#) (VisualElement parent, [ElementConfig](#) config)
- override void [Draw](#) ()
- void [Clear](#) ()
- override void [Dispose](#) ()

Public Member Functions inherited from [Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer](#)

- [BaseSettingsDrawer](#) (VisualElement parent, [ElementConfig](#) config)
- void [Draw](#) ()
- virtual void [Hide](#) ()
- virtual void [Show](#) ()

Protected Member Functions

- virtual void [OnClear](#) ()
- override void [OnExpand](#) (bool expand)
- override VisualTreeAsset [GetUxml](#) ()

Protected Member Functions inherited from [Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer](#)

- void [Expand](#) (bool expand)
- void [OnExpand](#) (bool expand)
- void [RotateArrow](#) (bool rotate)
- VisualTreeAsset [GetUxml](#) ()

Additional Inherited Members

Protected Attributes inherited from [Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer](#)

- readonly VisualElement [Parent](#)
- readonly VisualElement [Root](#)
- readonly [ElementConfig](#) [Config](#)
- Label [Arrow](#)

6.140.1 Detailed Description

Definition at line 11 of file [GeneratorSettingsBaseDrawer.cs](#).

6.140.2 Constructor & Destructor Documentation

6.140.2.1 [GeneratorSettingsBaseDrawer\(\)](#)

```
MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer.GeneratorSettingsBaseDrawer (
    VisualElement parent,
    ElementConfig config ) [inline]
```

Definition at line 23 of file [GeneratorSettingsBaseDrawer.cs](#).

```
00023     config)
00024     {
00025         _settings = config.GeneratorSettings;
00026     }
```

6.140.3 Member Function Documentation

6.140.3.1 Clear()

```
void MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer.Clear ( ) [inline]
```

Definition at line 58 of file [GeneratorSettingsBaseDrawer.cs](#).

```
00059     {
00060         OnClear ();
00061
00062         Parent.Remove (Root);
00063     }
```

6.140.3.2 Dispose()

```
override void MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer.Dispose ( )
[inline], [virtual]
```

Reimplemented from [MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer](#).

Definition at line 71 of file [GeneratorSettingsBaseDrawer.cs](#).

```
00072     {
00073         base.Dispose ();
00074
00075         Clear ();
00076     }
```

6.140.3.3 Draw()

```
override void MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer.Draw ( ) [inline]
```

Definition at line 28 of file [GeneratorSettingsBaseDrawer.cs](#).

```
00029     {
00030         _stackContainer = Root.Q<VisualElement>("StackContainer");
00031         _needOpenContainer = Root.Q<VisualElement>("NeedOpenContainer");
00032         _isUnlimitedToggle = Root.Q<Toggle>("IsUnlimited");
00033         _needOpenToggle = Root.Q<Toggle>("NeedOpen");
00034
00035         var skipOpenParams = Root.Q<CurrencyComponent>("SkipOpenParameters");
00036         skipOpenParams.Bind(new SerializedObject(Config));
00037
00038         var skipChargeParams = Root.Q<CurrencyComponent>("SkipChargeParameters");
00039         skipChargeParams.Bind(new SerializedObject(Config));
00040
00041         _generatorSettings = Root.Q<Box>("GeneratorSettings");
00042
00043         _isUnlimitedToggle.RegisterValueChangedCallback(OnUnlimitedChanged);
00044         _needOpenToggle.RegisterValueChangedCallback(OnNeedOpenChanged);
00045
00046         _stackContainer.style.display = _settings.IsUnlimited ? StylesConstants.DisplayNone :
00047             StylesConstants.DisplayFlex;
00048         _needOpenContainer.style.display = _settings.NeedOpen ? StylesConstants.DisplayFlex :
00049             StylesConstants.DisplayNone;
00050
00051         _expanded = Root.Q<Toggle>("GeneratorSettingsFoldout");
00052         _expanded.value = _settings.Expanded;
00053         _expanded.RegisterValueChangedCallback(OnExpandedChanged);
00054
00055         Arrow = Root.Q<Label>("Arrow");
00056         Expand(_settings.Expanded);
00057     }
```

6.140.3.4 GetUxml()

```
override VisualTreeAsset MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer.GetUxml ( ) [inline], [protected]
```

Definition at line 100 of file [GeneratorSettingsBaseDrawer.cs](#).

```
00101     {
00102         return
00103             (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.InspectorResourcesRoot,
00104                 "GeneratorInspector.uxml"));
00105     }
```

6.140.3.5 OnClear()

```
virtual void MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer.OnClear ( ) [inline],
[protected], [virtual]
```

Definition at line 65 of file [GeneratorSettingsBaseDrawer.cs](#).

```
00066     {
00067         _isUnlimitedToggle.UnregisterValueChangedCallback(OnUnlimitedChanged);
00068         _needOpenToggle.UnregisterValueChangedCallback(OnNeedOpenChanged);
00069     }
```

6.140.3.6 OnExpand()

```
override void MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer.OnExpand (
    bool expand) [inline], [protected]
```

Definition at line 93 of file [GeneratorSettingsBaseDrawer.cs](#).

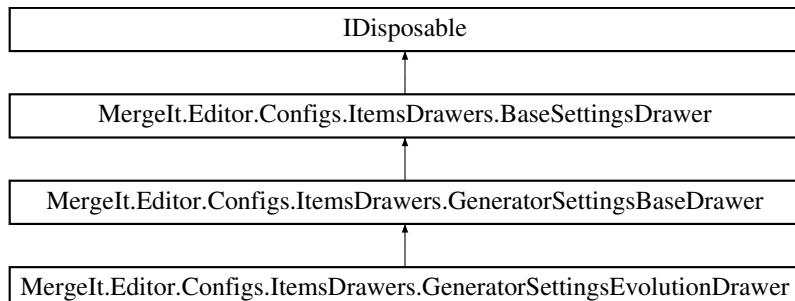
```
00094     {
00095         _settings.Expanded = expand;
00096         _generatorSettings.style.display =
00097             expand ? StylesConstants.DisplayFlex : StylesConstants.DisplayNone;
00098     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Editor/Core/Configs/ItemsDrawers/GeneratorSettingsBaseDrawer.cs

6.141 MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsEvolutionDrawer Class Reference

Inheritance diagram for MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsEvolutionDrawer:



Public Member Functions

- [GeneratorSettingsEvolutionDrawer](#) (VisualElement parent, ElementConfig config, EvolutionsGraphView evolutionsGraphView)
- override void [Draw](#) ()

Public Member Functions inherited from

[Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer](#)

- [GeneratorSettingsBaseDrawer](#) (VisualElement parent, ElementConfig config)
- override void [Draw](#) ()
- void [Clear](#) ()
- override void [Dispose](#) ()

Public Member Functions inherited from

[Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer](#)

- [BaseSettingsDrawer](#) (VisualElement parent, ElementConfig config)
- void [Draw](#) ()
- virtual void [Hide](#) ()
- virtual void [Show](#) ()

Protected Member Functions

- override void [OnClear](#) ()

Protected Member Functions inherited from

[Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer](#)

- override void [OnExpand](#) (bool expand)
- override VisualTreeAsset [GetUxml](#) ()

Protected Member Functions inherited from

[Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer](#)

- void [Expand](#) (bool expand)
- void [OnExpand](#) (bool expand)
- void [RotateArrow](#) (bool rotate)
- VisualTreeAsset [GetUxml](#) ()

Additional Inherited Members

Protected Attributes inherited from

[Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer](#)

- readonly VisualElement [Parent](#)
- readonly VisualElement [Root](#)
- readonly ElementConfig [Config](#)
- Label [Arrow](#)

6.141.1 Detailed Description

Definition at line 9 of file [GeneratorSettingsEvolutionDrawer.cs](#).

6.141.2 Constructor & Destructor Documentation

6.141.2.1 GeneratorSettingsEvolutionDrawer()

```
MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsEvolutionDrawer.GeneratorSettingsEvolutionDrawer (
    VisualElement parent,
    ElementConfig config,
    EvolutionsGraphView evolutionsGraphView ) [inline]
```

Definition at line 15 of file [GeneratorSettingsEvolutionDrawer.cs](#).

```
00016         : base(parent, config)
00017         {
00018             _graphView = evolutionsGraphView;
00019         }
```

6.141.3 Member Function Documentation

6.141.3.1 Draw()

```
override void MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsEvolutionDrawer.Draw ( )
[inline]
```

Definition at line 21 of file [GeneratorSettingsEvolutionDrawer.cs](#).

```
00022         {
00023             base.Draw();
00024
00025             var genItemsContainer = Root.Q<VisualElement>("GenItemsContainer");
00026             var genItemsControls = Root.Q<VisualElement>("GeneratedItemsControls");
00027             _creator = new GeneratedItemsCreator(genItemsControls, genItemsContainer, Config,
00028             _graphView.EvolutionsData, _graphView.Configs);
00029         }
```

6.141.3.2 OnClear()

```
override void MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsEvolutionDrawer.OnClear ( )
[inline], [protected], [virtual]
```

Reimplemented from [MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer](#).

Definition at line 31 of file [GeneratorSettingsEvolutionDrawer.cs](#).

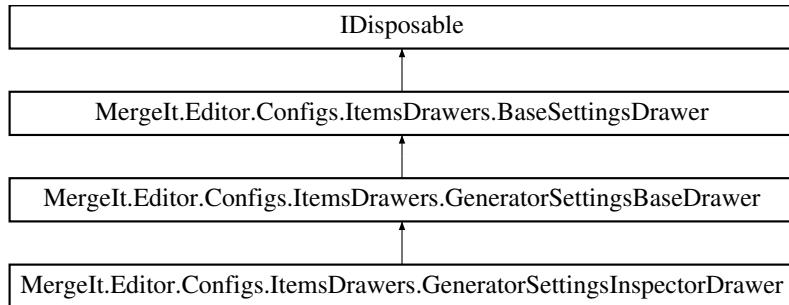
```
00032         {
00033             base.OnClear();
00034
00035             _creator.Dispose();
00036         }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Editor/Core/Configs/ItemsDrawers/GeneratorSettingsEvolutionDrawer.cs

6.142 MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsInspectorDrawer Class Reference

Inheritance diagram for MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsInspectorDrawer:



Public Member Functions

- [GeneratorSettingsInspectorDrawer](#) (VisualElement parent, [ElementConfig](#) config)
- override void [Draw](#) ()

Public Member Functions inherited from MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer

- [GeneratorSettingsBaseDrawer](#) (VisualElement parent, [ElementConfig](#) config)
- override void [Draw](#) ()
- void [Clear](#) ()
- override void [Dispose](#) ()

Public Member Functions inherited from MergeIt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer

- [BaseSettingsDrawer](#) (VisualElement parent, [ElementConfig](#) config)
- void [Draw](#) ()
- virtual void [Hide](#) ()
- virtual void [Show](#) ()

Additional Inherited Members

Protected Member Functions inherited from MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer

- virtual void [OnClear](#) ()
- override void [OnExpand](#) (bool expand)
- override VisualTreeAsset [GetUxml](#) ()

Protected Member Functions inherited from Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer

- void `Expand` (bool expand)
- void `OnExpand` (bool expand)
- void `RotateArrow` (bool rotate)
- VisualTreeAsset `GetUxml` ()

Protected Attributes inherited from Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer

- readonly VisualElement `Parent`
- readonly VisualElement `Root`
- readonly ElementConfig `Config`
- Label `Arrow`

6.142.1 Detailed Description

Definition at line 8 of file [GeneratorSettingsInspectorDrawer.cs](#).

6.142.2 Constructor & Destructor Documentation

6.142.2.1 GeneratorSettingsInspectorDrawer()

```
MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsInspectorDrawer.GeneratorSettingsInspectorDrawer ( 
    VisualElement parent,
    ElementConfig config ) [inline]
```

Definition at line 10 of file [GeneratorSettingsInspectorDrawer.cs](#).

```
00010     base(parent, config)
00011     {
00012     }
```

:

6.142.3 Member Function Documentation

6.142.3.1 Draw()

```
override void MergeIt.Editor.Configs.ItemsDrawers.GeneratorSettingsInspectorDrawer.Draw ( )
[inline]
```

Definition at line 14 of file [GeneratorSettingsInspectorDrawer.cs](#).

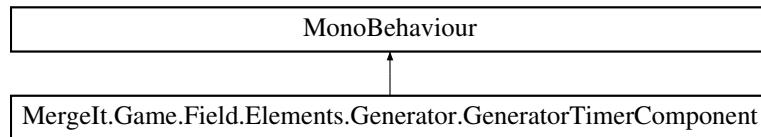
```
00015     {
00016         base.Draw();
00017
00018         var cantGenerateSection = Root.Q<VisualElement>("CantGenerateSection");
00019         var helpBox = new HelpBox("Editing this section is available only from the 'Evolutions
editor', " +
00020                               "that you can find in the menu 'Window/Merge Toolkit/Evolutions
editor'", HelpBoxMessageType.Info);
00021         cantGenerateSection.Add(helpBox);
00022     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/GeneratorSettingsInspectorDrawer.cs

6.143 Mergelt.Game.Field.Elements.Generator.GeneratorTimerComponent Class Reference

Inheritance diagram for Mergelt.Game.Field.Elements.Generator.GeneratorTimerComponent:



Public Member Functions

- void [StartTimer \(Bindable< float > remainChargeTime, float fullRemainTime\)](#)

6.143.1 Detailed Description

Definition at line 9 of file [GeneratorTimerComponent.cs](#).

6.143.2 Member Function Documentation

6.143.2.1 StartTimer()

```
void MergeIt.Game.Field.Elements.Generator.GeneratorTimerComponent.StartTimer (
    Bindable< float > remainChargeTime,
    float fullRemainTime ) [inline]
```

Definition at line 17 of file [GeneratorTimerComponent.cs](#).

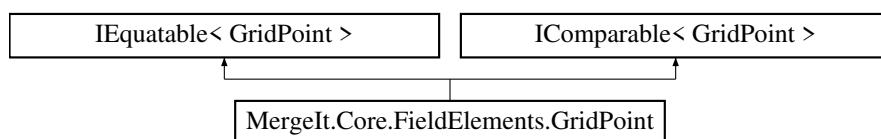
```
00018     {
00019         if (remainChargeTime.Value > 0f)
00020         {
00021             gameObject.SetActive(true);
00022
00023             _remainFullTime = fullRemainTime;
00024
00025             _remainChargeTime?.Unsubscribe(OnRemainTimeChanged);
00026
00027             _remainChargeTime = remainChargeTime;
00028             _remainChargeTime.Subscribe(OnRemainTimeChanged, true);
00029         }
00030     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Elements/Generator/GeneratorTimerComponent.cs

6.144 Mergelt.Core.FieldElements.GridPoint Struct Reference

Inheritance diagram for Mergelt.Core.FieldElements.GridPoint:



Public Member Functions

- `GridPoint (int x, int y)`
- `GridPoint Copy ()`
- `bool Equals (GridPoint other)`
- `int CompareTo (GridPoint other)`
- `override bool Equals (object obj)`
- `override int GetHashCode ()`
- `override string ToString ()`

Static Public Member Functions

- `static bool operator== (GridPoint p1, GridPoint p2)`
- `static bool operator!= (GridPoint p1, GridPoint p2)`
- `static GridPoint Create (int row, int column)`

Static Public Attributes

- `static readonly GridPoint Default = new GridPoint(-1, -1)`

Properties

- `readonly int X [get]`
- `readonly int Y [get]`

6.144.1 Detailed Description

Definition at line 10 of file [GridPoint.cs](#).

6.144.2 Constructor & Destructor Documentation

6.144.2.1 GridPoint()

```
MergeIt.Core.FieldElements.GridPoint.GridPoint (
    int x,
    int y )  [inline]
```

Definition at line 40 of file [GridPoint.cs](#).

```
00041     {
00042         _x = x;
00043         _y = y;
00044     }
```

6.144.3 Member Function Documentation

6.144.3.1 CompareTo()

```
int MergeIt.Core.FieldElements.GridPoint.CompareTo (
    GridPoint other )  [inline]
```

Definition at line 61 of file [GridPoint.cs](#).

```
00062     {
00063         int xComparison = _x.CompareTo(other._x);
00064         if (xComparison != 0) return xComparison;
00065         return _y.CompareTo(other._y);
00066     }
```

6.144.3.2 Copy()

```
GridPoint MergeIt.Core.FieldElements.GridPoint.Copy ( ) [inline]
```

Definition at line 46 of file GridPoint.cs.

```
00047     {  
00048         return new GridPoint(_x, _y);  
00049     }
```

6.144.3.3 Create()

```
static GridPoint MergeIt.Core.FieldElements.GridPoint.Create (  
    int row,  
    int column) [inline], [static]
```

Definition at line 51 of file GridPoint.cs.

```
00052     {  
00053         return new GridPoint(row, column);  
00054     }
```

6.144.3.4 Equals() [1/2]

```
bool MergeIt.Core.FieldElements.GridPoint.Equals (  
    GridPoint other) [inline]
```

Definition at line 56 of file GridPoint.cs.

```
00057     {  
00058         return _x == other._x && _y == other._y;  
00059     }
```

6.144.3.5 Equals() [2/2]

```
override bool MergeIt.Core.FieldElements.GridPoint.Equals (  
    object obj) [inline]
```

Definition at line 68 of file GridPoint.cs.

```
00069     {  
00070         return obj is GridPoint other && Equals(other);  
00071     }
```

6.144.3.6 GetHashCode()

```
override int MergeIt.Core.FieldElements.GridPoint.GetHashCode ( ) [inline]
```

Definition at line 73 of file GridPoint.cs.

```
00074     {  
00075         unchecked  
00076         {  
00077             return _x * 397 ^ _y;  
00078         }  
00079     }
```

6.144.3.7 operator"!=()

```
static bool MergeIt.Core.FieldElements.GridPoint.operator!= (
    GridPoint p1,
    GridPoint p2 ) [inline], [static]
```

Definition at line 17 of file [GridPoint.cs](#).

```
00018     {
00019         return !p1.Equals(p2);
00020     }
```

6.144.3.8 operator==()

```
static bool MergeIt.Core.FieldElements.GridPoint.operator== (
    GridPoint p1,
    GridPoint p2 ) [inline], [static]
```

Definition at line 12 of file [GridPoint.cs](#).

```
00013     {
00014         return p1.Equals(p2);
00015     }
```

6.144.3.9 ToString()

```
override string MergeIt.Core.FieldElements.GridPoint.ToString ( ) [inline]
```

Definition at line 81 of file [GridPoint.cs](#).

```
00082     {
00083         return $"(Row: {_x}, Column: {_y})";
00084     }
```

6.144.4 Member Data Documentation

6.144.4.1 Default

```
readonly GridPoint MergeIt.Core.FieldElements.GridPoint.Default = new GridPoint(-1, -1) [static]
```

Definition at line 22 of file [GridPoint.cs](#).

6.144.5 Property Documentation

6.144.5.1 X

```
readonly int MergeIt.Core.FieldElements.GridPoint.X [get]
```

Definition at line 30 of file [GridPoint.cs](#).

```
00031     {
00032         get => _x;
00033     }
```

6.144.5.2 Y

```
readonly int MergeIt.Core.FieldElements.GridPoint.Y [get]
```

Definition at line 35 of file [GridPoint.cs](#).

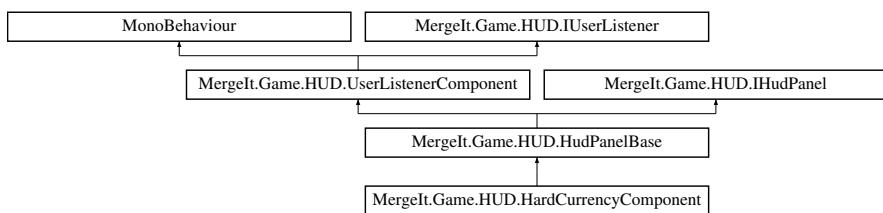
```
00036     {
00037         get => _y;
00038     }
```

The documentation for this struct was generated from the following file:

- [Assets/Mergelt/Source/Core/FieldElements/GridPoint.cs](#)

6.145 Mergelt.Game.HUD.HardCurrencyComponent Class Reference

Inheritance diagram for Mergelt.Game.HUD.HardCurrencyComponent:



Protected Member Functions

- [override void OnApplyModel \(\[UserServiceModel\]\(#\) userServiceModel\)](#)

Protected Member Functions inherited from [MergeIt.Game.HUD.HudPanelBase](#)

- [virtual void Start \(\)](#)

Protected Member Functions inherited from [MergeIt.Game.HUD.UserListenerComponent](#)

- [void OnApplyModel \(\[UserServiceModel\]\(#\) userServiceModel\)](#)

Properties

- [override HudPanelType Type \[get\]](#)

Properties inherited from [MergeIt.Game.HUD.HudPanelBase](#)

- [virtual HudPanelType Type \[get\]](#)

Properties inherited from [MergeIt.Game.HUD.UserListenerComponent](#)

- [UserServiceModel UserServiceModel \[get\]](#)

Properties inherited from [Mergelt.Game.HUD.IHudPanel](#)

Additional Inherited Members

Public Member Functions inherited from [Mergelt.Game.HUD.UserListenerComponent](#)

- void [ApplyModel](#) ([UserServiceModel](#) userServiceModel)

6.145.1 Detailed Description

Definition at line 10 of file [HardCurrencyComponent.cs](#).

6.145.2 Member Function Documentation

6.145.2.1 [OnApplyModel\(\)](#)

```
override void MergeIt.Game.HUD.HardCurrencyComponent.OnApplyModel (
    UserServiceModel userServiceModel ) [inline], [protected]
```

Definition at line 20 of file [HardCurrencyComponent.cs](#).

```
00021     {
00022         userServiceModel.HardCurrency.Subscribe(OnHardCurrencyChanged, true);
00023     }
```

6.145.3 Property Documentation

6.145.3.1 [Type](#)

```
override HudPanelType MergeIt.Game.HUD.HardCurrencyComponent.Type [get]
```

Implements [Mergelt.Game.HUD.IHudPanel](#).

Definition at line 15 of file [HardCurrencyComponent.cs](#).

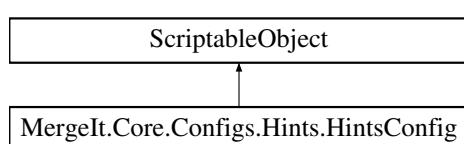
```
00016     {
00017         get => HudPanelType.HardCurrency;
00018     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/HUD/HardCurrencyComponent.cs

6.146 Mergelt.Core.Configs.Hints.HintsConfig Class Reference

Inheritance diagram for Mergelt.Core.Configs.Hints.HintsConfig:



Properties

- float [Interval](#) [get]

6.146.1 Detailed Description

Definition at line 8 of file [HintsConfig.cs](#).

6.146.2 Property Documentation

6.146.2.1 Interval

```
float MergeIt.Core.Configs.Hints.HintsConfig.Interval [get]
```

Definition at line 13 of file [HintsConfig.cs](#).

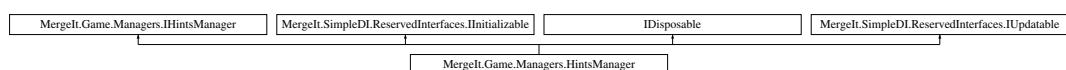
```
00014     {  
00015         get => _interval;  
00016     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Configs/Hints/HintsConfig.cs

6.147 Mergelt.Game.Managers.HintsManager Class Reference

Inheritance diagram for Mergelt.Game.Managers.HintsManager:



Public Member Functions

- void [Initialize](#) ()
- void [Dispose](#) ()
- void [Update](#) ()

6.147.1 Detailed Description

Definition at line 19 of file [HintsManager.cs](#).

6.147.2 Member Function Documentation

6.147.2.1 Dispose()

```
void MergeIt.Game.Managers.HintsManager.Dispose () [inline]
```

Definition at line 46 of file [HintsManager.cs](#).

```
00047     {
00048         _messageBus.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00049         _messageBus.RemoveListener<ActivateHintsMessage>(OnActivateHintsMessageHandler);
00050         _messageBus.RemoveListener<ResetHintsMessage>(OnResetHintsMessageHandler);
00051     }
```

6.147.2.2 Initialize()

```
void MergeIt.Game.Managers.HintsManager.Initialize () [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 39 of file [HintsManager.cs](#).

```
00040     {
00041         _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00042         _messageBus.AddListener<ActivateHintsMessage>(OnActivateHintsMessageHandler);
00043         _messageBus.AddListener<ResetHintsMessage>(OnResetHintsMessageHandler);
00044     }
```

6.147.2.3 Update()

```
void MergeIt.Game.Managers.HintsManager.Update () [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IUpdatable](#).

Definition at line 53 of file [HintsManager.cs](#).

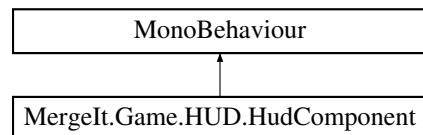
```
00054     {
00055         if (Available && _candidate1 == null && _candidate2 == null)
00056     {
00057         _time += Time.deltaTime;
00058
00059         if (_time >= _configsService.HintsConfig.Interval)
00060     {
00061             _time = 0f;
00062             FindCandidates();
00063         }
00064     }
00065     else if (_candidate1 != null && _candidate2 != null)
00066     {
00067         if (_candidate1.State == FieldElementState.Idle &&
00068             _candidate2.State == FieldElementState.Idle)
00069     {
00070         _candidate1 = null;
00071         _candidate2 = null;
00072
00073         _animationInProcess = false;
00074     }
00075 }
00076 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Managers/HintsManager.cs

6.148 Mergelt.Game.HUD.HudComponent Class Reference

Inheritance diagram for Mergelt.Game.HUD.HudComponent:



6.148.1 Detailed Description

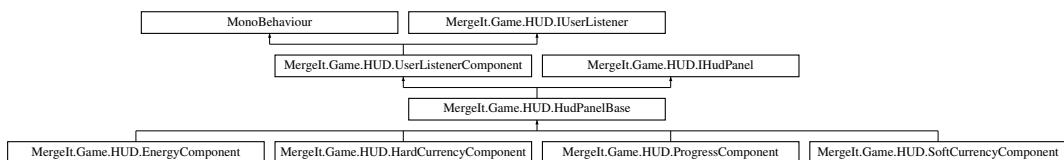
Definition at line 13 of file [HudComponent.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/HUD/HudComponent.cs

6.149 Mergelt.Game.HUD.HudPanelBase Class Reference

Inheritance diagram for Mergelt.Game.HUD.HudPanelBase:



Protected Member Functions

- virtual void [Start \(\)](#)

Protected Member Functions inherited from [Mergelt.Game.HUD.UserListenerComponent](#)

- void [OnApplyModel \(UserServiceModel userServiceModel\)](#)

Properties

- virtual HudPanelType [Type \[get\]](#)

Properties inherited from [Mergelt.Game.HUD.UserListenerComponent](#)

- [UserServiceModel UserServiceModel \[get\]](#)

Properties inherited from [Mergelt.Game.HUD.IHudPanel](#)

Additional Inherited Members

Public Member Functions inherited from [Mergelt.Game.HUD.UserListenerComponent](#)

- void [ApplyModel](#) ([UserServiceModel](#) userServiceModel)

6.149.1 Detailed Description

Definition at line 8 of file [HudPanelBase.cs](#).

6.149.2 Member Function Documentation

6.149.2.1 Start()

```
virtual void MergeIt.Game.HUD.HudPanelBase.Start ( ) [inline], [protected], [virtual]
```

Definition at line 12 of file [HudPanelBase.cs](#).

```
00013     {
00014         HudTargets.AddTarget(Type, transform);
00015     }
```

6.149.3 Property Documentation

6.149.3.1 Type

```
virtual HudPanelType MergeIt.Game.HUD.HudPanelBase.Type [get]
```

Implements [Mergelt.Game.HUD.IHudPanel](#).

Definition at line 10 of file [HudPanelBase.cs](#).

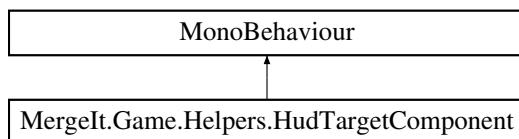
```
00010 { get; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/HUD/HudPanelBase.cs

6.150 Mergelt.Game.Helpers.HudTargetComponent Class Reference

Inheritance diagram for Mergelt.Game.Helpers.HudTargetComponent:



6.150.1 Detailed Description

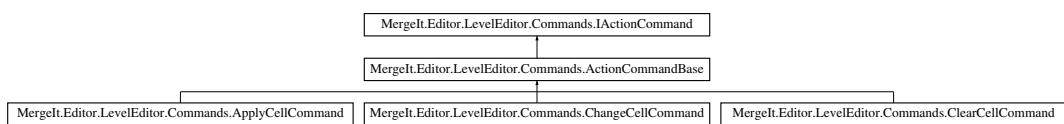
Definition at line 8 of file [HudTargetComponent.cs](#).

The documentation for this class was generated from the following file:

- [Assets/Mergelt/Source/Game/Helpers/HudTargetComponent.cs](#)

6.151 Mergelt.Editor.LevelEditor.Commands.IActionCommand Interface Reference

Inheritance diagram for Mergelt.Editor.LevelEditor.Commands.IActionCommand:



Public Member Functions

- void **Execute ()**
- void **Undo ()**

6.151.1 Detailed Description

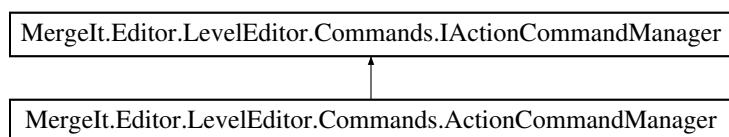
Definition at line 5 of file [IActionCommand.cs](#).

The documentation for this interface was generated from the following file:

- [Assets/Mergelt/Editor/Core/LevelEditor/Commands/IActionCommand.cs](#)

6.152 Mergelt.Editor.LevelEditor.Commands.IActionCommandManager Interface Reference

Inheritance diagram for Mergelt.Editor.LevelEditor.Commands.IActionCommandManager:



Public Member Functions

- void **ExecuteCommand** ([IActionCommand](#) command)
- void **Undo** ()
- void **Redo** ()

6.152.1 Detailed Description

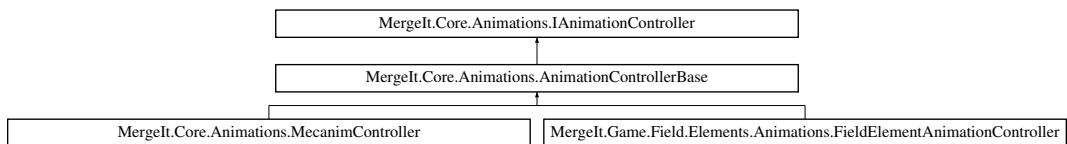
Definition at line [5](#) of file [IActionCommandManager.cs](#).

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Editor/Core/LevelEditor/Commands/IActionCommandManager.cs

6.153 MergeIt.Core.Animations.IAnimationController Interface Reference

Inheritance diagram for MergeIt.Core.Animations.IAnimationController:



Public Member Functions

- void **Initialize** ([IAnimationListener](#) listener)
- void **SetState** (string state)
- void **SetState** (int state)
- void **SetState< T >** (T state)

6.153.1 Detailed Description

Definition at line [7](#) of file [IAnimationController.cs](#).

6.153.2 Member Function Documentation

6.153.2.1 SetState< T >()

```
void MergeIt.Core.Animations.IAnimationController.SetState< T > (
    T state )
```

Type Constraints

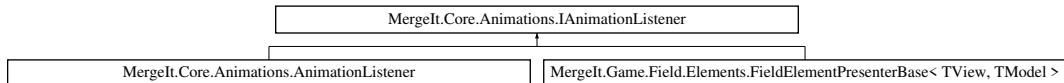
T : *Enum*

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/Animations/IAnimationController.cs

6.154 Mergelt.Core.Animations.IAnimationListener Interface Reference

Inheritance diagram for Mergelt.Core.Animations.IAnimationListener:



Public Member Functions

- void **ResetAnimationState ()**

6.154.1 Detailed Description

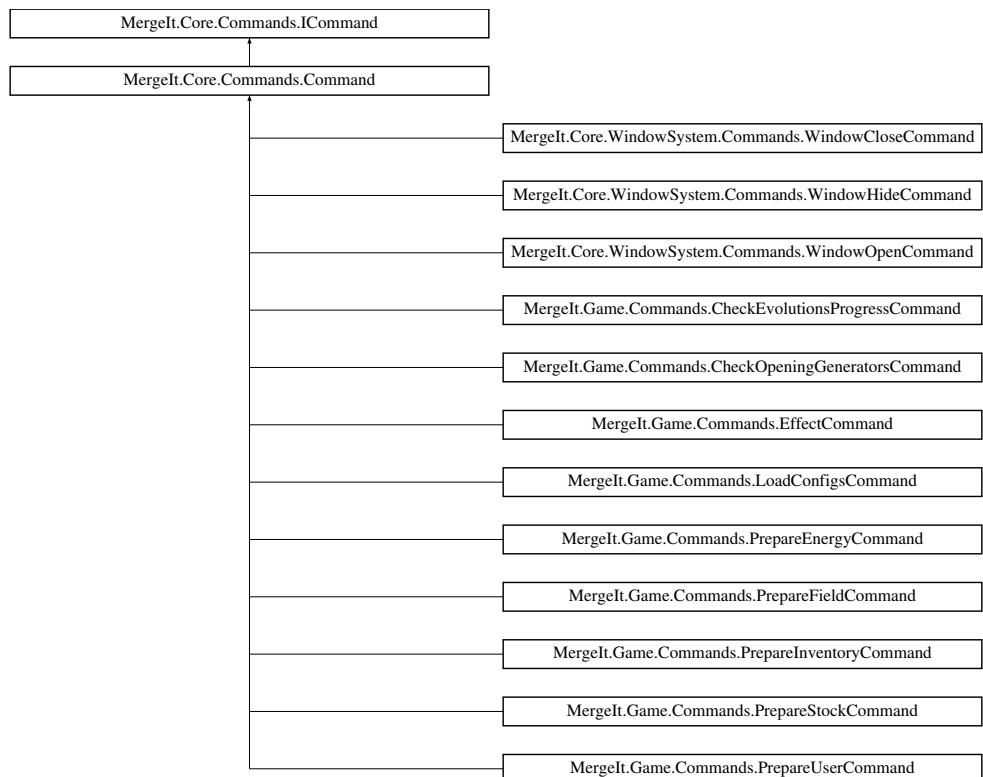
Definition at line 5 of file [IAnimationListener.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Animations/IAnimationListener.cs

6.155 Mergelt.Core.Commands.ICommand Interface Reference

Inheritance diagram for Mergelt.Core.Commands.ICommand:



Public Member Functions

- void **Execute** ()
- UniTask **ExecuteAsync** ()

Events

- Action< [ICommand](#) > **Finished**

6.155.1 Detailed Description

Definition at line 8 of file [ICommand.cs](#).

6.155.2 Event Documentation

6.155.2.1 Finished

Action< [ICommand](#)> MergeIt.Core.Commands.ICommand.Finished

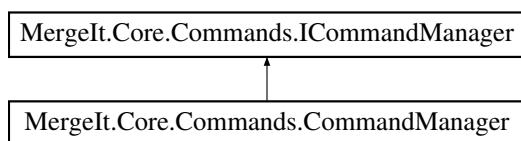
Definition at line 10 of file [ICommand.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Commands/ICommand.cs

6.156 Mergelt.Core.Commands.ICommandManager Interface Reference

Inheritance diagram for MergeIt.Core.Commands.ICommandManager:



Public Member Functions

- void **Run** ()
- void **RunSimultaneously** ()
- void **Add** ([ICommand](#) command)

Properties

- bool **Executing** [get]

Events

- Action< [ICommandManager](#) > Finished

6.156.1 Detailed Description

Definition at line 7 of file [ICommandManager.cs](#).

6.156.2 Property Documentation

6.156.2.1 Executing

```
bool MergeIt.Core.Commands.ICommandManager.Executing [get]
```

Definition at line 11 of file [ICommandManager.cs](#).

```
00011 { get; }
```

6.156.3 Event Documentation

6.156.3.1 Finished

```
Action<ICommandManager> MergeIt.Core.Commands.ICommandManager.Finished
```

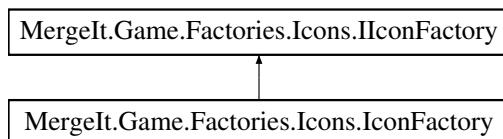
Definition at line 9 of file [ICommandManager.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Commands/ICommandManager.cs

6.157 Mergelt.Game.Factories.Icons.IconFactory Class Reference

Inheritance diagram for Mergelt.Game.Factories.Icons.IconFactory:



Public Member Functions

- [FieldElementIconComponent CreateIcon \(ElementConfig config, Transform parent=null\)](#)

6.157.1 Detailed Description

Definition at line 10 of file [IconFactory.cs](#).

6.157.2 Member Function Documentation

6.157.2.1 CreateIcon()

```
FieldElementIconComponent MergeIt.Game.Factories.IIconFactory.CreateIcon (
    ElementConfig config,
    Transform parent = null ) [inline]
```

Implements [Mergelt.Game.Factories.IIconFactory](#).

Definition at line 14 of file [IconFactory.cs](#).

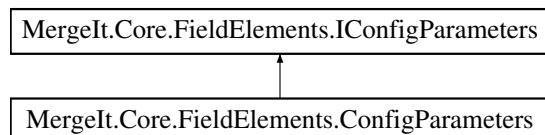
```
00015     {
00016         FieldElementIconComponent iconComponent;
00017         if (!icons.TryGetValue(config, out IObjectPool<FieldElementIconComponent> iconsPool))
00018         {
00019             iconsPool = new ObjectPool<FieldElementIconComponent>(
00020                 () =>
00021                 {
00022                     FieldElementIconComponent iconPrototype = config.GetIconComponent();
00023                     GameObject iconContainer = Object.Instantiate(iconPrototype.gameObject);
00024                     iconContainer.TryGetComponent(out FieldElementIconComponent icon);
00025
00026                     return icon;
00027                 },
00028                 OnGetIcon,
00029                 OnReleaseIcon,
00030                 OnDestroyIcon,
00031                 defaultCapacity: 2,
00032                 maxSize: 20);
00033
00034             icons[config] = iconsPool;
00035         }
00036
00037         iconComponent = GetIconComponent(iconsPool, parent);
00038
00039         return iconComponent;
00040     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Factories/Icons/IconFactory.cs

6.158 Mergelt.Core.FieldElements.IConfigParameters Interface Reference

Inheritance diagram for Mergelt.Core.FieldElements.IConfigParameters:



Properties

- `ElementConfig ElementConfig [get, set]`
- `EvolutionData EvolutionData [get, set]`

6.158.1 Detailed Description

Definition at line 8 of file [IConfigParameters.cs](#).

6.158.2 Property Documentation

6.158.2.1 ElementConfig

`ElementConfig` MergeIt.Core.FieldElements.IConfigParameters.ElementConfig [get], [set]

Definition at line 10 of file [IConfigParameters.cs](#).

```
00010 { get; set; }
```

6.158.2.2 EvolutionData

`EvolutionData` MergeIt.Core.FieldElements.IConfigParameters.EvolutionData [get], [set]

Definition at line 11 of file [IConfigParameters.cs](#).

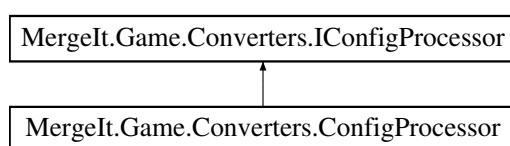
```
00011 { get; set; }
```

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/Parameters/IConfigParameters.cs

6.159 Mergelt.Game.Converters.IConfigProcessor Interface Reference

Inheritance diagram for Mergelt.Game.Converters.IConfigProcessor:



Public Member Functions

- `FieldData Convert (LevelConfig levelConfig)`
- `FieldElementData ConvertToFieldElementData (IFieldElement fieldElement)`
- `FieldElementData ConvertToFieldElementData (LevelElementData levelElementData)`
- `FieldData BuildLevel ()`
- `IFieldElement ConvertToFieldElement (FieldElementData data)`

6.159.1 Detailed Description

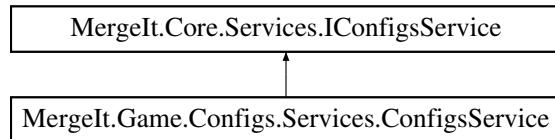
Definition at line 10 of file [IConfigProcessor.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/Converters/IConfigProcessor.cs

6.160 Mergelt.Core.Services.IConfigsService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IConfigsService:



Public Member Functions

- UniTask **Load** ()
- Sprite **GetCurrencyIcon** (CurrencyType type)
- EvolutionData **GetEvolutionData** (string evolutionId)
- string **GetEvolutionIdByElement** (ElementConfig element)
- EvolutionData **GetEvolutionByElement** (ElementConfig element)
- T **GetEffectConfig< T >** (string type)
- LevelUpParameters **GetLevelUpData** (int currentLevel)
- IFieldElementView **GetElementPrefab** (ElementType type)
- IEnumerable< ElementConfig > **GetConfigs** (Func< ElementConfig, bool > predicate=null)
- ElementConfig **GetConfig** (string id)

Properties

- LevelConfig **LevelConfig** [get]
- GameConfig **GameConfig** [get]
- InventoryConfig **InventoryConfig** [get]
- HintsConfig **HintsConfig** [get]

6.160.1 Detailed Description

Definition at line 19 of file [IConfigsService.cs](#).

6.160.2 Member Function Documentation

6.160.2.1 GetEffectConfig< T >()

```
T MergeIt.Core.Services.IConfigsService.GetEffectConfig< T > (
    string type )
```

Type Constraints

T : EffectConfig

6.160.3 Property Documentation

6.160.3.1 GameConfig

`GameConfig MergeIt.Core.Services.IConfigsService.GameConfig [get]`

Definition at line 23 of file [IConfigsService.cs](#).

```
00023 { get; }
```

6.160.3.2 HintsConfig

`HintsConfig MergeIt.Core.Services.IConfigsService.HintsConfig [get]`

Definition at line 25 of file [IConfigsService.cs](#).

```
00025 { get; }
```

6.160.3.3 InventoryConfig

`InventoryConfig MergeIt.Core.Services.IConfigsService.InventoryConfig [get]`

Definition at line 24 of file [IConfigsService.cs](#).

```
00024 { get; }
```

6.160.3.4 LevelConfig

`LevelConfig MergeIt.Core.Services.IConfigsService.LevelConfig [get]`

Definition at line 22 of file [IConfigsService.cs](#).

```
00022 { get; }
```

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IConfigsService.cs

6.161 Mergelt.Core.Configs.Icons.IconParameters Class Reference

Properties

- CurrencyType `CurrencyType [get]`
- Sprite `CurrencyIcon [get]`

6.161.1 Detailed Description

Definition at line 10 of file [IconParameters.cs](#).

6.161.2 Property Documentation

6.161.2.1 CurrencyIcon

Sprite MergeIt.Core.Configs.Icons.IconParameters.CurrencyIcon [get]

Definition at line 23 of file [IconParameters.cs](#).

```
00024     {
00025         get => _currencyIcon;
00026     }
```

6.161.2.2 CurrencyType

CurrencyType MergeIt.Core.Configs.Icons.IconParameters.CurrencyType [get]

Definition at line 18 of file [IconParameters.cs](#).

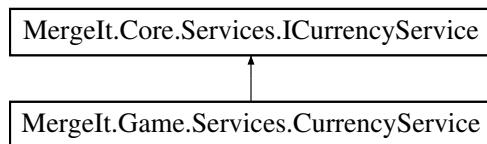
```
00019     {
00020         get => _currencyType;
00021     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Configs/Icons/IconParameters.cs

6.162 Mergelt.Core.Services.ICurrencyService Interface Reference

Inheritance diagram for Mergelt.Core.Services.ICurrencyService:



Public Member Functions

- bool [TryPay](#) ([CurrencySettings](#) currencySettings)
- void [Sell](#) ([CurrencySettings](#) currencySettings)

6.162.1 Detailed Description

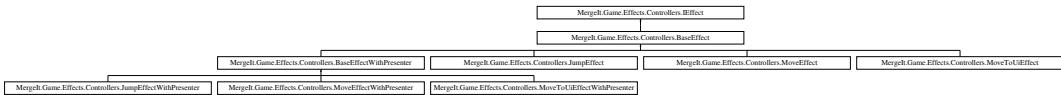
Definition at line 7 of file [ICurrencyService.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/ICurrencyService.cs

6.163 Mergelt.Game.Effects.Controllers.IEffect Interface Reference

Inheritance diagram for Mergelt.Game.Effects.Controllers.IEffect:



Public Member Functions

- void **Start** ()
- void **Update** ()
- void **Setup** (Transform target, [IEffectParameters](#) effectParameters=null, Action finishedCallback=null)

6.163.1 Detailed Description

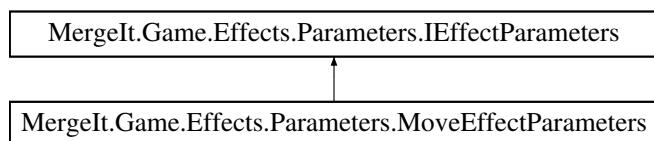
Definition at line 9 of file [IEffect.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/Effects/Controllers/IEffect.cs

6.164 Mergelt.Game.Effects.Parameters.IEffectParameters Interface Reference

Inheritance diagram for Mergelt.Game.Effects.Parameters.IEffectParameters:



Properties

- string [OverriddenName](#) [get]

6.164.1 Detailed Description

Definition at line 5 of file [IEffectParameters.cs](#).

6.164.2 Property Documentation

6.164.2.1 OverriddenName

```
string MergeIt.Game.Effects.Parameters.IEffectParameters.OverrideName [get]
```

Definition at line 7 of file [IEffectParameters.cs](#).

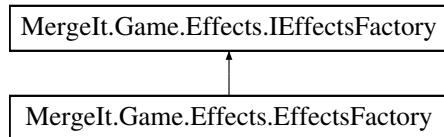
```
00007 { get; }
```

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Game/Effects/Parameters/IEffectParameters.cs

6.165 MergeIt.Game.Effects.IEffectsFactory Interface Reference

Inheritance diagram for MergeIt.Game.Effects.IEffectsFactory:



Public Member Functions

- void [CreateEffect< T >](#) (IFieldElementPresenter target, IEffectParameters effectParameters=null, Action finishedCallback=null)
- void [CreateEffect< T >](#) (RectTransform target, IEffectParameters effectParameters=null, Action finishedCallback=null)

6.165.1 Detailed Description

Definition at line 11 of file [IEffectsFactory.cs](#).

6.165.2 Member Function Documentation

6.165.2.1 CreateEffect< T >()

```
void MergeIt.Game.Effects.IEffectsFactory.CreateEffect< T > (
    IFieldElementPresenter target,
    IEffectParameters effectParameters = null,
    Action finishedCallback = null )
```

Type Constraints

T : IEffectWithPresenter

T : IEffect

T : new()

6.165.2.2 CreateEffect< T >() [2/2]

```
void MergeIt.Game.Effects.IEffectsFactory.CreateEffect< T > (
    RectTransform target,
    IEffectParameters effectParameters = null,
    Action finishedCallback = null )
```

Type Constraints

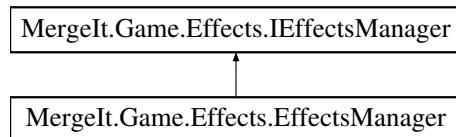
T : IEffect
T : new()

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/Effects/IEffectsFactory.cs

6.166 Mergelt.Game.Effects.IEffectsManager Interface Reference

Inheritance diagram for Mergelt.Game.Effects.IEffectsManager:



Public Member Functions

- void **AddEffect** (IEffect effectController)

6.166.1 Detailed Description

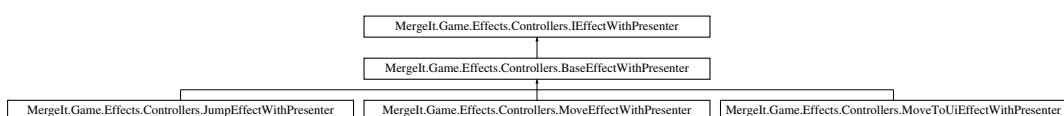
Definition at line 7 of file [IEffectsManager.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/Effects/IEffectsManager.cs

6.167 Mergelt.Game.Effects.Controllers.IEffectWithPresenter Interface Reference

Inheritance diagram for Mergelt.Game.Effects.Controllers.IEffectWithPresenter:



Public Member Functions

- void **Setup** ([IFieldElementPresenter](#) presenter, [IEffectParameters](#) effectParameters=null, Action finished←
Callback=null)
- [IFieldElementPresenter](#) **GetPresenter** ()

6.167.1 Detailed Description

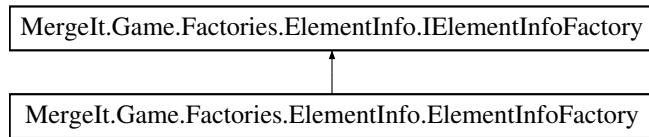
Definition at line 9 of file [IEffectWithPresenter.cs](#).

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Game/Effects/Controllers/IEffectWithPresenter.cs

6.168 MergeIt.Game.Factories.ElementInfo.IElementInfoFactory Interface Reference

Inheritance diagram for MergeIt.Game.Factories.ElementInfo.IElementInfoFactory:



Public Member Functions

- [ElementInfoItemComponent](#) **CreateElementWindowItem** ([ElementConfig](#) elementConfig, [ElementInfoType](#) infoType=[ElementInfoType.InfoWindow](#), bool isLocked=false)
- [ElementInfoItemComponent](#) **CreateUnknownElementWindowItem** ([ElementInfoType](#) infoType=[ElementInfoType.InfoWindow](#))

6.168.1 Detailed Description

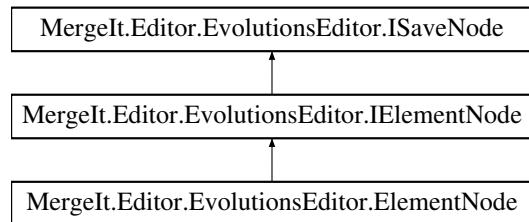
Definition at line 8 of file [IElementInfoFactory.cs](#).

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Game/Factories/ElementInfo/IElementInfoFactory.cs

6.169 Mergelt.Editor.EvolutionsEditor.IElementNode Interface Reference

Inheritance diagram for Mergelt.Editor.EvolutionsEditor.IElementNode:



Public Member Functions

- bool **Validate** ()

Public Member Functions inherited from [Mergelt.Editor.EvolutionsEditor.ISaveNode](#)

- void **SaveData** ()
- Port **GetPort** (string portName)

Properties

- [ElementConfig Config](#) [get]
- [IElementNodeData Data](#) [get]

Properties inherited from [Mergelt.Editor.EvolutionsEditor.ISaveNode](#)

- string **Id** [get]

6.169.1 Detailed Description

Definition at line 8 of file [IElementNode.cs](#).

6.169.2 Property Documentation

6.169.2.1 Config

[ElementConfig](#) MergeIt.Editor.EvolutionsEditor.IElementNode.Config [get]

Definition at line 10 of file [IElementNode.cs](#).
00010 { get; }

6.169.2.2 Data

`IElementNodeData MergeIt.Editor.EvolutionsEditor.IElementNode.Data [get]`

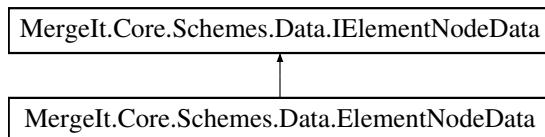
Definition at line 12 of file `IElementNode.cs`.
 00012 { `get;` }

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Editor/Core/EvolutionsEditor/ItemsNodes/IElementNode.cs

6.170 Mergelt.Core.Schemes.Data.IElementNodeData Interface Reference

Inheritance diagram for Mergelt.Core.Schemes.Data.IElementNodeData:



Public Member Functions

- `T Copy< T > ()`

Properties

- string `Id` [get, set]
- `ElementConfig ElementConfig` [get, set]
- `Rect Position` [get, set]

6.170.1 Detailed Description

Definition at line 8 of file `IElementNodeData.cs`.

6.170.2 Member Function Documentation

6.170.2.1 `Copy< T >()`

`T MergeIt.Core.Schemes.Data.IElementNodeData.Copy< T > ()`

Type Constraints

- `T : IElementNodeData`**
`T : new()`

6.170.3 Property Documentation

6.170.3.1 ElementConfig

```
ElementConfig MergeIt.Core.Schemes.Data.IElementNodeData.ElementConfig [get], [set]
```

Definition at line 11 of file [IElementNodeData.cs](#).

```
00011 { get; set; }
```

6.170.3.2 Id

```
string MergeIt.Core.Schemes.Data.IElementNodeData.Id [get], [set]
```

Definition at line 10 of file [IElementNodeData.cs](#).

```
00010 { get; set; }
```

6.170.3.3 Position

```
Rect MergeIt.Core.Schemes.Data.IElementNodeData.Position [get], [set]
```

Definition at line 12 of file [IElementNodeData.cs](#).

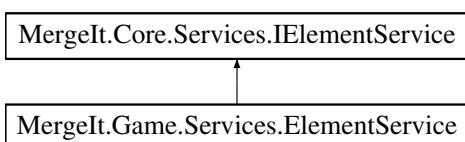
```
00012 { get; set; }
```

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Schemes/Data/IElementNodeData.cs

6.171 Mergelt.Core.Services.IElementService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IElementService:



Public Member Functions

- void **TrySell** ([IFieldElement](#) fieldElement)
- void **TryUnlock** ([IFieldElement](#) fieldElement)
- void **TrySplit** ([IFieldElement](#) fieldElement)

6.171.1 Detailed Description

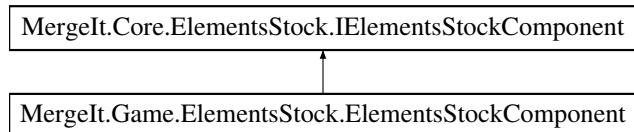
Definition at line 7 of file [IElementService.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IElementService.cs

6.172 Mergelt.Core.ElementsStock.IElementsStockComponent Interface Reference

Inheritance diagram for Mergelt.Core.ElementsStock.IElementsStockComponent:



Public Member Functions

- void **SetupElement** ([ElementConfig](#) elementConfig)
- void **PopElement** ()

6.172.1 Detailed Description

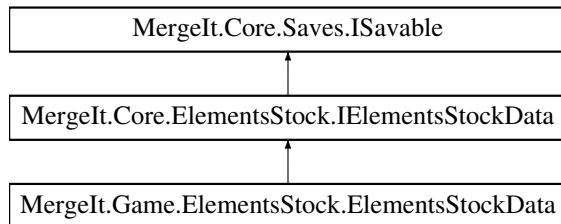
Definition at line [7](#) of file [IElementsStockComponent.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/ElementsStock/IElementsStockComponent.cs

6.173 Mergelt.Core.ElementsStock.IElementsStockData Interface Reference

Inheritance diagram for Mergelt.Core.ElementsStock.IElementsStockData:



Properties

- string[] **Elements** [get, set]

6.173.1 Detailed Description

Definition at line [7](#) of file [IElementsStockData.cs](#).

6.173.2 Property Documentation

6.173.2.1 Elements

```
string [] MergeIt.Core.ElementsStock.IElementsStockData.Elements [get], [set]
```

Definition at line 9 of file [IElementsStockData.cs](#).

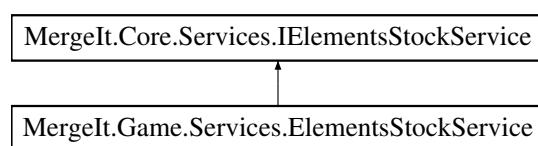
```
00009 { get; set; }
```

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/ElementsStock/IElementsStockData.cs

6.174 Mergelt.Core.Services.IElementsStockService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IElementsStockService:



Public Member Functions

- void **CreateStock** ()
- void **SetupStock** ([IElementsStockData](#) stockData)
- void **Add** ([ElementConfig](#) elementConfig)
- bool **Remove** ()
- [ElementConfig](#) **GetNext** ()
- [ElementConfig](#) **GetCurrent** ()
- [IElementsStockData](#) **GetData** ()

6.174.1 Detailed Description

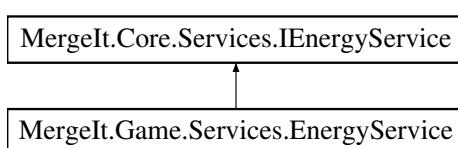
Definition at line 8 of file [IElementsStockService.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IElementsStockService.cs

6.175 Mergelt.Core.Services.IEnergyService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IEnergyService:



6.175.1 Detailed Description

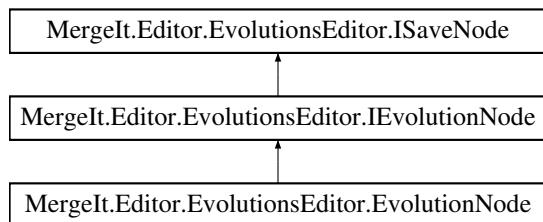
Definition at line 5 of file [IEnergyService.cs](#).

The documentation for this interface was generated from the following file:

- [Assets/MergeIt/Source/Core/Services/IEnergyService.cs](#)

6.176 [MergeIt.Editor.EvolutionsEditor.IEvolutionNode](#) Interface Reference

Inheritance diagram for [MergeIt.Editor.EvolutionsEditor.IEvolutionNode](#):



Properties

- string [Name](#) [get]
- string [Description](#) [get]
- bool [Discovered](#) [get]
- [IEvolutionNodeData](#) [Data](#) [get]

Properties inherited from [MergeIt.Editor.EvolutionsEditor.ISaveNode](#)

- string [Id](#) [get]

Additional Inherited Members

Public Member Functions inherited from [MergeIt.Editor.EvolutionsEditor.ISaveNode](#)

- void [SaveData](#) ()
- Port [GetPort](#) (string portName)

6.176.1 Detailed Description

Definition at line 7 of file [IEvolutionNode.cs](#).

6.176.2 Property Documentation

6.176.2.1 Data

```
IEvolutionNodeData MergeIt.Editor.EvolutionsEditor.IEvolutionNode.Data [get]
```

Definition at line 12 of file [IEvolutionNode.cs](#).
00012 { **get**; }

6.176.2.2 Description

```
string MergeIt.Editor.EvolutionsEditor.IEvolutionNode.Description [get]
```

Definition at line 10 of file [IEvolutionNode.cs](#).
00010 { **get**; }

6.176.2.3 Discovered

```
bool MergeIt.Editor.EvolutionsEditor.IEvolutionNode.Discovered [get]
```

Definition at line 11 of file [IEvolutionNode.cs](#).
00011 { **get**; }

6.176.2.4 Name

```
string MergeIt.Editor.EvolutionsEditor.IEvolutionNode.Name [get]
```

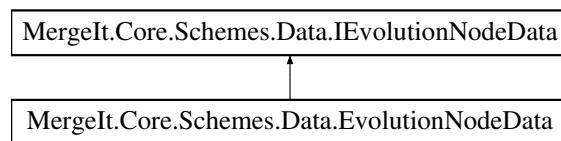
Definition at line 9 of file [IEvolutionNode.cs](#).
00009 { **get**; }

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Editor/Core/EvolutionsEditor/ItemsNodes/IEvolutionNode.cs

6.177 Mergelt.Core.Schemes.Data.IEvolutionNodeData Interface Reference

Inheritance diagram for Mergelt.Core.Schemes.Data.IEvolutionNodeData:



Public Member Functions

- T [Copy< T >](#) ()

Properties

- string `Id` [get, set]
- string `Name` [get, set]
- string `Description` [get, set]
- bool `Discovered` [get, set]
- Rect `Position` [get, set]

6.177.1 Detailed Description

Definition at line 7 of file [IEvolutionNodeData.cs](#).

6.177.2 Member Function Documentation

6.177.2.1 `Copy< T >()`

`T MergeIt.Core.Schemes.Data.IEvolutionNodeData.Copy< T > ()`

Type Constraints

`T : IEvolutionNodeData`

`T : new()`

6.177.3 Property Documentation

6.177.3.1 Description

`string MergeIt.Core.Schemes.Data.IEvolutionNodeData.Description [get], [set]`

Definition at line 11 of file [IEvolutionNodeData.cs](#).

`00011 { get; set; }`

6.177.3.2 Discovered

`bool MergeIt.Core.Schemes.Data.IEvolutionNodeData.Discovered [get], [set]`

Definition at line 12 of file [IEvolutionNodeData.cs](#).

`00012 { get; set; }`

6.177.3.3 Id

`string MergeIt.Core.Schemes.Data.IEvolutionNodeData.Id [get], [set]`

Definition at line 9 of file [IEvolutionNodeData.cs](#).

`00009 { get; set; }`

6.177.3.4 Name

```
string MergeIt.Core.Schemes.Data.IEvolutionNodeData.Name [get], [set]
```

Definition at line 10 of file [IEvolutionNodeData.cs](#).
00010 { get; set; }

6.177.3.5 Position

```
Rect MergeIt.Core.Schemes.Data.IEvolutionNodeData.Position [get], [set]
```

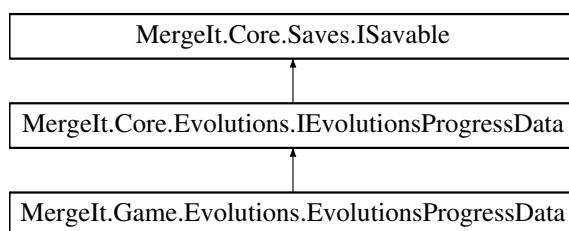
Definition at line 14 of file [IEvolutionNodeData.cs](#).
00014 { get; set; }

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Schemes/Data/IEvolutionNodeData.cs

6.178 Mergelt.Core.Evolutions.IEvolutionsProgressData Interface Reference

Inheritance diagram for Mergelt.Core.Evolutions.IEvolutionsProgressData:



Properties

- List< [EvolutionProgressData](#) > [EvolutionsProgress](#) [get]

6.178.1 Detailed Description

Definition at line 8 of file [IEvolutionsProgressData.cs](#).

6.178.2 Property Documentation

6.178.2.1 EvolutionsProgress

```
List<EvolutionProgressData> MergeIt.Core.Evolutions.IEvolutionsProgressData.EvolutionsProgress  
[get]
```

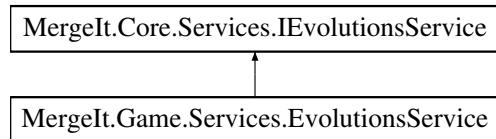
Definition at line 10 of file [IEvolutionsProgressData.cs](#).
00010 { get; }

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Evolutions/IEvolutionsProgressData.cs

6.179 Mergelt.Core.Services.IEvolutionsService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IEvolutionsService:



Public Member Functions

- List<[ElementConfig](#)> **GetEvolutionChain** ([EvolutionData](#) evolutionData)
- List<[ElementConfig](#)> **GetGeneratedBy** ([ElementConfig](#) config)
- List<[ElementConfig](#)> **GetGenerates** ([ElementConfig](#) config)
- void **SetupEvolutionsProgress** ([IEvolutionsProgressData](#) data)
- [IEvolutionsProgressData](#) **GetData** ()
- void **CreateEvolutionsProgress** ()
- void **UpdateProgress** ([ElementConfig](#) elementId)
- int **GetEvolutionProgress** (string id)

6.179.1 Detailed Description

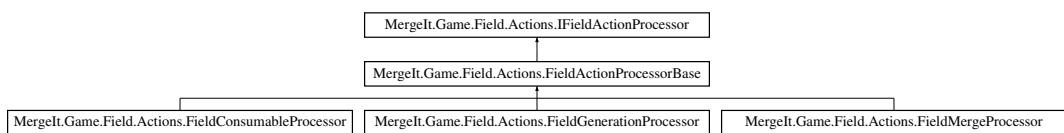
Definition at line 10 of file [IEvolutionsService.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IEvolutionsService.cs

6.180 Mergelt.Game.Field.Actions.IFieldActionProcessor Interface Reference

Inheritance diagram for Mergelt.Game.Field.Actions.IFieldActionProcessor:



Public Member Functions

- void **ProcessClick** ([FieldCellComponent](#) cellComponent)
- void **ProcessEndDrag** ([GridPoint](#) fromPoint, [GameObject](#) toGameObject)

6.180.1 Detailed Description

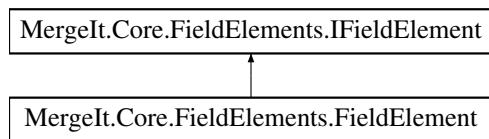
Definition at line 8 of file [IFieldActionProcessor.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/Field/Actions/IFieldActionProcessor.cs

6.181 Mergelt.Core.FieldElements.IFieldElement Interface Reference

Inheritance diagram for Mergelt.Core.FieldElements.IFieldElement:



Properties

- [IConfigParameters ConfigParameters \[get, set\]](#)
- [IInfoParameters InfoParameters \[get, set\]](#)
- [IGeneratorParameters GeneratorParameters \[get, set\]](#)
- [IGeneratorOpenParameters GeneratorOpenParameters \[get, set\]](#)
- [IProduceParameters ProduceParameters \[get, set\]](#)
- [IProducedByParameters ProducedByParameters \[get, set\]](#)

6.181.1 Detailed Description

Definition at line 5 of file [IFieldElement.cs](#).

6.181.2 Property Documentation

6.181.2.1 ConfigParameters

[IConfigParameters](#) MergeIt.Core.FieldElements.IFieldElement.ConfigParameters [get], [set]

Definition at line 7 of file [IFieldElement.cs](#).

```
00007 { get; set; }
```

6.181.2.2 GeneratorOpenParameters

[IGeneratorOpenParameters](#) MergeIt.Core.FieldElements.IFieldElement.GeneratorOpenParameters [get], [set]

Definition at line 10 of file [IFieldElement.cs](#).

```
00010 { get; set; }
```

6.181.2.3 GeneratorParameters

`IGeneratorParameters MergeIt.Core.FieldElements.IFieldElement.GeneratorParameters [get], [set]`

Definition at line 9 of file [IFieldElement.cs](#).
00009 { get; set; }

6.181.2.4 InfoParameters

`IIInfoParameters MergeIt.Core.FieldElements.IFieldElement.InfoParameters [get], [set]`

Definition at line 8 of file [IFieldElement.cs](#).
00008 { get; set; }

6.181.2.5 ProducedByParameters

`IProducedByParameters MergeIt.Core.FieldElements.IFieldElement.ProducedByParameters [get], [set]`

Definition at line 12 of file [IFieldElement.cs](#).
00012 { get; set; }

6.181.2.6 ProduceParameters

`IProduceParameters MergeIt.Core.FieldElements.IFieldElement.ProduceParameters [get], [set]`

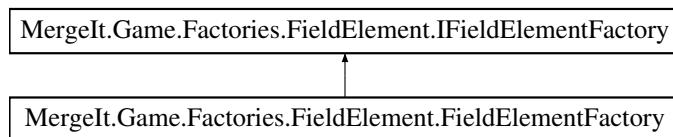
Definition at line 11 of file [IFieldElement.cs](#).
00011 { get; set; }

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/FieldElements/IFieldElement.cs

6.182 MergeIt.Game.Factories.FieldElement.IFieldElementFactory Interface Reference

Inheritance diagram for MergeIt.Game.Factories.FieldElement.IFieldElementFactory:



Public Member Functions

- `IFieldElement CreateFieldElement (FieldElementData data)`
- `IFieldElement CreateFieldElement (LevelElementData data)`
- `IFieldElement CreateFieldElement (ElementConfig elementConfig, GridPoint point, bool isBlocked=false)`

6.182.1 Detailed Description

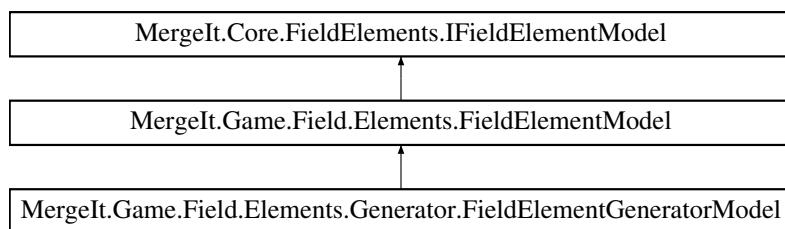
Definition at line 9 of file [IFieldElementFactory.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/Factories/FieldElement/IFieldElementFactory.cs

6.183 Mergelt.Core.FieldElements.IFieldElementModel Interface Reference

Inheritance diagram for Mergelt.Core.FieldElements.IFieldElementModel:



Properties

- bool [IsBusy](#) [get, set]
- bool [IsLocked](#) [get, set]
- bool [Selected](#) [get, set]
- int [ClicksCount](#) [get, set]
- GridPoint [Point](#) [get, set]

6.183.1 Detailed Description

Definition at line 5 of file [IFieldElementModel.cs](#).

6.183.2 Property Documentation

6.183.2.1 ClicksCount

```
int MergeIt.Core.FieldElements.IFieldElementModel.ClicksCount [get], [set]
```

Definition at line 10 of file [IFieldElementModel.cs](#).

```
00010 { get; set; }
```

6.183.2.2 IsBusy

```
bool MergeIt.Core.FieldElements.IFieldElementModel.IsBusy [get], [set]
```

Definition at line 7 of file [IFieldElementModel.cs](#).

```
00007 { get; set; }
```

6.183.2.3 IsLocked

```
bool MergeIt.Core.FieldElements.IFieldElementModel.IsLocked [get], [set]
```

Definition at line 8 of file [IFieldElementModel.cs](#).

```
00008 { get; set; }
```

6.183.2.4 Point

```
GridPoint MergeIt.Core.FieldElements.IFieldElementModel.Point [get], [set]
```

Definition at line 11 of file [IFieldElementModel.cs](#).

```
00011 { get; set; }
```

6.183.2.5 Selected

```
bool MergeIt.Core.FieldElements.IFieldElementModel.Selected [get], [set]
```

Definition at line 9 of file [IFieldElementModel.cs](#).

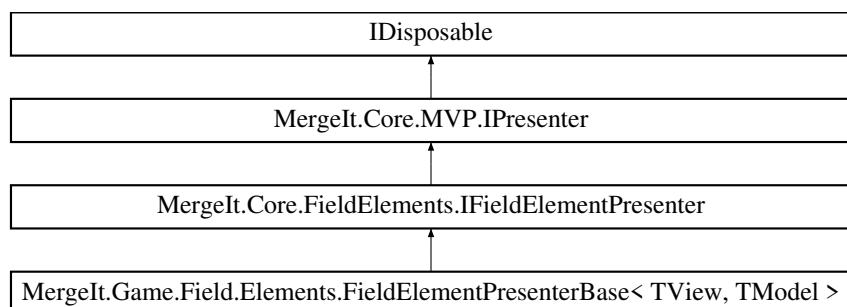
```
00009 { get; set; }
```

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/FieldElements/IFieldElementModel.cs

6.184 MergeIt.Core.FieldElements.IFieldElementPresenter Interface Reference

Inheritance diagram for MergeIt.Core.FieldElements.IFieldElementPresenter:



Public Member Functions

- void **Update** ([IFieldElement](#) fieldElement)
- void **Activate** (bool isActive)
- void **Release** ()
- void **SetBusy** (bool isBusy)
- void **Select** (bool isSelected)
- void **StartDrag** ()
- void **EndDrag** ()
- void **SetParent** (Transform parent, bool resetPosition)
- void **SetPoint** ([GridPoint](#) point)
- void **SetLock** (bool block)
- void **Remove** ()
- void **ResetPosition** ()
- [IFieldElementView](#) **GetView** ()
- [IFieldElementModel](#) **GetModel** ()
- void **SetState** (FieldElementState state)

Public Member Functions inherited from [MergeIt.Core.MVP.IPresenter](#)

- void [Initialize](#) ([IView](#) view)

Properties

- FieldElementState [State](#) [get]
- Transform [Transform](#) [get]
- RectTransform [RectTransform](#) [get]
- Canvas [Canvas](#) [get]
- bool [IsAvailable](#) [get]
- bool [IsBusy](#) [get]
- bool [IsLocked](#) [get]

6.184.1 Detailed Description

Definition at line 8 of file [IFieldElementPresenter.cs](#).

6.184.2 Property Documentation

6.184.2.1 [Canvas](#)

Canvas MergeIt.Core.FieldElements.IFieldElementPresenter.Canvas [get]

Definition at line 13 of file [IFieldElementPresenter.cs](#).
00013 { [get](#); }

6.184.2.2 [IsAvailable](#)

bool MergeIt.Core.FieldElements.IFieldElementPresenter.IsAvailable [get]

Definition at line 14 of file [IFieldElementPresenter.cs](#).
00014 { [get](#); }

6.184.2.3 [IsBusy](#)

bool MergeIt.Core.FieldElements.IFieldElementPresenter.IsBusy [get]

Definition at line 15 of file [IFieldElementPresenter.cs](#).
00015 { [get](#); }

6.184.2.4 [IsLocked](#)

bool MergeIt.Core.FieldElements.IFieldElementPresenter.IsLocked [get]

Definition at line 16 of file [IFieldElementPresenter.cs](#).
00016 { [get](#); }

6.184.2.5 RectTransform

RectTransform MergeIt.Core.FieldElements.IFieldElementPresenter.RectTransform [get]

Definition at line 12 of file [IFieldElementPresenter.cs](#).

```
00012 { get; }
```

6.184.2.6 State

FieldElementState MergeIt.Core.FieldElements.IFieldElementPresenter.State [get]

Definition at line 10 of file [IFieldElementPresenter.cs](#).

```
00010 { get; }
```

6.184.2.7 Transform

Transform MergeIt.Core.FieldElements.IFieldElementPresenter.Transform [get]

Definition at line 11 of file [IFieldElementPresenter.cs](#).

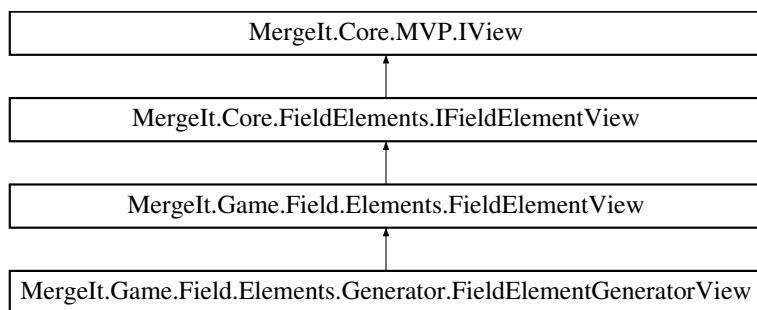
```
00011 { get; }
```

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/FieldElements/IFieldElementPresenter.cs

6.185 MergeIt.Core.FieldElements.IFieldElementView Interface Reference

Inheritance diagram for MergeIt.Core.FieldElements.IFieldElementView:



Public Member Functions

- void **Lock** (bool isLocked)
- void **ResetState** ()

Public Member Functions inherited from [MergeIt.Core.MVP.IView](#)

- void **Initialize** ()

Properties

- RectTransform [RectTransform](#) [get]
- Canvas [Canvas](#) [get]
- GraphicRaycaster [GraphicRaycaster](#) [get]

Properties inherited from [Mergelt.Core.MVP.IView](#)

- GameObject [GameObject](#) [get]

Additional Inherited Members

Events inherited from [Mergelt.Core.MVP.IView](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

6.185.1 Detailed Description

Definition at line 9 of file [IFieldElementView.cs](#).

6.185.2 Property Documentation

6.185.2.1 Canvas

Canvas [MergeIt.Core.FieldElements.IFieldElementView.Canvas](#) [get]

Definition at line 12 of file [IFieldElementView.cs](#).

```
00012 { get; }
```

6.185.2.2 GraphicRaycaster

GraphicRaycaster [MergeIt.Core.FieldElements.IFieldElementView.GraphicRaycaster](#) [get]

Definition at line 13 of file [IFieldElementView.cs](#).

```
00013 { get; }
```

6.185.2.3 RectTransform

RectTransform [MergeIt.Core.FieldElements.IFieldElementView.RectTransform](#) [get]

Definition at line 11 of file [IFieldElementView.cs](#).

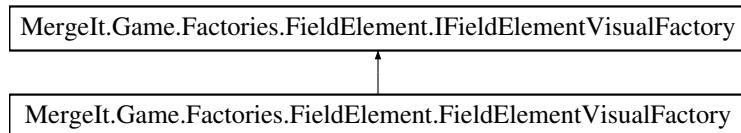
```
00011 { get; }
```

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/IFieldElementView.cs

6.186 Mergelt.Game.Factories.FieldElement.IFieldElementVisualFactory Interface Reference

Inheritance diagram for Mergelt.Game.Factories.FieldElement.IFieldElementVisualFactory:



Public Member Functions

- [IFieldElementPresenter CreateFieldElement](#) ([IFieldElement](#) element)

6.186.1 Detailed Description

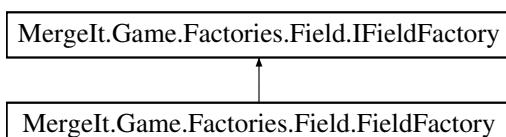
Definition at line 7 of file [IFieldElementVisualFactory.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/Factories/FieldElement/IFieldElementVisualFactory.cs

6.187 Mergelt.Game.Factories.Field.IFieldFactory Interface Reference

Inheritance diagram for Mergelt.Game.Factories.Field.IFieldFactory:



Public Member Functions

- [FieldPresenter CreateField](#) (Transform parent)
- [FieldCellComponent CreateFieldCell](#) (Transform parent, [GridPoint](#) gridPoint)

6.187.1 Detailed Description

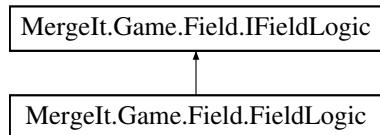
Definition at line 9 of file [IFieldFactory.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/Factories/Field/IFieldFactory.cs

6.188 Mergelt.Game.Field.IFieldLogic Interface Reference

Inheritance diagram for Mergelt.Game.Field.IFieldLogic:



6.188.1 Detailed Description

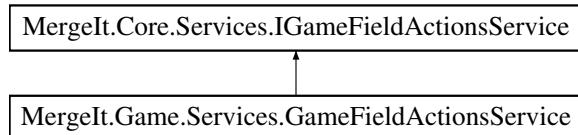
Definition at line 5 of file [IFieldLogic.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/Field/IFieldLogic.cs

6.189 Mergelt.Core.Services.IGameFieldActionsService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IGameFieldActionsService:



6.189.1 Detailed Description

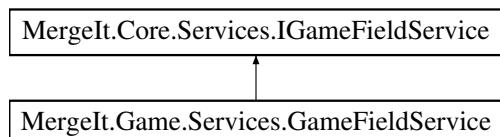
Definition at line 5 of file [IGameFieldActionsService.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IGameFieldActionsService.cs

6.190 Mergelt.Core.Services.IGameFieldService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IGameFieldService:



Public Member Functions

- `GridPoint? GetFreeCell ()`
- `IFieldElement CreateNewElement (ElementConfig config, GridPoint point, bool blocked=false)`

6.190.1 Detailed Description

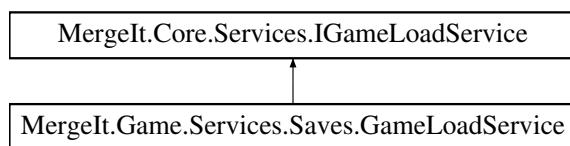
Definition at line 8 of file [IGameFieldService.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IGameFieldService.cs

6.191 Mergelt.Core.Services.IGameLoadService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IGameLoadService:



Public Member Functions

- `T Load< T > ()`

6.191.1 Detailed Description

Definition at line 7 of file [IGameLoadService.cs](#).

6.191.2 Member Function Documentation

6.191.2.1 Load< T >()

`T MergeIt.Core.Services.IGameLoadService.Load< T > ()`

Type Constraints

`T : class`

`T : ISavable`

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IGameLoadService.cs

6.192 Mergelt.Core.Services.IGameSaveService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IGameSaveService:



Public Member Functions

- UniTask **Save** (GameSaveType saveType)

6.192.1 Detailed Description

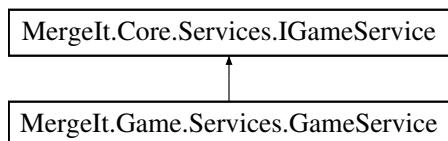
Definition at line 8 of file [IGameSaveService.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IGameSaveService.cs

6.193 Mergelt.Core.Services.IGameService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IGameService:



6.193.1 Detailed Description

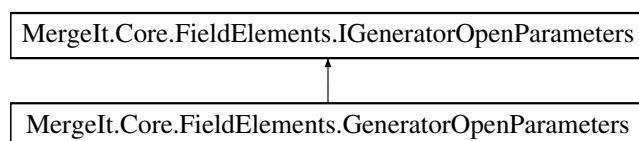
Definition at line 5 of file [IGameService.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IGameService.cs

6.194 Mergelt.Core.FieldElements.IGeneratorOpenParameters Interface Reference

Inheritance diagram for Mergelt.Core.FieldElements.IGeneratorOpenParameters:



Public Member Functions

- void **CopyFrom** ([SavedGeneratorOpenParameters](#) other)

Properties

- bool **IsOpening** [get]
- long **StartOpeningTime** [get, set]
- **Bindable**<float> **RemainingTime** [get, set]

6.194.1 Detailed Description

Definition at line [7](#) of file [IGeneratorOpenParameters.cs](#).

6.194.2 Property Documentation

6.194.2.1 IsOpening

```
bool MergeIt.Core.FieldElements.IGeneratorOpenParameters.IsOpening [get]
```

Definition at line [9](#) of file [IGeneratorOpenParameters.cs](#).
00009 { **get**; }

6.194.2.2 RemainingTime

```
Bindable<float> MergeIt.Core.FieldElements.IGeneratorOpenParameters.RemainingTime [get],  
[set]
```

Definition at line [11](#) of file [IGeneratorOpenParameters.cs](#).
00011 { **get**; **set**; }

6.194.2.3 StartOpeningTime

```
long MergeIt.Core.FieldElements.IGeneratorOpenParameters.StartOpeningTime [get], [set]
```

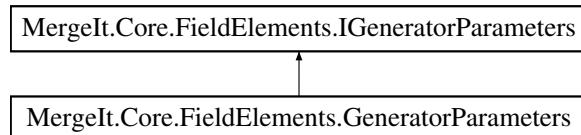
Definition at line [10](#) of file [IGeneratorOpenParameters.cs](#).
00010 { **get**; **set**; }

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/FieldElements/Parameters/IGeneratorOpenParameters.cs

6.195 Mergelt.Core.FieldElements.IGeneratorParameters Interface Reference

Inheritance diagram for Mergelt.Core.FieldElements.IGeneratorParameters:



Public Member Functions

- void [CopyFrom](#) ([SavedGeneratorParameters](#) other)

Properties

- bool [Charging](#) [get]
- int [AvailableToDrop](#) [get, set]
- int [ChargedCount](#) [get, set]
- long [StartChargingTime](#) [get, set]
- [Bindable< float > MinDropChargeTime](#) [get, set]
- float [MinDropFullChargeTime](#) [get, set]
- float [RemainChargeTime](#) [get, set]
- int [DroppedElements](#) [get, set]

6.195.1 Detailed Description

Definition at line 7 of file [IGeneratorParameters.cs](#).

6.195.2 Property Documentation

6.195.2.1 AvailableToDrop

```
int MergeIt.Core.FieldElements.IGeneratorParameters.AvailableToDrop [get], [set]
```

Definition at line 11 of file [IGeneratorParameters.cs](#).
00011 { get; set; }

6.195.2.2 ChargedCount

```
int MergeIt.Core.FieldElements.IGeneratorParameters.ChargedCount [get], [set]
```

Definition at line 12 of file [IGeneratorParameters.cs](#).
00012 { get; set; }

6.195.2.3 Charging

```
bool MergeIt.Core.FieldElements.IGeneratorParameters.Charging [get]
```

Definition at line 9 of file [IGeneratorParameters.cs](#).
00009 { get; }

6.195.2.4 DroppedElements

```
int MergeIt.Core.FieldElements.IGeneratorParameters.DroppedElements [get], [set]
```

Definition at line 17 of file [IGeneratorParameters.cs](#).
00017 { get; set; }

6.195.2.5 MinDropChargeTime

```
Bindable<float> MergeIt.Core.FieldElements.IGeneratorParameters.MinDropChargeTime [get], [set]
```

Definition at line 14 of file [IGeneratorParameters.cs](#).
00014 { get; set; }

6.195.2.6 MinDropFullChargeTime

```
float MergeIt.Core.FieldElements.IGeneratorParameters.MinDropFullChargeTime [get], [set]
```

Definition at line 15 of file [IGeneratorParameters.cs](#).
00015 { get; set; }

6.195.2.7 RemainChargeTime

```
float MergeIt.Core.FieldElements.IGeneratorParameters.RemainChargeTime [get], [set]
```

Definition at line 16 of file [IGeneratorParameters.cs](#).
00016 { get; set; }

6.195.2.8 StartChargingTime

```
long MergeIt.Core.FieldElements.IGeneratorParameters.StartChargingTime [get], [set]
```

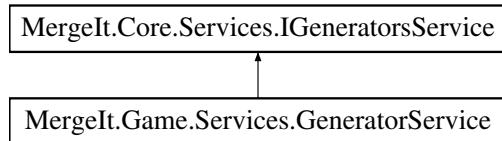
Definition at line 13 of file [IGeneratorParameters.cs](#).
00013 { get; set; }

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/FieldElements/Parameters/IGeneratorParameters.cs

6.196 Mergelt.Core.Services.IGeneratorsService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IGeneratorsService:



Public Member Functions

- void **TryOpen** ([IFieldElement](#) generator)
- void **TrySkipOpening** ([IFieldElement](#) generator)
- void **TrySkipCharging** ([IFieldElement](#) generator)

6.196.1 Detailed Description

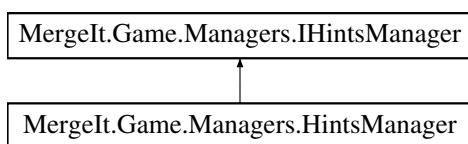
Definition at line [7](#) of file [IGeneratorsService.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IGeneratorsService.cs

6.197 Mergelt.Game.Managers.IHintsManager Interface Reference

Inheritance diagram for Mergelt.Game.Managers.IHintsManager:



6.197.1 Detailed Description

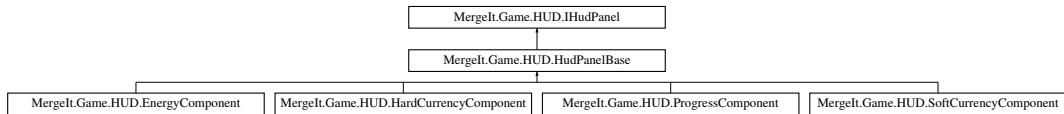
Definition at line [5](#) of file [IHintsManager.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/Managers/IHintsManager.cs

6.198 Mergelt.Game.HUD.IHudPanel Interface Reference

Inheritance diagram for Mergelt.Game.HUD.IHudPanel:



Properties

- HudPanelType [Type](#) [get]

6.198.1 Detailed Description

Definition at line 7 of file [IHudPanel.cs](#).

6.198.2 Property Documentation

6.198.2.1 Type

HudPanelType MergeIt.Game.HUD.IHudPanel.Type [get]

Definition at line 9 of file [IHudPanel.cs](#).

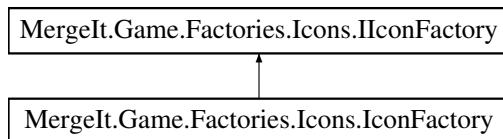
```
00009 { get; }
```

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/HUD/IHudPanel.cs

6.199 Mergelt.Game.Factories.Icons.IIconFactory Interface Reference

Inheritance diagram for Mergelt.Game.Factories.Icons.IIconFactory:



Public Member Functions

- [FieldElementIconComponent CreateIcon](#) ([ElementConfig](#) config, Transform parent=null)

6.199.1 Detailed Description

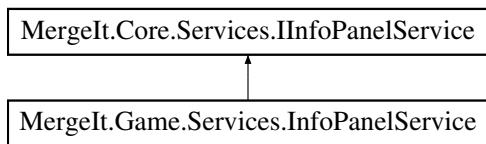
Definition at line 8 of file [IIconFactory.cs](#).

The documentation for this interface was generated from the following file:

- [Assets/Mergelt/Source/Game/Factories/Icons/IIconFactory.cs](#)

6.200 Mergelt.Core.Services.IInfoPanelService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IInfoPanelService:



6.200.1 Detailed Description

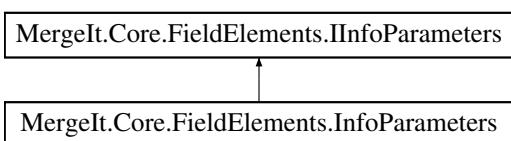
Definition at line 5 of file [IInfoPanelService.cs](#).

The documentation for this interface was generated from the following file:

- [Assets/Mergelt/Source/Core/Services/IInfoPanelService.cs](#)

6.201 Mergelt.Core.FieldElements.IInfoParameters Interface Reference

Inheritance diagram for Mergelt.Core.FieldElements.IInfoParameters:



Properties

- `GridPoint LogicPosition [get, set]`
- `bool IsBlocked [get, set]`
- `ElementType Type [get, set]`
- `string Name [get, set]`
- `string Description [get, set]`

6.201.1 Detailed Description

Definition at line 7 of file [IInfoParameters.cs](#).

6.201.2 Property Documentation

6.201.2.1 Description

```
string MergeIt.Core.FieldElements.IInfoParameters.Description [get], [set]
```

Definition at line 13 of file [IInfoParameters.cs](#).

```
00013 { get; set; }
```

6.201.2.2 IsBlocked

```
bool MergeIt.Core.FieldElements.IInfoParameters.IsBlocked [get], [set]
```

Definition at line 10 of file [IInfoParameters.cs](#).

```
00010 { get; set; }
```

6.201.2.3 LogicPosition

```
GridPoint MergeIt.Core.FieldElements.IInfoParameters.LogicPosition [get], [set]
```

Definition at line 9 of file [IInfoParameters.cs](#).

```
00009 { get; set; }
```

6.201.2.4 Name

```
string MergeIt.Core.FieldElements.IInfoParameters.Name [get], [set]
```

Definition at line 12 of file [IInfoParameters.cs](#).

```
00012 { get; set; }
```

6.201.2.5 Type

```
ElementType MergeIt.Core.FieldElements.IInfoParameters.Type [get], [set]
```

Definition at line 11 of file [IInfoParameters.cs](#).

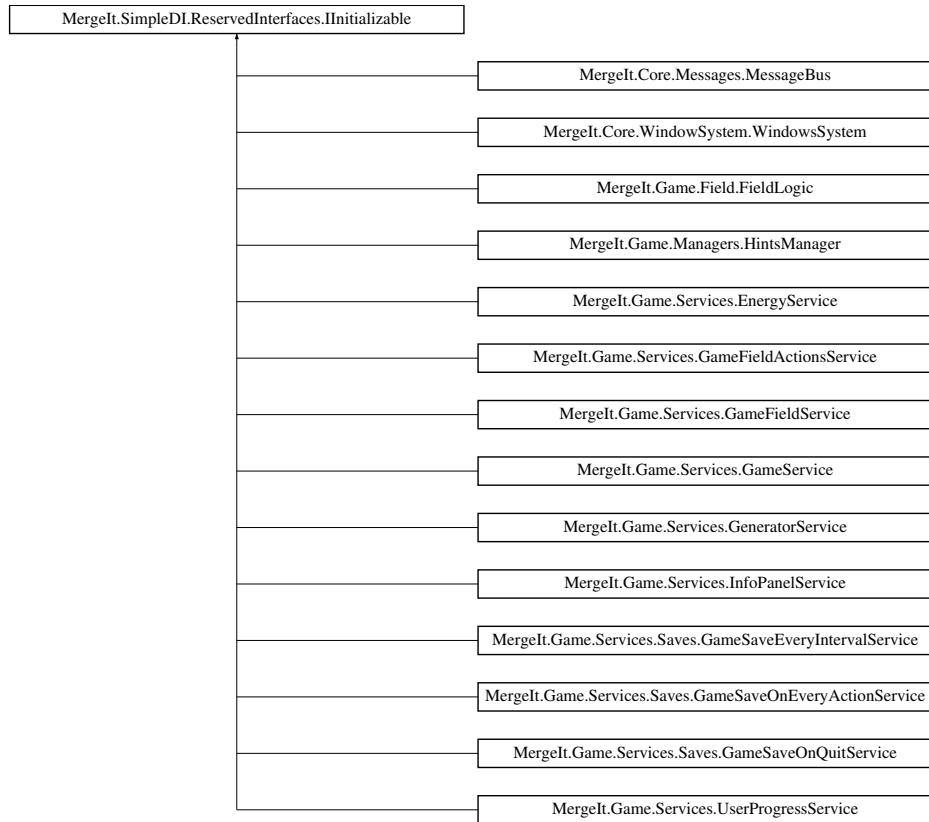
```
00011 { get; set; }
```

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/FieldElements/Parameters/IInfoParameters.cs

6.202 Mergelt.SimpleDI.ReservedInterfaces.IInitializable Interface Reference

Inheritance diagram for Mergelt.SimpleDI.ReservedInterfaces.IInitializable:



Public Member Functions

- void **Initialize** ()

6.202.1 Detailed Description

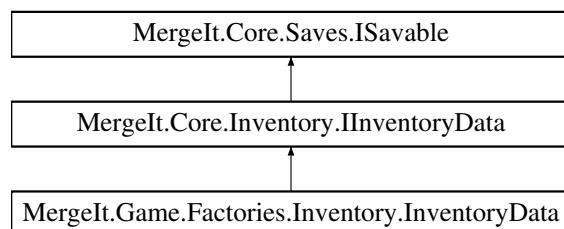
Definition at line 5 of file [IInitializable.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/SimpleDI/ReservedInterfaces/IInitializable.cs

6.203 Mergelt.Core.Inventory.IInventoryData Interface Reference

Inheritance diagram for Mergelt.Core.Inventory.IInventoryData:



Properties

- int `InventorySize` [get, set]
- `FieldElementData[] InventoryElements` [get, set]

6.203.1 Detailed Description

Definition at line 8 of file `IInventoryData.cs`.

6.203.2 Property Documentation

6.203.2.1 `InventoryElements`

`FieldElementData[] MergeIt.Core.Inventory.IInventoryData.InventoryElements` [get], [set]

Definition at line 11 of file `IInventoryData.cs`.
`00011 { get; set; }`

6.203.2.2 `InventorySize`

`int MergeIt.Core.Inventory.IInventoryData.InventorySize` [get], [set]

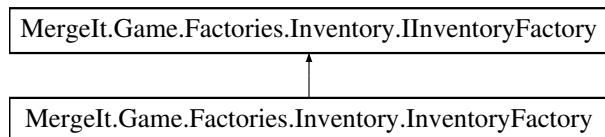
Definition at line 10 of file `IInventoryData.cs`.
`00010 { get; set; }`

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/Inventory/IInventoryData.cs

6.204 MergeIt.Game.Factories.Inventory.IInventoryFactory Interface Reference

Inheritance diagram for MergeIt.Game.Factories.Inventory.IInventoryFactory:



Public Member Functions

- `InventoryPanelItemComponent CreateInventoryPanelItem (IFieldElement fieldElement)`
- `InventoryWindowItemComponent CreateInventoryWindowItem (IFieldElement fieldElement)`
- `InventoryWindowPaidCellComponent CreateWindowPaidCell ()`
- `GameObject CreateWindowEmptyCell ()`

6.204.1 Detailed Description

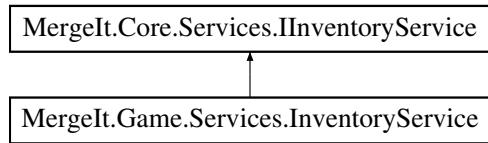
Definition at line 10 of file [IInventoryFactory.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/Factories/Inventory/IInventoryFactory.cs

6.205 Mergelt.Core.Services.IInventoryService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IInventoryService:



Public Member Functions

- void **CreateInventory** ()
- void **SetupInventory** ([IInventoryData](#) inventoryData)
- [IInventoryData](#) **GetData** ()
- void **Add** ([IFieldElement](#) element)
- bool **Remove** ([IFieldElement](#) fieldElement)
- void **OpenWindow** ()

6.205.1 Detailed Description

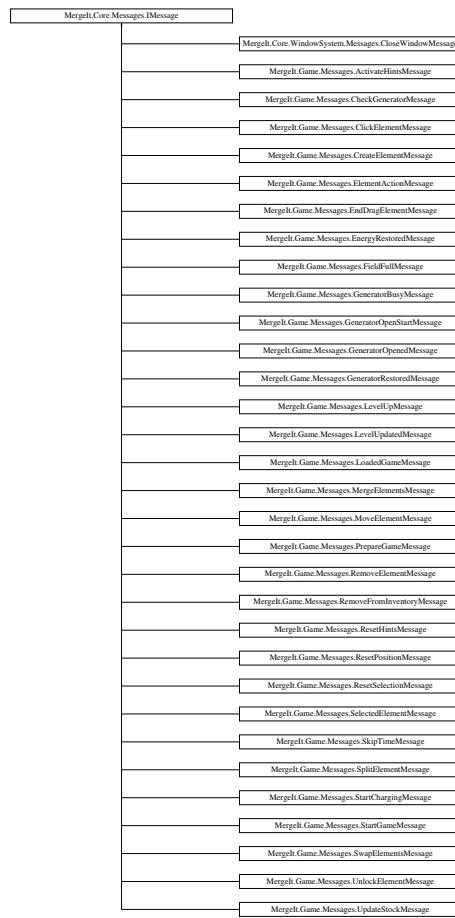
Definition at line 8 of file [IInventoryService.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IInventoryService.cs

6.206 Mergelt.Core.Messages.IMessage Interface Reference

Inheritance diagram for Mergelt.Core.Messages.IMessage:



6.206.1 Detailed Description

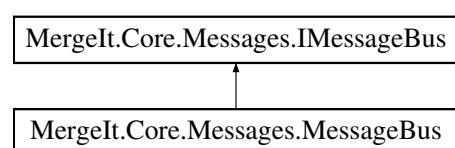
Definition at line 5 of file [IMessage.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Messages/IMessage.cs

6.207 Mergelt.Core.Messages.IMessageBus Interface Reference

Inheritance diagram for Mergelt.Core.Messages.IMessageBus:



Public Member Functions

- void `DefineMessage< T >()`
- void `AddListener< T >(Action< T > callback)`
- void `RemoveListener< T >(Action< T > callback)`
- void `Fire< T >(T messageData)`
- void `Fire< T >()`

6.207.1 Detailed Description

Definition at line 7 of file `IMessageBus.cs`.

6.207.2 Member Function Documentation

6.207.2.1 AddListener< T >()

```
void MergeIt.Core.Messages.IMessageBus.AddListener< T >(
    Action< T > callback )
```

Type Constraints

T : IMessage

6.207.2.2 DefineMessage< T >()

```
void MergeIt.Core.Messages.IMessageBus.DefineMessage< T >()
```

Type Constraints

T : IMessage

6.207.2.3 Fire< T >() [1/2]

```
void MergeIt.Core.Messages.IMessageBus.Fire< T >()
```

Type Constraints

T : IMessage

T : new()

6.207.2.4 Fire< T >() [2/2]

```
void MergeIt.Core.Messages.IMessageBus.Fire< T >(
    T messageData )
```

Type Constraints

T : IMessage

6.207.2.5 RemoveListener< T >()

```
void MergeIt.Core.Messages.IMessageBus.RemoveListener< T > (
    Action< T > callback )
```

Type Constraints

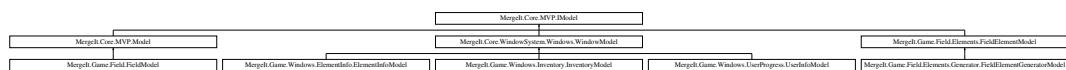
T : IMessages

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/Messages/IMessageBus.cs

6.208 MergeIt.Core.MVP.IModel Interface Reference

Inheritance diagram for MergeIt.Core.MVP.IModel:



6.208.1 Detailed Description

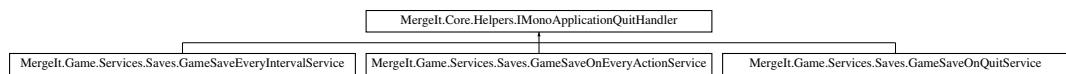
Definition at line 5 of file [IModel.cs](#).

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/MVP/IModel.cs

6.209 MergeIt.Core.Helpers.IMonoApplicationQuitHandler Interface Reference

Inheritance diagram for MergeIt.Core.Helpers.IMonoApplicationQuitHandler:



Public Member Functions

- void **OnApplicationQuit ()**

6.209.1 Detailed Description

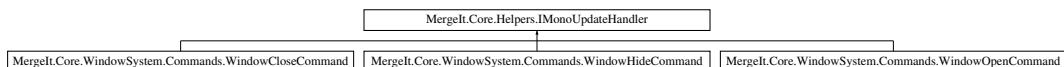
Definition at line 5 of file [IMonoApplicationQuitHandler.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Utils/IMonoApplicationQuitHandler.cs

6.210 Mergelt.Core.Helpers.IMonoUpdateHandler Interface Reference

Inheritance diagram for Mergelt.Core.Helpers.IMonoUpdateHandler:



Public Member Functions

- void **Update** ()

6.210.1 Detailed Description

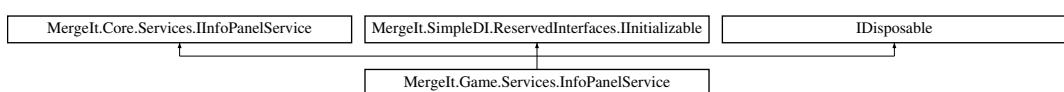
Definition at line 5 of file [IMonoUpdateHandler.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Utils/IMonoUpdateHandler.cs

6.211 Mergelt.Game.Services.InfoPanelService Class Reference

Inheritance diagram for Mergelt.Game.Services.InfoPanelService:



Public Member Functions

- void [Dispose](#) ()
- void [Initialize](#) ()

6.211.1 Detailed Description

Definition at line 14 of file [InfoPanelService.cs](#).

6.211.2 Member Function Documentation

6.211.2.1 Dispose()

```
void MergeIt.Game.Services.InfoPanelService.Dispose ( ) [inline]
```

Definition at line 25 of file [InfoPanelService.cs](#).

```
00026     {
00027         _messageBus.RemoveListener<ElementActionMessage>(OnElementActionMessageHandler);
00028     }
```

6.211.2.2 Initialize()

```
void MergeIt.Game.Services.InfoPanelService.Initialize ( ) [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 30 of file [InfoPanelService.cs](#).

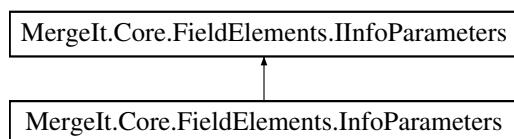
```
00031     {
00032         _messageBus.AddListener<ElementActionMessage>(OnElementActionMessageHandler);
00033     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/InfoPanelService.cs

6.212 Mergelt.Core.FieldElements.InfoParameters Class Reference

Inheritance diagram for Mergelt.Core.FieldElements.InfoParameters:



Properties

- [GridPoint LogicPosition](#) [get, set]
- bool [IsBlocked](#) [get, set]
- [ElementType Type](#) [get, set]
- string [Name](#) [get, set]
- string [Description](#) [get, set]

Properties inherited from [Mergelt.Core.FieldElements.IInfoParameters](#)

6.212.1 Detailed Description

Definition at line 7 of file [InfoParameters.cs](#).

6.212.2 Property Documentation

6.212.2.1 Description

```
string MergeIt.Core.FieldElements.InfoParameters.Description [get], [set]
```

Implements [Mergelt.Core.FieldElements.IInfoParameters](#).

Definition at line 13 of file [InfoParameters.cs](#).

```
00013 { get; set; }
```

6.212.2.2 IsBlocked

```
bool MergeIt.Core.FieldElements.InfoParameters.IsBlocked [get], [set]
```

Implements [Mergelt.Core.FieldElements.IInfoParameters](#).

Definition at line 10 of file [InfoParameters.cs](#).

```
00010 { get; set; }
```

6.212.2.3 LogicPosition

```
GridPoint MergeIt.Core.FieldElements.InfoParameters.LogicPosition [get], [set]
```

Implements [Mergelt.Core.FieldElements.IInfoParameters](#).

Definition at line 9 of file [InfoParameters.cs](#).

```
00009 { get; set; }
```

6.212.2.4 Name

```
string MergeIt.Core.FieldElements.InfoParameters.Name [get], [set]
```

Implements [Mergelt.Core.FieldElements.IInfoParameters](#).

Definition at line 12 of file [InfoParameters.cs](#).

```
00012 { get; set; }
```

6.212.2.5 Type

```
ElementType MergeIt.Core.FieldElements.InfoParameters.Type [get], [set]
```

Implements [Mergelt.Core.FieldElements.IInfoParameters](#).

Definition at line 11 of file [InfoParameters.cs](#).

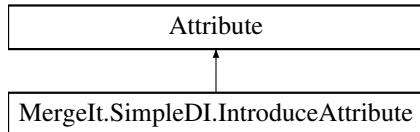
```
00011 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/Parameters/InfoParameters.cs

6.213 Mergelt.SimpleDI.IntroduceAttribute Class Reference

Inheritance diagram for Mergelt.SimpleDI.IntroduceAttribute:



Public Member Functions

- [IntroduceAttribute \(string key=""\)](#)

6.213.1 Detailed Description

Definition at line 7 of file [IntroduceAttribute.cs](#).

6.213.2 Constructor & Destructor Documentation

6.213.2.1 IntroduceAttribute()

```
MergeIt.SimpleDI.IntroduceAttribute.IntroduceAttribute (
    string key = "") [inline]
```

Definition at line 11 of file [IntroduceAttribute.cs](#).

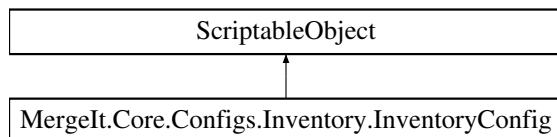
```
00012     {
00013         _key = key;
00014     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/SimpleDI/IntroduceAttribute.cs

6.214 Mergelt.Core.Configs.Inventory.InventoryConfig Class Reference

Inheritance diagram for Mergelt.Core.Configs.Inventory.InventoryConfig:



Public Attributes

- int [InitialCapacity](#)
- [CurrencySettings\[\] PaidCells](#)

6.214.1 Detailed Description

Definition at line 9 of file [InventoryConfig.cs](#).

6.214.2 Member Data Documentation

6.214.2.1 InitialCapacity

```
int MergeIt.Core.Configs.Inventory.InventoryConfig.InitialCapacity
```

Definition at line 11 of file [InventoryConfig.cs](#).

6.214.2.2 PaidCells

```
CurrencySettings [] MergeIt.Core.Configs.Inventory.InventoryConfig.PaidCells
```

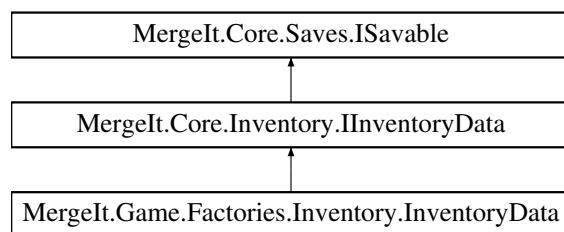
Definition at line 13 of file [InventoryConfig.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Configs/Inventory/InventoryConfig.cs

6.215 Mergelt.Game.Factories.Inventory.InventoryData Class Reference

Inheritance diagram for Mergelt.Game.Factories.Inventory.InventoryData:



Properties

- int [InventorySize](#) [get, set]
- [FieldElementData\[\]](#) [InventoryElements](#) [get, set]

Properties inherited from [Mergelt.Core.Inventory.IInventoryData](#)

6.215.1 Detailed Description

Definition at line 12 of file [InventoryData.cs](#).

6.215.2 Property Documentation

6.215.2.1 InventoryElements

```
FieldElementData [ ] MergeIt.Game.Factories.Inventory.InventoryData.InventoryElements [get],  
[set]
```

Implements [Mergelt.Core.Inventory.IInventoryData](#).

Definition at line 17 of file [InventoryData.cs](#).
00017 { get; set; }

6.215.2.2 InventorySize

```
int MergeIt.Game.Factories.Inventory.InventoryData.InventorySize [get], [set]
```

Implements [Mergelt.Core.Inventory.IInventoryData](#).

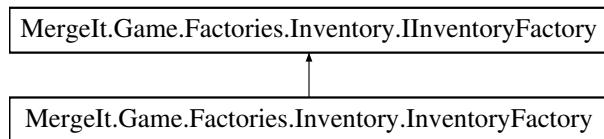
Definition at line 15 of file [InventoryData.cs](#).
00015 { get; set; }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Factories/Inventory/InventoryData.cs

6.216 Mergelt.Game.Factories.Inventory.InventoryFactory Class Reference

Inheritance diagram for Mergelt.Game.Factories.Inventory.InventoryFactory:



Public Member Functions

- [InventoryPanelItemComponent CreateInventoryPanelItem \(IFieldElement fieldElement\)](#)
- [InventoryWindowItemComponent CreateInventoryWindowItem \(IFieldElement fieldElement\)](#)
- [InventoryWindowPaidCellComponent CreateWindowPaidCell \(\)](#)
- [GameObject CreateWindowEmptyCell \(\)](#)

6.216.1 Detailed Description

Definition at line 11 of file [InventoryFactory.cs](#).

6.216.2 Member Function Documentation

6.216.2.1 CreateInventoryPanelItem()

```
InventoryPanelItemComponent MergeIt.Game.Factories.Inventory.InventoryFactory.CreateInventory←
PanelItem (
    IFieldElement fieldElement ) [inline]
```

Implements [Mergelt.Game.Factories.Inventory.IInventoryFactory](#).

Definition at line 18 of file [InventoryFactory.cs](#).

```
00019     {
00020         GameObject itemPanelObject = Resources.Load<GameObject>(PanelItemPath);
00021
00022         if (itemPanelObject)
00023         {
00024             GameObject panelItemObject = Object.Instantiate(itemPanelObject);
00025             FieldElementIconComponent iconPrototype =
00026                 fieldElement.ConfigParameters.ElementConfig.GetIconComponent();
00027
00028             var icon = Object.Instantiate(iconPrototype, panelItemObject.transform);
00029             var iconRectTransform = icon.GetComponent<RectTransform>();
00030             iconRectTransform.SetAsFirstSibling();
00031
00032             var resultComponent = panelItemObject.GetComponent<InventoryPanelItemComponent>();
00033             return resultComponent;
00034         }
00035
00036     } [return null];
```

6.216.2.2 CreateInventoryWindowItem()

```
InventoryWindowItemComponent MergeIt.Game.Factories.Inventory.InventoryFactory.CreateInventory←
WindowItem (
    IFieldElement fieldElement ) [inline]
```

Implements [Mergelt.Game.Factories.Inventory.IInventoryFactory](#).

Definition at line 38 of file [InventoryFactory.cs](#).

```
00039     {
00040         GameObject itemPanelObject = Resources.Load<GameObject>(WindowItemPath);
00041
00042         if (itemPanelObject)
00043         {
00044             var panelItemObject = Object.Instantiate(itemPanelObject);
00045             if (panelItemObject.TryGetComponent(out InventoryWindowItemComponent resultComponent))
00046             {
00047                 FieldElementIconComponent iconPrototype =
00048                     fieldElement.ConfigParameters.ElementConfig.GetIconComponent();
00049
00050                 var icon = Object.Instantiate(iconPrototype, panelItemObject.transform);
00051                 var iconRectTransform = icon.GetComponent<RectTransform>();
00052
00053                 resultComponent.SetIcon(iconRectTransform);
00054             }
00055
00056             return resultComponent;
00057         }
00058
00059     } [return null];
```

6.216.2.3 CreateWindowEmptyCell()

```
GameObject MergeIt.Game.Factories.Inventory.InventoryFactory.CreateWindowEmptyCell () [inline]
```

Implements [Mergelt.Game.Factories.Inventory.IInventoryFactory](#).

Definition at line 76 of file [InventoryFactory.cs](#).

```
00077     {
00078         GameObject itemPanelObject = Resources.Load<GameObject>(WindowEmptyItemPath);
00079
00080         if (itemPanelObject)
00081         {
00082             var panelItemObject = Object.Instantiate(itemPanelObject);
00083             return panelItemObject.gameObject;
00084         }
00085
00086         return null;
00087     }
```

6.216.2.4 CreateWindowPaidCell()

```
InventoryWindowPaidCellComponent MergeIt.Game.Factories.Inventory.InventoryFactory.CreateWindowPaidCell () [inline] ←
```

Implements [Mergelt.Game.Factories.Inventory.IInventoryFactory](#).

Definition at line 61 of file [InventoryFactory.cs](#).

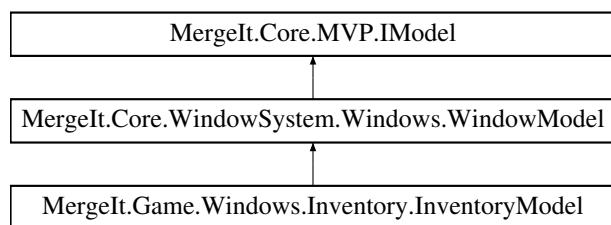
```
00062     {
00063         GameObject itemPanelObject = Resources.Load<GameObject>(WindowPaidCellPath);
00064
00065         if (itemPanelObject)
00066         {
00067             var panelItemObject = Object.Instantiate(itemPanelObject);
00068
00069             var resultComponent =
00070                 panelItemObject.GetComponent<InventoryWindowPaidCellComponent>();
00071             return resultComponent;
00072         }
00073
00074         return null;
00075     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Factories/Inventory/InventoryFactory.cs

6.217 Mergelt.Game.Windows.Inventory.InventoryModel Class Reference

Inheritance diagram for [Mergelt.Game.Windows.Inventory.InventoryModel](#):



6.217.1 Detailed Description

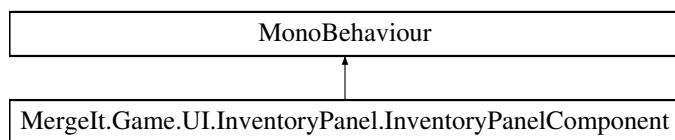
Definition at line 7 of file [InventoryModel.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Windows/Inventory/InventoryModel.cs

6.218 Mergelt.Game.UI.InventoryPanel.InventoryPanelComponent Class Reference

Inheritance diagram for Mergelt.Game.UI.InventoryPanel.InventoryPanelComponent:



6.218.1 Detailed Description

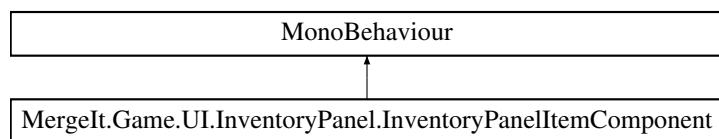
Definition at line 18 of file [InventoryPanelComponent.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/UI/InventoryPanel/InventoryPanelComponent.cs

6.219 Mergelt.Game.UI.InventoryPanel.InventoryPanelItemComponent Class Reference

Inheritance diagram for Mergelt.Game.UI.InventoryPanel.InventoryPanelItemComponent:



Properties

- RectTransform [RectTransform](#) [get]

Events

- Action< [InventoryPanelItemComponent](#) > [ClickEvent](#)

6.219.1 Detailed Description

Definition at line 9 of file [InventoryPanelItemComponent.cs](#).

6.219.2 Property Documentation

6.219.2.1 RectTransform

```
RectTransform MergeIt.Game.UI.InventoryPanel.InventoryPanelItemComponent.RectTransform [get]
```

Definition at line 19 of file [InventoryPanelItemComponent.cs](#).

```
00020     {
00021         get
00022     {
00023         if (!_rectTransform)
00024         {
00025             _rectTransform = GetComponent<RectTransform>();
00026         }
00027         return _rectTransform;
00028     }
00029 }
00030 }
```

6.219.3 Event Documentation

6.219.3.1 ClickEvent

```
Action<InventoryPanelItemComponent> MergeIt.Game.UI.InventoryPanel.InventoryPanelItemComponent.ClickEvent
```

Definition at line 41 of file [InventoryPanelItemComponent.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/UI/InventoryPanel/InventoryPanelItemComponent.cs

6.220 MergeIt.Game.UI.InventoryPanel.InventoryPanelItemPair Class Reference

Public Member Functions

- [InventoryPanelItemPair](#) ([InventoryPanelItemComponent](#) component, [IFieldElement](#) fieldElement)

Properties

- [InventoryPanelItemComponent Component](#) [get]
- [IFieldElement Element](#) [get]

6.220.1 Detailed Description

Definition at line 7 of file [InventoryPanelItemPair.cs](#).

6.220.2 Constructor & Destructor Documentation

6.220.2.1 InventoryPanelItemPair()

```
MergeIt.Game.UI.InventoryPanel.InventoryPanelItemPair.InventoryPanelItemPair (
    InventoryPanelItemComponent component,
    IFieldElement fieldElement ) [inline]
```

Definition at line 9 of file [InventoryPanelItemPair.cs](#).

```
00010     {
00011         Component = component;
00012         Element = fieldElement;
00013     }
```

6.220.3 Property Documentation

6.220.3.1 Component

```
InventoryPanelItemComponent MergeIt.Game.UI.InventoryPanel.InventoryPanelItemPair.Component
[get]
```

Definition at line 15 of file [InventoryPanelItemPair.cs](#).

```
00015 { get; }
```

6.220.3.2 Element

```
IFieldElement MergeIt.Game.UI.InventoryPanel.InventoryPanelItemPair.Element [get]
```

Definition at line 17 of file [InventoryPanelItemPair.cs](#).

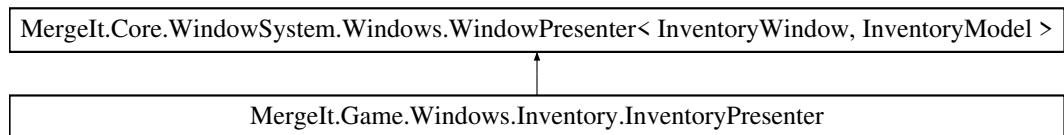
```
00017 { get; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/UI/InventoryPanel/InventoryPanelItemPair.cs

6.221 Mergelt.Game.Windows.Inventory.InventoryPresenter Class Reference

Inheritance diagram for Mergelt.Game.Windows.Inventory.InventoryPresenter:



Protected Member Functions

- override void [OnInitialize \(IWindowArgs args=null\)](#)
- override void [OnDestroyWindow \(\)](#)

Protected Member Functions inherited from**MergeIt.Core.WindowSystem.Windows.WindowPresenter< InventoryWindow, InventoryModel >**

- virtual void [OnInitialize \(IWindowArgs args=null\)](#)
- override void [OnDispose \(\)](#)
- virtual void [OnWindowStartShowing \(\)](#)
- virtual void [OnWindowShown \(\)](#)
- virtual void [OnWindowStartClosing \(\)](#)
- virtual void [OnWindowClosed \(\)](#)

Additional Inherited Members**Public Member Functions inherited from****MergeIt.Core.WindowSystem.Windows.WindowPresenter< InventoryWindow, InventoryModel >**

- void [Initialize \(IView view, string layer, IWindowArgs windowArgs=null\)](#)
- void [Show \(\)](#)
- void [Hide \(\)](#)
- void [Close \(\)](#)
- void [SetWindowActive \(bool active\)](#)
- void [SetWindowLayer \(\)](#)
- void [DestroyWindow \(\)](#)

Protected Attributes inherited from**MergeIt.Core.WindowSystem.Windows.WindowPresenter< InventoryWindow, InventoryModel >**

- [IMessageBus MessageBus](#)
- [IWindowSystem WindowSystem](#)

Properties inherited from**MergeIt.Core.WindowSystem.Windows.WindowPresenter< InventoryWindow, InventoryModel >**

- string [Layer \[get\]](#)
- [WindowState State \[get\]](#)

6.221.1 Detailed Description

Definition at line 19 of file [InventoryPresenter.cs](#).

6.221.2 Member Function Documentation

6.221.2.1 OnDestroyWindow()

```
override void MergeIt.Game.Windows.Inventory.InventoryPresenter.OnDestroyWindow () [inline],  
[protected], [virtual]
```

Reimplemented from [Mergelt.Core.WindowSystem.Windows.WindowPresenter< InventoryWindow, InventoryModel >](#).

Definition at line 50 of file [InventoryPresenter.cs](#).

```
00051      {  
00052          base.OnDestroyWindow();  
00053  
00054          for (int i = 0; i < _itemComponents.Count; i++)  
00055          {  
00056              InventoryWindowItemPair item = _itemComponents[i];  
00057              item.Component.InfoClickEvent -= OnItemInfoClick;  
00058              item.Component.ItemClickEvent -= OnItemClick;  
00059          }  
00060  
00061          if (_paidCell)  
00062          {  
00063              _paidCell.BuyCellEvent -= OnBuyCellClick;  
00064          }  
00065      }
```

6.221.2.2 OnInitialize()

```
override void MergeIt.Game.Windows.Inventory.InventoryPresenter.OnInitialize (  
    IWindowArgs args = null ) [inline], [protected]
```

Definition at line 33 of file [InventoryPresenter.cs](#).

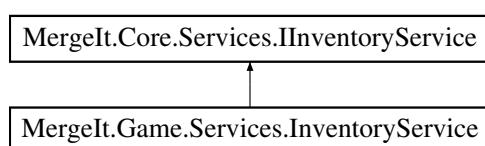
```
00034      {  
00035          base.OnInitialize(args);  
00036  
00037          _inventoryService = DiContainer.Get<IIInventoryService>();  
00038          _inventoryServiceModel = DiContainer.Get<InventoryServiceModel>();  
00039          _configsService = DiContainer.Get<IConfigsService>();  
00040          _inventoryFactory = DiContainer.Get<IIInventoryFactory>();  
00041          _currencyService = DiContainer.Get<ICurrencyService>();  
00042          _fieldService = DiContainer.Get<IGameFieldService>();  
00043  
00044          _paidCellsPrices = _configsService.InventoryConfig.PaidCells;  
00045          _initialSize = _configsService.InventoryConfig.InitialCapacity;  
00046  
00047          FillGrid();  
00048      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Windows/Inventory/InventoryPresenter.cs

6.222 Mergelt.Game.Services.InventoryService Class Reference

Inheritance diagram for Mergelt.Game.Services.InventoryService:



Public Member Functions

- void [CreateInventory \(\)](#)
- void [SetupInventory \(IInventoryData inventoryData\)](#)
- [IInventoryData GetData \(\)](#)
- void [Add \(IFieldElement fieldElement\)](#)
- bool [Remove \(IFieldElement fieldElement\)](#)
- void [OpenWindow \(\)](#)

6.222.1 Detailed Description

Definition at line 17 of file [InventoryService.cs](#).

6.222.2 Member Function Documentation

6.222.2.1 Add()

```
void MergeIt.Game.Services.InventoryService.Add (
    IFIELDELEMENT fieldElement ) [inline]
```

Implements [Mergelt.Core.Services.IInventoryService](#).

Definition at line 85 of file [InventoryService.cs](#).

```
00086     {
00087         _serviceModel.InventoryElements.Add(fieldElement);
00088         _saveService.Save(GameSaveType.Inventory);
00089     }
```

6.222.2.2 CreateInventory()

```
void MergeIt.Game.Services.InventoryService.CreateInventory ( ) [inline]
```

Implements [Mergelt.Core.Services.IInventoryService](#).

Definition at line 34 of file [InventoryService.cs](#).

```
00035     {
00036         InventoryConfig inventoryConfig = _configsService.InventoryConfig;
00037         var inventoryData = new InventoryData
00038         {
00039             InventorySize = inventoryConfig.InitialCapacity
00040         };
00041         SetupInventory(inventoryData);
00042         _saveService.Save(GameSaveType.Inventory);
00043     }
00044 }
```

6.222.2.3 GetData()

```
IInventoryData MergeIt.Game.Services.InventoryService.GetData () [inline]
```

Implements [MergIt.Core.Services.IInventoryService](#).

Definition at line 65 of file [InventoryService.cs](#).

```
00066      {
00067          var data = new InventoryData();
00068          data.InventorySize = _serviceModel.InventorySize;
00069
00070          if (_serviceModel.InventoryElements != null)
00071          {
00072              data.InventoryElements = new FieldElementData[_serviceModel.InventoryElements.Count];
00073
00074              for (int i = 0; i < _serviceModel.InventoryElements.Count; i++)
00075              {
00076                  IFieldElement element = _serviceModel.InventoryElements[i];
00077                  FieldElementData elementData =
00078                      _configProcessor.ConvertToFieldElementData(element);
00079                  data.InventoryElements[i] = elementData;
00080              }
00081
00082          return data;
00083      }
```

6.222.2.4 OpenWindow()

```
void MergeIt.Game.Services.InventoryService.OpenWindow () [inline]
```

Implements [MergIt.Core.Services.IInventoryService](#).

Definition at line 103 of file [InventoryService.cs](#).

```
00104      {
00105          _windowSystem.OpenWindow<InventoryPresenter>(enableBlackout: true);
00106      }
```

6.222.2.5 Remove()

```
bool MergeIt.Game.Services.InventoryService.Remove (
    IFieldElement fieldElement ) [inline]
```

Implements [MergIt.Core.Services.IInventoryService](#).

Definition at line 91 of file [InventoryService.cs](#).

```
00092      {
00093          bool result = _serviceModel.InventoryElements.Remove(fieldElement);
00094
00095          if (result)
00096          {
00097              _saveService.Save(GameSaveType.Inventory);
00098          }
00099
00100      return result;
00101  }
```

6.222.2.6 SetupInventory()

```
void MergeIt.Game.Services.InventoryService.SetupInventory (
    IInventoryData inventoryData ) [inline]
```

Implements [Mergelt.Core.Services.IInventoryService](#).

Definition at line 47 of file [InventoryService.cs](#).

```
00048     {
00049         _serviceModel.InventorySize = inventoryData.InventorySize;
00050         var elements = new List<IFieldElement>();
00051
00052         if (inventoryData.InventoryElements != null)
00053         {
00054             for (int i = 0; i < inventoryData.InventoryElements.Length; i++)
00055             {
00056                 FieldElementData elementData = inventoryData.InventoryElements[i];
00057                 IFieldElement element = _configProcessor.ConvertToFieldElement(elementData);
00058                 elements.Add(element);
00059             }
00060
00061             _serviceModel.InventoryElements.AddRange(elements);
00062         }
00063     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/InventoryService.cs

6.223 Mergelt.Game.Services.InventoryServiceModel Class Reference

Public Member Functions

- bool [IsFull](#) ()

Properties

- int [InventorySize](#) [get, set]
- List<[IFieldElement](#)> [InventoryElements](#) = new() [get]

6.223.1 Detailed Description

Definition at line 8 of file [InventoryServiceModel.cs](#).

6.223.2 Member Function Documentation

6.223.2.1 IsFull()

```
bool MergeIt.Game.Services.InventoryServiceModel.IsFull ( ) [inline]
```

Definition at line 13 of file [InventoryServiceModel.cs](#).

```
00014     {
00015         return InventoryElements.Count == InventorySize;
00016     }
```

6.223.3 Property Documentation

6.223.3.1 InventoryElements

```
List<IFieldElement> MergeIt.Game.Services.InventoryServiceModel.InventoryElements = new()
[get]
```

Definition at line 11 of file [InventoryServiceModel.cs](#).
00011 { get; } = new();

6.223.3.2 InventorySize

```
int MergeIt.Game.Services.InventoryServiceModel.InventorySize [get], [set]
```

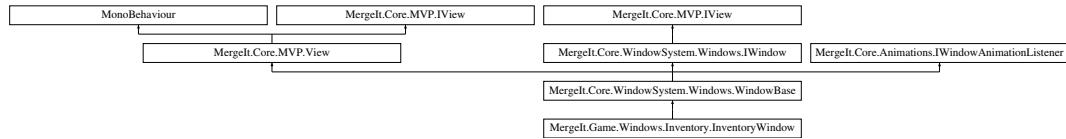
Definition at line 10 of file [InventoryServiceModel.cs](#).
00010 { get; set; }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/InventoryServiceModel.cs

6.224 Mergelt.Game.Windows.Inventory.InventoryWindow Class Reference

Inheritance diagram for Mergelt.Game.Windows.Inventory.InventoryWindow:



Properties

- RectTransform [ItemsGrid](#) [get]

Properties inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- Canvas [Canvas](#) [get]
- [IWindowAnimationController](#) [AnimationController](#) [get]

Properties inherited from [Mergelt.Core.MVP.View](#)

- GameObject [GameObject](#) [get]

Properties inherited from [Mergelt.Core.MVP.IView](#)

Additional Inherited Members

Public Member Functions inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- virtual void [Show \(\)](#)
- virtual void [Close \(\)](#)
- virtual void [SetLayer \(string layer\)](#)
- virtual void [OnOpenStarted \(\)](#)
- virtual void [OnOpenFinished \(\)](#)
- virtual void [OnCloseStarted \(\)](#)
- virtual void [OnCloseFinished \(\)](#)

Public Member Functions inherited from [Mergelt.Core.MVP.View](#)

- virtual void [Initialize \(\)](#)

Protected Member Functions inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- override void [OnDestroy \(\)](#)

Protected Attributes inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- Button [CloseButton](#)
- Button [ClickOutsideArea](#)

Events inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- Action [InitiateCloseEvent](#)
- Action [ShowStartEvent](#)
- Action [ShowEndEvent](#)
- Action [CloseStartEvent](#)
- Action [CloseEndEvent](#)

Events inherited from [Mergelt.Core.MVP.View](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

Events inherited from [Mergelt.Core.MVP.IView](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

Events inherited from [MergeIt.Core.WindowSystem.Windows.IWindow](#)

- Action [InitiateCloseEvent](#)
- Action [ShowStartEvent](#)
- Action [ShowEndEvent](#)
- Action [CloseStartEvent](#)
- Action [CloseEndEvent](#)

6.224.1 Detailed Description

Definition at line 8 of file [InventoryWindow.cs](#).

6.224.2 Property Documentation

6.224.2.1 ItemsGrid

```
RectTransform MergeIt.Game.Windows.Inventory.InventoryWindow.ItemsGrid [get]
```

Definition at line 13 of file [InventoryWindow.cs](#).

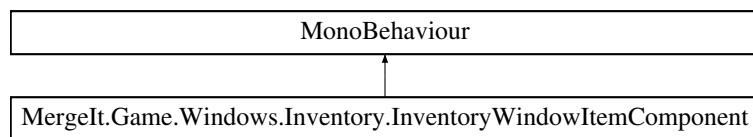
```
00014     {  
00015         get => _itemsGrid;  
00016     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Windows/Inventory/InventoryWindow.cs

6.225 MergeIt.Game.Windows.Inventory.InventoryWindowItemComponent Class Reference

Inheritance diagram for MergeIt.Game.Windows.Inventory.InventoryWindowItemComponent:



Public Member Functions

- void [Awake \(\)](#)
- void [OnDestroy \(\)](#)
- void [SetIcon \(RectTransform rectTransform\)](#)

Events

- Action< [InventoryWindowItemComponent](#) > [ItemClickEvent](#)
- Action< [InventoryWindowItemComponent](#) > [InfoClickEvent](#)

6.225.1 Detailed Description

Definition at line 9 of file [InventoryWindowItemComponent.cs](#).

6.225.2 Member Function Documentation

6.225.2.1 Awake()

```
void MergeIt.Game.Windows.Inventory.InventoryWindowItemComponent.Awake () [inline]
```

Definition at line 21 of file [InventoryWindowItemComponent.cs](#).

```
00022     {
00023         _itemButton.onClick.AddListener(OnItemClick);
00024         _infoButton.onClick.AddListener(OnInfoClick);
00025     }
```

6.225.2.2 OnDestroy()

```
void MergeIt.Game.Windows.Inventory.InventoryWindowItemComponent.OnDestroy () [inline]
```

Definition at line 27 of file [InventoryWindowItemComponent.cs](#).

```
00028     {
00029         _itemButton.onClick.RemoveListener(OnItemClick);
00030         _infoButton.onClick.RemoveListener(OnInfoClick);
00031     }
```

6.225.2.3 SetIcon()

```
void MergeIt.Game.Windows.Inventory.InventoryWindowItemComponent.SetIcon (
    RectTransform rectTransform) [inline]
```

Definition at line 35 of file [InventoryWindowItemComponent.cs](#).

```
00036     {
00037         rectTransform.SetParent(_iconContainer);
00038         rectTransform.localScale = Vector3.one;
00039         ;
00040     }
```

6.225.3 Event Documentation

6.225.3.1 InfoClickEvent

```
Action<InventoryWindowItemComponent> MergeIt.Game.Windows.Inventory.InventoryWindowItemComponent.InfoClickEvent
```

Definition at line 33 of file [InventoryWindowItemComponent.cs](#).

6.225.3.2 ItemClickEvent

```
Action<InventoryWindowItemComponent> MergeIt.Game.Windows.Inventory.InventoryWindowItemComponent.ItemClickEvent
```

Definition at line 32 of file [InventoryWindowItemComponent.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Windows/Inventory/InventoryWindowItemComponent.cs

6.226 MergeIt.Game.Windows.Inventory.InventoryWindowItemPair Class Reference

Public Member Functions

- [InventoryWindowItemPair \(InventoryWindowItemComponent component, IFieldElement fieldElement\)](#)

Properties

- [InventoryWindowItemComponent Component \[get\]](#)
- [IFieldElement Element \[get\]](#)

6.226.1 Detailed Description

Definition at line 7 of file [InventoryWindowItemPair.cs](#).

6.226.2 Constructor & Destructor Documentation

6.226.2.1 InventoryWindowItemPair()

```
MergeIt.Game.Windows.Inventory.InventoryWindowItemPair.InventoryWindowItemPair (
    InventoryWindowItemComponent component,
    IFieldElement fieldElement ) [inline]
```

Definition at line 10 of file [InventoryWindowItemPair.cs](#).

```
00011 {
00012     Component = component;
00013     Element = fieldElement;
00014 }
```

6.226.3 Property Documentation

6.226.3.1 Component

```
InventoryWindowItemComponent MergeIt.Game.Windows.Inventory.InventoryWindowItemPair.Component
[get]
```

Definition at line 16 of file [InventoryWindowItemPair.cs](#).

```
00016 { get; }
```

6.226.3.2 Element

```
IFieldElement MergeIt.Game.Windows.Inventory.InventoryWindowItemPair.Element [get]
```

Definition at line 18 of file [InventoryWindowItemPair.cs](#).

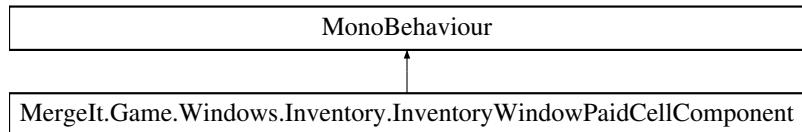
```
00018 { get; }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Windows/Inventory/InventoryWindowItemPair.cs

6.227 MergeIt.Game.Windows.Inventory.InventoryWindowPaidCellComponent Class Reference

Inheritance diagram for MergeIt.Game.Windows.Inventory.InventoryWindowPaidCellComponent:



Public Member Functions

- void [Awake \(\)](#)
- void [OnDestroy \(\)](#)
- void [Setup \(string priceText, Sprite priceImage\)](#)

Events

- Action [BuyCellEvent](#)

6.227.1 Detailed Description

Definition at line 10 of file [InventoryWindowPaidCellComponent.cs](#).

6.227.2 Member Function Documentation

6.227.2.1 Awake()

```
void MergeIt.Game.Windows.Inventory.InventoryWindowPaidCellComponent.Awake () [inline]
```

Definition at line 22 of file [InventoryWindowPaidCellComponent.cs](#).

```
00023     {
00024         _buyButton.onClick.AddListener(OnBuyButtonClick);
00025     }
```

6.227.2.2 OnDestroy()

```
void MergeIt.Game.Windows.Inventory.InventoryWindowPaidCellComponent.OnDestroy () [inline]
```

Definition at line 27 of file [InventoryWindowPaidCellComponent.cs](#).

```
00028     {
00029         _buyButton.onClick.RemoveListener(OnBuyButtonClick);
00030     }
```

6.227.2.3 Setup()

```
void MergeIt.Game.Windows.Inventory.InventoryWindowPaidCellComponent.Setup (
    string priceText,
    Sprite priceImage ) [inline]
```

Definition at line 33 of file [InventoryWindowPaidCellComponent.cs](#).

```
00034     {
00035         _priceLabel.text = priceText;
00036         _priceIcon.sprite = priceImage;
00037     }
```

6.227.3 Event Documentation

6.227.3.1 BuyCellEvent

Action `MergeIt.Game.Windows.Inventory.InventoryWindowPaidCellComponent.BuyCellEvent`

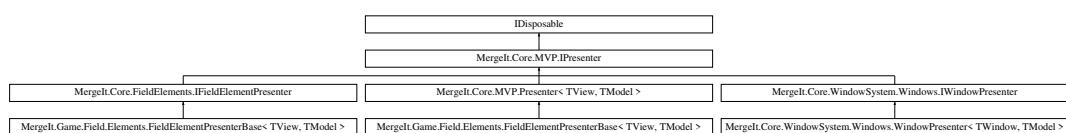
Definition at line 31 of file [InventoryWindowPaidCellComponent.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Windows/Inventory/InventoryWindowPaidCellComponent.cs

6.228 Mergelt.Core.MVP.IPresenter Interface Reference

Inheritance diagram for Mergelt.Core.MVP.IPresenter:



Public Member Functions

- void `Initialize (IView view)`

6.228.1 Detailed Description

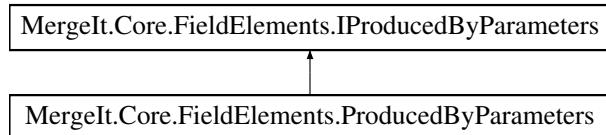
Definition at line 7 of file [IPresenter.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/MVP/IPresenter.cs

6.229 Mergelt.Core.FieldElements.IProducedByParameters Interface Reference

Inheritance diagram for Mergelt.Core.FieldElements.IProducedByParameters:



Properties

- List<[ElementConfig](#) > Elements [get, set]

6.229.1 Detailed Description

Definition at line 8 of file [IProducedByParameters.cs](#).

6.229.2 Property Documentation

6.229.2.1 Elements

List<[ElementConfig](#)> MergeIt.Core.FieldElements.IProducedByParameters.Elements [get], [set]

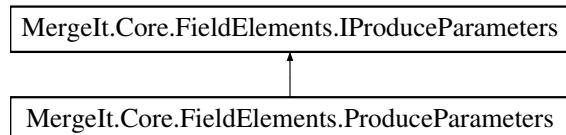
Definition at line 10 of file [IProducedByParameters.cs](#).
00010 { get; set; }

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/Parameters/IProducedByParameters.cs

6.230 Mergelt.Core.FieldElements.IProduceParameters Interface Reference

Inheritance diagram for Mergelt.Core.FieldElements.IProduceParameters:



Properties

- List<[GeneratableFieldElement](#) > Elements [get, set]

6.230.1 Detailed Description

Definition at line 7 of file [IProduceParameters.cs](#).

6.230.2 Property Documentation

6.230.2.1 Elements

```
List<GeneratableFieldElement> MergeIt.Core.FieldElements.IProduceParameters.Elements [get],  
[set]
```

Definition at line 9 of file [IProduceParameters.cs](#).

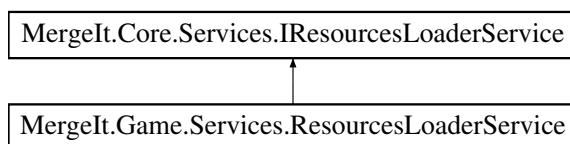
```
00009 { get; set; }
```

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/Parameters/IProduceParameters.cs

6.231 Mergelt.Core.Services.IResourcesLoaderService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IResourcesLoaderService:



Public Member Functions

- T [GetObject< T >](#) (string path)

6.231.1 Detailed Description

Definition at line 7 of file [IResourcesLoaderService.cs](#).

6.231.2 Member Function Documentation

6.231.2.1 GetObject< T >()

```
T MergeIt.Core.Services.IResourcesLoaderService.GetObject< T > (  
    string path )
```

Type Constraints

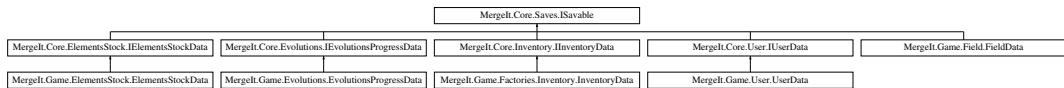
T : Object

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IResourcesLoaderService.cs

6.232 Mergelt.Core.Saves.ISavable Interface Reference

Inheritance diagram for Mergelt.Core.Saves.ISavable:



6.232.1 Detailed Description

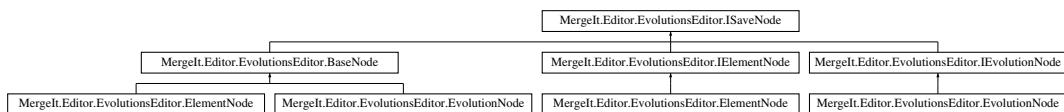
Definition at line 5 of file [ISavable.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Saves/ISavable.cs

6.233 Mergelt.Editor.EvolutionsEditor.ISaveNode Interface Reference

Inheritance diagram for Mergelt.Editor.EvolutionsEditor.ISaveNode:



Public Member Functions

- void **SaveData** ()
- Port **GetPort** (string portName)

Properties

- string **Id** [get]

6.233.1 Detailed Description

Definition at line 7 of file [ISaveNode.cs](#).

6.233.2 Property Documentation

6.233.2.1 Id

string Mergelt.Editor.EvolutionsEditor.ISaveNode.Id [get]

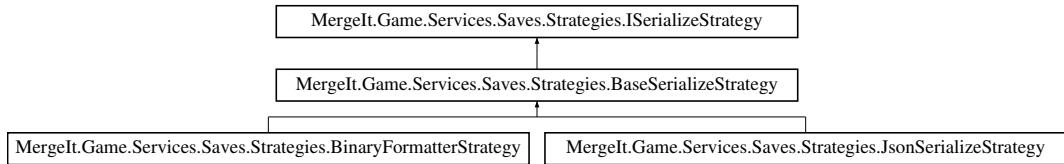
Definition at line 9 of file [ISaveNode.cs](#).
00009 { get; }

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Editor/Core/EvolutionsEditor/ItemsNodes/ISaveNode.cs

6.234 Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy Interface Reference

Inheritance diagram for Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy:



Public Member Functions

- UniTask [Save< T >](#) (T data)
- T [Load< T >](#) ()

Properties

- string [SaveDir](#) [get]

6.234.1 Detailed Description

Definition at line 8 of file [ISerializeStrategy.cs](#).

6.234.2 Member Function Documentation

6.234.2.1 Load< T >()

T [MergeIt.Game.Services.Saves.Strategies.ISerializeStrategy.Load< T >](#) ()

Type Constraints

T : class

T : ISavable

6.234.2.2 Save< T >()

UniTask [MergeIt.Game.Services.Saves.Strategies.ISerializeStrategy.Save< T >](#) (T data)

Type Constraints

T : class

T : ISavable

6.234.3 Property Documentation

6.234.3.1 SaveDir

```
string MergeIt.Game.Services.Saves.Strategies.ISerializeStrategy.SaveDir [get]
```

Definition at line 10 of file [ISerializeStrategy.cs](#).

```
00010 { get; }
```

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/Services/Saves/Strategies/ISerializeStrategy.cs

6.235 Mergelt.SimpleDI.ReservedInterfaces.IUpdatable Interface Reference

Inheritance diagram for Mergelt.SimpleDI.ReservedInterfaces.IUpdatable:



Public Member Functions

- void **Update** ()

6.235.1 Detailed Description

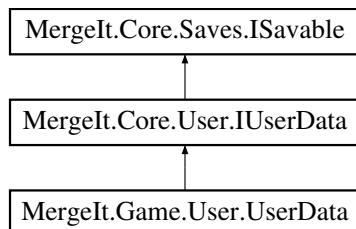
Definition at line 5 of file [IUpdatable.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/SimpleDI/ReservedInterfaces/IUpdatable.cs

6.236 Mergelt.Core.User.IUserData Interface Reference

Inheritance diagram for Mergelt.Core.User.IUserData:



Properties

- string `Name` [get, set]
- int `Energy` [get, set]
- int `SoftCurrency` [get, set]
- int `HardCurrency` [get, set]
- int `Splitters` [get, set]
- int `Level` [get, set]
- int `Experience` [get, set]
- long `EnergyRestoringStartTime` [get, set]

6.236.1 Detailed Description

Definition at line 7 of file [IUserData.cs](#).

6.236.2 Property Documentation

6.236.2.1 Energy

```
int MergeIt.Core.User.IUserData.Energy [get], [set]
```

Definition at line 10 of file [IUserData.cs](#).

```
00010 { get; set; }
```

6.236.2.2 EnergyRestoringStartTime

```
long MergeIt.Core.User.IUserData.EnergyRestoringStartTime [get], [set]
```

Definition at line 16 of file [IUserData.cs](#).

```
00016 { get; set; }
```

6.236.2.3 Experience

```
int MergeIt.Core.User.IUserData.Experience [get], [set]
```

Definition at line 15 of file [IUserData.cs](#).

```
00015 { get; set; }
```

6.236.2.4 HardCurrency

```
int MergeIt.Core.User.IUserData.HardCurrency [get], [set]
```

Definition at line 12 of file [IUserData.cs](#).

```
00012 { get; set; }
```

6.236.2.5 Level

```
int MergeIt.Core.User.IUserData.Level [get], [set]
```

Definition at line 14 of file [IUserData.cs](#).
00014 { get; set; }

6.236.2.6 Name

```
string MergeIt.Core.User.IUserData.Name [get], [set]
```

Definition at line 9 of file [IUserData.cs](#).
00009 { get; set; }

6.236.2.7 SoftCurrency

```
int MergeIt.Core.User.IUserData.SoftCurrency [get], [set]
```

Definition at line 11 of file [IUserData.cs](#).
00011 { get; set; }

6.236.2.8 Splitters

```
int MergeIt.Core.User.IUserData.Splitters [get], [set]
```

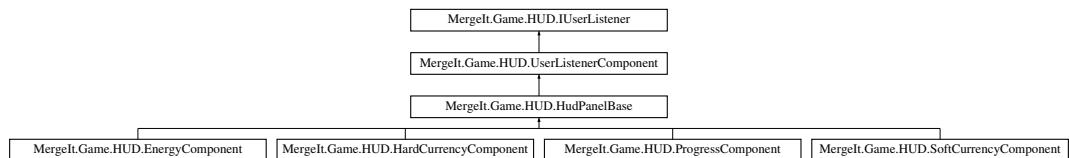
Definition at line 13 of file [IUserData.cs](#).
00013 { get; set; }

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/User/IUserData.cs

6.237 MergeIt.Game.HUD.IUserListener Interface Reference

Inheritance diagram for MergeIt.Game.HUD.IUserListener:



Public Member Functions

- void **ApplyModel** ([UserServiceModel](#) userServiceModel)

6.237.1 Detailed Description

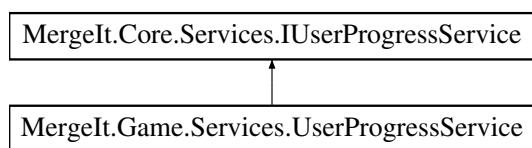
Definition at line 7 of file [IUserListener.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Game/HUD/IUserListener.cs

6.238 Mergelt.Core.Services.IUserProgressService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IUserProgressService:



Public Member Functions

- int **GetCurrentLevelMaxExp ()**
- bool **CanLevelUp ()**
- [**ElementConfig\[\] GetLevelUpPrizes \(\)**](#)

6.238.1 Detailed Description

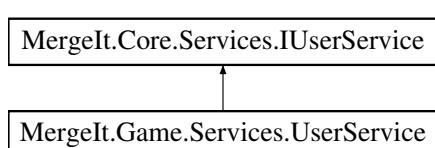
Definition at line 7 of file [IUserProgressService.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Services/IUserProgressService.cs

6.239 Mergelt.Core.Services.IUserService Interface Reference

Inheritance diagram for Mergelt.Core.Services.IUserService:



Public Member Functions

- void **CreateUser ()**
- void **SetupUser ([IUserData](#) userData)**

6.239.1 Detailed Description

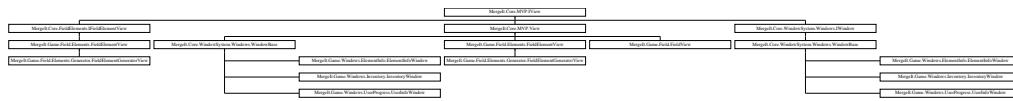
Definition at line 7 of file [IUserService.cs](#).

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/Services/IUserService.cs

6.240 Mergelt.Core.MVP.IView Interface Reference

Inheritance diagram for Mergelt.Core.MVP.IView:



Public Member Functions

- void **Initialize** ()

Properties

- **GameObject** [GameObject](#) [get]

Events

- Action `InitializeEvent`
 - Action `DestroyEvent`

6.240.1 Detailed Description

Definition at line 8 of file [IView.cs](#).

6.240.2 Property Documentation

6.240.2.1 GameObject

GameObject MergeIt.Core.MVP.IView.GameObject [get]

Definition at line 13 of file `IView.cs`.

00013 { get; }

6.240.3 Event Documentation

6.240.3.1 DestroyEvent

Action MergeIt.Core.MVP.IView.DestroyEvent

Definition at line 11 of file [IView.cs](#).

6.240.3.2 InitializeEvent

Action MergeIt.Core.MVP.IView.InitializeEvent

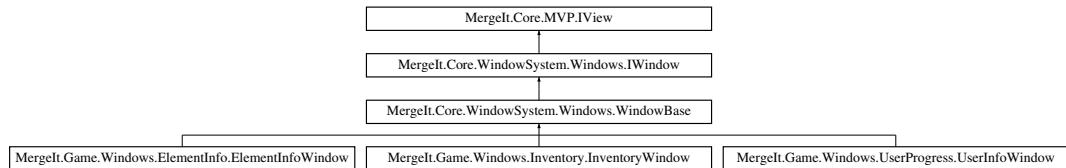
Definition at line 10 of file [IView.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/MVP/IView.cs

6.241 Mergelt.Core.WindowSystem.Windows.IWindow Interface Reference

Inheritance diagram for Mergelt.Core.WindowSystem.Windows.IWindow:



Public Member Functions

- void **Show** ()
- void **Close** ()
- void **SetLayer** (string layer)

Public Member Functions inherited from [Mergelt.Core.MVP.IView](#)

- void **Initialize** ()

Events

- Action [InitiateCloseEvent](#)
- Action [ShowStartEvent](#)
- Action [ShowEndEvent](#)
- Action [CloseStartEvent](#)
- Action [CloseEndEvent](#)

Events inherited from [Mergelt.Core.MVP.IView](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

Additional Inherited Members

Properties inherited from [Mergelt.Core.MVP.IView](#)

- GameObject [GameObject](#) [get]

6.241.1 Detailed Description

Definition at line 8 of file [IWindow.cs](#).

6.241.2 Event Documentation

6.241.2.1 CloseEndEvent

Action MergeIt.Core.WindowSystem.Windows.IWindow.CloseEndEvent

Definition at line 15 of file [IWindow.cs](#).

6.241.2.2 CloseStartEvent

Action MergeIt.Core.WindowSystem.Windows.IWindow.CloseStartEvent

Definition at line 14 of file [IWindow.cs](#).

6.241.2.3 InitiateCloseEvent

Action MergeIt.Core.WindowSystem.Windows.IWindow.InitiateCloseEvent

Definition at line 10 of file [IWindow.cs](#).

6.241.2.4 ShowEndEvent

Action MergeIt.Core.WindowSystem.Windows.IWindow.ShowEndEvent

Definition at line 13 of file [IWindow.cs](#).

6.241.2.5 ShowStartEvent

Action MergeIt.Core.WindowSystem.Windows.IWindow.ShowStartEvent

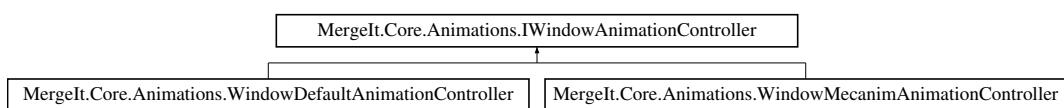
Definition at line 12 of file [IWindow.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/Windows/IWindow.cs

6.242 Mergelt.Core.Animations.IWindowAnimationController Interface Reference

Inheritance diagram for Mergelt.Core.Animations.IWindowAnimationController:



Public Member Functions

- void **Initialize** ([IWindowAnimationListener](#) listener)
- void **OpenWindow** ()
- void **CloseWindow** ()
- void **OnOpenEnd** ()
- void **OnCloseEnd** ()

6.242.1 Detailed Description

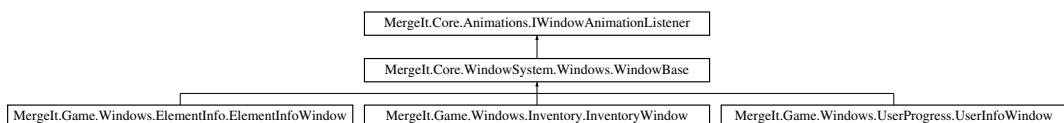
Definition at line 5 of file [IWindowAnimationController.cs](#).

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/Animations/IWindowAnimationController.cs

6.243 Mergelt.Core.Animations.IWindowAnimationListener Interface Reference

Inheritance diagram for Mergelt.Core.Animations.IWindowAnimationListener:



Public Member Functions

- void **OnOpenStarted** ()
- void **OnOpenFinished** ()
- void **OnCloseStarted** ()
- void **OnCloseFinished** ()

6.243.1 Detailed Description

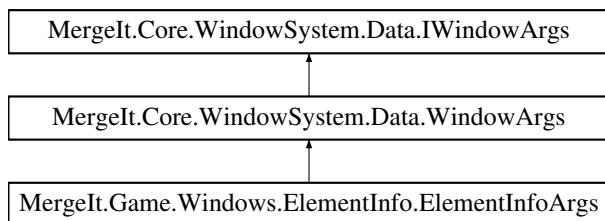
Definition at line 5 of file [IWindowAnimationListener.cs](#).

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/Animations/IWindowAnimationListener.cs

6.244 MergeIt.Core.WindowSystem.Data.IWindowArgs Interface Reference

Inheritance diagram for MergeIt.Core.WindowSystem.Data.IWindowArgs:



6.244.1 Detailed Description

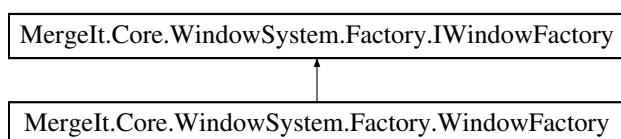
Definition at line 5 of file [IWindowArgs.cs](#).

The documentation for this interface was generated from the following file:

- Assets/MergeIt/Source/Core/WindowSystem/Data/IWindowArgs.cs

6.245 MergeIt.Core.WindowSystem.Factory.IWindowFactory Interface Reference

Inheritance diagram for MergeIt.Core.WindowSystem.Factory.IWindowFactory:



Public Member Functions

- RectTransform **GetRoot** ()
- BlackoutComponent **GetBlackout** (RectTransform parent)
- TPresenter **CreateWindow< TPresenter >** (WindowCreateInfo prefabPath, Transform parent, IWindowArgs windowArgs=null)

6.245.1 Detailed Description

Definition at line 9 of file [IWindowFactory.cs](#).

6.245.2 Member Function Documentation

6.245.2.1 CreateWindow< TPresenter >()

```
TPresenter MergeIt.Core.WindowSystem.Factory.IWindowFactory.CreateWindow< TPresenter > (
    WindowCreateInfo prefabPath,
    Transform parent,
    IWindowArgs windowArgs = null )
```

Type Constraints

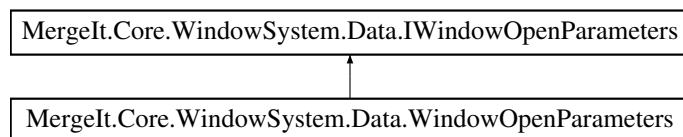
TPresenter : class
TPresenter : [IWindowPresenter](#)
TPresenter : new()

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/Factory/IWindowFactory.cs

6.246 Mergelt.Core.WindowSystem.Data.IWindowOpenParameters Interface Reference

Inheritance diagram for Mergelt.Core.WindowSystem.Data.IWindowOpenParameters:



Properties

- bool **ClosePrevious** [get, set]
- bool **NeedBlackout** [get, set]
- [IWindowPresenter](#) **Presenter** [get, set]

6.246.1 Detailed Description

Definition at line 7 of file [IWindowOpenParameters.cs](#).

6.246.2 Property Documentation

6.246.2.1 ClosePrevious

```
bool MergeIt.Core.WindowSystem.Data.IWindowOpenParameters.ClosePrevious [get], [set]
```

Definition at line 9 of file [IWindowOpenParameters.cs](#).

```
00009 { get; set; }
```

6.246.2.2 NeedBlackout

```
bool MergeIt.Core.WindowSystem.Data.IWindowOpenParameters.NeedBlackout [get], [set]
```

Definition at line 10 of file [IWindowOpenParameters.cs](#).

```
00010 { get; set; }
```

6.246.2.3 Presenter

```
IWindowPresenter MergeIt.Core.WindowSystem.Data.IWindowOpenParameters.Presenter [get], [set]
```

Definition at line 11 of file [IWindowOpenParameters.cs](#).

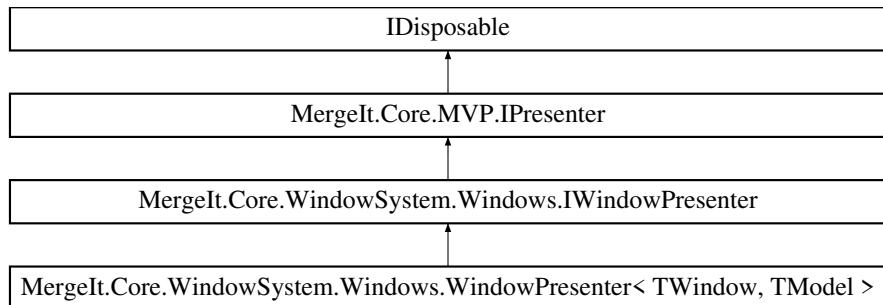
```
00011 { get; set; }
```

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/Data/IWindowOpenParameters.cs

6.247 MergeIt.Core.WindowSystem.Windows.IWindowPresenter Interface Reference

Inheritance diagram for MergeIt.Core.WindowSystem.Windows.IWindowPresenter:



Public Member Functions

- void **Initialize** ([IView](#) view, string layer, [IWindowArgs](#) windowArgs=null)
- void **Show** ()
- void **Hide** ()
- void **Close** ()
- void **SetWindowActive** (bool active)
- void **SetWindowLayer** ()

Public Member Functions inherited from [Mergelt.Core.MVP.IPresenter](#)

- void **Initialize** ([IView](#) view)

Properties

- string **Layer** [get]
- WindowState **State** [get]

6.247.1 Detailed Description

Definition at line 8 of file [IWindowPresenter.cs](#).

6.247.2 Property Documentation

6.247.2.1 Layer

```
string MergeIt.Core.WindowSystem.Windows.IWindowPresenter.Layer [get]
```

Definition at line 10 of file [IWindowPresenter.cs](#).
00010 { [get](#); }

6.247.2.2 State

```
WindowState MergeIt.Core.WindowSystem.Windows.IWindowPresenter.State [get]
```

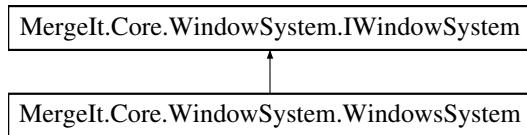
Definition at line 11 of file [IWindowPresenter.cs](#).
00011 { [get](#); }

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/Windows/IWindowPresenter.cs

6.248 Mergelt.Core.WindowSystem.IWindowSystem Interface Reference

Inheritance diagram for Mergelt.Core.WindowSystem.IWindowSystem:



Public Member Functions

- void `OpenWindow< TPresenter >` (bool `closePrevious=false`, bool `enableBlackout=false`, `IWindowArgs args=null`)
- void `CloseWindow` (`IWindowPresenter` presenter)

6.248.1 Detailed Description

Definition at line 8 of file [IWindowSystem.cs](#).

6.248.2 Member Function Documentation

6.248.2.1 OpenWindow< TPresenter >()

```
void MergeIt.Core.WindowSystem.IWindowSystem.OpenWindow< TPresenter > (
    bool closePrevious = false,
    bool enableBlackout = false,
    IWindowArgs args = null )
```

Type Constraints

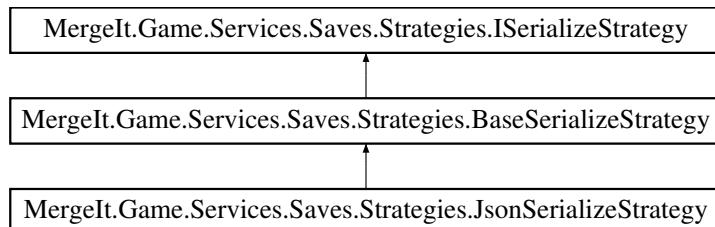
`TPresenter : class`
`TPresenter : IWindowPresenter`
`TPresenter : new()`

The documentation for this interface was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/IWindowSystem.cs

6.249 Mergelt.Game.Services.Saves.Strategies.JsonSerializeStrategy Class Reference

Inheritance diagram for Mergelt.Game.Services.Saves.Strategies.JsonSerializeStrategy:



Public Member Functions

- `async override UniTask Save< T > (T data)`
- `override T Load< T > ()`

Public Member Functions inherited from [Mergelt.Game.Services.Saves.Strategies.BaseSerializeStrategy](#)

- `UniTask Save< T > (T data)`
- `T Load< T > ()`

Additional Inherited Members

Properties inherited from [Mergelt.Game.Services.Saves.Strategies.BaseSerializeStrategy](#)

- `virtual string SaveDir [get]`

Properties inherited from [Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy](#)

6.249.1 Detailed Description

Definition at line 11 of file [JsonSerializeStrategy.cs](#).

6.249.2 Member Function Documentation

6.249.2.1 Load< T >()

```
override T MergeIt.Game.Services.Saves.Strategies.JsonSerializeStrategy.Load< T > () [inline]
```

Implements [Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy](#).

Definition at line 29 of file [JsonSerializeStrategy.cs](#).

```
00030     {
00031         string fileName = SavesHelper.GetFileNameWithExtension<T>();
00032         string loadPath = Path.Combine(Application.streamingAssetsPath, SaveDir, fileName);
00033         if (File.Exists(loadPath))
00034         {
00035             string fileStream = File.ReadAllText(loadPath);
00036             var loadedData = JsonConvert.DeserializeObject<T>(fileStream);
00037             return loadedData;
00038         }
00039         Debug.Log($"There is no saved {fileName} found.");
00040         return null;
00041     }
```

6.249.2.2 Save< T >()

```
async override UniTask MergeIt.Game.Services.Saves.Strategies.JsonSerializeStrategy.Save< T >
(
    T data ) [inline]
```

Implements [MergeIt.Game.Services.Saves.Strategies.ISerializeStrategy](#).

Definition at line 13 of file [JsonSerializeStrategy.cs](#).

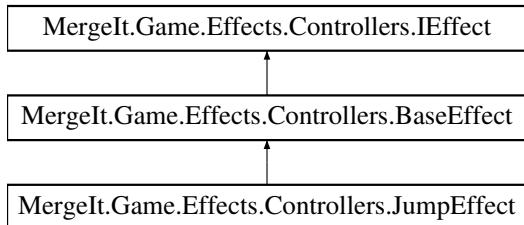
```
00014     {
00015         string fileName = SavesHelper.GetFileNameWithExtension<T>();
00016         string serialized = JsonConvert.SerializeObject(data);
00017
00018         string path = Path.Combine(Application.streamingAssetsPath, SaveDir);
00019         if (!Directory.Exists(path))
00020         {
00021             Directory.CreateDirectory(path);
00022         }
00023
00024         path = Path.Combine(path, fileName);
00025
00026         await File.WriteAllTextAsync(path, serialized);
00027     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Services/Saves/Strategies/JsonSerializeStrategy.cs

6.250 [MergeIt.Game.Effects.Controllers.JumpEffect](#) Class Reference

Inheritance diagram for [MergeIt.Game.Effects.Controllers.JumpEffect](#):



Public Member Functions

- [override void Update \(\)](#)
- [override void Setup \(Transform target, \[IEffectParameters\]\(#\) effectParameters=null, Action finishedCallback=null\)](#)

Public Member Functions inherited from [MergeIt.Game.Effects.Controllers.BaseEffect](#)

- [virtual void Start \(\)](#)
- [void Update \(\)](#)

Additional Inherited Members

Protected Member Functions inherited from [MergeIt.Game.Effects.Controllers.BaseEffect](#)

- [void Finish \(\)](#)
- [virtual void OnStarted \(\)](#)
- [virtual void OnFinished \(\)](#)

Protected Attributes inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- Transform [Target](#)
- Animator [Animator](#)
- Action [FinishedCallback](#)
- readonly [IConfigsService](#) [ConfigsService](#) = DiContainer.Get<[IConfigsService](#)>()

Properties inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- bool [Started](#) [get, set]

6.250.1 Detailed Description

Definition at line 11 of file [JumpEffect.cs](#).

6.250.2 Constructor & Destructor Documentation

6.250.2.1 [JumpEffect\(\)](#)

MergeIt.Game.Effects.Controllers.JumpEffect.JumpEffect () [inline]

Definition at line 22 of file [JumpEffect.cs](#).

```
00023     {
00024         _effectConfig = ConfigsService.GetEffectConfig<JumpEffectConfig>(EffectName.JumpElement);
00025     }
```

6.250.3 Member Function Documentation

6.250.3.1 [Setup\(\)](#)

```
override void MergeIt.Game.Effects.Controllers.JumpEffect.Setup (
    Transform target,
    IEffectParameters effectParameters = null,
    Action finishedCallback = null ) [inline], [virtual]
```

Reimplemented from [Mergelt.Game.Effects.Controllers.BaseEffect](#).

Definition at line 55 of file [JumpEffect.cs](#).

```
00056     {
00057         base.Setup(target, effectParameters, finishedCallback);
00058
00059         _initialPosition = target.position;
00060         _initialScale = target.localScale;
00061
00062         _jumpCurve = CurvesHelper.CopyCurveFrom(_effectConfig.CurveY, _initialPosition.y);
00063         _scaleCurve = CurvesHelper.CopyCurveFrom(_effectConfig.CurveY, _initialScale.x);
00064     }
```

6.250.3.2 Update()

```
override void MergeIt.Game.Effects.Controllers.JumpEffect.Update ( ) [inline]
```

Implements [Mergelt.Game.Effects.Controllers.IEffect](#).

Definition at line 27 of file [JumpEffect.cs](#).

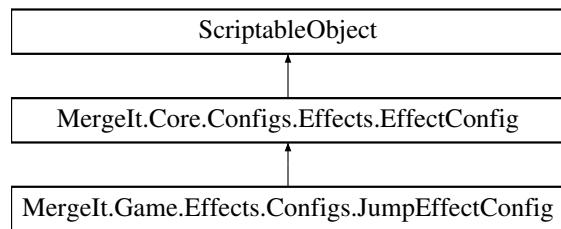
```
00028     {
00029         if (Started)
00030         {
00031             _progress += Time.deltaTime / _effectConfig.Duration;
00032
00033             float posY = _jumpCurve.Evaluate(_progress);
00034             float scale = _scaleCurve.Evaluate(_progress);
00035
00036             Vector3 position = Target.position;
00037             Vector3 localScale = Target.localScale;
00038             position.y = posY;
00039             localScale.x = scale;
00040             localScale.y = scale;
00041
00042             Target.position = position;
00043             Target.localScale = localScale;
00044
00045             if (_progress >= 1f)
00046             {
00047                 Target.position = _initialPosition;
00048                 Target.localScale = _initialScale;
00049
00050                 Finish();
00051             }
00052         }
00053     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Effects/Controllers/JumpEffect.cs

6.251 Mergelt.Game.Effects.Configs.JumpEffectConfig Class Reference

Inheritance diagram for Mergelt.Game.Effects.Configs.JumpEffectConfig:



Public Attributes

- AnimationCurve [CurveY](#)
- AnimationCurve [ScaleCurve](#)

Public Attributes inherited from [Mergelt.Core.Configs.Effects.EffectConfig](#)

- string [Name](#)
- float [Duration](#)

6.251.1 Detailed Description

Definition at line 9 of file [JumpEffectConfig.cs](#).

6.251.2 Member Data Documentation

6.251.2.1 CurveY

AnimationCurve `MergeIt.Game.Effects.Configs.JumpEffectConfig.CurveY`

Definition at line 11 of file [JumpEffectConfig.cs](#).

6.251.2.2 ScaleCurve

AnimationCurve `MergeIt.Game.Effects.Configs.JumpEffectConfig.ScaleCurve`

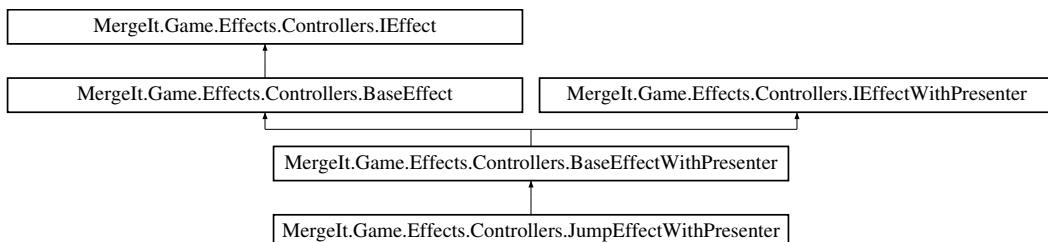
Definition at line 12 of file [JumpEffectConfig.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Effects/Configs/JumpEffectConfig.cs

6.252 MergeIt.Game.Effects.Controllers.JumpEffectWithPresenter Class Reference

Inheritance diagram for MergeIt.Game.Effects.Controllers.JumpEffectWithPresenter:



Public Member Functions

- override void [Start \(\)](#)
- override void [Setup \(IFieldElementPresenter presenter, IEffectParameters effectParameters=null, Action finishedCallback=null\)](#)
- override void [Update \(\)](#)

Public Member Functions inherited from MergeIt.Game.Effects.Controllers.BaseEffectWithPresenter

- [IFieldElementPresenter GetPresenter \(\)](#)

Public Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- virtual void [Setup](#) (Transform target, [IEffectParameters](#) effectParameters=null, Action finishedCallback=null)
- void [Update](#) ()

Protected Member Functions

- override void [OnStarted](#) ()

Protected Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter](#)

- virtual void [FinishCallbackHandler](#) ()

Protected Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- void [Finish](#) ()
- virtual void [OnFinished](#) ()

Additional Inherited Members

Protected Attributes inherited from [Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter](#)

- [IFieldElementPresenter](#) Presenter

Protected Attributes inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- Transform [Target](#)
- Animator [Animator](#)
- Action [FinishedCallback](#)
- readonly [IConfigsService](#) [ConfigsService](#) = DiContainer.Get<[IConfigsService](#)>()

Properties inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- bool [Started](#) [get, set]

6.252.1 Detailed Description

Definition at line 9 of file [JumpEffectWithPresenter.cs](#).

6.252.2 Member Function Documentation

6.252.2.1 OnStarted()

```
override void MergeIt.Game.Effects.Controllers.JumpEffectWithPresenter.OnStarted ( ) [inline],  
[protected], [virtual]
```

Reimplemented from [MergeIt.Game.Effects.Controllers.BaseEffect](#).

Definition at line 35 of file [JumpEffectWithPresenter.cs](#).

```
00036     {  
00037         base.OnStarted();  
00038  
00039         Presenter.Canvas.sortingOrder = 2;  
00040         Presenter.SetBusy(true);  
00041     }
```

6.252.2.2 Setup()

```
override void MergeIt.Game.Effects.Controllers.JumpEffectWithPresenter.Setup (   
    IFieldElementPresenter presenter,  
    IEfectParameters effectParameters = null,  
    Action finishedCallback = null ) [inline], [virtual]
```

Reimplemented from [MergeIt.Game.Effects.Controllers.BaseEffectWithPresenter](#).

Definition at line 20 of file [JumpEffectWithPresenter.cs](#).

```
00021     {  
00022         base.Setup(presenter, effectParameters, finishedCallback);  
00023  
00024         _effect.Setup(presenter.RectTransform, effectParameters, FinishCallbackHandler);  
00025     }
```

6.252.2.3 Start()

```
override void MergeIt.Game.Effects.Controllers.JumpEffectWithPresenter.Start ( ) [inline],  
[virtual]
```

Reimplemented from [MergeIt.Game.Effects.Controllers.BaseEffect](#).

Definition at line 13 of file [JumpEffectWithPresenter.cs](#).

```
00014     {  
00015         base.Start();  
00016  
00017         _effect.Start();  
00018     }
```

6.252.2.4 Update()

```
override void MergeIt.Game.Effects.Controllers.JumpEffectWithPresenter.Update ( ) [inline]
```

Implements [MergeIt.Game.Effects.Controllers.IEffect](#).

Definition at line 27 of file [JumpEffectWithPresenter.cs](#).

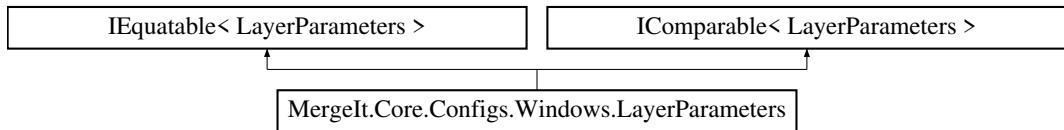
```
00028     {  
00029         if (_effect.Started)  
00030         {  
00031             _effect.Update();  
00032         }  
00033     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Effects/Controllers/JumpEffectWithPresenter.cs

6.253 MergIt.Core.Configs.Windows.LayerParameters Class Reference

Inheritance diagram for MergIt.Core.Configs.Windows.LayerParameters:



Public Member Functions

- bool Equals (LayerParameters other)
- override bool Equals (object obj)
- override int GetHashCode ()
- int CompareTo (LayerParameters other)

Properties

- string Name [get]
- int Order [get]

6.253.1 Detailed Description

Definition at line 9 of file [LayerParameters.cs](#).

6.253.2 Member Function Documentation

6.253.2.1 CompareTo()

```
int MergeIt.Core.Configs.Windows.LayerParameters.CompareTo (
    LayerParameters other ) [inline]
```

Definition at line 46 of file [LayerParameters.cs](#).

```
00047     {
00048         if (ReferenceEquals(this, other)) return 0;
00049         if (ReferenceEquals(null, other)) return 1;
00050         return _order.CompareTo(other._order);
00051     }
```

6.253.2.2 Equals() [1/2]

```
bool MergeIt.Core.Configs.Windows.LayerParameters.Equals (
    LayerParameters other ) [inline]
```

Definition at line 27 of file [LayerParameters.cs](#).

```
00028     {
00029         if (ReferenceEquals(null, other)) return false;
00030         if (ReferenceEquals(this, other)) return true;
00031         return _name == other._name && _order == other._order;
00032     }
```

6.253.2.3 Equals() [2/2]

```
override bool MergeIt.Core.Configs.Windows.LayerParameters.Equals (
    object obj) [inline]
```

Definition at line 33 of file [LayerParameters.cs](#).

```
00034     {
00035         if (ReferenceEquals(null, obj)) return false;
00036         if (ReferenceEquals(this, obj)) return true;
00037         if (obj.GetType() != this.GetType()) return false;
00038         return Equals((LayerParameters)obj);
00039     }
```

6.253.2.4 GetHashCode()

```
override int MergeIt.Core.Configs.Windows.LayerParameters.GetHashCode () [inline]
```

Definition at line 41 of file [LayerParameters.cs](#).

```
00042     {
00043         return HashCode.Combine(_name, _order);
00044     }
```

6.253.3 Property Documentation

6.253.3.1 Name

```
string MergeIt.Core.Configs.Windows.LayerParameters.Name [get]
```

Definition at line 17 of file [LayerParameters.cs](#).

```
00018     {
00019         get => _name;
00020     }
```

6.253.3.2 Order

```
int MergeIt.Core.Configs.Windows.LayerParameters.Order [get]
```

Definition at line 22 of file [LayerParameters.cs](#).

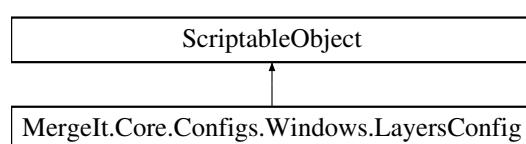
```
00023     {
00024         get => _order;
00025     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Configs/Windows/LayerParameters.cs

6.254 Mergelt.Core.Configs.Windows.LayersConfig Class Reference

Inheritance diagram for Mergelt.Core.Configs.Windows.LayersConfig:



Properties

- `LayerParameters[] Layers [get]`

6.254.1 Detailed Description

Definition at line 8 of file [LayersConfig.cs](#).

6.254.2 Property Documentation

6.254.2.1 Layers

`LayerParameters [] MergeIt.Core.Configs.Windows.LayersConfig.Layers [get]`

Definition at line 13 of file [LayersConfig.cs](#).

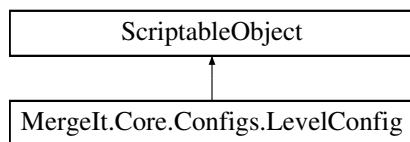
```
00014     {  
00015         get => _layers;  
00016     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Configs/Windows/LayersConfig.cs

6.255 MergeIt.Core.Configs.LevelConfig Class Reference

Inheritance diagram for MergeIt.Core.Configs.LevelConfig:



Public Attributes

- int `FieldWidth`
- int `FieldHeight`
- `SchemeObject EvolutionsScheme`
- List<`LevelElementData`> `FieldElementsData`

6.255.1 Detailed Description

Definition at line 11 of file [LevelConfig.cs](#).

6.255.2 Member Data Documentation

6.255.2.1 EvolutionsScheme

```
SchemeObject MergeIt.Core.Configs.LevelConfig.EvolutionsScheme
```

Definition at line 15 of file [LevelConfig.cs](#).

6.255.2.2 FieldElementsData

```
List<LevelElementData> MergeIt.Core.Configs.LevelConfig.FieldElementsData
```

Definition at line 16 of file [LevelConfig.cs](#).

6.255.2.3 FieldHeight

```
int MergeIt.Core.Configs.LevelConfig.FieldHeight
```

Definition at line 14 of file [LevelConfig.cs](#).

6.255.2.4 FieldWidth

```
int MergeIt.Core.Configs.LevelConfig.FieldWidth
```

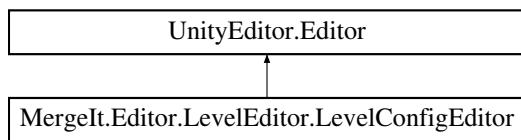
Definition at line 13 of file [LevelConfig.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Configs/LevelConfig.cs

6.256 Mergelt.Editor.LevelEditor.LevelConfigEditor Class Reference

Inheritance diagram for Mergelt.Editor.LevelEditor.LevelConfigEditor:



Public Member Functions

- override void [OnInspectorGUI \(\)](#)

6.256.1 Detailed Description

Definition at line 10 of file [LevelConfigEditor.cs](#).

6.256.2 Member Function Documentation

6.256.2.1 OnInspectorGUI()

```
override void MergeIt.Editor.LevelEditor.LevelConfigEditor.OnInspectorGUI () [inline]
```

Definition at line 12 of file [LevelConfigEditor.cs](#).

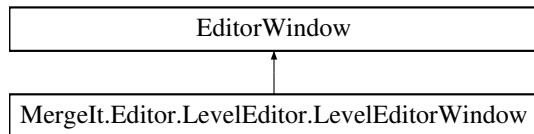
```
00013     {
00014         if (GUILayout.Button("Open"))
00015     {
00016         LevelEditorWindow.Show(target as LevelConfig);
00017     }
00018 }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Editor/Core/LevelEditor/LevelConfigEditor.cs

6.257 MergeIt.Editor.LevelEditor.LevelEditorWindow Class Reference

Inheritance diagram for MergeIt.Editor.LevelEditor.LevelEditorWindow:



Public Member Functions

- void [CreateGUI \(\)](#)
- void [ApplyCell \(LevelElementData newData, bool wasChanged\)](#)
- void [UndoApplyCell \(LevelElementData previousData, LevelElementData newData\)](#)
- void [ClearCell \(GridPoint point\)](#)
- void [UndoClearCell \(LevelElementData previousData\)](#)

Static Public Member Functions

- static void [ShowWindow \(\)](#)
- static void [Show \(LevelConfig levelConfig\)](#)

6.257.1 Detailed Description

Definition at line 25 of file [LevelEditorWindow.cs](#).

6.257.2 Member Function Documentation

6.257.2.1 ApplyCell()

```
void MergeIt.Editor.LevelEditor.LevelEditorWindow.ApplyCell (
    LevelElementData newData,
    bool wasChanged ) [inline]
```

Definition at line 225 of file [LevelEditorWindow.cs](#).

```
00226     {
00227         SetupElementCell(newData);
00228         CellSelected(true, newData.Position.X, newData.Position.Y);
00229         DrawElementCell(newData, _selectedToggle);
00230
00231         if (!wasChanged)
00232         {
00233             SetItemData();
00234         }
00235     }
```

6.257.2.2 ClearCell()

```
void MergeIt.Editor.LevelEditor.LevelEditorWindow.ClearCell (
    GridPoint point ) [inline]
```

Definition at line 250 of file [LevelEditorWindow.cs](#).

```
00251     {
00252         CellSelected(true, point.X, point.Y);
00253         _fieldElements.Remove(point);
00254         _selectedToggle.style.backgroundImage = null;
00255
00256         ChangeVisualLock(false, _selectedToggle);
00257     }
```

6.257.2.3 CreateGUI()

```
void MergeIt.Editor.LevelEditor.LevelEditorWindow.CreateGUI ( ) [inline]
```

Definition at line 92 of file [LevelEditorWindow.cs](#).

```
00093     {
00094         _window = this;
00095         _commandManager = new ActionCommandManager();
00096
00097         VisualElement root = rootVisualElement;
00098         root.focusable = true;
00099         root.RegisterCallback<KeyDownEvent>(evt =>
00100         {
00101             if (evt.commandKey || evt.ctrlKey)
00102             {
00103                 switch (evt.keyCode)
00104                 {
00105                     case KeyCode.Z:
00106                         _commandManager.Undo();
00107                         evt.StopPropagation();
00108                         break;
00109
00110                     case KeyCode.Y:
00111                         _commandManager.Redo();
00112                         evt.StopPropagation();
00113                         break;
00114                 }
00115             }
00116         });
00117
00118         _lockTexture =
00119             AssetDatabase.LoadAssetAtPath<Texture>("Assets/MergeIt/Content/Images/Common/lock.png");
00120
00121         var visualTree =
00122             (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.LevelEditorResourcesRoot,
```

```

00122             "LevelEditorWindow.uxml");
00123     VisualElement windowRoot = visualTree.CloneTree();
00124     root.Add(windowRoot);
00125
00126     _globalStyle =
00127         (StyleSheet)EditorGUIUtility.Load(Path.Combine(Constants.LevelEditorResourcesRoot,
00128             "LevelEditorWindow.uss"));
00129     root.styleSheets.Add(_globalStyle);
00130     root.RegisterCallback<KeyUpEvent>(evt =>
00131     {
00132         if (evt.keyCode == KeyCode.S && (evt.commandKey || evt.ctrlKey))
00133         {
00134             SaveButtonClicked();
00135         }
00136     });
00137
00138     _evolutionsField = root.Q<ObjectField>("EvolutionsField");
00139     _evolutionsField.objectType = typeof(SchemeObject);
00140     _evolutionsField.RegisterValueChangedCallback(OnEvolutionsConfigChanged);
00141
00142     ToolbarMenu toolbarMenu = windowRoot.Q<ToolbarMenu>("FileMenu");
00143     toolbarMenu.menu.AppendAction("Open...", _ => LoadButtonClicked());
00144     toolbarMenu.menu.AppendAction("Save", _ => SaveButtonClicked());
00145     toolbarMenu.menu.AppendAction("Save As...", _ => SaveAsButtonClicked());
00146
00147     _createButton = root.Q<ToolbarButton>("CreateButton");
00148     _createButton.clicked += CreateButtonClicked;
00149
00150     _createLevelButton = root.Q<Button>("CreateLevelButton");
00151     _createLevelButton.clicked += CreateLevelButtonClicked;
00152
00153     _createLevelPopup = root.Q<PopupWindow>("CreateLevelPopup");
00154     _createLevelPopup.visible = false;
00155
00156     _createFieldHeight = root.Q<IntegerField>("FieldHeight");
00157     _createFieldWidth = root.Q<IntegerField>("FieldWidth");
00158
00159     _createFieldHeight.RegisterValueChangedCallback(OnFieldSizeChanged);
00160     _createFieldWidth.RegisterValueChangedCallback(OnFieldSizeChanged);
00161
00162     _warnLabel = root.Q<Label>("WarningLabel");
00163     _warnLabel.visible = false;
00164
00165     _grid = root.Q<VisualElement>("ItemsGrid");
00166     _grid.focusable = true;
00167     _grid.RegisterCallback<KeyUpEvent>(evt =>
00168     {
00169         if (evt.keyCode is KeyCode.Delete or KeyCode.Backspace)
00170         {
00171             if (_selectedToggle != null)
00172             {
00173                 ClearButtonClicked();
00174             }
00175         }
00176         else if (evt.commandKey || evt.ctrlKey)
00177         {
00178             switch (evt.keyCode)
00179             {
00180                 case KeyCode.C:
00181                     if (_selectedCell != GridPoint.Default)
00182                     {
00183                         _copiedCell = _selectedCell;
00184                     }
00185
00186                     break;
00187
00188                 case KeyCode.V:
00189                     if (_selectedToggle != null &&
00190                         _copiedCell != GridPoint.Default &&
00191                         _selectedCell != _copiedCell)
00192                     {
00193                         CopyCell(_copiedCell.X + 1, _copiedCell.Y + 1);
00194                     }
00195
00196                     break;
00197                 }
00198             }
00199         });
00200
00201     _itemFrameSettings = root.Q<VisualElement>("ItemFrameSettings");
00202     SwitchSideBarVisibility(false);
00203     _isLockedToggle = _itemFrameSettings.Q<Toggle>("IsLocked");
00204     _isLockedToggle.RegisterValueChangedCallback(OnIsLockedChanged);
00205
00206     _itemFrameEvolutions = _itemFrameSettings.Q<VisualElement>("EvoPanel");
00207
00208     _itemImage = _itemFrameSettings.Q<Image>("ItemIcon");

```

```

00209
00210     _applyButton = _itemFrameSettings.Q<Button>("ApplyButton");
00211     _applyButton.clicked += ApplyButtonClicked;
00212
00213     _clearButton = _itemFrameSettings.Q<Button>("ClearButton");
00214     _clearButton.clicked += ClearButtonClicked;
00215
00216     _copyButton = _itemFrameSettings.Q<Button>("CopyButton");
00217     _copyRow = _itemFrameSettings.Q<IntegerField>("CopyRow");
00218     _copyColumn = _itemFrameSettings.Q<IntegerField>("CopyColumn");
00219
00220     _copyButton.clicked += CopyButtonClicked;
00221
00222     CheckCreateButton();
00223 }
```

6.257.2.4 Show()

```
static void MergeIt.Editor.LevelEditor.LevelEditorWindow.Show (
    LevelConfig levelConfig) [inline], [static]
```

Definition at line 80 of file [LevelEditorWindow.cs](#).

```

00081     {
00082         _window = GetWindow<LevelEditorWindow>(true);
00083         _window.minSize = new Vector2(800, 600);
00084         _window.titleContent = new GUIContent("New level");
00085
00086         if (levelConfig != null)
00087         {
00088             _window.Reload(levelConfig);
00089         }
00090     }
```

6.257.2.5 ShowWindow()

```
static void MergeIt.Editor.LevelEditor.LevelEditorWindow.ShowWindow () [inline], [static]
```

Definition at line 75 of file [LevelEditorWindow.cs](#).

```

00076     {
00077         Show(null);
00078     }
```

6.257.2.6 UndoApplyCell()

```
void MergeIt.Editor.LevelEditor.LevelEditorWindow.UndoApplyCell (
    LevelElementData previousData,
    LevelElementData newData) [inline]
```

Definition at line 237 of file [LevelEditorWindow.cs](#).

```

00238     {
00239         if (previousData != null)
00240         {
00241             CellSelected(true, previousData.Position.X, previousData.Position.Y);
00242             ApplyCell(previousData, true);
00243         }
00244         else
00245         {
00246             ClearCell(newData.Position);
00247         }
00248     }
```

6.257.2.7 UndoClearCell()

```
void MergeIt.Editor.LevelEditor.LevelEditorWindow.UndoClearCell (
    LevelElementData previousData ) [inline]
```

Definition at line 259 of file [LevelEditorWindow.cs](#).

```
00260     {
00261         SetupElementCell(previousData);
00262         CellSelected(true, previousData.Position.X, previousData.Position.Y);
00263         DrawElementCell(previousData, _selectedToggle);
00264     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Editor/Core/LevelEditor/LevelEditorWindow.cs

6.258 MergeIt.Core.Configs.Data.LevelElementData Class Reference

Public Member Functions

- void [CopyFrom \(LevelElementData itemData, bool copyPosition=true\)](#)
- [LevelElementData GetClone \(bool copyPosition=true\)](#)

Public Attributes

- string [EvolutionId](#)
- [ElementConfig Element](#)
- bool [IsBlocked](#)
- [GridPoint Position](#)

6.258.1 Detailed Description

Definition at line 11 of file [LevelElementData.cs](#).

6.258.2 Member Function Documentation

6.258.2.1 CopyFrom()

```
void MergeIt.Core.Configs.Data.LevelElementData.CopyFrom (
    LevelElementData itemData,
    bool copyPosition = true ) [inline]
```

Definition at line 18 of file [LevelElementData.cs](#).

```
00019     {
00020         EvolutionId = itemData.EvolutionId;
00021         Element = itemData.Element;
00022         IsBlocked = itemData.IsBlocked;
00023
00024         if (copyPosition)
00025         {
00026             Position = itemData.Position.Copy();
00027         }
00028     }
```

6.258.2.2 GetClone()

```
LevelElementData MergeIt.Core.Configs.Data.LevelElementData.GetClone (
    bool copyPosition = true ) [inline]
```

Definition at line 30 of file [LevelElementData.cs](#).

```
00031     {
00032         var fieldCellData = new LevelElementData();
00033         fieldCellData.CopyFrom(this, copyPosition);
00034         return fieldCellData;
00036     }
```

6.258.3 Member Data Documentation

6.258.3.1 Element

```
ElementConfig MergeIt.Core.Configs.Data.LevelElementData.Element
```

Definition at line 14 of file [LevelElementData.cs](#).

6.258.3.2 EvolutionId

```
string MergeIt.Core.Configs.Data.LevelElementData.EvolutionId
```

Definition at line 13 of file [LevelElementData.cs](#).

6.258.3.3 IsBlocked

```
bool MergeIt.Core.Configs.Data.LevelElementData.IsBlocked
```

Definition at line 15 of file [LevelElementData.cs](#).

6.258.3.4 Position

```
GridPoint MergeIt.Core.Configs.Data.LevelElementData.Position
```

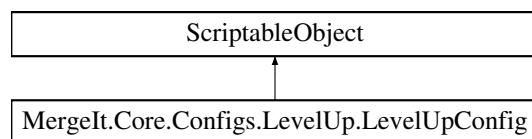
Definition at line 16 of file [LevelElementData.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Configs/Data/LevelElementData.cs

6.259 Mergelt.Core.Configs.LevelUp.LevelUpConfig Class Reference

Inheritance diagram for Mergelt.Core.Configs.LevelUp.LevelUpConfig:



Public Attributes

- [LevelUpParameters\[\] LevelUp](#)

6.259.1 Detailed Description

Definition at line 8 of file [LevelUpConfig.cs](#).

6.259.2 Member Data Documentation

6.259.2.1 LevelUp

[LevelUpParameters \[\] MergeIt.Core.Configs.LevelUp.LevelUpConfig.LevelUp](#)

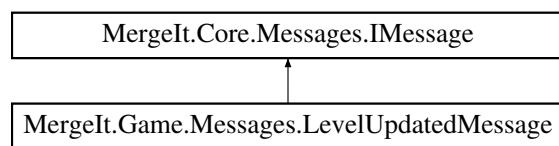
Definition at line 10 of file [LevelUpConfig.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Configs/LevelUp/LevelUpConfig.cs

6.260 MergeIt.Game.Messages.LevelUpdatedMessage Class Reference

Inheritance diagram for MergeIt.Game.Messages.LevelUpdatedMessage:



Properties

- int [NextLevelExp \[get, set\]](#)

6.260.1 Detailed Description

Definition at line 7 of file [LevelUpdatedMessage.cs](#).

6.260.2 Property Documentation

6.260.2.1 NextLevelExp

`int MergeIt.Game.Messages.LevelUpdatedMessage.NextLevelExp [get], [set]`

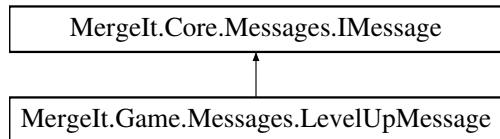
Definition at line 9 of file [LevelUpdatedMessage.cs](#).
00009 { get; set; }

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Messages/LevelUpdatedMessage.cs

6.261 Mergelt.Game.Messages.LevelUpMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.LevelUpMessage:



6.261.1 Detailed Description

Definition at line 7 of file [LevelUpMessage.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/LevelUpMessage.cs

6.262 Mergelt.Core.Configs.LevelUp.LevelUpParameters Class Reference

Public Attributes

- int [Experience](#)
- [ElementConfig\[\] Bonuses](#)

6.262.1 Detailed Description

Definition at line 9 of file [LevelUpParameters.cs](#).

6.262.2 Member Data Documentation

6.262.2.1 Bonuses

[ElementConfig](#) [] MergeIt.Core.Configs.LevelUp.LevelUpParameters.Bonuses

Definition at line 12 of file [LevelUpParameters.cs](#).

6.262.2.2 Experience

int MergeIt.Core.Configs.LevelUp.LevelUpParameters.Experience

Definition at line 11 of file [LevelUpParameters.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Configs/LevelUp/LevelUpParameters.cs

6.263 MergeIt.Editor.Core.LevelEditor.Commands.LimitedStack< T > Class Template Reference

Public Member Functions

- `LimitedStack` (int `maxSize`)
- void `Push` (T `item`)
- T `Pop` ()
- T `Peek` ()
- bool `Any` ()
- void `Clear` ()

Properties

- int `Count` [get]

6.263.1 Detailed Description

Definition at line 8 of file [LimitedStack.cs](#).

6.263.2 Constructor & Destructor Documentation

6.263.2.1 LimitedStack()

```
MergeIt.Editor.Core.LevelEditor.Commands.LimitedStack< T >.LimitedStack (
    int maxSize ) [inline]
```

Definition at line 15 of file [LimitedStack.cs](#).

```
00016     {
00017         _maxSize = maxSize;
00018     }
```

6.263.3 Member Function Documentation

6.263.3.1 Any()

```
bool MergeIt.Editor.Core.LevelEditor.Commands.LimitedStack< T >.Any ( ) [inline]
```

Definition at line 44 of file [LimitedStack.cs](#).

```
00045     {
00046         return _list.Any();
00047     }
```

6.263.3.2 Clear()

```
void MergeIt.Editor.Core.LevelEditor.Commands.LimitedStack< T >.Clear ( ) [inline]
```

Definition at line 49 of file [LimitedStack.cs](#).

```
00050     {
00051         _list.Clear();
00052     }
```

6.263.3.3 Peek()

```
T MergeIt.Editor.Core.LevelEditor.Commands.LimitedStack< T >.Peek ( ) [inline]
```

Definition at line 39 of file [LimitedStack.cs](#).

```
00040     {
00041         return _list.Count > 0 ? _list.First.Value : default;
00042     }
```

6.263.3.4 Pop()

```
T MergeIt.Editor.Core.LevelEditor.Commands.LimitedStack< T >.Pop ( ) [inline]
```

Definition at line 29 of file [LimitedStack.cs](#).

```
00030     {
00031         if (_list.Count == 0)
00032             return default;
00033
00034         var value = _list.First.Value;
00035         _list.RemoveFirst();
00036         return value;
00037     }
```

6.263.3.5 Push()

```
void MergeIt.Editor.Core.LevelEditor.Commands.LimitedStack< T >.Push (
    T item) [inline]
```

Definition at line 20 of file [LimitedStack.cs](#).

```
00021     {
00022         if (_list.Count >= _maxSize)
00023         {
00024             _list.RemoveLast();
00025         }
00026         _list.AddFirst(item);
00027     }
```

6.263.4 Property Documentation

6.263.4.1 Count

```
int MergeIt.Editor.Core.LevelEditor.Commands.LimitedStack< T >.Count [get]
```

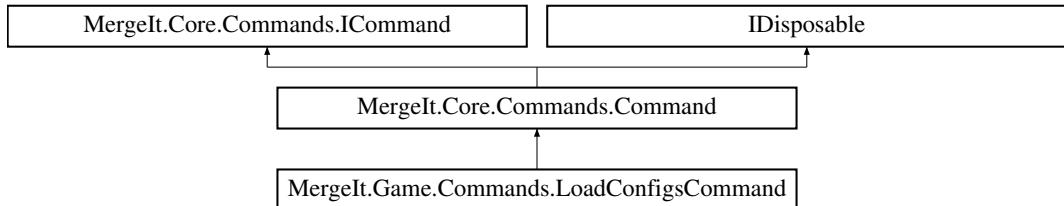
Definition at line 13 of file [LimitedStack.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/LevelEditor/Commands/LimitedStack.cs

6.264 Mergelt.Game.Commands.LoadConfigsCommand Class Reference

Inheritance diagram for Mergelt.Game.Commands.LoadConfigsCommand:



Public Member Functions

- `async override UniTask ExecuteAsync ()`

Public Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- `virtual void Execute ()`
- `void Dispose ()`

Additional Inherited Members

Protected Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- `virtual void Finish ()`
- `virtual void OnDispose ()`

Events inherited from [Mergelt.Core.Commands.Command](#)

- `Action< ICommand > Finished`

Events inherited from [Mergelt.Core.Commands.ICommand](#)

- `Action< ICommand > Finished`

6.264.1 Detailed Description

Definition at line 10 of file [LoadConfigsCommand.cs](#).

6.264.2 Member Function Documentation

6.264.2.1 ExecuteAsync()

```
async override UniTask MergeIt.Game.Commands.LoadConfigsCommand.ExecuteAsync ( ) [inline],  
[virtual]
```

Reimplemented from [Mergelt.Core.Commands.Command](#).

Definition at line 14 of file [LoadConfigsCommand.cs](#).

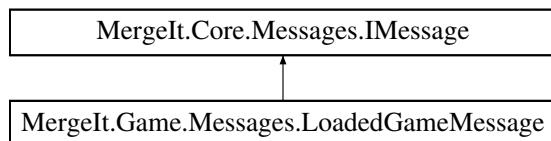
```
00015      {  
00016          await _configsService.Load();  
00017      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Commands/LoadConfigsCommand.cs

6.265 Mergelt.Game.Messages.LoadedGameMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.LoadedGameMessage:



6.265.1 Detailed Description

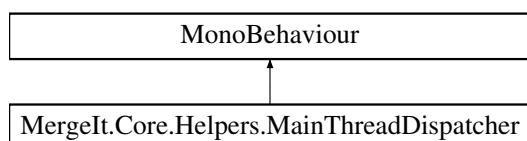
Definition at line 7 of file [LoadedGameMessage.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/LoadedGameMessage.cs

6.266 Mergelt.Core.Helpers.MainThreadDispatcher Class Reference

Inheritance diagram for Mergelt.Core.Helpers.MainThreadDispatcher:



Public Member Functions

- void [Enqueue](#) (Action action)
- void [RunOnMainThread](#) (Action action)
- bool [IsMainThread](#) ()

Properties

- static [MainThreadDispatcher Instance](#) [get]

6.266.1 Detailed Description

Definition at line 10 of file [MainThreadDispatcher.cs](#).

6.266.2 Member Function Documentation

6.266.2.1 Enqueue()

```
void MergeIt.Core.Helpers.MainThreadDispatcher.Enqueue (
    Action action) [inline]
```

Definition at line 47 of file [MainThreadDispatcher.cs](#).

```
00048     {
00049         if (action == null)
00050         {
00051             Debug.LogError("No action to enqueue.");
00052             return;
00053         }
00054
00055         lock (ExecutionQueue)
00056         {
00057             ExecutionQueue.Enqueue(action);
00058         }
00059     }
```

6.266.2.2 IsMainThread()

```
bool MergeIt.Core.Helpers.MainThreadDispatcher.IsMainThread () [inline]
```

Definition at line 79 of file [MainThreadDispatcher.cs](#).

```
00080     {
00081         return Thread.CurrentThread.ManagedThreadId == _mainThreadId;
00082     }
```

6.266.2.3 RunOnMainThread()

```
void MergeIt.Core.Helpers.MainThreadDispatcher.RunOnMainThread (
    Action action) [inline]
```

Definition at line 61 of file [MainThreadDispatcher.cs](#).

```
00062     {
00063         if (action == null)
00064         {
00065             Debug.LogError("No action to run on main thread.");
00066             return;
00067         }
00068
00069         if (IsMainThread())
00070         {
00071             action();
00072         }
00073         else
00074         {
00075             Enqueue(action);
00076         }
00077     }
```

6.266.3 Property Documentation

6.266.3.1 Instance

```
MainThreadDispatcher MergeIt.Core.Helpers.MainThreadDispatcher.Instance [static], [get]
```

Definition at line 13 of file [MainThreadDispatcher.cs](#).

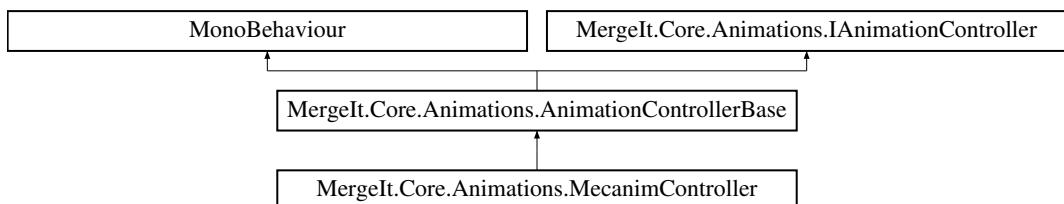
```
00014     {
00015         get
00016     {
00017         if (!_instance)
00018         {
00019             var updater = new GameObject($"{nameof(MainThreadDispatcher)}");
00020             _instance = updater.AddComponent<MainThreadDispatcher>();
00021             DontDestroyOnLoad(updater);
00022         }
00023         return _instance;
00024     }
00025 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Utils/MainThreadDispatcher.cs

6.267 Mergelt.Core.Animations.MecanimController Class Reference

Inheritance diagram for Mergelt.Core.Animations.MecanimController:



Public Member Functions

- override void [Initialize \(IAnimationListener listener\)](#)
- override void [SetState \(string state\)](#)
- override void [SetState \(int state\)](#)
- override void [SetState< T > \(T state\)](#)

Public Member Functions inherited from Mergelt.Core.Animations.AnimationControllerBase

- void [Initialize \(IAnimationListener listener\)](#)

6.267.1 Detailed Description

Definition at line 8 of file [MecanimController.cs](#).

6.267.2 Member Function Documentation

6.267.2.1 Initialize()

```
override void MergeIt.Core.Animations.MecanimController.Initialize (
    IAnimationListener listener) [inline]
```

Implements [MergeIt.Core.Animations.IAnimationController](#).

Definition at line 18 of file [MecanimController.cs](#).

```
00019     {
00020         _listener = listener;
00021     }
```

6.267.2.2 SetState() [1/2]

```
override void MergeIt.Core.Animations.MecanimController.SetState (
    int state) [inline], [virtual]
```

Reimplemented from [MergeIt.Core.Animations.AnimationControllerBase](#).

Definition at line 28 of file [MecanimController.cs](#).

```
00029     {
00030         _animator.Play(state);
00031     }
```

6.267.2.3 SetState() [2/2]

```
override void MergeIt.Core.Animations.MecanimController.SetState (
    string state) [inline], [virtual]
```

Reimplemented from [MergeIt.Core.Animations.AnimationControllerBase](#).

Definition at line 23 of file [MecanimController.cs](#).

```
00024     {
00025         _animator.Play(state);
00026     }
```

6.267.2.4 SetState< T >()

```
override void MergeIt.Core.Animations.MecanimController.SetState< T > (
    T state) [inline], [virtual]
```

Reimplemented from [MergeIt.Core.Animations.AnimationControllerBase](#).

Definition at line 33 of file [MecanimController.cs](#).

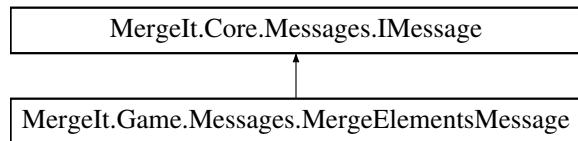
```
00034     {
00035         _animator.Play(state.ToString());
00036     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Animations/MecanimController.cs

6.268 Mergelt.Game.Messages.MergeElementsMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.MergeElementsMessage:



Properties

- `GridPoint From [get, set]`
- `IFieldElement NewElement [get, set]`

6.268.1 Detailed Description

Definition at line 8 of file [MergeElementsMessage.cs](#).

6.268.2 Property Documentation

6.268.2.1 From

`GridPoint MergeIt.Game.Messages.MergeElementsMessage.From [get], [set]`

Definition at line 10 of file [MergeElementsMessage.cs](#).
00010 { get; set; }

6.268.2.2 NewElement

`IFieldElement MergeIt.Game.Messages.MergeElementsMessage.NewElement [get], [set]`

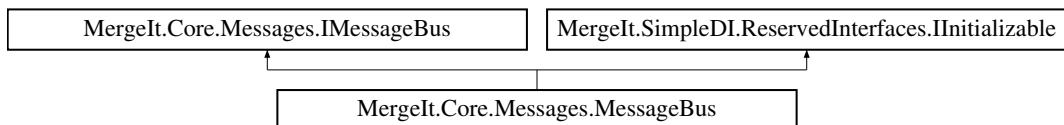
Definition at line 11 of file [MergeElementsMessage.cs](#).
00011 { get; set; }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/MergeElementsMessage.cs

6.269 Mergelt.Core.Messages.MessageBus Class Reference

Inheritance diagram for Mergelt.Core.Messages.MessageBus:



Public Member Functions

- void `Initialize()`
- void `DefineMessage< T >()`
- void `AddListener< T >(Action< T > callback)`
- void `RemoveListener< T >(Action< T > callback)`
- void `Fire< T >(T messageData)`
- void `Fire< T >()`

6.269.1 Detailed Description

Definition at line 11 of file [MessageBus.cs](#).

6.269.2 Member Function Documentation

6.269.2.1 AddListener< T >()

```
void MergeIt.Core.Messages.MessageBus.AddListener< T >(
    Action< T > callback) [inline]
```

Implements [Mergelt.Core.Messages.IMessageBus](#).

Type Constraints

T : IMessage

Definition at line 38 of file [MessageBus.cs](#).

```
00038
00039     {
00040         Type type = typeof(T);
00041
00042         if (!Messages.ContainsKey(type))
00043         {
00044             Debug.LogWarning($"Message with type {type} was not defined and will added to
cache.");
00045
00046             Messages[type] = new List<object>();
00047         }
00048
00049         Messages[type].Add(callback);
00050     }
```

6.269.2.2 DefineMessage< T >()

```
void MergeIt.Core.Messages.MessageBus.DefineMessage< T >() [inline]
```

Implements [Mergelt.Core.Messages.IMessageBus](#).

Type Constraints

T : IMessage

Definition at line 28 of file [MessageBus.cs](#).

```
00028
00029     {
00030         Type type = typeof(T);
00031
00032         if (!Messages.ContainsKey(type))
00033         {
00034             Messages.Add(type, new List<object>());
00035         }
00036     }
```

6.269.2.3 Fire< T >() [1/2]

```
void MergeIt.Core.Messages.MessageBus.Fire< T > ( ) [inline]
```

Implements [MergeIt.Core.Messages.IMessageBus](#).

Type Constraints

T : IMessage

T : new()

Definition at line 85 of file [MessageBus.cs](#).

```
00085                               : IMessage, new()
00086     {
00087         Type type = typeof(T);
00088
00089         if (Messages.TryGetValue(type, out var message))
00090         {
00091             for (var index = 0; index < message.Count;)
00092             {
00093                 var callback = message[index];
00094                 Action<T> action = callback as Action<T>;
00095
00096                 if (action == null)
00097                 {
00098                     message.RemoveAt(index);
00099
00100                     continue;
00101                 }
00102
00103                 action(new T());
00104
00105                 index++;
00106             }
00107         }
00108     }
```

6.269.2.4 Fire< T >() [2/2]

```
void MergeIt.Core.Messages.MessageBus.Fire< T > (
    T messageData ) [inline]
```

Implements [MergeIt.Core.Messages.IMessageBus](#).

Type Constraints

T : IMessage

Definition at line 60 of file [MessageBus.cs](#).

```
00060                               : IMessage
00061     {
00062         Type type = typeof(T);
00063
00064         if (Messages.TryGetValue(type, out var message))
00065         {
00066             for (var index = 0; index < message.Count;)
00067             {
00068                 var callback = message[index];
00069                 Action<T> action = callback as Action<T>;
00070
00071                 if (action == null)
00072                 {
00073                     message.RemoveAt(index);
00074
00075                     continue;
00076                 }
00077
00078                 action(messageData);
00079
00080                 index++;
00081             }
00082         }
00083     }
```

6.269.2.5 Initialize()

```
void MergeIt.Core.Messages.MessageBus.Initialize ( ) [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 15 of file [MessageBus.cs](#).

```
00016      {
00017          Type interfaceType = typeof(IMessage);
00018          IEnumerable<Type> allTypes = AppDomain.CurrentDomain.GetAssemblies()
00019              .SelectMany(assembly => assembly.GetTypes())
00020                  .Where(type => !type.IsAbstract && interfaceType.IsAssignableFrom(type));
00021
00022          foreach (Type type in allTypes)
00023          {
00024              Messages[type] = new List<object>();
00025          }
00026      }
```

6.269.2.6 RemoveListener< T >()

```
void MergeIt.Core.Messages.MessageBus.RemoveListener< T > (
    Action< T > callback ) [inline]
```

Implements [Mergelt.Core.Messages.IMessageBus](#).

Type Constraints

T : IMessage

Definition at line 52 of file [MessageBus.cs](#).

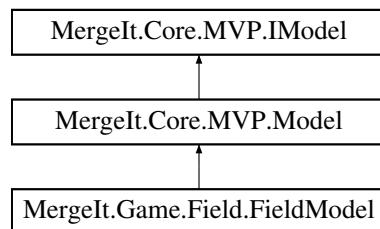
```
00052      : IMessage
00053      {
00054          if (Messages.TryGetValue(typeof(T), out List<object> callbacks))
00055          {
00056              callbacks.Remove(callback);
00057          }
00058      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Messages/MessageBus.cs

6.270 Mergelt.Core.MVP.Model Class Reference

Inheritance diagram for Mergelt.Core.MVP.Model:



6.270.1 Detailed Description

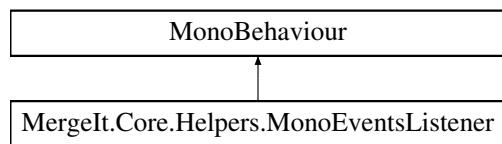
Definition at line 5 of file [Model.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/MVP/Model.cs

6.271 MergeIt.Core.Helpers.MonoEventsListener Class Reference

Inheritance diagram for MergeIt.Core.Helpers.MonoEventsListener:



Public Member Functions

- void [SubscribeOnUpdate](#) ([IMonoUpdateHandler](#) monoUpdateHandler)
- void [UnsubscribeFromUpdate](#) ([IMonoUpdateHandler](#) monoUpdateHandler)
- void [SubscribeOnApplicationQuit](#) ([IMonoApplicationQuitHandler](#) monoApplicationQuitHandler)
- void [UnsubscribeFromApplicationQuit](#) ([IMonoApplicationQuitHandler](#) monoApplicationQuitHandler)

Properties

- static [MonoEventsListener](#) [Instance](#) [get]

6.271.1 Detailed Description

Definition at line 8 of file [MonoEventsListener.cs](#).

6.271.2 Member Function Documentation

6.271.2.1 SubscribeOnApplicationQuit()

```
void MergeIt.Core.Helpers.MonoEventsListener.SubscribeOnApplicationQuit (
    IMonoApplicationQuitHandler monoApplicationQuitHandler ) [inline]
```

Definition at line 39 of file [MonoEventsListener.cs](#).

```
00040      {
00041          ApplicationQuitHandlers.Add(monoApplicationQuitHandler);
00042      }
```

6.271.2.2 SubscribeOnUpdate()

```
void MergeIt.Core.Helpers.MonoEventsListener.SubscribeOnUpdate (
    IMonoUpdateHandler monoUpdateHandler ) [inline]
```

Definition at line 29 of file [MonoEventsListener.cs](#).

```
00030     {
00031         UpdatableHandlers.Add(monoUpdateHandler);
00032     }
```

6.271.2.3 UnsubscribeFromApplicationQuit()

```
void MergeIt.Core.Helpers.MonoEventsListener.UnsubscribeFromApplicationQuit (
    IMonoApplicationQuitHandler monoApplicationQuitHandler ) [inline]
```

Definition at line 44 of file [MonoEventsListener.cs](#).

```
00045     {
00046         ApplicationQuitHandlers.Remove(monoApplicationQuitHandler);
00047     }
```

6.271.2.4 UnsubscribeFromUpdate()

```
void MergeIt.Core.Helpers.MonoEventsListener.UnsubscribeFromUpdate (
    IMonoUpdateHandler monoUpdateHandler ) [inline]
```

Definition at line 34 of file [MonoEventsListener.cs](#).

```
00035     {
00036         UpdatableHandlers.Remove(monoUpdateHandler);
00037     }
```

6.271.3 Property Documentation

6.271.3.1 Instance

`MonoEventsListener` MergeIt.Core.Helpers.MonoEventsListener.Instance [static], [get]

Definition at line 11 of file [MonoEventsListener.cs](#).

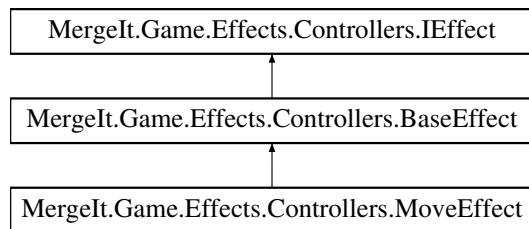
```
00012     {
00013         get
00014     {
00015         if (!_instance)
00016         {
00017             var updater = new GameObject($"{nameof(MonoEventsListener)}");
00018             _instance = updater.AddComponent<MonoEventsListener>();
00019             DontDestroyOnLoad(updater);
00020         }
00021         return _instance;
00022     }
00023 }
00024 }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Utils/MonoEventsListener.cs

6.272 Mergelt.Game.Effects.Controllers.MoveEffect Class Reference

Inheritance diagram for Mergelt.Game.Effects.Controllers.MoveEffect:



Public Member Functions

- override void [Setup](#) (Transform target, [IEffectParameters](#) effectParameters=null, Action finishedCallback=null)
- void [Setup](#) (Vector3 moveFrom, Vector3 moveTo, HudPanelType targetInHud)
- override void [Update](#) ()

Public Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- virtual void [Start](#) ()
- void [Update](#) ()

Additional Inherited Members

Protected Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- void [Finish](#) ()
- virtual void [OnStarted](#) ()
- virtual void [OnFinished](#) ()

Protected Attributes inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- Transform [Target](#)
- Animator [Animator](#)
- Action [FinishedCallback](#)
- readonly [IConfigsService](#) [ConfigsService](#) = DiContainer.Get<[IConfigsService](#)>()

Properties inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- bool [Started](#) [get, set]

6.272.1 Detailed Description

Definition at line 13 of file [MoveEffect.cs](#).

6.272.2 Constructor & Destructor Documentation

6.272.2.1 MoveEffect()

```
MergeIt.Game.Effects.Controllers.MoveEffect.MoveEffect ( ) [inline]
```

Definition at line 23 of file [MoveEffect.cs](#).

```
00024     {
00025         _effectConfig = ConfigsService.GetEffectConfig<MoveEffectConfig>(EffectName.MoveElement);
00026     }
```

6.272.3 Member Function Documentation

6.272.3.1 Setup() [1/2]

```
override void MergeIt.Game.Effects.Controllers.MoveEffect.Setup (
    Transform target,
    IEffectParameters effectParameters = null,
    Action finishedCallback = null ) [inline], [virtual]
```

Reimplemented from [MergeIt.Game.Effects.Controllers.BaseEffect](#).

Definition at line 28 of file [MoveEffect.cs](#).

```
00029     {
00030         base.Setup(target, effectParameters, finishedCallback);
00031
00032         if (effectParameters is MoveEffectParameters moveEffectParameters)
00033         {
00034             Setup(moveEffectParameters.FromPosition, moveEffectParameters.ToPosition,
00035                  moveEffectParameters.TargetInHud);
00036         }
00037     }
```

6.272.3.2 Setup() [2/2]

```
void MergeIt.Game.Effects.Controllers.MoveEffect.Setup (
    Vector3 moveFrom,
    Vector3 moveTo,
    HudPanelType targetInHud ) [inline]
```

Definition at line 38 of file [MoveEffect.cs](#).

```
00039     {
00040         _moveFrom = moveFrom;
00041         _moveTo = moveTo;
00042
00043         Target.position = _moveFrom;
00044
00045         if (targetInHud != 0)
00046         {
00047             _moveTo = HudTargets.GetTarget(targetInHud).position;
00048         }
00049
00050         _scaleCurve = CurvesHelper.CopyCurveFrom(_effectConfig.ScaleCurve, Target.localScale.x);
00051     }
```

6.272.3.3 Update()

```
override void MergeIt.Game.Effects.Controllers.MoveEffect.Update ( ) [inline]
```

Implements [Mergelt.Game.Effects.Controllers.IEffect](#).

Definition at line 53 of file [MoveEffect.cs](#).

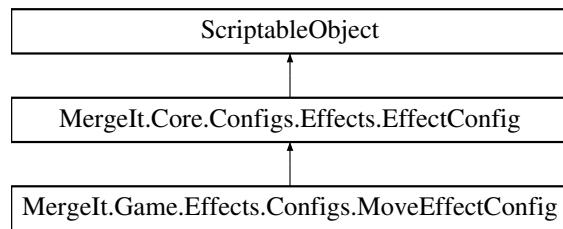
```
00054     {
00055         if (Started)
00056         {
00057             _progress += Time.deltaTime / _effectConfig.Duration;
00058
00059             float posX = _effectConfig.PositionXCurve.Evaluate(_progress);
00060             float posY = _effectConfig.PositionYCurve.Evaluate(_progress);
00061             float scale = _scaleCurve.Evaluate(_progress);
00062
00063             Vector3 position = Vector3.Lerp(_moveFrom, _moveTo, _progress);
00064             position.x += posX;
00065             position.y += posY;
00066
00067             Target.position = position;
00068             Target.localScale = new Vector3(scale, scale, 1f);
00069
00070             if (_progress >= 1f)
00071             {
00072                 Target.position = _moveTo;
00073                 Target.localScale = Vector3.one;
00074
00075                 Finish();
00076             }
00077         }
00078     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Effects/Controllers/MoveEffect.cs

6.273 Mergelt.Game.Effects.Configs.MoveEffectConfig Class Reference

Inheritance diagram for Mergelt.Game.Effects.Configs.MoveEffectConfig:



Public Attributes

- AnimationCurve [PositionXCurve](#)
- AnimationCurve [PositionYCurve](#)
- AnimationCurve [ScaleCurve](#)

Public Attributes inherited from [MergeIt.Core.Configs.Effects.EffectConfig](#)

- string [Name](#)
- float [Duration](#)

6.273.1 Detailed Description

Definition at line 9 of file [MoveEffectConfig.cs](#).

6.273.2 Member Data Documentation

6.273.2.1 PositionXCurve

AnimationCurve `MergeIt.Game.Effects.Configs.MoveEffectConfig.PositionXCurve`

Definition at line 11 of file [MoveEffectConfig.cs](#).

6.273.2.2 PositionYCurve

AnimationCurve `MergeIt.Game.Effects.Configs.MoveEffectConfig.PositionYCurve`

Definition at line 12 of file [MoveEffectConfig.cs](#).

6.273.2.3 ScaleCurve

AnimationCurve `MergeIt.Game.Effects.Configs.MoveEffectConfig.ScaleCurve`

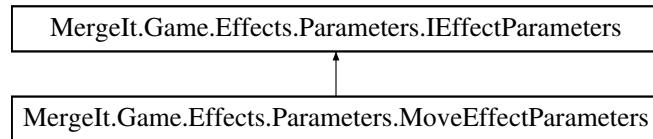
Definition at line 13 of file [MoveEffectConfig.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Effects/Configs/MoveEffectConfig.cs

6.274 MergeIt.Game.Effects.Parameters.MoveEffectParameters Class Reference

Inheritance diagram for MergeIt.Game.Effects.Parameters.MoveEffectParameters:



Public Member Functions

- [MoveEffectParameters](#) (Vector3 fromPosition, Vector3 toPosition=default, HudPanelType targetInHud=0, string overriddenName=null)

Properties

- string `OverriddenName` [get]
- Vector3 `FromPosition` [get]
- Vector3 `ToPosition` [get]
- HudPanelType `TargetInHud` [get]

Properties inherited from [Mergelt.Game.Effects.Parameters.IEffectParameters](#)

6.274.1 Detailed Description

Definition at line 8 of file [MoveEffectParameters.cs](#).

6.274.2 Constructor & Destructor Documentation

6.274.2.1 MoveEffectParameters()

```
MergeIt.Game.Effects.Parameters.MoveEffectParameters.MoveEffectParameters (
    Vector3 fromPosition,
    Vector3 toPosition = default,
    HudPanelType targetInHud = 0,
    string overriddenName = null ) [inline]
```

Definition at line 31 of file [MoveEffectParameters.cs](#).

```
00032     {
00033         OverriddenName = overriddenName;
00034
00035         _fromPosition = fromPosition;
00036         _toPosition = toPosition;
00037         _targetInHud = targetInHud;
00038     }
```

6.274.3 Property Documentation

6.274.3.1 FromPosition

```
Vector3 MergeIt.Game.Effects.Parameters.MoveEffectParameters.FromPosition [get]
```

Definition at line 16 of file [MoveEffectParameters.cs](#).

```
00017     {
00018         get => _fromPosition;
00019     }
```

6.274.3.2 OverriddenName

```
string MergeIt.Game.Effects.Parameters.MoveEffectParameters.OverriddenName [get]
```

Implements [Mergelt.Game.Effects.Parameters.IEffectParameters](#).

Definition at line 14 of file [MoveEffectParameters.cs](#).

```
00014 { get; }
```

6.274.3.3 TargetInHud

```
HudPanelType MergeIt.Game.Effects.Parameters.MoveEffectParameters.TargetInHud [get]
```

Definition at line 26 of file [MoveEffectParameters.cs](#).

```
00027     {
00028         get => _targetInHud;
00029     }
```

6.274.3.4 ToPosition

```
Vector3 MergeIt.Game.Effects.Parameters.MoveEffectParameters.ToPosition [get]
```

Definition at line 21 of file [MoveEffectParameters.cs](#).

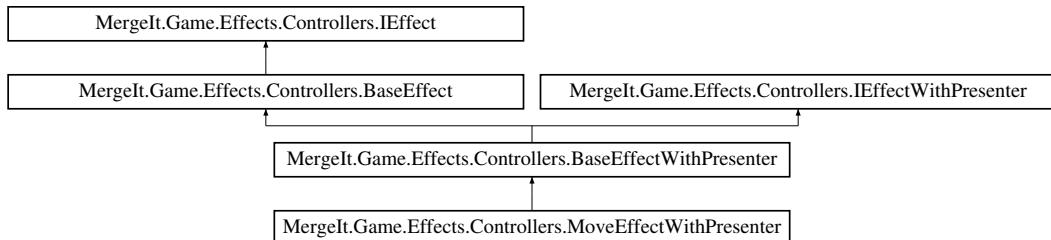
```
00022     {
00023         get => _toPosition;
00024     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Effects/Parameters/MoveEffectParameters.cs

6.275 Mergelt.Game.Effects.Controllers.MoveEffectWithPresenter Class Reference

Inheritance diagram for Mergelt.Game.Effects.Controllers.MoveEffectWithPresenter:



Public Member Functions

- override void [Start \(\)](#)
- override void [Setup \(IFieldElementPresenter presenter, IEffectParameters effectParameters=null, Action finishedCallback=null\)](#)
- override void [Update \(\)](#)

Public Member Functions inherited from Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter

- [IFieldElementPresenter GetPresenter \(\)](#)

Public Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- virtual void [Setup](#) (Transform target, [IEffectParameters](#) effectParameters=null, Action finishedCallback=null)
- void [Update](#) ()

Protected Member Functions

- override void [OnStarted](#) ()
- override void [OnFinished](#) ()

Protected Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter](#)

- virtual void [FinishCallbackHandler](#) ()

Protected Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- void [Finish](#) ()

Additional Inherited Members

Protected Attributes inherited from [Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter](#)

- [IFieldElementPresenter](#) Presenter

Protected Attributes inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- Transform [Target](#)
- Animator [Animator](#)
- Action [FinishedCallback](#)
- readonly [IConfigsService](#) ConfigsService = DiContainer.Get<[IConfigsService](#)>()

Properties inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- bool [Started](#) [get, set]

6.275.1 Detailed Description

Definition at line 9 of file [MoveEffectWithPresenter.cs](#).

6.275.2 Member Function Documentation

6.275.2.1 OnFinished()

```
override void MergeIt.Game.Effects.Controllers.MoveEffectWithPresenter.OnFinished () [inline],  
[protected], [virtual]
```

Reimplemented from [MergeIt.Game.Effects.Controllers.BaseEffect](#).

Definition at line 43 of file [MoveEffectWithPresenter.cs](#).

```
00044     {  
00045         base.OnFinished();  
00046  
00047         Presenter.Canvas.sortingOrder = 1;  
00048         // \_presenter.UpdateInitialPosition\(\);  
00049         Presenter.SetBusy(false);  
00050     }
```

6.275.2.2 OnStarted()

```
override void MergeIt.Game.Effects.Controllers.MoveEffectWithPresenter.OnStarted () [inline],  
[protected], [virtual]
```

Reimplemented from [MergeIt.Game.Effects.Controllers.BaseEffect](#).

Definition at line 35 of file [MoveEffectWithPresenter.cs](#).

```
00036     {  
00037         base.OnStarted();  
00038  
00039         Presenter.Canvas.sortingOrder = 2;  
00040         Presenter.SetBusy(true);  
00041     }
```

6.275.2.3 Setup()

```
override void MergeIt.Game.Effects.Controllers.MoveEffectWithPresenter.Setup (  
    IFieldElementPresenter presenter,  
    IEffectParameters effectParameters = null,  
    Action finishedCallback = null ) [inline], [virtual]
```

Reimplemented from [MergeIt.Game.Effects.Controllers.BaseEffectWithPresenter](#).

Definition at line 20 of file [MoveEffectWithPresenter.cs](#).

```
00021     {  
00022         base.Setup(presenter, effectParameters, finishedCallback);  
00023  
00024         \_effect.Setup(presenter.RectTransform, effectParameters, FinishCallbackHandler);  
00025     }
```

6.275.2.4 Start()

```
override void MergeIt.Game.Effects.Controllers.MoveEffectWithPresenter.Start () [inline],  
[virtual]
```

Reimplemented from [MergeIt.Game.Effects.Controllers.BaseEffect](#).

Definition at line 13 of file [MoveEffectWithPresenter.cs](#).

```
00014     {  
00015         base.Start();  
00016  
00017         \_effect.Start();  
00018     }
```

6.275.2.5 Update()

```
override void MergeIt.Game.Effects.Controllers.MoveEffectWithPresenter.Update () [inline]
```

Implements [Mergelt.Game.Effects.Controllers.IEffect](#).

Definition at line 27 of file [MoveEffectWithPresenter.cs](#).

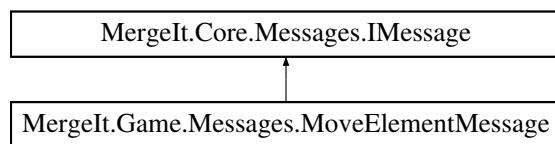
```
00028     {
00029         if (_effect.Started)
00030         {
00031             _effect.Update();
00032         }
00033     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Effects/Controllers/MoveEffectWithPresenter.cs

6.276 Mergelt.Game.Messages.MoveElementMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.MoveElementMessage:



Properties

- [IFieldElementView FieldElementView](#) [get, set]
- [GridPoint FromPoint](#) [get, set]
- [GridPoint ToPoint](#) [get, set]

6.276.1 Detailed Description

Definition at line 8 of file [MoveElementMessage.cs](#).

6.276.2 Property Documentation

6.276.2.1 FieldElementView

```
IFieldElementView MergeIt.Game.Messages.MoveElementMessage.FieldElementView [get], [set]
```

Definition at line 10 of file [MoveElementMessage.cs](#).

```
00010 { get; set; }
```

6.276.2.2 FromPoint

`GridPoint` MergeIt.Game.Messages.MoveElementMessage.FromPoint [get], [set]

Definition at line 11 of file [MoveElementMessage.cs](#).

```
00011 { get; set; }
```

6.276.2.3 ToPoint

`GridPoint` MergeIt.Game.Messages.MoveElementMessageToPoint [get], [set]

Definition at line 12 of file [MoveElementMessage.cs](#).

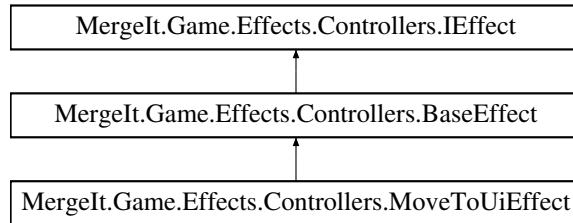
```
00012 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Messages/MoveElementMessage.cs

6.277 MergeIt.Game.Effects.Controllers.MoveToUiEffect Class Reference

Inheritance diagram for MergeIt.Game.Effects.Controllers.MoveToUiEffect:



Public Member Functions

- override void [Setup](#) (Transform target, [IEffectParameters](#) effectParameters=null, Action finishedCallback=null)
- override void [Update](#) ()

Public Member Functions inherited from [MergeIt.Game.Effects.Controllers.BaseEffect](#)

- virtual void [Start](#) ()
- void [Update](#) ()

Additional Inherited Members

Protected Member Functions inherited from [MergeIt.Game.Effects.Controllers.BaseEffect](#)

- void [Finish](#) ()
- virtual void [OnStarted](#) ()
- virtual void [OnFinished](#) ()

Protected Attributes inherited from Mergelt.Game.Effects.Controllers.BaseEffect

- Transform [Target](#)
- Animator [Animator](#)
- Action [FinishedCallback](#)
- readonly [IConfigsService](#) ConfigsService = DiContainer.Get<IConfigsService>()

Properties inherited from Mergelt.Game.Effects.Controllers.BaseEffect

- bool [Started](#) [get, set]

6.277.1 Detailed Description

Definition at line 12 of file [MoveToUiEffect.cs](#).

6.277.2 Member Function Documentation

6.277.2.1 Setup()

```
override void MergeIt.Game.Effects.Controllers.MoveToUiEffect.Setup (
    Transform target,
    IEFFECTPARAMETERS effectParameters = null,
    Action finishedCallback = null )  [inline], [virtual]
```

Reimplemented from [Mergelt.Game.Effects.Controllers.BaseEffect](#).

Definition at line 22 of file [MoveToUiEffect.cs](#).

```
00023     {
00024         base.Setup(target, effectParameters, finishedCallback);
00025
00026         if (effectParameters?.OverriddenName != null)
00027         {
00028             _effectConfig =
00029                 ConfigsService.GetEffectConfig<MoveToUiEffectConfig>(effectParameters.OverriddenName);
00030         }
00031         else
00032         {
00033             _effectConfig =
00034                 ConfigsService.GetEffectConfig<MoveToUiEffectConfig>(EffectName.MoveToUiElement);
00035         }
00036         if (effectParameters is MoveEffectParameters moveEffectParameters)
00037         {
00038             _moveTo = HudTargets.GetTarget(moveEffectParameters.TargetInHud).position;
00039             _moveFrom = moveEffectParameters.FromPosition;
00040         }
00041         _scaleCurve = CurvesHelper.CopyCurveFrom(_effectConfig.ScaleCurve, Target.localScale.x);
00042     }
```

6.277.2.2 Update()

```
override void MergeIt.Game.Effects.Controllers.MoveToUiEffect.Update () [inline]
```

Implements [MergeIt.Game.Effects.Controllers.IEffect](#).

Definition at line 44 of file [MoveToUiEffect.cs](#).

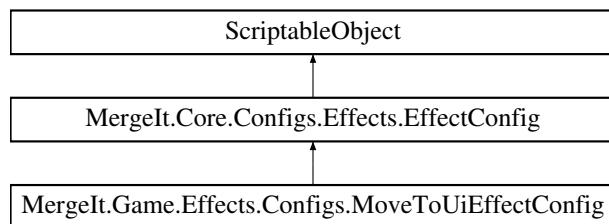
```
00045     {
00046         if (_started)
00047         {
00048             _progress += Time.deltaTime / _effectConfig.Duration;
00049
00050             float posX = _effectConfig.PositionXCurve.Evaluate(_progress);
00051             float posY = _effectConfig.PositionYCurve.Evaluate(_progress);
00052             float scale = _scaleCurve.Evaluate(_progress);
00053
00054             Vector3 position = Vector3.Lerp(_moveFrom, _moveTo, _progress);
00055             position.x += posX;
00056             position.y += posY;
00057
00058             Target.position = position;
00059             Target.localScale = new Vector3(scale, scale, 1f);
00060
00061             if (_progress >= 1f)
00062             {
00063                 Finish();
00064             }
00065         }
00066     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Effects/Controllers/MoveToUiEffect.cs

6.278 MergeIt.Game.Effects.Configs.MoveToUiEffectConfig Class Reference

Inheritance diagram for MergeIt.Game.Effects.Configs.MoveToUiEffectConfig:



Public Attributes

- AnimationCurve [PositionXCurve](#)
- AnimationCurve [PositionYCurve](#)
- AnimationCurve [ScaleCurve](#)

Public Attributes inherited from [MergeIt.Core.Configs.Effects.EffectConfig](#)

- string [Name](#)
- float [Duration](#)

6.278.1 Detailed Description

Definition at line 9 of file [MoveToUiEffectConfig.cs](#).

6.278.2 Member Data Documentation

6.278.2.1 PositionXCurve

`AnimationCurve MergeIt.Game.Effects.Configs.MoveToUiEffectConfig.PositionXCurve`

Definition at line 11 of file [MoveToUiEffectConfig.cs](#).

6.278.2.2 PositionYCurve

`AnimationCurve MergeIt.Game.Effects.Configs.MoveToUiEffectConfig.PositionYCurve`

Definition at line 12 of file [MoveToUiEffectConfig.cs](#).

6.278.2.3 ScaleCurve

`AnimationCurve MergeIt.Game.Effects.Configs.MoveToUiEffectConfig.ScaleCurve`

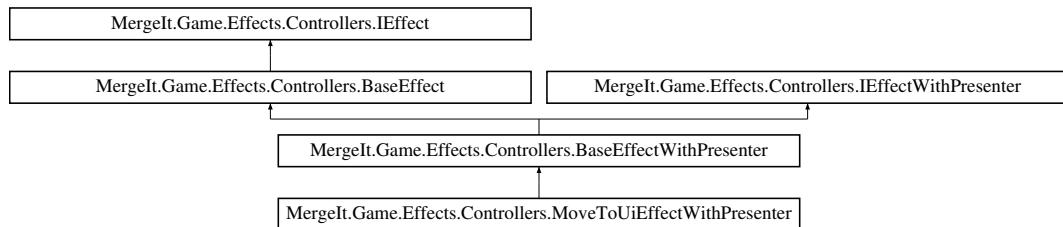
Definition at line 13 of file [MoveToUiEffectConfig.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Effects/Configs/MoveToUiEffectConfig.cs

6.279 Mergelt.Game.Effects.Controllers.MoveToUiEffectWithPresenter Class Reference

Inheritance diagram for `Mergelt.Game.Effects.Controllers.MoveToUiEffectWithPresenter`:



Public Member Functions

- `override void Start ()`
- `override void Setup (IFieldElementPresenter presenter, IEffectParameters effectParameters=null, Action finishedCallback=null)`
- `override void Update ()`

Public Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter](#)

- [IFieldElementPresenter GetPresenter \(\)](#)

Public Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- virtual void [Setup](#) (Transform target, [IEffectParameters](#) effectParameters=null, Action finishedCallback=null)
- void [Update \(\)](#)

Protected Member Functions

- override void [OnStarted \(\)](#)

Protected Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter](#)

- virtual void [FinishCallbackHandler \(\)](#)

Protected Member Functions inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- void [Finish \(\)](#)
- virtual void [OnFinished \(\)](#)

Additional Inherited Members

Protected Attributes inherited from [Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter](#)

- [IFieldElementPresenter](#) Presenter

Protected Attributes inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- Transform [Target](#)
- Animator [Animator](#)
- Action [FinishedCallback](#)
- readonly [IConfigsService](#) ConfigsService = DiContainer.Get<[IConfigsService](#)>()

Properties inherited from [Mergelt.Game.Effects.Controllers.BaseEffect](#)

- bool [Started](#) [get, set]

6.279.1 Detailed Description

Definition at line 9 of file [MoveToUiEffectWithPresenter.cs](#).

6.279.2 Member Function Documentation

6.279.2.1 OnStarted()

```
override void MergeIt.Game.Effects.Controllers.MoveToUiEffectWithPresenter.OnStarted ( ) [inline],  
[protected], [virtual]
```

Reimplemented from [MergeIt.Game.Effects.Controllers.BaseEffect](#).

Definition at line 35 of file [MoveToUiEffectWithPresenter.cs](#).

```
00036      {  
00037          base.OnStarted();  
00038  
00039          Presenter.Canvas.sortingOrder = 2;  
00040          Presenter.SetBusy(true);  
00041      }
```

6.279.2.2 Setup()

```
override void MergeIt.Game.Effects.Controllers.MoveToUiEffectWithPresenter.Setup (  
    IFieldElementPresenter presenter,  
    IEfectParameters effectParameters = null,  
    Action finishedCallback = null ) [inline], [virtual]
```

Reimplemented from [MergeIt.Game.Effects.Controllers.BaseEffectWithPresenter](#).

Definition at line 20 of file [MoveToUiEffectWithPresenter.cs](#).

```
00021      {  
00022          base.Setup(presenter, effectParameters, finishedCallback);  
00023  
00024          _effect.Setup(presenter.RectTransform, effectParameters, FinishCallbackHandler);  
00025      }
```

6.279.2.3 Start()

```
override void MergeIt.Game.Effects.Controllers.MoveToUiEffectWithPresenter.Start ( ) [inline],  
[virtual]
```

Reimplemented from [MergeIt.Game.Effects.Controllers.BaseEffect](#).

Definition at line 13 of file [MoveToUiEffectWithPresenter.cs](#).

```
00014      {  
00015          base.Start();  
00016  
00017          _effect.Start();  
00018      }
```

6.279.2.4 Update()

```
override void MergeIt.Game.Effects.Controllers.MoveToUiEffectWithPresenter.Update ( ) [inline]
```

Implements [MergeIt.Game.Effects.Controllers.IEffect](#).

Definition at line 27 of file [MoveToUiEffectWithPresenter.cs](#).

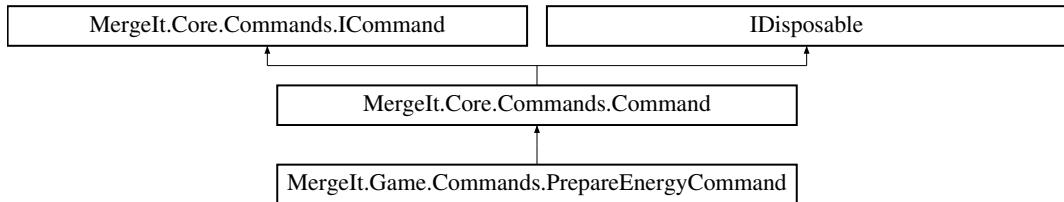
```
00028      {  
00029          if (_effect.Started)  
00030          {  
00031              _effect.Update();  
00032          }  
00033      }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Effects/Controllers/MoveToUiEffectWithPresenter.cs

6.280 Mergelt.Game.Commands.PrepareEnergyCommand Class Reference

Inheritance diagram for Mergelt.Game.Commands.PrepareEnergyCommand:



Public Member Functions

- override void [Execute \(\)](#)

Public Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual async UniTask [ExecuteAsync \(\)](#)
- void [Dispose \(\)](#)

Additional Inherited Members

Protected Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual void [Finish \(\)](#)
- virtual void [OnDispose \(\)](#)

Events inherited from [Mergelt.Core.Commands.Command](#)

- Action< [ICommand](#) > [Finished](#)

Events inherited from [Mergelt.Core.Commands.ICommand](#)

- Action< [ICommand](#) > [Finished](#)

6.280.1 Detailed Description

Definition at line 12 of file [PrepareEnergyCommand.cs](#).

6.280.2 Member Function Documentation

6.280.2.1 Execute()

```
override void MergeIt.Game.Commands.PrepareEnergyCommand.Execute () [inline], [virtual]
```

Reimplemented from [Mergelt.Core.Commands.Command](#).

Definition at line 17 of file [PrepareEnergyCommand.cs](#).

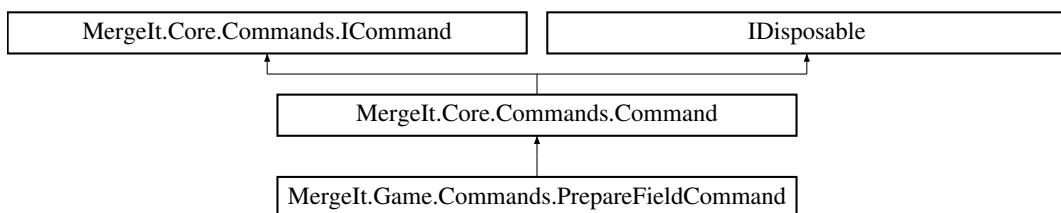
```
00018      {
00019          CheckEnergy ();
00020      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Commands/PrepareEnergyCommand.cs

6.281 Mergelt.Game.Commands.PrepareFieldCommand Class Reference

Inheritance diagram for Mergelt.Game.Commands.PrepareFieldCommand:



Public Member Functions

- `override void Execute ()`

Public Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- `virtual async UniTask ExecuteAsync ()`
- `void Dispose ()`

Additional Inherited Members

Protected Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- `virtual void Finish ()`
- `virtual void OnDispose ()`

Events inherited from [Mergelt.Core.Commands.Command](#)

- `Action< ICommand > Finished`

Events inherited from [Mergelt.Core.Commands.ICommand](#)

- Action< [ICommand](#) > Finished

6.281.1 Detailed Description

Definition at line 18 of file [PrepareFieldCommand.cs](#).

6.281.2 Member Function Documentation

6.281.2.1 Execute()

```
override void MergeIt.Game.Commands.PrepareFieldCommand.Execute ( ) [inline], [virtual]
```

Reimplemented from [Mergelt.Core.Commands.Command](#).

Definition at line 28 of file [PrepareFieldCommand.cs](#).

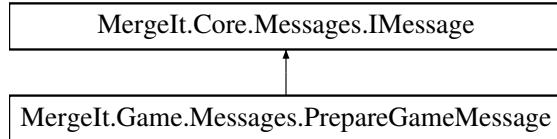
```
00029     {
00030         var loadedLevel = _gameLoadService.Load<FieldData>();
00031
00032         if (loadedLevel != null)
00033     {
00034             _fieldLogicModel.FieldHeight = loadedLevel.FieldHeight;
00035             _fieldLogicModel.FieldWidth = loadedLevel.FieldWidth;
00036
00037             Dictionary<GridPoint, IFieldElement> fieldElements = _fieldLogicModel.FieldElements;
00038
00039             foreach (FieldElementData fieldElementData in loadedLevel.SavedElementsData)
00040             {
00041                 IFieldElement fieldElement =
00042                     _fieldElementFactory.CreateFieldElement(fieldElementData);
00043
00044                 fieldElements.Add(fieldElement.InfoParameters.LogicPosition, fieldElement);
00045             }
00046         else
00047     {
00048         LevelConfig levelConfig = _configsService.LevelConfig;
00049         _fieldLogicModel.FieldHeight = levelConfig.FieldHeight;
00050         _fieldLogicModel.FieldWidth = levelConfig.FieldWidth;
00051
00052         Dictionary<GridPoint, IFieldElement> fieldElements = _fieldLogicModel.FieldElements;
00053
00054         foreach (LevelElementData fieldElementData in levelConfig.FieldElementsData)
00055         {
00056             IFieldElement fieldElement =
00057                 _fieldElementFactory.CreateFieldElement(fieldElementData);
00058
00059                 fieldElements.Add(fieldElement.InfoParameters.LogicPosition, fieldElement);
00060             }
00061
00062             loadedLevel = _configProcessor.BuildLevel();
00063
00064             _gameServiceModel.LoadedLevel = loadedLevel;
00065             _saveService.Save(GameSaveType.Field);
00066         }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Commands/PrepareFieldCommand.cs

6.282 Mergelt.Game.Messages.PrepareGameMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.PrepareGameMessage:



6.282.1 Detailed Description

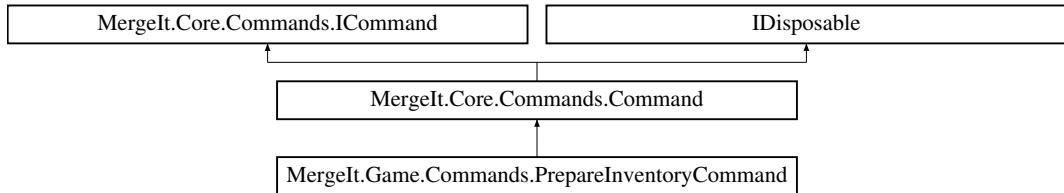
Definition at line 7 of file [PrepareGameMessage.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/PrepareGameMessage.cs

6.283 Mergelt.Game.Commands.PrepareInventoryCommand Class Reference

Inheritance diagram for Mergelt.Game.Commands.PrepareInventoryCommand:



Public Member Functions

- override void [Execute \(\)](#)

Public Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual async UniTask [ExecuteAsync \(\)](#)
- void [Dispose \(\)](#)

Additional Inherited Members

Protected Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual void [Finish \(\)](#)
- virtual void [OnDispose \(\)](#)

Events inherited from [Mergelt.Core.Commands.Command](#)

- Action< [ICommand](#) > Finished

Events inherited from [Mergelt.Core.Commands.ICommand](#)

- Action< [ICommand](#) > Finished

6.283.1 Detailed Description

Definition at line 10 of file [PrepareInventoryCommand.cs](#).

6.283.2 Member Function Documentation

6.283.2.1 Execute()

```
override void MergeIt.Game.Commands.PrepareInventoryCommand.Execute ( ) [inline], [virtual]
```

Reimplemented from [Mergelt.Core.Commands.Command](#).

Definition at line 15 of file [PrepareInventoryCommand.cs](#).

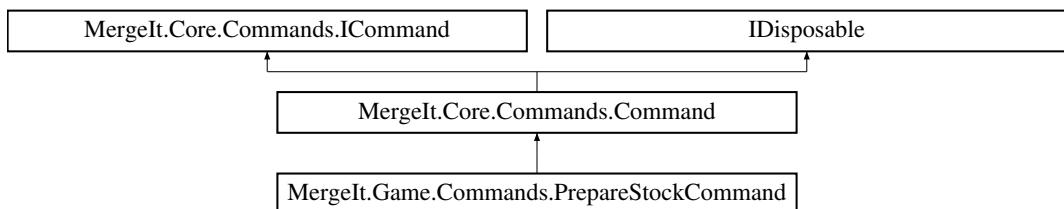
```
00016     {
00017         var inventoryData = _gameLoadService.Load<InventoryData>();
00018
00019         if (inventoryData == null)
00020         {
00021             _inventoryService.CreateInventory();
00022         }
00023         else
00024         {
00025             _inventoryService.SetupInventory(inventoryData);
00026         }
00027     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Commands/PrepareInventoryCommand.cs

6.284 [Mergelt.Game.Commands.PrepareStockCommand](#) Class Reference

Inheritance diagram for [Mergelt.Game.Commands.PrepareStockCommand](#):



Public Member Functions

- override void [Execute \(\)](#)

Public Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual async UniTask [ExecuteAsync \(\)](#)
- void [Dispose \(\)](#)

Additional Inherited Members

Protected Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual void [Finish \(\)](#)
- virtual void [OnDispose \(\)](#)

Events inherited from [Mergelt.Core.Commands.Command](#)

- Action< [ICommand](#) > [Finished](#)

Events inherited from [Mergelt.Core.Commands.ICommand](#)

- Action< [ICommand](#) > [Finished](#)

6.284.1 Detailed Description

Definition at line 10 of file [PrepareStockCommand.cs](#).

6.284.2 Member Function Documentation

6.284.2.1 Execute()

```
override void MergeIt.Game.Commands.PrepareStockCommand.Execute ( ) [inline], [virtual]
```

Reimplemented from [Mergelt.Core.Commands.Command](#).

Definition at line 15 of file [PrepareStockCommand.cs](#).

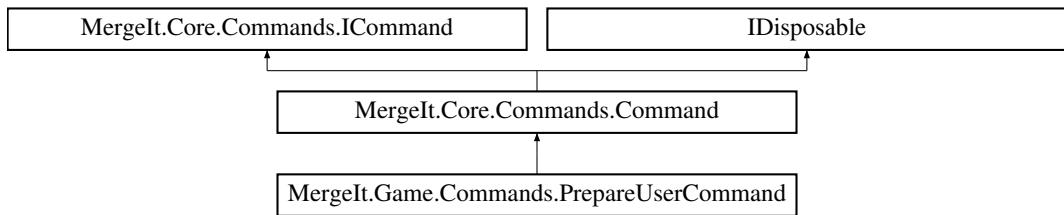
```
00016     {
00017         var stockData = _gameLoadService.Load<ElementsStockData>();
00018
00019         if (stockData == null)
00020         {
00021             _stockService.CreateStock();
00022         }
00023         else
00024         {
00025             _stockService.SetupStock(stockData);
00026         }
00027     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Commands/PrepareStockCommand.cs

6.285 Mergelt.Game.Commands.PrepareUserCommand Class Reference

Inheritance diagram for Mergelt.Game.Commands.PrepareUserCommand:



Public Member Functions

- override void [Execute \(\)](#)

Public Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual async UniTask [ExecuteAsync \(\)](#)
- void [Dispose \(\)](#)

Additional Inherited Members

Protected Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual void [Finish \(\)](#)
- virtual void [OnDispose \(\)](#)

Events inherited from [Mergelt.Core.Commands.Command](#)

- Action< [ICommand](#) > [Finished](#)

Events inherited from [Mergelt.Core.Commands.ICommand](#)

- Action< [ICommand](#) > [Finished](#)

6.285.1 Detailed Description

Definition at line 10 of file [PrepareUserCommand.cs](#).

6.285.2 Member Function Documentation

6.285.2.1 Execute()

```
override void MergeIt.Game.Commands.PrepareUserCommand.Execute () [inline], [virtual]
```

Reimplemented from [Mergelt.Core.Commands.Command](#).

Definition at line 15 of file [PrepareUserCommand.cs](#).

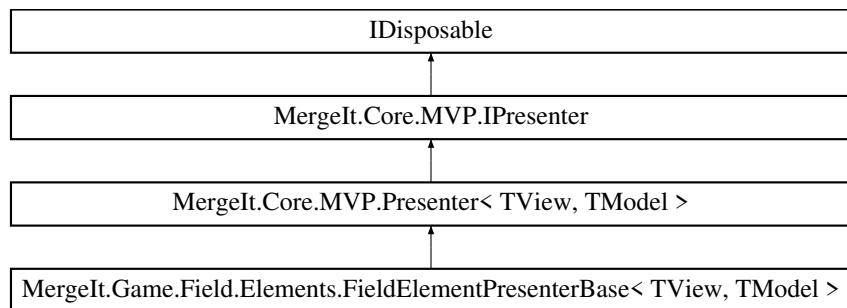
```
00016      {
00017          var userData = _gameLoadService.Load<UserData>();
00018
00019          if (userData == null)
00020          {
00021              _userService.CreateUser();
00022          }
00023          else
00024          {
00025              _userService.SetupUser(userData);
00026          }
00027      }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Commands/PrepareUserCommand.cs

6.286 Mergelt.Core.MVP.Presenter< TView, TModel > Class Template Reference

Inheritance diagram for Mergelt.Core.MVP.Presenter< TView, TModel >:



Public Member Functions

- void [Initialize \(IView view\)](#)
- void [Dispose \(\)](#)

Protected Member Functions

- virtual void [OnInitialize \(TView view\)](#)
- virtual void [OnDispose \(\)](#)

Properties

- TView `View` [get]
- TModel `Model = new()` [get]

6.286.1 Detailed Description

Type Constraints

TView : Component

TView : IView

TModel : IModel

TModel : new()

Definition at line 7 of file `Presenter.cs`.

6.286.2 Member Function Documentation

6.286.2.1 Dispose()

```
void MergeIt.Core.MVP.Presenter< TView, TModel >.Dispose ( ) [inline]
```

Definition at line 22 of file `Presenter.cs`.

```
00023     {
00024         if (View)
00025         {
00026             View.DestroyEvent -= Dispose;
00027         }
00028         OnDispose();
00029     }
```

6.286.2.2 Initialize()

```
void MergeIt.Core.MVP.Presenter< TView, TModel >.Initialize (
    IView view) [inline]
```

Implements `MergeIt.Core.MVP.IPresenter`.

Definition at line 14 of file `Presenter.cs`.

```
00015     {
00016         View = view as TView;
00017         View.DestroyEvent += Dispose;
00018         OnInitialize(View);
00020     }
```

6.286.2.3 OnDispose()

```
virtual void MergeIt.Core.MVP.Presenter< TView, TModel >.OnDispose ( ) [inline], [protected],
[virtual]
```

Definition at line 37 of file `Presenter.cs`.

```
00038     {
00039
00040     }
```

6.286.2.4 OnInitialize()

```
virtual void MergeIt.Core.MVP.Presenter< TView, TModel >.OnInitialize (
    TView view ) [inline], [protected], [virtual]
```

Definition at line 32 of file [Presenter.cs](#).

```
00033     {
00034     }
00035 }
```

6.286.3 Property Documentation

6.286.3.1 Model

```
TModel MergeIt.Core.MVP.Presenter< TView, TModel >.Model = new() [get]
```

Definition at line 12 of file [Presenter.cs](#).

```
00012 { get; } = new();
```

6.286.3.2 View

```
TView MergeIt.Core.MVP.Presenter< TView, TModel >.View [get]
```

Definition at line 11 of file [Presenter.cs](#).

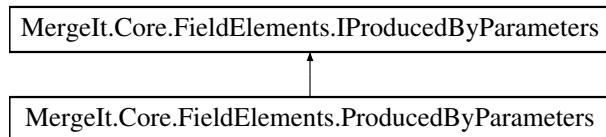
```
00011 { get; private set; }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/MVP/Presenter.cs

6.287 MergeIt.Core.FieldElements.ProducedByParameters Class Reference

Inheritance diagram for MergeIt.Core.FieldElements.ProducedByParameters:



Properties

- List< [ElementConfig](#) > Elements [get, set]

Properties inherited from [Mergelt.Core.FieldElements.IProducedByParameters](#)

6.287.1 Detailed Description

Definition at line 8 of file [ProducedByParameters.cs](#).

6.287.2 Property Documentation

6.287.2.1 Elements

`List<ElementConfig> MergeIt.Core.FieldElements.ProducedByParameters.Elements [get], [set]`

Implements [Mergelt.Core.FieldElements.IProducedByParameters](#).

Definition at line 10 of file [ProducedByParameters.cs](#).

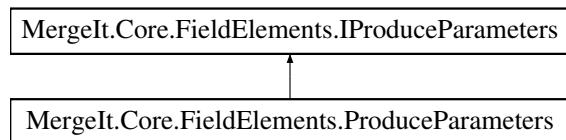
`00010 { get; set; }`

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/Parameters/ProducedByParameters.cs

6.288 Mergelt.Core.FieldElements.ProduceParameters Class Reference

Inheritance diagram for Mergelt.Core.FieldElements.ProduceParameters:



Properties

- `List<GeneratableFieldElement> Elements [get, set]`

Properties inherited from [Mergelt.Core.FieldElements.IProduceParameters](#)

6.288.1 Detailed Description

Definition at line 7 of file [ProduceParameters.cs](#).

6.288.2 Property Documentation

6.288.2.1 Elements

```
List<GeneratableFieldElement> MergeIt.Core.FieldElements.ProduceParameters.Elements [get],  
[set]
```

Implements [MergeIt.Core.FieldElements.IProduceParameters](#).

Definition at line 9 of file [ProduceParameters.cs](#).

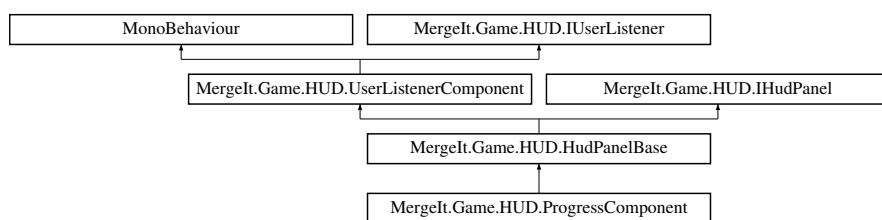
```
00009 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/Parameters/ProduceParameters.cs

6.289 Mergelt.Game.HUD.ProgressComponent Class Reference

Inheritance diagram for Mergelt.Game.HUD.ProgressComponent:



Public Member Functions

- void [OnDestroy](#) ()
- void [Initialize](#) ([IWindowSystem](#) windowSystem)
- void [SetMaxProgress](#) (int nextExperience)
- void [UpdateProgress](#) ()

Public Member Functions inherited from [Mergelt.Game.HUD.UserListenerComponent](#)

- void [ApplyModel](#) ([UserServiceModel](#) userServiceModel)

Protected Member Functions

- override void [Start](#) ()
- override void [OnApplyModel](#) ([UserServiceModel](#) userServiceModel)

Protected Member Functions inherited from [Mergelt.Game.HUD.UserListenerComponent](#)

- void [OnApplyModel](#) ([UserServiceModel](#) userServiceModel)

Properties

- override HudPanelType **Type** [get]

Properties inherited from [Mergelt.Game.HUD.HudPanelBase](#)

- virtual HudPanelType **Type** [get]

Properties inherited from [Mergelt.Game.HUD.UserListenerComponent](#)

- [UserServiceModel](#) **UserServiceModel** [get]

Properties inherited from [Mergelt.Game.HUD.IHudPanel](#)

6.289.1 Detailed Description

Definition at line 14 of file [ProgressComponent.cs](#).

6.289.2 Member Function Documentation

6.289.2.1 Initialize()

```
void MergeIt.Game.HUD.ProgressComponent.Initialize (
    IWindowSystem windowSystem ) [inline]
```

Definition at line 56 of file [ProgressComponent.cs](#).

```
00057     {
00058         _windowSystem = windowSystem;
00059     }
```

6.289.2.2 OnApplyModel()

```
override void MergeIt.Game.HUD.ProgressComponent.OnApplyModel (
    UserServiceModel userServiceModel ) [inline], [protected]
```

Definition at line 71 of file [ProgressComponent.cs](#).

```
00072     {
00073         userServiceModel.Level.Subscribe(OnLevelChanged, true);
00074         userServiceModel.Experience.Subscribe(OnExperienceChanged, true);
00075     }
```

6.289.2.3 OnDestroy()

```
void MergeIt.Game.HUD.ProgressComponent.OnDestroy () [inline]
```

Definition at line 48 of file [ProgressComponent.cs](#).

```
00049     {
00050         _progressButton.onClick.RemoveListener(OnProgressBarClick);
00051
00052         userServiceModel.Level.Unsubscribe(OnLevelChanged);
00053         userServiceModel.Experience.Unsubscribe(OnExperienceChanged);
00054     }
```

6.289.2.4 SetMaxProgress()

```
void MergeIt.Game.HUD.ProgressComponent.SetMaxProgress (
    int nextExperience ) [inline]
```

Definition at line 61 of file [ProgressComponent.cs](#).

```
00062     {
00063         _nextExperience = nextExperience;
00064     }
```

6.289.2.5 Start()

```
override void MergeIt.Game.HUD.ProgressComponent.Start () [inline], [protected], [virtual]
```

Reimplemented from [Mergelt.Game.HUD.HudPanelBase](#).

Definition at line 41 of file [ProgressComponent.cs](#).

```
00042     {
00043         base.Start ();
00044
00045         _progressButton.onClick.AddListener(OnProgressButtonClick);
00046     }
```

6.289.2.6 UpdateProgress()

```
void MergeIt.Game.HUD.ProgressComponent.UpdateProgress () [inline]
```

Definition at line 66 of file [ProgressComponent.cs](#).

```
00067     {
00068         SetProgress (_nextExperience, UserServiceModel.Experience.Value, false);
00069     }
```

6.289.3 Property Documentation

6.289.3.1 Type

```
override HudPanelType MergeIt.Game.HUD.ProgressComponent.Type [get]
```

Implements [Mergelt.Game.HUD.IHudPanel](#).

Definition at line 28 of file [ProgressComponent.cs](#).

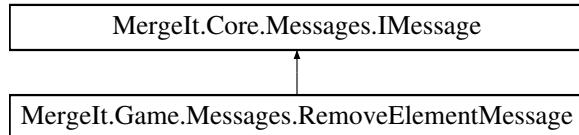
```
00029     {
00030         get => HudPanelType.UserProgress;
00031     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/HUD/ProgressComponent.cs

6.290 Mergelt.Game.Messages.RemoveElementMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.RemoveElementMessage:



Properties

- `GridPoint RemoveAtPoint [get, set]`

6.290.1 Detailed Description

Definition at line 8 of file [RemoveElementMessage.cs](#).

6.290.2 Property Documentation

6.290.2.1 RemoveAtPoint

`GridPoint MergeIt.Game.Messages.RemoveElementMessage.RemoveAtPoint [get], [set]`

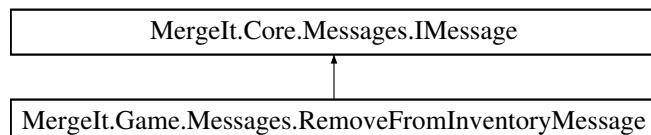
Definition at line 10 of file [RemoveElementMessage.cs](#).
00010 { `get; set;` }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/RemoveElementMessage.cs

6.291 Mergelt.Game.Messages.RemoveFromInventoryMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.RemoveFromInventoryMessage:



Properties

- `IFieldElement FieldElement [get, set]`

6.291.1 Detailed Description

Definition at line 8 of file [RemoveFromInventoryMessage.cs](#).

6.291.2 Property Documentation

6.291.2.1 FieldElement

```
IFieldElement MergeIt.Game.Messages.RemoveFromInventoryMessage.FieldElement [get], [set]
```

Definition at line 10 of file [RemoveFromInventoryMessage.cs](#).

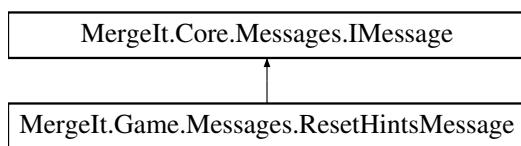
```
00010 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/RemoveFromInventoryMessage.cs

6.292 Mergelt.Game.Messages.ResetHintsMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.ResetHintsMessage:



6.292.1 Detailed Description

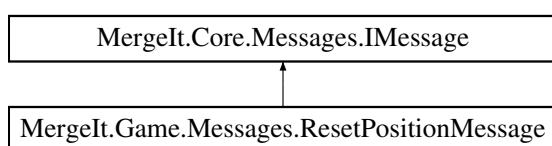
Definition at line 7 of file [ResetHintsMessage.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/ResetHintsMessage.cs

6.293 Mergelt.Game.Messages.ResetPositionMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.ResetPositionMessage:



Properties

- `GridPoint From [get, set]`

6.293.1 Detailed Description

Definition at line 8 of file [ResetPositionMessage.cs](#).

6.293.2 Property Documentation

6.293.2.1 From

`GridPoint MergeIt.Game.Messages.ResetPositionMessage.From [get, set]`

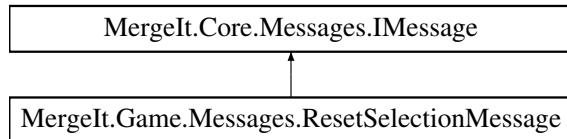
Definition at line 10 of file [ResetPositionMessage.cs](#).
`00010 { get; set; }`

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Messages/ResetPositionMessage.cs

6.294 MergeIt.Game.Messages.ResetSelectionMessage Class Reference

Inheritance diagram for MergeIt.Game.Messages.ResetSelectionMessage:



6.294.1 Detailed Description

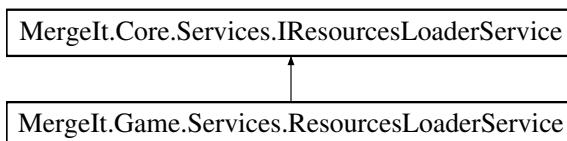
Definition at line 7 of file [ResetSelectionMessage.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Messages/ResetSelectionMessage.cs

6.295 MergeIt.Game.Services.ResourcesLoaderService Class Reference

Inheritance diagram for MergeIt.Game.Services.ResourcesLoaderService:



Public Member Functions

- TResource [GetObject< TResource >\(string path\)](#)

Public Member Functions inherited from [Mergelt.Core.Services.IResourcesLoaderService](#)

- T [GetObject< T >\(string path\)](#)

6.295.1 Detailed Description

Definition at line 8 of file [ResourcesLoaderService.cs](#).

6.295.2 Member Function Documentation

6.295.2.1 [GetObject< TResource >\(\)](#)

```
TResource MergeIt.Game.Services.ResourcesLoaderService.GetObject< TResource > (
    string path ) [inline]
```

Type Constraints

TResource : Object

Definition at line 10 of file [ResourcesLoaderService.cs](#).

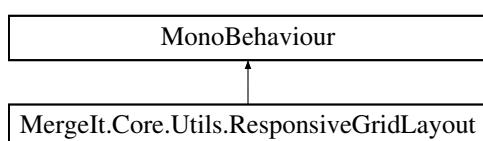
```
00010 : Object
00011 {
00012     var resource = Resources.Load<TResource>(path);
00013
00014     return resource;
00015 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/ResourcesLoaderService.cs

6.296 Mergelt.Core.Utils.ResponsiveGridLayout Class Reference

Inheritance diagram for Mergelt.Core.Utils.ResponsiveGridLayout:



6.296.1 Detailed Description

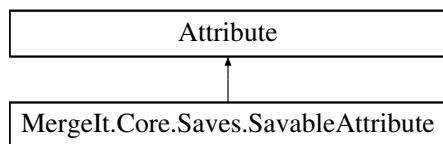
Definition at line 9 of file [ResponsiveGridLayout.cs](#).

The documentation for this class was generated from the following file:

- [Assets/MergeIt/Source/Utils/ResponsiveGridLayout.cs](#)

6.297 MergeIt.Core.Saves.SavableAttribute Class Reference

Inheritance diagram for MergeIt.Core.Saves.SavableAttribute:



Public Member Functions

- [SavableAttribute \(string name, string extension\)](#)

Properties

- string [Name](#) [get]
- string [Extension](#) [get]

6.297.1 Detailed Description

Definition at line 7 of file [SavableAttribute.cs](#).

6.297.2 Constructor & Destructor Documentation

6.297.2.1 SavableAttribute()

```
MergeIt.Core.Saves.SavableAttribute.SavableAttribute (
    string name,
    string extension ) [inline]
```

Definition at line 12 of file [SavableAttribute.cs](#).

```
00013     {
00014         Name = name;
00015         Extension = extension;
00016     }
```

6.297.3 Property Documentation

6.297.3.1 Extension

```
string MergeIt.Core.Saves.SavableAttribute.Extension [get]
```

Definition at line 10 of file [SavableAttribute.cs](#).

```
00010 { get; }
```

6.297.3.2 Name

```
string MergeIt.Core.Saves.SavableAttribute.Name [get]
```

Definition at line 9 of file [SavableAttribute.cs](#).

```
00009 { get; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Saves/SavableAttribute.cs

6.298 Mergelt.Core.FieldElements.SavedConfigParameters Class Reference

Properties

- string [ElementId](#) [get, set]
- string [EvolutionId](#) [get, set]

6.298.1 Detailed Description

Definition at line 9 of file [SavedConfigParameters.cs](#).

6.298.2 Property Documentation

6.298.2.1 ElementId

```
string MergeIt.Core.FieldElements.SavedConfigParameters.ElementId [get], [set]
```

Definition at line 12 of file [SavedConfigParameters.cs](#).

```
00012 { get; set; }
```

6.298.2.2 EvolutionId

```
string MergeIt.Core.FieldElements.SavedConfigParameters.EvolutionId [get], [set]
```

Definition at line 14 of file [SavedConfigParameters.cs](#).

```
00014 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/Parameters/Serializable/SavedConfigParameters.cs

6.299 Mergelt.Core.FieldElements.SavedGeneratorOpenParameters Class Reference

Public Member Functions

- void [CopyFrom \(IGeneratorOpenParameters other\)](#)

Properties

- long [StartOpeningTime \[get, set\]](#)

6.299.1 Detailed Description

Definition at line 9 of file [SavedGeneratorOpenParameters.cs](#).

6.299.2 Member Function Documentation

6.299.2.1 CopyFrom()

```
void MergeIt.Core.FieldElements.SavedGeneratorOpenParameters.CopyFrom (
    IGeneratorOpenParameters other) [inline]
```

Definition at line 14 of file [SavedGeneratorOpenParameters.cs](#).
00015 {
00016 StartOpeningTime = other.StartOpeningTime;
00017 }

6.299.3 Property Documentation

6.299.3.1 StartOpeningTime

```
long MergeIt.Core.FieldElements.SavedGeneratorOpenParameters.StartOpeningTime [get], [set]
```

Definition at line 12 of file [SavedGeneratorOpenParameters.cs](#).
00012 { get; set; }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/FieldElements/Parameters/Serializable/SavedGeneratorOpenParameters.cs

6.300 Mergelt.Core.FieldElements.SavedGeneratorParameters Class Reference

Public Member Functions

- void [CopyFrom \(IGeneratorParameters other\)](#)

Properties

- int [AvailableToDrop](#) [get, set]
- long [StartChargingTime](#) [get, set]
- int [DroppedElements](#) [get, set]
- int [ChargedCount](#) [get, set]

6.300.1 Detailed Description

Definition at line [9](#) of file [SavedGeneratorParameters.cs](#).

6.300.2 Member Function Documentation

6.300.2.1 CopyFrom()

```
void MergeIt.Core.FieldElements.SavedGeneratorParameters.CopyFrom (
    IGeneratorParameters other) [inline]
```

Definition at line [20](#) of file [SavedGeneratorParameters.cs](#).

```
00021     {
00022         AvailableToDrop = other.AvailableToDrop;
00023         StartChargingTime = other.StartChargingTime;
00024         DroppedElements = other.DroppedElements;
00025         ChargedCount = other.ChargedCount;
00026     }
```

6.300.3 Property Documentation

6.300.3.1 AvailableToDrop

```
int MergeIt.Core.FieldElements.SavedGeneratorParameters.AvailableToDrop [get], [set]
```

Definition at line [12](#) of file [SavedGeneratorParameters.cs](#).

```
00012 { get; set; }
```

6.300.3.2 ChargedCount

```
int MergeIt.Core.FieldElements.SavedGeneratorParameters.ChargedCount [get], [set]
```

Definition at line [18](#) of file [SavedGeneratorParameters.cs](#).

```
00018 { get; set; }
```

6.300.3.3 DroppedElements

```
int MergeIt.Core.FieldElements.SavedGeneratorParameters.DroppedElements [get], [set]
```

Definition at line [16](#) of file [SavedGeneratorParameters.cs](#).

```
00016 { get; set; }
```

6.300.3.4 StartChargingTime

```
long MergeIt.Core.FieldElements.SavedGeneratorParameters.StartChargingTime [get], [set]
```

Definition at line 14 of file [SavedGeneratorParameters.cs](#).
00014 { get; set; }

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/FieldElements/Parameters/Serializable/SavedGeneratorParameters.cs

6.301 MergeIt.Core.FieldElements.SavedInfoParameters Class Reference

Properties

- [GridPoint LogicPosition](#) [get, set]
- bool [IsBlocked](#) [get, set]

6.301.1 Detailed Description

Definition at line 9 of file [SavedInfoParameters.cs](#).

6.301.2 Property Documentation

6.301.2.1 IsBlocked

```
bool MergeIt.Core.FieldElements.SavedInfoParameters.IsBlocked [get], [set]
```

Definition at line 14 of file [SavedInfoParameters.cs](#).
00014 { get; set; }

6.301.2.2 LogicPosition

```
GridPoint MergeIt.Core.FieldElements.SavedInfoParameters.LogicPosition [get], [set]
```

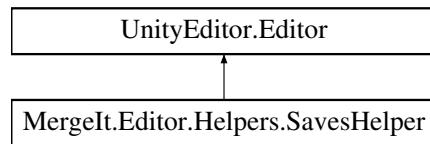
Definition at line 12 of file [SavedInfoParameters.cs](#).
00012 { get; set; }

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/FieldElements/Parameters/Serializable/SavedInfoParameters.cs

6.302 Mergelt.Editor.Helpers.SavesHelper Class Reference

Inheritance diagram for Mergelt.Editor.Helpers.SavesHelper:



Static Public Member Functions

- static void [ClearSaves \(\)](#)

6.302.1 Detailed Description

Definition at line 9 of file [SavesHelper.cs](#).

6.302.2 Member Function Documentation

6.302.2.1 ClearSaves()

```
static void MergeIt.Editor.Helpers.SavesHelper.ClearSaves ( ) [inline], [static]
```

Definition at line 12 of file [SavesHelper.cs](#).

```

00013     {
00014         string path = Path.Combine(Application.streamingAssetsPath, "Saves");
00015
00016         if (Directory.Exists(path))
00017         {
00018             Directory.Delete(path, true);
00019             string meta = Path.ChangeExtension(path, "meta");
00020
00021             if (File.Exists(meta))
00022             {
00023                 File.Delete(meta);
00024             }
00025
00026             AssetDatabase.Refresh();
00027         }
00028     }
  
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Helpers/SavesHelper.cs

6.303 Mergelt.Core.Schemes.SchemeData Class Reference

Public Attributes

- List< [ElementNodeData](#) > [ElementsNodesData](#)
- List< [EvolutionNodeData](#) > [EvolutionsNodesData](#)
- List< [EdgeData](#) > [Edges](#)
- List< [EvolutionData](#) > [EvolutionsData](#)

6.303.1 Detailed Description

Definition at line 11 of file [SchemeData.cs](#).

6.303.2 Member Data Documentation

6.303.2.1 Edges

`List<EdgeData> MergeIt.Core.Schemes.SchemeData.Edges`

Definition at line 15 of file [SchemeData.cs](#).

6.303.2.2 ElementsNodesData

`List<ElementNodeData> MergeIt.Core.Schemes.SchemeData.ElementsNodesData`

Definition at line 13 of file [SchemeData.cs](#).

6.303.2.3 EvolutionsData

`List<EvolutionData> MergeIt.Core.Schemes.SchemeData.EvolutionsData`

Definition at line 17 of file [SchemeData.cs](#).

6.303.2.4 EvolutionsNodesData

`List<EvolutionNodeData> MergeIt.Core.Schemes.SchemeData.EvolutionsNodesData`

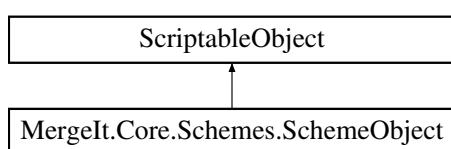
Definition at line 14 of file [SchemeData.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Schemes/SchemeData.cs

6.304 MergeIt.Core.Schemes.SchemeObject Class Reference

Inheritance diagram for MergeIt.Core.Schemes.SchemeObject:



Public Attributes

- [SchemeData SchemeData](#)

Properties

- [List< EvolutionData > Evolution \[get\]](#)

6.304.1 Detailed Description

Definition at line 9 of file [SchemeObject.cs](#).

6.304.2 Member Data Documentation

6.304.2.1 SchemeData

[SchemeData](#) MergeIt.Core.Schemes.SchemeObject.SchemeData

Definition at line 11 of file [SchemeObject.cs](#).

6.304.3 Property Documentation

6.304.3.1 Evolution

[List<EvolutionData>](#) MergeIt.Core.Schemes.SchemeObject.Evolution [get]

Definition at line 13 of file [SchemeObject.cs](#).

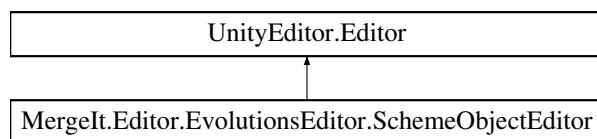
```
00014     {  
00015         get => SchemeData?.EvolutionsData;  
00016     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Schemes/SchemeObject.cs

6.305 Mergelt.Editor.EvolutionsEditor.SchemeObjectEditor Class Reference

Inheritance diagram for Mergelt.Editor.EvolutionsEditor.SchemeObjectEditor:



Public Member Functions

- override void [OnInspectorGUI \(\)](#)

6.305.1 Detailed Description

Definition at line 10 of file [SchemeObjectEditor.cs](#).

6.305.2 Member Function Documentation

6.305.2.1 OnInspectorGUI()

```
override void MergeIt.Editor.EvolutionsEditor.SchemeObjectEditor.OnInspectorGUI () [inline]
```

Definition at line 12 of file [SchemeObjectEditor.cs](#).

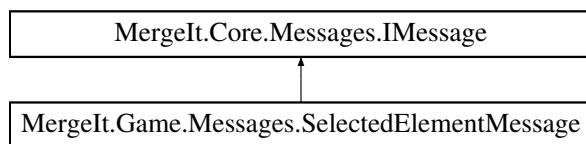
```
00013     {
00014         if (GUILayout.Button("Open"))
00015     {
00016         var window = EditorWindow.GetWindow<EvolutionsEditorWindow>(true, "Evolutions
Editor");
00017
00018         if (!window.Initialized)
00019         {
00020             window.Initialize();
00021         }
00022
00023         window.Show();
00024         window.Reload(target as SchemeObject);
00025     }
00026 }
```

The documentation for this class was generated from the following file:

- [Assets/MergeIt/Editor/Core/EvolutionsEditor/SchemeObjectEditor.cs](#)

6.306 MergeIt.Game.Messages.SelectedElementMessage Class Reference

Inheritance diagram for MergeIt.Game.Messages.SelectedElementMessage:



Properties

- [GridPoint Point \[get, set\]](#)

6.306.1 Detailed Description

Definition at line 8 of file [SelectedElementMessage.cs](#).

6.306.2 Property Documentation

6.306.2.1 Point

`GridPoint` MergeIt.Game.Messages.SelectedElementMessage.Point [get], [set]

Definition at line 10 of file [SelectedElementMessage.cs](#).
00010 { get; set; }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/SelectedElementMessage.cs

6.307 Mergelt.Editor.EvolutionsEditor.SerializedElementNodes Class Reference

Public Attributes

- `List< EvolutionNodeData > EvolutionNodesData`

6.307.1 Detailed Description

Definition at line 10 of file [SerializedElementNodes.cs](#).

6.307.2 Member Data Documentation

6.307.2.1 EvolutionNodesData

`List<EvolutionNodeData>` MergeIt.Editor.EvolutionsEditor.SerializedElementNodes.Evolution←→
NodesData

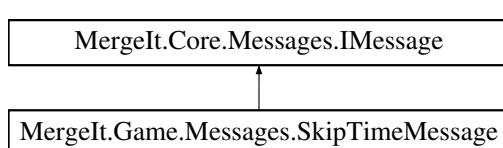
Definition at line 12 of file [SerializedElementNodes.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/EvolutionsEditor/ItemsNodes/CopyPaste/SerializedElementNodes.cs

6.308 Mergelt.Game.Messages.SkipTimeMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.SkipTimeMessage:



Properties

- int **Seconds** [get, set]

6.308.1 Detailed Description

Definition at line 7 of file [SkipTimeMessage.cs](#).

6.308.2 Property Documentation

6.308.2.1 Seconds

```
int MergeIt.Game.Messages.SkipTimeMessage.Seconds [get], [set]
```

Definition at line 9 of file [SkipTimeMessage.cs](#).

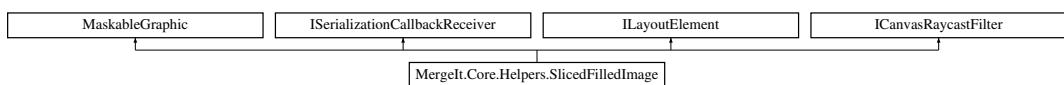
```
00009 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Messages/SkipTimeMessage.cs

6.309 MergeIt.Core.Helpers.SlicedFilledImage Class Reference

Inheritance diagram for MergeIt.Core.Helpers.SlicedFilledImage:



Public Types

- enum **FillDirection** { **Right** = 0 , **Left** = 1 , **Up** = 2 , **Down** = 3 }

Protected Member Functions

- override void **OnEnable** ()
- override void **OnDisable** ()
- override void **OnPopulateMesh** (VertexHelper vh)
- override void **UpdateMaterial** ()

Update the renderer's material.

Properties

- Sprite `sprite` [get, set]
- FillDirection `fillDirection` [get, set]
- float `fillAmount` [get, set]
- bool `fillCenter` [get, set]
- float `pixelsPerUnitMultiplier` [get, set]
- float `pixelsPerUnit` [get]
- Sprite `overrideSprite` [get, set]
- override Texture `mainTexture` [get]
- bool `hasBorder` [get]
- override Material `material` [get, set]
- float `alphaHitTestMinimumThreshold` [get, set]

6.309.1 Detailed Description

Definition at line 17 of file [SlicedFilledImage.cs](#).

6.309.2 Member Enumeration Documentation

6.309.2.1 FillDirection

```
enum MergeIt.Core.Helpers.SlicedFilledImage.FillDirection
```

Definition at line 40 of file [SlicedFilledImage.cs](#).

```
00040 { Right = 0, Left = 1, Up = 2, Down = 3 }
```

6.309.3 Constructor & Destructor Documentation

6.309.3.1 SlicedFilledImage()

```
MergeIt.Core.Helpers.SlicedFilledImage.SlicedFilledImage () [inline], [protected]
```

Definition at line 193 of file [SlicedFilledImage.cs](#).

```
00194     {  
00195         useLegacyMeshGeneration = false;  
00196     }
```

6.309.4 Member Function Documentation

6.309.4.1 OnDisable()

```
override void MergeIt.Core.Helpers.SlicedFilledImage.OnDisable () [inline], [protected]
```

Definition at line 204 of file [SlicedFilledImage.cs](#).

```
00205     {  
00206         base.OnDisable();  
00207  
00208         if( m_Tracked )  
00209             UnTrackImage();  
00210     }
```

6.309.4.2 OnEnable()

```
override void MergeIt.Core.Helpers.SlicedFilledImage.OnEnable ( ) [inline], [protected]
```

Definition at line 198 of file [SlicedFilledImage.cs](#).

```
00199     {
00200         base.OnEnable();
00201         TrackImage();
00202     }
```

6.309.4.3 OnPopulateMesh()

```
override void MergeIt.Core.Helpers.SlicedFilledImage.OnPopulateMesh (
    VertexHelper vh) [inline], [protected]
```

Definition at line 220 of file [SlicedFilledImage.cs](#).

```
00221     {
00222         if( activeSprite == null )
00223         {
00224             base.OnPopulateMesh( vh );
00225             return;
00226         }
00227         GenerateSlicedFilledSprite( vh );
00228     }
```

6.309.4.4 UpdateMaterial()

```
override void MergeIt.Core.Helpers.SlicedFilledImage.UpdateMaterial ( ) [inline], [protected]
```

Update the renderer's material.

Definition at line 234 of file [SlicedFilledImage.cs](#).

```
00235     {
00236         base.UpdateMaterial();
00237
00238         // Check if this sprite has an associated alpha texture (generated when splitting RGBA =
00239         // RGB + A as two textures without alpha)
00240         if( activeSprite == null )
00241         {
00242             canvasRenderer.SetAlphaTexture( null );
00243             return;
00244         }
00245         Texture2D alphaTex = activeSprite.associatedAlphaSplitTexture;
00246         if( alphaTex != null )
00247             canvasRenderer.SetAlphaTexture( alphaTex );
00248     }
```

6.309.5 Property Documentation

6.309.5.1 alphaHitTestMinimumThreshold

```
float MergeIt.Core.Helpers.SlicedFilledImage.alphaHitTestMinimumThreshold [get], [set]
```

Definition at line 189 of file [SlicedFilledImage.cs](#).

```
00189 { get; set; }
```

6.309.5.2 fillAmount

```
float MergeIt.Core.Helpers.SlicedFilledImage.fillAmount [get], [set]
```

Definition at line 80 of file [SlicedFilledImage.cs](#).

```
00081     {
00082         get { return m_FillAmount; }
00083         set
00084     {
00085         if( SetPropertyUtility.SetStruct( ref m_FillAmount, Mathf.Clamp01( value ) ) )
00086             SetVerticesDirty();
00087     }
00088 }
```

6.309.5.3 fillCenter

```
bool MergeIt.Core.Helpers.SlicedFilledImage.fillCenter [get], [set]
```

Definition at line 92 of file [SlicedFilledImage.cs](#).

```
00093     {
00094         get { return m_FillCenter; }
00095         set
00096     {
00097         if( SetPropertyUtility.SetStruct( ref m_FillCenter, value ) )
00098             SetVerticesDirty();
00099     }
00100 }
```

6.309.5.4 fillDirection

```
FillDirection MergeIt.Core.Helpers.SlicedFilledImage.fillDirection [get], [set]
```

Definition at line 67 of file [SlicedFilledImage.cs](#).

```
00068     {
00069         get { return m_FillDirection; }
00070         set
00071     {
00072         if( SetPropertyUtility.SetStruct( ref m_FillDirection, value ) )
00073             SetVerticesDirty();
00074     }
00075 }
```

6.309.5.5 hasBorder

```
bool MergeIt.Core.Helpers.SlicedFilledImage.hasBorder [get]
```

Definition at line 155 of file [SlicedFilledImage.cs](#).

```
00156     {
00157         get
00158     {
00159         if( activeSprite != null )
00160         {
00161             Vector4 v = activeSprite.border;
00162             return v.sqrMagnitude > 0f;
00163         }
00164
00165         return false;
00166     }
00167 }
```

6.309.5.6 mainTexture

```
override Texture MergeIt.Core.Helpers.SlicedFilledImage.mainTexture [get]
```

Definition at line 144 of file [SlicedFilledImage.cs](#).

```
00145         {
00146             get
00147             {
00148                 if( activeSprite != null )
00149                     return activeSprite.texture;
00150
00151                 return material != null && material.mainTexture != null ? material.mainTexture :
00152                     s_WhiteTexture;
00153             }
00153         }
```

6.309.5.7 material

```
override Material MergeIt.Core.Helpers.SlicedFilledImage.material [get], [set]
```

Definition at line 169 of file [SlicedFilledImage.cs](#).

```
00170         {
00171             get
00172             {
00173                 if( m_Material != null )
00174                     return m_Material;
00175
00176                 if( activeSprite && activeSprite.associatedAlphaSplitTexture != null )
00177                 {
00178 #if UNITY_EDITOR
00179                     if( Application.isPlaying )
00180 #endif
00181                     return Image.defaultETC1GraphicMaterial;
00182                 }
00183
00184                 return defaultMaterial;
00185             }
00186             set { base.material = value; }
00187         }
```

6.309.5.8 overrideSprite

```
Sprite MergeIt.Core.Helpers.SlicedFilledImage.overrideSprite [get], [set]
```

Definition at line 129 of file [SlicedFilledImage.cs](#).

```
00130         {
00131             get { return activeSprite; }
00132             set
00133             {
00134                 if( SetPropertyUtility.SetClass( ref m_OverrideSprite, value ) )
00135                 {
00136                     SetAllDirty();
00137                     TrackImage();
00138                 }
00139             }
00140         }
```

6.309.5.9 pixelsPerUnit

```
float MergeIt.Core.Helpers.SlicedFilledImage.pixelsPerUnit [get]
```

Definition at line 110 of file [SlicedFilledImage.cs](#).

```
00111         {
00112             get
00113             {
00114                 float spritePixelsPerUnit = 100;
00115                 if( activeSprite )
00116                     spritePixelsPerUnit = activeSprite.pixelsPerUnit;
00117
00118                 float referencePixelsPerUnit = 100;
00119                 if( canvas )
00120                     referencePixelsPerUnit = canvas.referencePixelsPerUnit;
00121
00122                 return m_PixelsPerUnitMultiplier * spritePixelsPerUnit / referencePixelsPerUnit;
00123             }
00124         }
```

6.309.5.10 pixelsPerUnitMultiplier

```
float MergeIt.Core.Helpers.SlicedFilledImage.pixelsPerUnitMultiplier [get], [set]
```

Definition at line 104 of file [SlicedFilledImage.cs](#).

```
00105     {
00106         get { return m_PixelsPerUnitMultiplier; }
00107         set { m_PixelsPerUnitMultiplier = Mathf.Max( 0.01f, value ); }
00108     }
```

6.309.5.11 sprite

```
Sprite MergeIt.Core.Helpers.SlicedFilledImage.sprite [get], [set]
```

Definition at line 52 of file [SlicedFilledImage.cs](#).

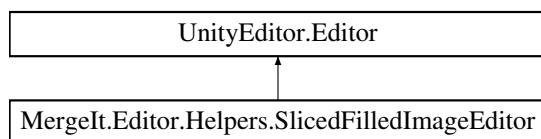
```
00053     {
00054         get { return m_Sprite; }
00055         set
00056     {
00057         if( SetPropertyUtility.SetClass( ref m_Sprite, value ) )
00058         {
00059             SetAllDirty();
00060             TrackImage();
00061         }
00062     }
00063 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Utils/SlicedFilledImage.cs

6.310 Mergelt.Editor.Helpers.SlicedFilledImageEditor Class Reference

Inheritance diagram for Mergelt.Editor.Helpers.SlicedFilledImageEditor:



Public Member Functions

- override void [OnInspectorGUI \(\)](#)

6.310.1 Detailed Description

Definition at line 10 of file [SlicedFilledImageEditor.cs](#).

6.310.2 Member Function Documentation

6.310.2.1 OnInspectorGUI()

```
override void MergeIt.Editor.Helpers.SlicedFilledImageEditor.OnInspectorGUI () [inline]
```

Definition at line 22 of file [SlicedFilledImageEditor.cs](#).

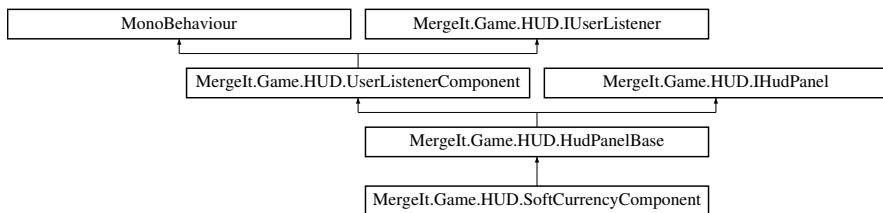
```
00023     {
00024         serializedObject.Update ();
00025
00026         EditorGUILayout.PropertyField( _spriteProp, _spriteLabel );
00027         EditorGUILayout.PropertyField( _colorProp );
00028         DrawPropertiesExcluding( serializedObject, "m_Script", "m_Sprite", "m_Color",
00029             "m_OnCullStateChanged" );
00030         serializedObject.ApplyModifiedProperties ();
00031     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Editor/Helpers/SlicedFilledImageEditor.cs

6.311 MergeIt.Game.HUD.SoftCurrencyComponent Class Reference

Inheritance diagram for MergeIt.Game.HUD.SoftCurrencyComponent:



Protected Member Functions

- override void [OnApplyModel](#) ([UserServiceModel](#) userServiceModel)

Protected Member Functions inherited from [MergeIt.Game.HUD.HudPanelBase](#)

- virtual void [Start](#) ()

Protected Member Functions inherited from [MergeIt.Game.HUD.UserListenerComponent](#)

- void [OnApplyModel](#) ([UserServiceModel](#) userServiceModel)

Properties

- override HudPanelType [Type](#) [get]

Properties inherited from [Mergelt.Game.HUD.HudPanelBase](#)

- virtual HudPanelType [Type](#) [get]

Properties inherited from [Mergelt.Game.HUD.UserListenerComponent](#)

- [UserServiceModel](#) [UserServiceModel](#) [get]

Properties inherited from [Mergelt.Game.HUD.IHudPanel](#)

Additional Inherited Members

Public Member Functions inherited from [Mergelt.Game.HUD.UserListenerComponent](#)

- void [ApplyModel](#) ([UserServiceModel](#) userServiceModel)

6.311.1 Detailed Description

Definition at line 10 of file [SoftCurrencyComponent.cs](#).

6.311.2 Member Function Documentation

6.311.2.1 OnApplyModel()

```
override void MergeIt.Game.HUD.SoftCurrencyComponent.OnApplyModel (
    UserServiceModel userServiceModel ) [inline], [protected]
```

Definition at line 20 of file [SoftCurrencyComponent.cs](#).

```
00021     {
00022         UserServiceModel userServiceModel .SoftCurrency.Subscribe(OnSoftCurrencyChanged, true);
00023     }
```

6.311.3 Property Documentation

6.311.3.1 Type

```
override HudPanelType MergeIt.Game.HUD.SoftCurrencyComponent.Type [get]
```

Implements [Mergelt.Game.HUD.IHudPanel](#).

Definition at line 15 of file [SoftCurrencyComponent.cs](#).

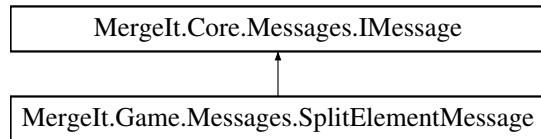
```
00016     {
00017         get => HudPanelType.SoftCurrency;
00018     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/HUD/SoftCurrencyComponent.cs

6.312 Mergelt.Game.Messages.SplitElementMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.SplitElementMessage:



Properties

- `IFieldElement SplitElement1 [get, set]`
- `IFieldElement SplitElement2 [get, set]`

6.312.1 Detailed Description

Definition at line 8 of file [SplitElementMessage.cs](#).

6.312.2 Property Documentation

6.312.2.1 SplitElement1

`IFieldElement MergeIt.Game.Messages.SplitElementMessage.SplitElement1 [get], [set]`

Definition at line 10 of file [SplitElementMessage.cs](#).
00010 { `get; set;` }

6.312.2.2 SplitElement2

`IFieldElement MergeIt.Game.Messages.SplitElementMessage.SplitElement2 [get], [set]`

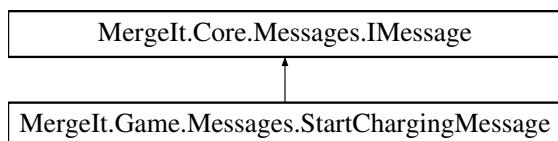
Definition at line 11 of file [SplitElementMessage.cs](#).
00011 { `get; set;` }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/SplitElementMessage.cs

6.313 Mergelt.Game.Messages.StartChargingMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.StartChargingMessage:



Properties

- long [RemainingTime](#) [get, set]

6.313.1 Detailed Description

Definition at line 7 of file [StartChargingMessage.cs](#).

6.313.2 Property Documentation

6.313.2.1 RemainingTime

long MergeIt.Game.Messages.StartChargingMessage.RemainingTime [get], [set]

Definition at line 9 of file [StartChargingMessage.cs](#).

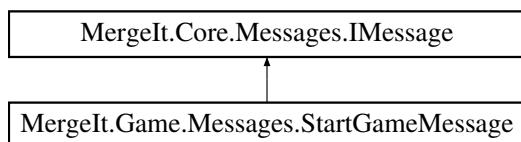
```
00009 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/StartChargingMessage.cs

6.314 Mergelt.Game.Messages.StartGameMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.StartGameMessage:



6.314.1 Detailed Description

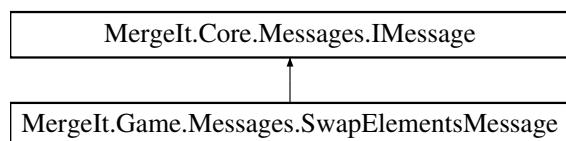
Definition at line 7 of file [StartGameMessage.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/StartGameMessage.cs

6.315 Mergelt.Game.Messages.SwapElementsMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.SwapElementsMessage:



Properties

- `GridPoint From [get, set]`
- `GridPoint To [get, set]`

6.315.1 Detailed Description

Definition at line 8 of file [SwapElementsMessage.cs](#).

6.315.2 Property Documentation

6.315.2.1 From

`GridPoint MergeIt.Game.Messages.SwapElementsMessage.From [get], [set]`

Definition at line 10 of file [SwapElementsMessage.cs](#).

```
00010 { get; set; }
```

6.315.2.2 To

`GridPoint MergeIt.Game.Messages.SwapElementsMessage.To [get], [set]`

Definition at line 11 of file [SwapElementsMessage.cs](#).

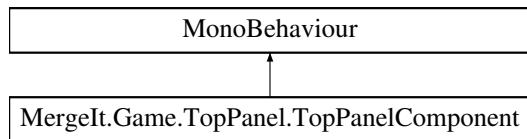
```
00011 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Messages/SwapElementsMessage.cs

6.316 MergeIt.Game.TopPanel.TopPanelComponent Class Reference

Inheritance diagram for MergeIt.Game.TopPanel.TopPanelComponent:



6.316.1 Detailed Description

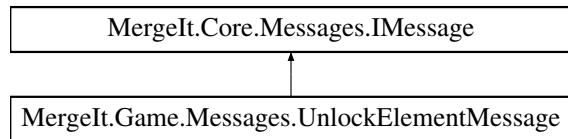
Definition at line 11 of file [TopPanelComponent.cs](#).

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/TopPanel/TopPanelComponent.cs

6.317 Mergelt.Game.Messages.UnlockElementMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.UnlockElementMessage:



Properties

- `IFieldElement Element [get, set]`

6.317.1 Detailed Description

Definition at line 8 of file [UnlockElementMessage.cs](#).

6.317.2 Property Documentation

6.317.2.1 Element

`IFieldElement MergeIt.Game.Messages.UnlockElementMessage.Element [get], [set]`

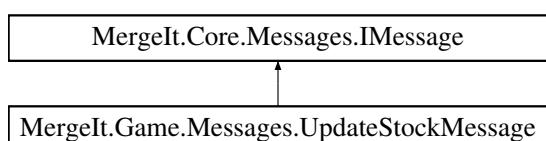
Definition at line 10 of file [UnlockElementMessage.cs](#).
00010 { `get; set;` }

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/UnlockElementMessage.cs

6.318 Mergelt.Game.Messages.UpdateStockMessage Class Reference

Inheritance diagram for Mergelt.Game.Messages.UpdateStockMessage:



6.318.1 Detailed Description

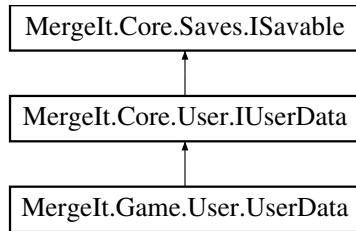
Definition at line 7 of file [UpdateStockMessage.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Messages/UpdateStockMessage.cs

6.319 Mergelt.Game.User(userData Class Reference)

Inheritance diagram for Mergelt.Game.User.userData:



Properties

- string [Name](#) [get, set]
- int [Energy](#) [get, set]
- int [SoftCurrency](#) [get, set]
- int [HardCurrency](#) [get, set]
- int [Splitters](#) [get, set]
- int [Level](#) [get, set]
- int [Experience](#) [get, set]
- long [EnergyRestoringStartTime](#) [get, set]

Properties inherited from [Mergelt.Core.User.IUserData](#)

6.319.1 Detailed Description

Definition at line 11 of file [UserData.cs](#).

6.319.2 Property Documentation

6.319.2.1 Energy

`int MergeIt.Game.User(userData).Energy [get], [set]`

Implements [Mergelt.Core.User.IUserData](#).

Definition at line 16 of file [UserData.cs](#).

`00016 { get; set; }`

6.319.2.2 EnergyRestoringStartTime

`long MergeIt.Game.User(userData).EnergyRestoringStartTime [get], [set]`

Implements [Mergelt.Core.User.IUserData](#).

Definition at line 28 of file [UserData.cs](#).

`00028 { get; set; }`

6.319.2.3 Experience

```
int MergeIt.Game.User.userData.Experience [get], [set]
```

Implements [Mergelt.Core.User.IUserData](#).

Definition at line 26 of file [UserData.cs](#).

```
00026 { get; set; }
```

6.319.2.4 HardCurrency

```
int MergeIt.Game.User.userData.HardCurrency [get], [set]
```

Implements [Mergelt.Core.User.IUserData](#).

Definition at line 20 of file [UserData.cs](#).

```
00020 { get; set; }
```

6.319.2.5 Level

```
int MergeIt.Game.User.userData.Level [get], [set]
```

Implements [Mergelt.Core.User.IUserData](#).

Definition at line 24 of file [UserData.cs](#).

```
00024 { get; set; }
```

6.319.2.6 Name

```
string MergeIt.Game.User.userData.Name [get], [set]
```

Implements [Mergelt.Core.User.IUserData](#).

Definition at line 14 of file [UserData.cs](#).

```
00014 { get; set; }
```

6.319.2.7 SoftCurrency

```
int MergeIt.Game.User.userData.SoftCurrency [get], [set]
```

Implements [Mergelt.Core.User.IUserData](#).

Definition at line 18 of file [UserData.cs](#).

```
00018 { get; set; }
```

6.319.2.8 Splitters

```
int MergeIt.Game.User(userDataSplitters [get], [set]
```

Implements [Mergelt.Core.User.IUserData](#).

Definition at line 22 of file [UserData.cs](#).

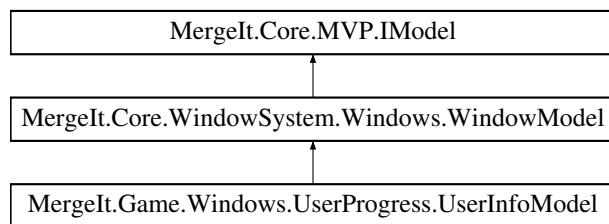
```
00022 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/User/UserData.cs

6.320 Mergelt.Game.Windows.UserProgress.UserInfoModel Class Reference

Inheritance diagram for Mergelt.Game.Windows.UserProgress.UserInfoModel:



6.320.1 Detailed Description

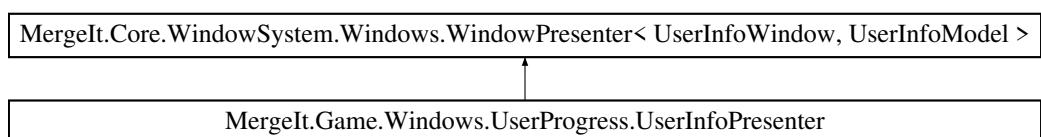
Definition at line 7 of file [UserInfoModel.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Windows/UserProgress/UserInfoModel.cs

6.321 Mergelt.Game.Windows.UserProgress.UserInfoPresenter Class Reference

Inheritance diagram for Mergelt.Game.Windows.UserProgress.UserInfoPresenter:



Protected Member Functions

- override void [OnInitialize](#) ([UserInfoWindow](#) view)
- override void [OnDestroyWindow](#) ()

Protected Member Functions inherited from

[Mergelt.Core.WindowSystem.Windows.WindowPresenter< UserInfoWindow, UserInfoModel >](#)

- virtual void [OnInitialize](#) ([IWindowArgs](#) args=null)
- override void [OnDispose](#) ()
- virtual void [OnWindowStartShowing](#) ()
- virtual void [OnWindowShown](#) ()
- virtual void [OnWindowStartClosing](#) ()
- virtual void [OnWindowClosed](#) ()

Additional Inherited Members

Public Member Functions inherited from

[Mergelt.Core.WindowSystem.Windows.WindowPresenter< UserInfoWindow, UserInfoModel >](#)

- void [Initialize](#) ([IView](#) view, string layer, [IWindowArgs](#) windowArgs=null)
- void [Show](#) ()
- void [Hide](#) ()
- void [Close](#) ()
- void [SetWindowActive](#) (bool active)
- void [SetWindowLayer](#) ()
- void [DestroyWindow](#) ()

Protected Attributes inherited from

[Mergelt.Core.WindowSystem.Windows.WindowPresenter< UserInfoWindow, UserInfoModel >](#)

- [IMessageBus](#) [MessageBus](#)
- [IWindowSystem](#) [WindowSystem](#)

Properties inherited from

[Mergelt.Core.WindowSystem.Windows.WindowPresenter< UserInfoWindow, UserInfoModel >](#)

- string [Layer](#) [get]
- [WindowState](#) [State](#) [get]

6.321.1 Detailed Description

Definition at line 21 of file [UserInfoPresenter.cs](#).

6.321.2 Member Function Documentation

6.321.2.1 OnDestroyWindow()

```
override void MergeIt.Game.Windows.UserProgress.UserInfoPresenter.OnDestroyWindow () [inline],  
[protected], [virtual]
```

Reimplemented from [Mergelt.Core.WindowSystem.Windows.WindowPresenter< UserInfoWindow, UserInfoModel >](#).

Definition at line 52 of file [UserInfoPresenter.cs](#).

```
00053     {  
00054         base.OnDestroyWindow();  
00055  
00056         _messageBus.RemoveListener<LevelUpdatedMessage>(LevelUpdatedMessageHandler);  
00057  
00058         View.LevelUpEvent -= LevelUpEventHandler;  
00059  
00060         base.OnDestroyWindow();  
00061  
00062         foreach (var elementConfig in _elementConfigs)  
00063         {  
00064             if (elementConfig.Key)  
00065             {  
00066                 elementConfig.Key.ClickEvent -= OnClickElement;  
00067             }  
00068         }  
00069  
00070         _elementConfigs.Clear();  
00071     }
```

6.321.2.2 OnInitialize()

```
override void MergeIt.Game.Windows.UserProgress.UserInfoPresenter.OnInitialize (  
    UserInfoWindow view) [inline], [protected]
```

Definition at line 32 of file [UserInfoPresenter.cs](#).

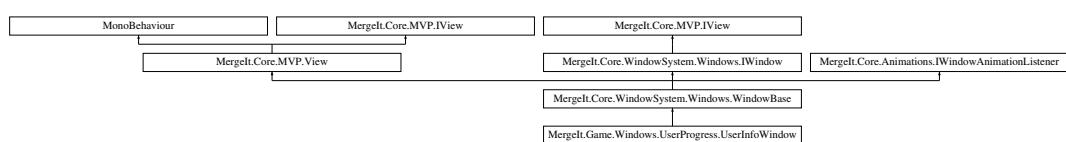
```
00033     {  
00034         base.OnInitialize(view);  
00035  
00036         _userServiceModel = DiContainer.Get<UserServiceModel>();  
00037         _userProgressService = DiContainer.Get<IUserProgressService>();  
00038         _elementsStockService = DiContainer.Get<IElementsStockService>();  
00039         _elementInfoFactory = DiContainer.Get<IElementInfoFactory>();  
00040         _effectsFactory = DiContainer.Get<IEffectsFactory>();  
00041  
00042         _messageBus = DiContainer.Get<IMessageBus>();  
00043         _messageBus.AddListener<LevelUpdatedMessage>(LevelUpdatedMessageHandler);  
00044  
00045         View.LevelUpEvent += LevelUpEventHandler;  
00046  
00047         CheckLevelUpButtonState();  
00048         FillElements();  
00049         UpdateProgress();  
00050     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Windows/UserProgress/UserInfoPresenter.cs

6.322 Mergelt.Game.Windows.UserProgress.UserInfoWindow Class Reference

Inheritance diagram for Mergelt.Game.Windows.UserProgress.UserInfoWindow:



Public Member Functions

- override void [OnOpenFinished](#) ()
- override void [Close](#) ()
- void [EnableLevelUpButton](#) (bool enable)
- void [UpdateProgress](#) (int level, int currentExp, int nextExp)

Public Member Functions inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- virtual void [Show](#) ()
- virtual void [SetLayer](#) (string layer)
- virtual void [OnOpenStarted](#) ()
- virtual void [OnCloseStarted](#) ()
- virtual void [OnCloseFinished](#) ()

Public Member Functions inherited from [Mergelt.Core.MVP.View](#)

- virtual void [Initialize](#) ()

Public Attributes

- UnityAction [LevelUpEvent](#)

Properties

- RectTransform [PrizesContainer](#) [get]

Properties inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- Canvas [Canvas](#) [get]
- IWindowAnimationController [AnimationController](#) [get]

Properties inherited from [Mergelt.Core.MVP.View](#)

- GameObject [GameObject](#) [get]

Properties inherited from [Mergelt.Core.MVP.IView](#)

Additional Inherited Members

Protected Member Functions inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- override void [OnDestroy](#) ()

Protected Attributes inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- Button [CloseButton](#)
- Button [ClickOutsideArea](#)

Events inherited from [Mergelt.Core.WindowSystem.Windows.WindowBase](#)

- Action [InitiateCloseEvent](#)
- Action [ShowStartEvent](#)
- Action [ShowEndEvent](#)
- Action [CloseStartEvent](#)
- Action [CloseEndEvent](#)

Events inherited from [Mergelt.Core.MVP.View](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

Events inherited from [Mergelt.Core.MVP.IView](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

Events inherited from [Mergelt.Core.WindowSystem.Windows.IWindow](#)

- Action [InitiateCloseEvent](#)
- Action [ShowStartEvent](#)
- Action [ShowEndEvent](#)
- Action [CloseStartEvent](#)
- Action [CloseEndEvent](#)

6.322.1 Detailed Description

Definition at line 10 of file [UserInfoWindow.cs](#).

6.322.2 Member Function Documentation

6.322.2.1 Close()

```
override void MergeIt.Game.Windows.UserProgress.UserInfoWindow.Close ( ) [inline], [virtual]
```

Reimplemented from [Mergelt.Core.WindowSystem.Windows.WindowBase](#).

Definition at line 36 of file [UserInfoWindow.cs](#).

```
00037      {
00038          base.Close();
00039          _levelUpButton.onClick.removeListener(LevelUpEvent);
00040      }
```

6.322.2.2 EnableLevelUpButton()

```
void MergeIt.Game.Windows.UserProgress.UserInfoWindow.EnableLevelUpButton (
    bool enable ) [inline]
```

Definition at line 43 of file [UserInfoWindow.cs](#).

```
00044     {
00045         _levelUpButton.interactable = enable;
00046     }
```

6.322.2.3 OnOpenFinished()

```
override void MergeIt.Game.Windows.UserProgress.UserInfoWindow.OnOpenFinished ( ) [inline],
[virtual]
```

Reimplemented from [Mergelt.Core.WindowSystem.Windows.WindowBase](#).

Definition at line 29 of file [UserInfoWindow.cs](#).

```
00030     {
00031         base.OnOpenFinished();
00032
00033         _levelUpButton.onClick.AddListener(LevelUpEvent);
00034     }
```

6.322.2.4 UpdateProgress()

```
void MergeIt.Game.Windows.UserProgress.UserInfoWindow.UpdateProgress (
    int level,
    int currentExp,
    int nextExp ) [inline]
```

Definition at line 48 of file [UserInfoWindow.cs](#).

```
00049     {
00050         _userProgressComponent.UpdateProgress(level, currentExp, nextExp);
00051     }
```

6.322.3 Member Data Documentation

6.322.3.1 LevelUpEvent

UnityAction [MergeIt.Game.Windows.UserProgress.UserInfoWindow.LevelUpEvent](#)

Definition at line 22 of file [UserInfoWindow.cs](#).

6.322.4 Property Documentation

6.322.4.1 PrizesContainer

```
RectTransform MergeIt.Game.Windows.UserProgress.UserInfoWindow.PrizesContainer [get]
```

Definition at line 24 of file [UserInfoWindow.cs](#).

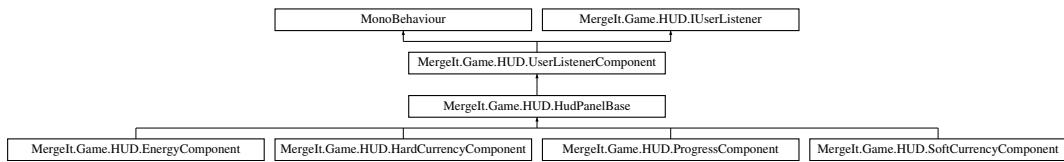
```
00025     {
00026         get => _prizesContainer;
00027     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Windows/UserProgress/UserInfoWindow.cs

6.323 Mergelt.Game.HUD.UserListenerComponent Class Reference

Inheritance diagram for Mergelt.Game.HUD.UserListenerComponent:



Public Member Functions

- void [ApplyModel](#) ([UserServiceModel](#) userServiceModel)

Protected Member Functions

- void [OnApplyModel](#) ([UserServiceModel](#) userServiceModel)

Properties

- [UserServiceModel](#) [UserServiceModel](#) [get]

6.323.1 Detailed Description

Definition at line 8 of file [UserListenerComponent.cs](#).

6.323.2 Member Function Documentation

6.323.2.1 ApplyModel()

```
void MergeIt.Game.HUD.UserListenerComponent.ApplyModel (
    UserServiceModel userServiceModel ) [inline]
```

Implements [Mergelt.Game.HUD.IUserListener](#).

Definition at line 12 of file [UserListenerComponent.cs](#).

```
00013     {
00014         UserServiceModel = userServiceModel;
00015         OnApplyModel(userServiceModel);
00017     }
```

6.323.3 Property Documentation

6.323.3.1 UserServiceModel

```
UserServiceModel MergeIt.Game.HUD.UserListenerComponent.UserServiceModel [get], [protected]
```

Definition at line 10 of file [UserListenerComponent.cs](#).

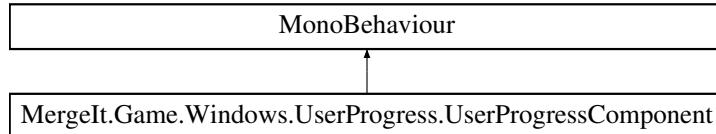
```
00010 { get; private set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/HUD/UserListenerComponent.cs

6.324 Mergelt.Game.Windows.UserProgress.UserProgressComponent Class Reference

Inheritance diagram for Mergelt.Game.Windows.UserProgress.UserProgressComponent:



Public Member Functions

- void [UpdateProgress](#) (int level, int currentExp, int nextExp)

6.324.1 Detailed Description

Definition at line [9](#) of file [UserProgressComponent.cs](#).

6.324.2 Member Function Documentation

6.324.2.1 UpdateProgress()

```
void MergeIt.Game.Windows.UserProgress.UserProgressComponent.UpdateProgress (
    int level,
    int currentExp,
    int nextExp ) [inline]
```

Definition at line [20](#) of file [UserProgressComponent.cs](#).

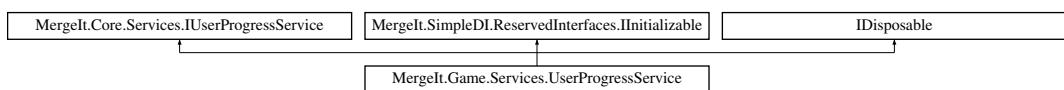
```
00021     {
00022         _levelValue.text = $"{level}";
00023         _progressText.text = $"{currentExp}/{nextExp}";
00024         _progressImage.fillAmount = (float)currentExp / nextExp;
00025     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Windows/UserProgress/UserProgressComponent.cs

6.325 Mergelt.Game.Services.UserProgressService Class Reference

Inheritance diagram for Mergelt.Game.Services.UserProgressService:



Public Member Functions

- void [Dispose \(\)](#)
- void [Initialize \(\)](#)
- int [GetCurrentLevelMaxExp \(\)](#)
- bool [CanLevelUp \(\)](#)
- [ElementConfig\[\] GetLevelUpPrizes \(\)](#)

6.325.1 Detailed Description

Definition at line 14 of file [UserProgressService.cs](#).

6.325.2 Member Function Documentation

6.325.2.1 CanLevelUp()

```
bool MergeIt.Game.Services.UserProgressService.CanLevelUp ( ) [inline]
```

Implements [Mergelt.Core.Services.IUserProgressService](#).

Definition at line 42 of file [UserProgressService.cs](#).

```
00043     {  
00044         return _userServiceModel.Experience.Value >= GetCurrentLevelMaxExp();  
00045     }
```

6.325.2.2 Dispose()

```
void MergeIt.Game.Services.UserProgressService.Dispose ( ) [inline]
```

Definition at line 25 of file [UserProgressService.cs](#).

```
00026     {  
00027         _messageBus.RemoveListener<LevelUpMessage>(OnLevelUpMessageHandler);  
00028     }
```

6.325.2.3 GetCurrentLevelMaxExp()

```
int MergeIt.Game.Services.UserProgressService.GetCurrentLevelMaxExp ( ) [inline]
```

Implements [Mergelt.Core.Services.IUserProgressService](#).

Definition at line 35 of file [UserProgressService.cs](#).

```
00036     {  
00037         LevelUpParameters levelUpParameters =  
00038             _configService.GetLevelUpData(_userServiceModel.Level.Value);  
00039         return levelUpParameters.Experience;  
00040     }
```

6.325.2.4 GetLevelUpPrizes()

```
ElementConfig[] MergeIt.Game.Services.UserProgressService.GetLevelUpPrizes () [inline]
```

Implements [Mergelt.Core.Services.IUserProgressService](#).

Definition at line 47 of file [UserProgressService.cs](#).

```
00048     {
00049         LevelUpParameters levelUpParameters =
00050             _configService.GetLevelUpData(_userServiceModel.Level.Value);
00051         if (levelUpParameters != null)
00052         {
00053             return levelUpParameters.Bonuses;
00054         }
00055     }
00056     return null;
00057 }
```

6.325.2.5 Initialize()

```
void MergeIt.Game.Services.UserProgressService.Initialize () [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 30 of file [UserProgressService.cs](#).

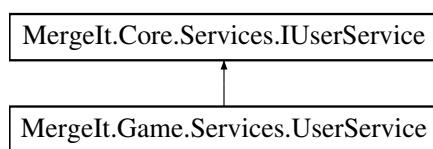
```
00031     {
00032         _messageBus.AddListener<LevelUpMessage>(OnLevelUpMessageHandler);
00033     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/UserProgressService.cs

6.326 Mergelt.Game.Services.UserService Class Reference

Inheritance diagram for Mergelt.Game.Services.UserService:



Public Member Functions

- void [CreateUser](#) ()
- void [SetupUser](#) ([IUserData](#) userData)

6.326.1 Detailed Description

Definition at line 14 of file [UserService.cs](#).

6.326.2 Member Function Documentation

6.326.2.1 CreateUser()

```
void MergeIt.Game.Services.UserService.CreateUser ( ) [inline]
```

Implements [Mergelt.Core.Services.IUserService](#).

Definition at line 28 of file [UserService.cs](#).

```
00029     {
00030         GameConfig config = _configsService.GameConfig;
00031         var userData = new UserData
00032         {
00033             Name = $"User{new Random().Next(ushort.MinValue, ushort.MaxValue)}",
00034             Energy = config.EnergyCap,
00035             Experience = 0,
00036             SoftCurrency = config.InitialSoftCurrency,
00037             HardCurrency = config.InitialHardCurrency,
00038             Splitters = config.InitialSplittersCount,
00039             Level = 1
00040         };
00041         SetupUser(userData);
00042         _saveService.Save (GameSaveType.User);
00043     }
00044 }
```

6.326.2.2 SetupUser()

```
void MergeIt.Game.Services.UserService.SetupUser (
    IUserData userData ) [inline]
```

Implements [Mergelt.Core.Services.IUserService](#).

Definition at line 47 of file [UserService.cs](#).

```
00048     {
00049         _userServiceModel.Set (userData);
00050     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Game/Services/UserService.cs

6.327 Mergelt.Game.Services.UserServiceModel Class Reference

Public Member Functions

- void [Set \(IUserData data\)](#)
- [UserData GetData \(\)](#)

Properties

- Bindable< string > [Name](#) = new() [get]
- Bindable< int > [Energy](#) = new() [get]
- Bindable< int > [SoftCurrency](#) = new() [get]
- Bindable< int > [HardCurrency](#) = new() [get]
- Bindable< int > [Splitters](#) = new() [get]
- Bindable< int > [Level](#) = new() [get]
- Bindable< int > [Experience](#) = new() [get]
- long [EnergyRestoringStartTime](#) [get, set]

6.327.1 Detailed Description

Definition at line 9 of file [UserServiceModel.cs](#).

6.327.2 Member Function Documentation

6.327.2.1 GetData()

```
UserData MergeIt.Game.Services.UserServiceModel.GetData ( ) [inline]
```

Definition at line 33 of file [UserServiceModel.cs](#).

```
00034     {
00035         var userData = new UserData
00036         {
00037             Name = Name.Value,
00038             Energy = Energy.Value,
00039             Experience = Experience.Value,
00040             SoftCurrency = SoftCurrency.Value,
00041             HardCurrency = HardCurrency.Value,
00042             Splitters = Splitters.Value,
00043             Level = Level.Value,
00044             EnergyRestoringStartTime = EnergyRestoringStartTime
00045         };
00046
00047         return userData;
00048     }
```

6.327.2.2 Set()

```
void MergeIt.Game.Services.UserServiceModel.Set (
    IUserData data ) [inline]
```

Definition at line 21 of file [UserServiceModel.cs](#).

```
00022     {
00023         Name.Value = data.Name;
00024         Energy.Value = data.Energy;
00025         SoftCurrency.Value = data.SoftCurrency;
00026         HardCurrency.Value = data.HardCurrency;
00027         Splitters.Value = data.Splitters;
00028         Level.Value = data.Level;
00029         Experience.Value = data.Experience;
00030         EnergyRestoringStartTime = data.EnergyRestoringStartTime;
00031     }
```

6.327.3 Property Documentation

6.327.3.1 Energy

```
Bindable<int> MergeIt.Game.Services.UserServiceModel.Energy = new() [get]
```

Definition at line 12 of file [UserServiceModel.cs](#).

```
00012 { get; } = new();
```

6.327.3.2 EnergyRestoringStartTime

```
long MergeIt.Game.Services.UserServiceModel.EnergyRestoringStartTime [get], [set]
```

Definition at line 19 of file [UserServiceModel.cs](#).

```
00019 { get; set; }
```

6.327.3.3 Experience

```
Bindable<int> MergeIt.Game.Services.UserServiceModel.Experience = new() [get]
```

Definition at line 17 of file [UserServiceModel.cs](#).
00017 { get; } = new();

6.327.3.4 HardCurrency

```
Bindable<int> MergeIt.Game.Services.UserServiceModel.HardCurrency = new() [get]
```

Definition at line 14 of file [UserServiceModel.cs](#).
00014 { get; } = new();

6.327.3.5 Level

```
Bindable<int> MergeIt.Game.Services.UserServiceModel.Level = new() [get]
```

Definition at line 16 of file [UserServiceModel.cs](#).
00016 { get; } = new();

6.327.3.6 Name

```
Bindable<string> MergeIt.Game.Services.UserServiceModel.Name = new() [get]
```

Definition at line 11 of file [UserServiceModel.cs](#).
00011 { get; } = new();

6.327.3.7 SoftCurrency

```
Bindable<int> MergeIt.Game.Services.UserServiceModel.SoftCurrency = new() [get]
```

Definition at line 13 of file [UserServiceModel.cs](#).
00013 { get; } = new();

6.327.3.8 Splitters

```
Bindable<int> MergeIt.Game.Services.UserServiceModel.Splitters = new() [get]
```

Definition at line 15 of file [UserServiceModel.cs](#).
00015 { get; } = new();

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Game/Services/UserServiceModel.cs

6.328 Mergelt.Editor.Configs.CurrencyComponent.UxmlFactory Class Reference

6.328.1 Detailed Description

Definition at line 12 of file [CurrencyComponent.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/Configs/Components/CurrencyComponent.cs

6.329 Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView.UxmlFactory Class Reference

6.329.1 Detailed Description

Definition at line 19 of file [EvolutionsGraphView.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Editor/Core/EvolutionsEditor/Components/EvolutionsGraphView.cs

6.330 Mergelt.Core.MVP.View Class Reference

Inheritance diagram for Mergelt.Core.MVP.View:



Public Member Functions

- virtual void [Initialize \(\)](#)

Protected Member Functions

- virtual void [OnDestroy \(\)](#)

Properties

- GameObject [GameObject](#) [get]

Properties inherited from [Mergelt.Core.MVP.IView](#)

Events

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

Events inherited from [Mergelt.Core.MVP.IView](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

6.330.1 Detailed Description

Definition at line 8 of file [View.cs](#).

6.330.2 Member Function Documentation

6.330.2.1 Initialize()

```
virtual void MergeIt.Core.MVP.View.Initialize ( ) [inline], [virtual]
```

Implements [Mergelt.Core.MVP.IView](#).

Definition at line 18 of file [View.cs](#).

```
00019      {  
00020          InitializeEvent?.Invoke();  
00021      }
```

6.330.2.2 OnDestroy()

```
virtual void MergeIt.Core.MVP.View.OnDestroy ( ) [inline], [protected], [virtual]
```

Definition at line 23 of file [View.cs](#).

```
00024      {  
00025          DestroyEvent?.Invoke();  
00026      }
```

6.330.3 Property Documentation

6.330.3.1 GameObject

```
GameObject MergeIt.Core.MVP.View.GameObject [get]
```

Implements [Mergelt.Core.MVP.IView](#).

Definition at line 13 of file [View.cs](#).

```
00014      {  
00015          get => gameObject;  
00016      }
```

6.330.4 Event Documentation

6.330.4.1 DestroyEvent

Action MergeIt.Core.MVP.View.DestroyEvent

Definition at line 11 of file [View.cs](#).

6.330.4.2 InitializeEvent

Action MergeIt.Core.MVP.View.InitializeEvent

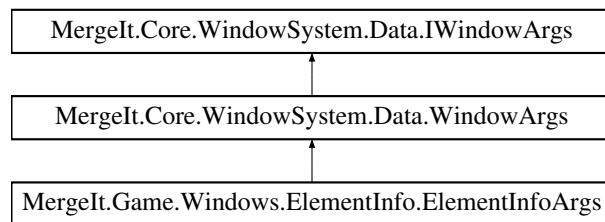
Definition at line 10 of file [View.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/MVP/View.cs

6.331 Mergelt.Core.WindowSystem.Data.WindowArgs Class Reference

Inheritance diagram for Mergelt.Core.WindowSystem.Data.WindowArgs:



6.331.1 Detailed Description

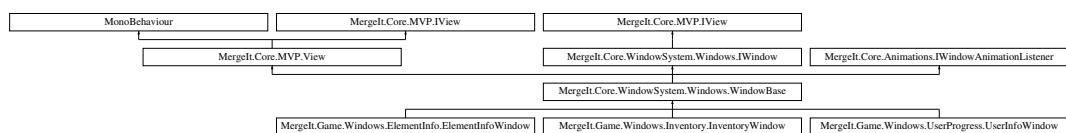
Definition at line 5 of file [WindowArgs.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/Data/WindowArgs.cs

6.332 Mergelt.Core.WindowSystem.Windows.WindowBase Class Reference

Inheritance diagram for Mergelt.Core.WindowSystem.Windows.WindowBase:



Public Member Functions

- virtual void [Show \(\)](#)
- virtual void [Close \(\)](#)
- virtual void [SetLayer \(string layer\)](#)
- virtual void [OnOpenStarted \(\)](#)
- virtual void [OnOpenFinished \(\)](#)
- virtual void [OnCloseStarted \(\)](#)
- virtual void [OnCloseFinished \(\)](#)

Public Member Functions inherited from [Mergelt.Core.MVP.View](#)

- virtual void [Initialize \(\)](#)

Protected Member Functions

- override void [OnDestroy \(\)](#)

Protected Attributes

- Button [CloseButton](#)
- Button [ClickOutsideArea](#)

Properties

- Canvas [Canvas \[get\]](#)
- IWindowAnimationController [AnimationController \[get\]](#)

Properties inherited from [Mergelt.Core.MVP.View](#)

- GameObject [GameObject \[get\]](#)

Properties inherited from [Mergelt.Core.MVP.IView](#)

Events

- Action [InitiateCloseEvent](#)
- Action [ShowStartEvent](#)
- Action [ShowEndEvent](#)
- Action [CloseStartEvent](#)
- Action [CloseEndEvent](#)

Events inherited from [Mergelt.Core.MVP.View](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

Events inherited from [MergeIt.Core.MVP.IView](#)

- Action [InitializeEvent](#)
- Action [DestroyEvent](#)

Events inherited from [MergeIt.Core.WindowSystem.Windows.IWindow](#)

- Action [InitiateCloseEvent](#)
- Action [ShowStartEvent](#)
- Action [ShowEndEvent](#)
- Action [CloseStartEvent](#)
- Action [CloseEndEvent](#)

6.332.1 Detailed Description

Definition at line 11 of file [WindowBase.cs](#).

6.332.2 Member Function Documentation

6.332.2.1 Close()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowBase.Close ( ) [inline], [virtual]
```

Implements [MergeIt.Core.WindowSystem.Windows.IWindow](#).

Definition at line 62 of file [WindowBase.cs](#).

```
00063     {  
00064         UnsubscribeFromClose();  
00065         AnimationController.CloseWindow();  
00066     }
```

6.332.2.2 OnCloseFinished()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowBase.OnCloseFinished ( ) [inline], [virtual]
```

Implements [MergeIt.Core.Animations.IWindowAnimationListener](#).

Definition at line 97 of file [WindowBase.cs](#).

```
00098     {  
00099         CloseEndEvent?.Invoke();  
00100     }
```

6.332.2.3 OnCloseStarted()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowBase.OnCloseStarted ( ) [inline], [virtual]
```

Implements [MergeIt.Core.Animations.IWindowAnimationListener](#).

Definition at line 92 of file [WindowBase.cs](#).

```
00093     {  
00094         CloseStartEvent?.Invoke();  
00095     }
```

6.332.2.4 OnDestroy()

```
override void MergeIt.Core.WindowSystem.Windows.WindowBase.OnDestroy ( ) [inline], [protected], [virtual]
```

Reimplemented from [MergeIt.Core.MVP.View](#).

Definition at line 102 of file [WindowBase.cs](#).

```
00103     {
00104         base.OnDestroy();
00105         UnsubscribeFromClose();
00106     }
```

6.332.2.5 OnOpenFinished()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowBase.OnOpenFinished ( ) [inline], [virtual]
```

Implements [MergeIt.Core.Animations.IWindowAnimationListener](#).

Definition at line 87 of file [WindowBase.cs](#).

```
00088     {
00089         ShowEndEvent?.Invoke();
00090     }
```

6.332.2.6 OnOpenStarted()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowBase.OnOpenStarted ( ) [inline], [virtual]
```

Implements [MergeIt.Core.Animations.IWindowAnimationListener](#).

Definition at line 82 of file [WindowBase.cs](#).

```
00083     {
00084         ShowStartEvent?.Invoke();
00085     }
```

6.332.2.7 SetLayer()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowBase.SetLayer (
    string layer ) [inline], [virtual]
```

Implements [MergeIt.Core.WindowSystem.Windows.IWindow](#).

Definition at line 68 of file [WindowBase.cs](#).

```
00069     {
00070         if (!string.IsNullOrEmpty(layer))
00071         {
00072             Canvas.overrideSorting = true;
00073             Canvas.sortingLayerName = layer;
00074         }
00075         else
00076         {
00077             Canvas.overrideSorting = false;
00078             Canvas.sortingLayerName = Configs.Windows.SortingLayers.Default;
00079         }
00080     }
```

6.332.2.8 Show()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowBase.Show () [inline], [virtual]
```

Implements [MergeIt.Core.WindowSystem.Windows.IWindow](#).

Definition at line 56 of file [WindowBase.cs](#).

```
00057     {
00058         SubscribeOnClose();
00059         AnimationController.OpenWindow();
00060     }
```

6.332.3 Member Data Documentation

6.332.3.1 ClickOutsideArea

```
Button MergeIt.Core.WindowSystem.Windows.WindowBase.ClickOutsideArea [protected]
```

Definition at line 24 of file [WindowBase.cs](#).

6.332.3.2 CloseButton

```
Button MergeIt.Core.WindowSystem.Windows.WindowBase.CloseButton [protected]
```

Definition at line 21 of file [WindowBase.cs](#).

6.332.4 Property Documentation

6.332.4.1 AnimationController

```
IWindowAnimationController MergeIt.Core.WindowSystem.Windows.WindowBase.AnimationController  
[get]
```

Definition at line 39 of file [WindowBase.cs](#).

```
00040     {
00041         get
00042         {
00043             if (_animationController == null)
00044             {
00045                 _animationController = GetComponent<IWindowAnimationController>() ?? new
00046                 WindowDefaultAnimationController();
00047                 _animationController.Initialize(this);
00048             }
00049             return _animationController;
00050         }
00051     }
```

6.332.4.2 Canvas

```
Canvas MergeIt.Core.WindowSystem.Windows.WindowBase.Canvas [get]
```

Definition at line 26 of file [WindowBase.cs](#).

```
00027     {
00028         get
00029         {
00030             if (!_canvas)
00031             {
00032                 _canvas = GetComponent<Canvas>();
00033             }
00034             return _canvas;
00035         }
00036     }
```

6.332.5 Event Documentation

6.332.5.1 CloseEndEvent

Action MergeIt.Core.WindowSystem.Windows.WindowBase.CloseEndEvent

Definition at line 18 of file [WindowBase.cs](#).

6.332.5.2 CloseStartEvent

Action MergeIt.Core.WindowSystem.Windows.WindowBase.CloseStartEvent

Definition at line 17 of file [WindowBase.cs](#).

6.332.5.3 InitiateCloseEvent

Action MergeIt.Core.WindowSystem.Windows.WindowBase.InitiateCloseEvent

Definition at line 13 of file [WindowBase.cs](#).

6.332.5.4 ShowEndEvent

Action MergeIt.Core.WindowSystem.Windows.WindowBase.ShowEndEvent

Definition at line 16 of file [WindowBase.cs](#).

6.332.5.5 ShowStartEvent

Action MergeIt.Core.WindowSystem.Windows.WindowBase.ShowStartEvent

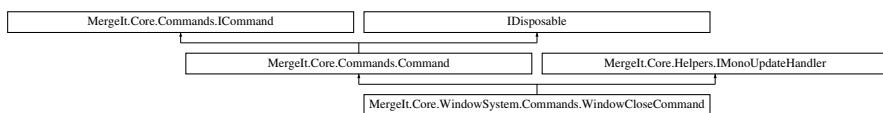
Definition at line 15 of file [WindowBase.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowBase.cs

6.333 Mergelt.Core.WindowSystem.Commands.WindowCloseCommand Class Reference

Inheritance diagram for Mergelt.Core.WindowSystem.Commands.WindowCloseCommand:



Public Member Functions

- [WindowCloseCommand \(IWindowPresenter presenter\)](#)
- [void Update \(\)](#)

Public Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- [virtual void Execute \(\)](#)
- [virtual async UniTask ExecuteAsync \(\)](#)
- [void Dispose \(\)](#)

Protected Member Functions

- [override void OnDispose \(\)](#)

Protected Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- [virtual void Finish \(\)](#)

Additional Inherited Members

Events inherited from [Mergelt.Core.Commands.Command](#)

- [Action< ICommand > Finished](#)

Events inherited from [Mergelt.Core.Commands.ICommand](#)

- [Action< ICommand > Finished](#)

6.333.1 Detailed Description

Definition at line 9 of file [WindowCloseCommand.cs](#).

6.333.2 Constructor & Destructor Documentation

6.333.2.1 WindowCloseCommand()

```
MergeIt.Core.WindowSystem.Commands.WindowCloseCommand.WindowCloseCommand (
    IWindowPresenter presenter) [inline]
```

Definition at line 13 of file [WindowCloseCommand.cs](#).

```
00014     {
00015         MonoEventsListener.Instance.SubscribeOnUpdate(this);
00016 
00017         _presenter = presenter;
00018         if (_presenter.State != WindowState.Closing &&
00019             _presenter.State != WindowState.Closed)
00020         {
00021             _presenter.Close();
00022         }
00023         else
00024         {
00025             Finish();
00026         }
00027     }
```

6.333.3 Member Function Documentation

6.333.3.1 OnDispose()

```
override void MergeIt.Core.WindowSystem.Commands.WindowCloseCommand.OnDispose ( ) [inline],  
[protected], [virtual]
```

Reimplemented from [MergeIt.Core.Commands.Command](#).

Definition at line 38 of file [WindowCloseCommand.cs](#).

```
00039     {  
00040         base.OnDispose();  
00041  
00042         MonoEventsListener.Instance.UnsubscribeFromUpdate(this);  
00043     }
```

6.333.3.2 Update()

```
void MergeIt.Core.WindowSystem.Commands.WindowCloseCommand.Update ( ) [inline]
```

Implements [MergeIt.Core.Helpers.IMonoUpdateHandler](#).

Definition at line 29 of file [WindowCloseCommand.cs](#).

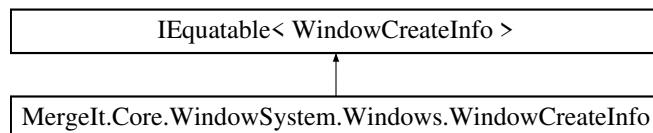
```
00030     {  
00031         if (_presenter is {State: WindowState.Closed})  
00032         {  
00033             _presenter = null;  
00034             Finish();  
00035         }  
00036     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/WindowSystem/Commands/WindowCloseCommand.cs

6.334 MergeIt.Core.WindowSystem.Windows.WindowCreateInfo Struct Reference

Inheritance diagram for MergeIt.Core.WindowSystem.Windows.WindowCreateInfo:



Public Member Functions

- bool [Equals \(WindowCreateInfo other\)](#)
- override bool [Equals \(object obj\)](#)
- override int [GetHashCode \(\)](#)

Properties

- string [LayerName](#) [get, set]
- string [PrefabPath](#) [get, set]
- Type [Type](#) [get, set]

6.334.1 Detailed Description

Definition at line 7 of file [WindowCreateInfo.cs](#).

6.334.2 Member Function Documentation

6.334.2.1 Equals() [1/2]

```
override bool MergeIt.Core.WindowSystem.Windows.WindowCreateInfo.Equals (
    object obj) [inline]
```

Definition at line 18 of file [WindowCreateInfo.cs](#).

```
00019     {
00020         return obj is WindowCreateInfo other && Equals(other);
00021     }
```

6.334.2.2 Equals() [2/2]

```
bool MergeIt.Core.WindowSystem.Windows.WindowCreateInfo.Equals (
    WindowCreateInfo other) [inline]
```

Definition at line 13 of file [WindowCreateInfo.cs](#).

```
00014     {
00015         return LayerName == other.LayerName && PrefabPath == other.PrefabPath && Type ==
00016             other.Type;
00017     }
```

6.334.2.3 GetHashCode()

```
override int MergeIt.Core.WindowSystem.Windows.WindowCreateInfo.GetHashCode () [inline]
```

Definition at line 23 of file [WindowCreateInfo.cs](#).

```
00024     {
00025         return HashCode.Combine(LayerName, PrefabPath, Type);
00026     }
```

6.334.3 Property Documentation

6.334.3.1 LayerName

```
string MergeIt.Core.WindowSystem.Windows.WindowCreateInfo.LayerName [get], [set]
```

Definition at line 9 of file [WindowCreateInfo.cs](#).

```
00009 { get; set; }
```

6.334.3.2 PrefabPath

string MergeIt.Core.WindowSystem.Windows.WindowCreateInfo.PrefabPath [get], [set]

Definition at line 10 of file [WindowCreateInfo.cs](#).

```
00010 { get; set; }
```

6.334.3.3 Type

Type MergeIt.Core.WindowSystem.Windows.WindowCreateInfo.Type [get], [set]

Definition at line 11 of file [WindowCreateInfo.cs](#).

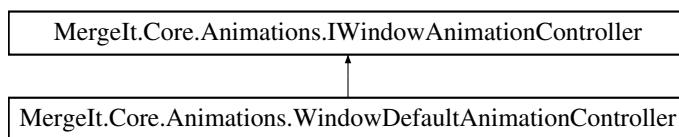
```
00011 { get; set; }
```

The documentation for this struct was generated from the following file:

- Assets/MergeIt/Source/Core/WindowSystem/Windows/WindowCreateInfo.cs

6.335 MergeIt.Core.Animations.WindowDefaultAnimationController Class Reference

Inheritance diagram for MergeIt.Core.Animations.WindowDefaultAnimationController:



Public Member Functions

- void [Initialize \(IWindowAnimationListener listener\)](#)
- void [OpenWindow \(\)](#)
- void [CloseWindow \(\)](#)
- void [OnOpenEnd \(\)](#)
- void [OnCloseEnd \(\)](#)

6.335.1 Detailed Description

Definition at line 5 of file [WindowDefaultAnimationController.cs](#).

6.335.2 Member Function Documentation

6.335.2.1 CloseWindow()

void MergeIt.Core.Animations.WindowDefaultAnimationController.CloseWindow () [inline]

Implements [MergeIt.Core.Animations.IWindowAnimationController](#).

Definition at line 20 of file [WindowDefaultAnimationController.cs](#).

```
00021 {
00022     _listener.OnCloseStarted();
00023     OnCloseEnd();
00024 }
```

6.335.2.2 Initialize()

```
void MergeIt.Core.Animations.WindowDefaultAnimationController.Initialize (
    IWindowAnimationListener listener ) [inline]
```

Implements [MergeIt.Core.Animations.IWindowAnimationController](#).

Definition at line 9 of file [WindowDefaultAnimationController.cs](#).

```
00010     {
00011         _listener = listener;
00012     }
```

6.335.2.3 OnCloseEnd()

```
void MergeIt.Core.Animations.WindowDefaultAnimationController.OnCloseEnd ( ) [inline]
```

Implements [MergeIt.Core.Animations.IWindowAnimationController](#).

Definition at line 31 of file [WindowDefaultAnimationController.cs](#).

```
00032     {
00033         _listener.OnCloseFinished();
00034     }
```

6.335.2.4 OnOpenEnd()

```
void MergeIt.Core.Animations.WindowDefaultAnimationController.OnOpenEnd ( ) [inline]
```

Implements [MergeIt.Core.Animations.IWindowAnimationController](#).

Definition at line 26 of file [WindowDefaultAnimationController.cs](#).

```
00027     {
00028         _listener.OnOpenFinished();
00029     }
```

6.335.2.5 OpenWindow()

```
void MergeIt.Core.Animations.WindowDefaultAnimationController.OpenWindow ( ) [inline]
```

Implements [MergeIt.Core.Animations.IWindowAnimationController](#).

Definition at line 14 of file [WindowDefaultAnimationController.cs](#).

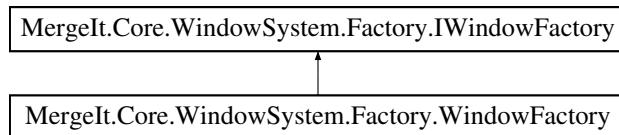
```
00015     {
00016         _listener.OnOpenStarted();
00017         OnOpenEnd();
00018     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/Animations/WindowDefaultAnimationController.cs

6.336 Mergelt.Core.WindowSystem.Factory.WindowFactory Class Reference

Inheritance diagram for Mergelt.Core.WindowSystem.Factory.WindowFactory:



Public Member Functions

- RectTransform [GetRoot \(\)](#)
- [BlackoutComponent GetBlackout](#) (RectTransform parent)
- TPresenter [CreateWindow< TPresenter >](#) ([WindowCreateInfo](#) createInfo, Transform parent, [IWindowArgs](#) windowArgs)

6.336.1 Detailed Description

Definition at line 10 of file [WindowFactory.cs](#).

6.336.2 Member Function Documentation

6.336.2.1 CreateWindow< TPresenter >()

```
TPresenter MergeIt.Core.WindowSystem.Factory.WindowFactory.CreateWindow< TPresenter > (
    WindowCreateInfo createInfo,
    Transform parent,
    IWindowArgs windowArgs ) [inline]
```

Implements [Mergelt.Core.WindowSystem.Factory.IWindowFactory](#).

Type Constraints

TPresenter : class
TPresenter : IWindowPresenter
TPresenter : new()

Definition at line 55 of file [WindowFactory.cs](#).

```
00056             : class, IWindowPresenter, new()
00057         {
00058             if (!string.IsNullOrEmpty(createInfo.PrefabPath))
00059             {
00060                 var presenter = new TPresenter();
00061                 GameObject windowPrototype = Resources.Load<GameObject>(createInfo.PrefabPath);
00062                 GameObject windowObject = Object.Instantiate(windowPrototype, parent);
00063
00064                 if ((windowObject ? windowObject.GetComponent(createInfo.Type) : null) is IWindow
00065                 window)
00066                 {
00067                     window.Initialize();
00068
00069                     presenter.Initialize(window, createInfo.LayerName, windowArgs);
00070                     presenter.SetWindowLayer();
00071                     presenter.SetWindowActive(false);
00072                 }
00073
00074             return presenter;
00075         }
00076
00077     return null;
00078 }
```

6.336.2.2 GetBlackout()

```
BlackoutComponent MergeIt.Core.WindowSystem.Factory.WindowFactory.GetBlackout (
    RectTransform parent ) [inline]
```

Implements [Mergelt.Core.WindowSystem.Factory.IWindowFactory](#).

Definition at line 30 of file [WindowFactory.cs](#).

```
00031     {
00032         var blackoutObject = Resources.Load<GameObject>(Blackout);
00033
00034         if (blackoutObject)
00035         {
00036             GameObject gameObject = Object.Instantiate(blackoutObject, parent);
00037
00038             if (gameObject)
00039             {
00040                 if (gameObject.TryGetComponent(out RectTransform rectTransform))
00041                 {
00042                     rectTransform.Stretch();
00043                     rectTransform.SetAsFirstSibling();
00044                 }
00045             }
00046
00047             gameObject.TryGetComponent(out BlackoutComponent blackoutComponent);
00048
00049             return blackoutComponent;
00050         }
00051
00052         return null;
00053     }
```

6.336.2.3 GetRoot()

```
RectTransform MergeIt.Core.WindowSystem.Factory.WindowFactory.GetRoot ( ) [inline]
```

Implements [Mergelt.Core.WindowSystem.Factory.IWindowFactory](#).

Definition at line 15 of file [WindowFactory.cs](#).

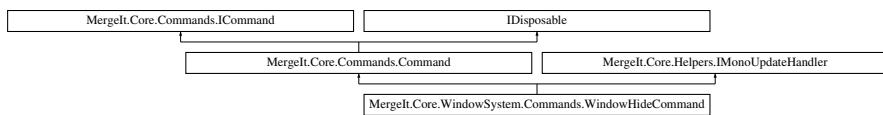
```
00016     {
00017         var root = Resources.Load<Canvas>(RootPath);
00018
00019         if (root)
00020         {
00021             root.worldCamera = Camera.main;
00022             GameObject gameObject = Object.Instantiate(root.gameObject);
00023
00024             return gameObject.GetComponent<RectTransform>();
00025         }
00026
00027         return null;
00028     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/Factory/WindowFactory.cs

6.337 Mergelt.Core.WindowSystem.Commands.WindowHideCommand Class Reference

Inheritance diagram for Mergelt.Core.WindowSystem.Commands.WindowHideCommand:



Public Member Functions

- [WindowHideCommand \(IWindowPresenter presenter\)](#)
- void [Update \(\)](#)

Public Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual void [Execute \(\)](#)
- virtual async UniTask [ExecuteAsync \(\)](#)
- void [Dispose \(\)](#)

Protected Member Functions

- override void [OnDispose \(\)](#)

Protected Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual void [Finish \(\)](#)

Additional Inherited Members

Events inherited from [Mergelt.Core.Commands.Command](#)

- Action< [ICommand >](#) Finished

Events inherited from [Mergelt.Core.Commands.ICommand](#)

- Action< [ICommand >](#) Finished

6.337.1 Detailed Description

Definition at line 9 of file [WindowHideCommand.cs](#).

6.337.2 Constructor & Destructor Documentation

6.337.2.1 [WindowHideCommand\(\)](#)

```
MergeIt.Core.WindowSystem.Commands.WindowHideCommand.WindowHideCommand (
    IWindowPresenter presenter) [inline]
```

Definition at line 13 of file [WindowHideCommand.cs](#).

```
00014     {
00015         MonoEventsListener.Instance.SubscribeOnUpdate(this);
00016 
00017         _presenter = presenter;
00018         if (_presenter.State != WindowState.Hiding &&
00019             _presenter.State != WindowState.Hidden)
00020         {
00021             _presenter.Hide();
00022         }
00023         else
00024         {
00025             Finish();
00026         }
00027 }
```

6.337.3 Member Function Documentation

6.337.3.1 OnDispose()

```
override void MergeIt.Core.WindowSystem.Commands.WindowHideCommand.OnDispose ( ) [inline],  
[protected], [virtual]
```

Reimplemented from [Mergelt.Core.Commands.Command](#).

Definition at line 38 of file [WindowHideCommand.cs](#).

```
00039     {  
00040         base.OnDispose();  
00041  
00042         MonoEventsListener.Instance.UnsubscribeFromUpdate(this);  
00043     }
```

6.337.3.2 Update()

```
void MergeIt.Core.WindowSystem.Commands.WindowHideCommand.Update ( ) [inline]
```

Implements [Mergelt.Core.Helpers.IMonoUpdateHandler](#).

Definition at line 29 of file [WindowHideCommand.cs](#).

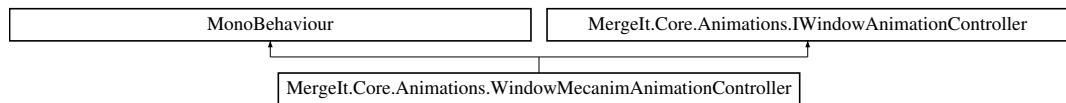
```
00030     {  
00031         if (_presenter is {State: WindowState.Hidden})  
00032         {  
00033             _presenter = null;  
00034             Finish();  
00035         }  
00036     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/Commands/WindowHideCommand.cs

6.338 Mergelt.Core.Animations.WindowMecanimAnimationController Class Reference

Inheritance diagram for Mergelt.Core.Animations.WindowMecanimAnimationController:



Public Member Functions

- void [Initialize \(IWindowAnimationListener listener\)](#)
- void [OpenWindow \(\)](#)
- void [CloseWindow \(\)](#)
- void [OnOpenEnd \(\)](#)
- void [OnCloseEnd \(\)](#)

6.338.1 Detailed Description

Definition at line 8 of file [WindowMecanimAnimationController.cs](#).

6.338.2 Member Function Documentation

6.338.2.1 CloseWindow()

```
void MergeIt.Core.Animations.WindowMecanimAnimationController.CloseWindow ( ) [inline]
```

Implements [MergeIt.Core.Animations.IWindowAnimationController](#).

Definition at line 42 of file [WindowMecanimAnimationController.cs](#).

```
00043     {
00044         _listener.OnCloseStarted();
00045         _animator.SetTrigger(_closeHash);
00046     }
```

6.338.2.2 Initialize()

```
void MergeIt.Core.Animations.WindowMecanimAnimationController.Initialize (
    IWindowAnimationListener listener ) [inline]
```

Implements [MergeIt.Core.Animations.IWindowAnimationController](#).

Definition at line 31 of file [WindowMecanimAnimationController.cs](#).

```
00032     {
00033         _listener = listener;
00034     }
```

6.338.2.3 OnCloseEnd()

```
void MergeIt.Core.Animations.WindowMecanimAnimationController.OnCloseEnd ( ) [inline]
```

Implements [MergeIt.Core.Animations.IWindowAnimationController](#).

Definition at line 53 of file [WindowMecanimAnimationController.cs](#).

```
00054     {
00055         _listener.OnCloseFinished();
00056     }
```

6.338.2.4 OnOpenEnd()

```
void MergeIt.Core.Animations.WindowMecanimAnimationController.OnOpenEnd ( ) [inline]
```

Implements [MergeIt.Core.Animations.IWindowAnimationController](#).

Definition at line 48 of file [WindowMecanimAnimationController.cs](#).

```
00049     {
00050         _listener.OnOpenFinished();
00051     }
```

6.338.2.5 OpenWindow()

```
void MergeIt.Core.Animations.WindowMecanimAnimationController.OpenWindow ( ) [inline]
```

Implements [Mergelt.Core.Animations.IWindowAnimationController](#).

Definition at line 36 of file [WindowMecanimAnimationController.cs](#).

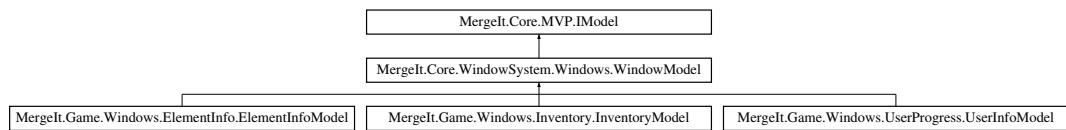
```
00037     {
00038         _listener.OnOpenStarted();
00039         _animator.SetTrigger(_openHash);
00040     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/Animations/WindowMecanimAnimationController.cs

6.339 Mergelt.Core.WindowSystem.Windows.WindowModel Class Reference

Inheritance diagram for Mergelt.Core.WindowSystem.Windows.WindowModel:



6.339.1 Detailed Description

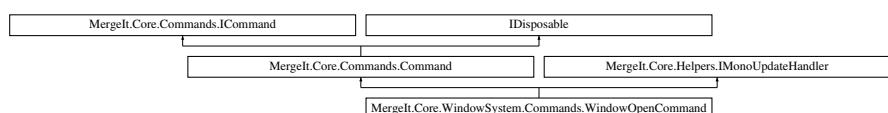
Definition at line 7 of file [WindowModel.cs](#).

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowModel.cs

6.340 Mergelt.Core.WindowSystem.Commands.WindowOpenCommand Class Reference

Inheritance diagram for Mergelt.Core.WindowSystem.Commands.WindowOpenCommand:



Public Member Functions

- [WindowOpenCommand](#) ([IWindowPresenter](#) presenter)
- void [Update](#) ()

Public Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual void [Execute \(\)](#)
- virtual async UniTask [ExecuteAsync \(\)](#)
- void [Dispose \(\)](#)

Protected Member Functions

- override void [OnDispose \(\)](#)

Protected Member Functions inherited from [Mergelt.Core.Commands.Command](#)

- virtual void [Finish \(\)](#)

Additional Inherited Members

Events inherited from [Mergelt.Core.Commands.Command](#)

- Action< [ICommand >](#) Finished

Events inherited from [Mergelt.Core.Commands.ICommand](#)

- Action< [ICommand >](#) Finished

6.340.1 Detailed Description

Definition at line 9 of file [WindowOpenCommand.cs](#).

6.340.2 Constructor & Destructor Documentation

6.340.2.1 [WindowOpenCommand\(\)](#)

```
MergeIt.Core.WindowSystem.Commands.WindowOpenCommand.WindowOpenCommand (
    IWindowPresenter presenter ) [inline]
```

Definition at line 13 of file [WindowOpenCommand.cs](#).

```
00014     {
00015         MonoEventsListener.Instance.SubscribeOnUpdate(this);
00016
00017         _presenter = presenter;
00018         if (_presenter.State != WindowState.Opening &&
00019             _presenter.State != WindowState.Opened)
00020         {
00021             _presenter.Show();
00022         }
00023         else
00024         {
00025             Finish();
00026         }
00027     }
```

6.340.3 Member Function Documentation

6.340.3.1 OnDispose()

```
override void MergeIt.Core.WindowSystem.Commands.WindowOpenCommand.OnDispose ( ) [inline],  
[protected], [virtual]
```

Reimplemented from [Mergelt.Core.Commands.Command](#).

Definition at line 38 of file [WindowOpenCommand.cs](#).

```
00039     {  
00040         base.OnDispose();  
00041  
00042         MonoEventsListener.Instance.UnsubscribeFromUpdate(this);  
00043     }
```

6.340.3.2 Update()

```
void MergeIt.Core.WindowSystem.Commands.WindowOpenCommand.Update ( ) [inline]
```

Implements [Mergelt.Core.Helpers.IMonoUpdateHandler](#).

Definition at line 29 of file [WindowOpenCommand.cs](#).

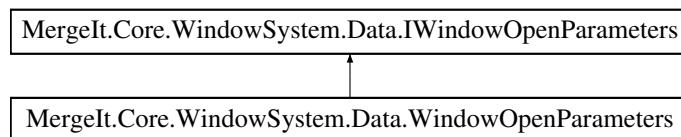
```
00030     {  
00031         if (_presenter is {State: WindowState.Opened})  
00032         {  
00033             _presenter = null;  
00034             Finish();  
00035         }  
00036     }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/Commands/WindowOpenCommand.cs

6.341 Mergelt.Core.WindowSystem.Data.WindowOpenParameters Class Reference

Inheritance diagram for Mergelt.Core.WindowSystem.Data.WindowOpenParameters:



Properties

- bool [ClosePrevious](#) [get, set]
- bool [NeedBlackout](#) [get, set]
- [IWindowPresenter](#) [Presenter](#) [get, set]

Properties inherited from [Mergelt.Core.WindowSystem.Data.IWindowOpenParameters](#)

6.341.1 Detailed Description

Definition at line 7 of file [WindowOpenParameters.cs](#).

6.341.2 Property Documentation

6.341.2.1 ClosePrevious

```
bool MergeIt.Core.WindowSystem.Data.WindowOpenParameters.ClosePrevious [get], [set]
```

Implements [Mergelt.Core.WindowSystem.Data.IWindowOpenParameters](#).

Definition at line 9 of file [WindowOpenParameters.cs](#).

```
00009 { get; set; }
```

6.341.2.2 NeedBlackout

```
bool MergeIt.Core.WindowSystem.Data.WindowOpenParameters.NeedBlackout [get], [set]
```

Implements [Mergelt.Core.WindowSystem.Data.IWindowOpenParameters](#).

Definition at line 10 of file [WindowOpenParameters.cs](#).

```
00010 { get; set; }
```

6.341.2.3 Presenter

```
IWindowPresenter MergeIt.Core.WindowSystem.Data.WindowOpenParameters.Presenter [get], [set]
```

Implements [Mergelt.Core.WindowSystem.Data.IWindowOpenParameters](#).

Definition at line 11 of file [WindowOpenParameters.cs](#).

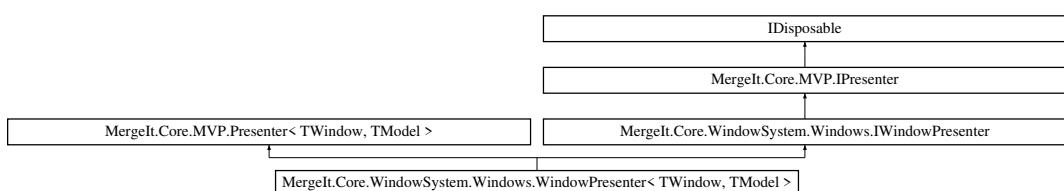
```
00011 { get; set; }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/Data/WindowOpenParameters.cs

6.342 Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel > Class Template Reference

Inheritance diagram for Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >:



Public Member Functions

- void [Initialize](#) ([IView](#) view, string layer, [IWindowArgs](#) windowArgs=null)
- void [Show](#) ()
- void [Hide](#) ()
- void [Close](#) ()
- void [SetWindowActive](#) (bool active)
- void [SetWindowLayer](#) ()
- void [DestroyWindow](#) ()

Public Member Functions inherited from [Mergelt.Core.MVP.Presenter< TWindow, TModel >](#)

- void [Initialize](#) ([IView](#) view)
- void [Dispose](#) ()

Public Member Functions inherited from [Mergelt.Core.MVP.IPresenter](#)

- void [Initialize](#) ([IView](#) view)

Protected Member Functions

- virtual void [OnInitialize](#) ([IWindowArgs](#) args=null)
- virtual void [OnDestroyWindow](#) ()
- override void [OnDispose](#) ()
- virtual void [OnWindowStartShowing](#) ()
- virtual void [OnWindowShown](#) ()
- virtual void [OnWindowStartClosing](#) ()
- virtual void [OnWindowClosed](#) ()

Protected Member Functions inherited from [Mergelt.Core.MVP.Presenter< TWindow, TModel >](#)

- virtual void [OnInitialize](#) ([TView](#) view)

Protected Attributes

- [IMessageBus](#) [MessageBus](#)
- [IWindowSystem](#) [WindowSystem](#)

Properties

- string [Layer](#) [get]
- [WindowState](#) [State](#) [get]

Properties inherited from [Mergelt.Core.MVP.Presenter< TWindow, TModel >](#)

- [TView](#) [View](#) [get]
- [TModel](#) [Model](#) [get]

Properties inherited from [MergeIt.Core.WindowSystem.Windows.IWindowPresenter](#)

6.342.1 Detailed Description

Type Constraints

TWindow : Component

TWindow : IWindow

TModel : WindowModel

TModel : new()

Definition at line 12 of file [WindowPresenter.cs](#).

6.342.2 Member Function Documentation

6.342.2.1 Close()

```
void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.Close ( ) [inline]
```

Implements [MergeIt.Core.WindowSystem.Windows.IWindowPresenter](#).

Definition at line 54 of file [WindowPresenter.cs](#).

```
00055      {
00056          _state = WindowState.Closing;
00057          InitiateClosing();
00058      }
```

6.342.2.2 DestroyWindow()

```
void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.DestroyWindow ( ) [inline]
```

Definition at line 70 of file [WindowPresenter.cs](#).

```
00071      {
00072          OnDestroyWindow();
00073
00074          View.InitiateCloseEvent -= OnInitiateCloseEvent;
00075
00076          Object.Destroy(View.gameObject);
00077      }
```

6.342.2.3 Hide()

```
void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.Hide ( ) [inline]
```

Implements [MergeIt.Core.WindowSystem.Windows.IWindowPresenter](#).

Definition at line 48 of file [WindowPresenter.cs](#).

```
00049      {
00050          _state = WindowState.Hiding;
00051          InitiateClosing();
00052      }
```

6.342.2.4 Initialize()

```
void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.Initialize (
    IView view,
    string layer,
    IWindowArgs windowArgs = null ) [inline]
```

Implements [MergeIt.Core.WindowSystem.Windows.IWindowPresenter](#).

Definition at line 28 of file [WindowPresenter.cs](#).

```
00029     {
00030         base.Initialize(view);
00031
00032         MessageBus = DiContainer.Get<IMessageBus>();
00033         WindowSystem = DiContainer.Get<IWindowSystem>();
00034
00035         Layer = layer;
00036
00037         OnInitialize(windowArgs);
00038     }
```

6.342.2.5 OnDestroyWindow()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.OnDestroy<-
Window ( ) [inline], [protected], [virtual]
```

Definition at line 83 of file [WindowPresenter.cs](#).

```
00084     {
00085 }
```

6.342.2.6 OnDispose()

```
override void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.OnDispose (
) [inline], [protected], [virtual]
```

Reimplemented from [MergeIt.Core.MVP.Presenter< TWindow, TModel >](#).

Definition at line 87 of file [WindowPresenter.cs](#).

```
00088     {
00089         base.OnDispose();
00090
00091         if (View)
00092         {
00093             View.InitiateCloseEvent -= OnInitiateCloseEvent;
00094         }
00095     }
```

6.342.2.7 OnInitialize()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.OnInitialize (
)
    IWindowArgs args = null ) [inline], [protected], [virtual]
```

Definition at line 79 of file [WindowPresenter.cs](#).

```
00080     {
00081 }
```

6.342.2.8 OnWindowClosed()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.OnWindowClosed () [inline], [protected], [virtual]
```

Definition at line 131 of file [WindowPresenter.cs](#).

```
00132     {
00133         View.CloseEndEvent -= OnWindowClosed;
00134
00135         if (_state == WindowState.Closing)
00136         {
00137             _state = WindowState.Closed;
00138             SetWindowActive(false);
00139             DestroyWindow();
00140         }
00141         else if (_state == WindowState.Hiding)
00142         {
00143             _state = WindowState.Hidden;
00144         }
00145     }
```

6.342.2.9 OnWindowShown()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.OnWindowShown () [inline], [protected], [virtual]
```

Definition at line 117 of file [WindowPresenter.cs](#).

```
00118     {
00119         View.ShowEndEvent -= OnWindowShown;
00120         View.InitiateCloseEvent += OnInitiateCloseEvent;
00121
00122         _state = WindowState.Opened;
00123     }
```

6.342.2.10 OnWindowStartClosing()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.OnWindowStartClosing () [inline], [protected], [virtual]
```

Definition at line 125 of file [WindowPresenter.cs](#).

```
00126     {
00127         View.CloseStartEvent -= OnWindowStartClosing;
00128         View.InitiateCloseEvent -= OnInitiateCloseEvent;
00129     }
```

6.342.2.11 OnWindowStartShowing()

```
virtual void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.OnWindowStartShowing () [inline], [protected], [virtual]
```

Definition at line 110 of file [WindowPresenter.cs](#).

```
00111     {
00112         SetWindowActive(true);
00113
00114         View.ShowStartEvent -= OnWindowStartShowing;
00115     }
```

6.342.2.12 SetWindowActive()

```
void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.SetWindowActive ( bool active ) [inline]
```

Implements [MergeIt.Core.WindowSystem.Windows.IWindowPresenter](#).

Definition at line 60 of file [WindowPresenter.cs](#).

```
00061     {  
00062         View.gameObject.SetActive(active);  
00063     }
```

6.342.2.13 SetWindowLayer()

```
void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.SetWindowLayer ( ) [inline]
```

Implements [MergeIt.Core.WindowSystem.Windows.IWindowPresenter](#).

Definition at line 65 of file [WindowPresenter.cs](#).

```
00066     {  
00067         View.SetLayer(Layer);  
00068     }
```

6.342.2.14 Show()

```
void MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.Show ( ) [inline]
```

Implements [MergeIt.Core.WindowSystem.Windows.IWindowPresenter](#).

Definition at line 40 of file [WindowPresenter.cs](#).

```
00041     {  
00042         _state = WindowState.Opening;  
00043         View.ShowStartEvent += OnWindowStartShowing;  
00044         View.ShowEndEvent += OnWindowShown;  
00045         View.Show();  
00046     }
```

6.342.3 Member Data Documentation

6.342.3.1 MessageBus

`IMessageBus MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.MessageBus` [protected]

Definition at line 16 of file [WindowPresenter.cs](#).

6.342.3.2 WindowSystem

`IWindowSystem MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.WindowSystem` [protected]

Definition at line 17 of file [WindowPresenter.cs](#).

6.342.4 Property Documentation

6.342.4.1 Layer

```
string MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.Layer [get]
```

Implements [MergeIt.Core.WindowSystem.Windows.IWindowPresenter](#).

Definition at line 21 of file [WindowPresenter.cs](#).

```
00021 { get; private set; }
```

6.342.4.2 State

```
WindowState MergeIt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >.State [get]
```

Implements [MergeIt.Core.WindowSystem.Windows.IWindowPresenter](#).

Definition at line 23 of file [WindowPresenter.cs](#).

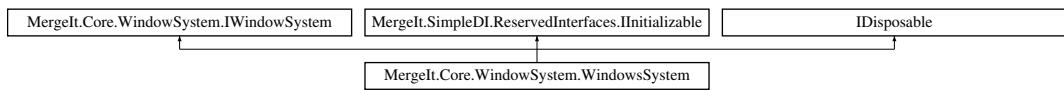
```
00024     {
00025         get => _state;
00026     }
```

The documentation for this class was generated from the following file:

- Assets/MergeIt/Source/Core/WindowSystem/Windows/WindowPresenter.cs

6.343 MergeIt.Core.WindowSystem.WindowsSystem Class Reference

Inheritance diagram for MergeIt.Core.WindowSystem.WindowsSystem:



Public Member Functions

- void [Initialize](#) ()
- void [Dispose](#) ()
- void [OpenWindow](#)< TPresenter > (bool closePrevious=false, bool enableBlackout=false, [IWindowArgs](#) args=null)
- void [CloseWindow](#) ([IWindowPresenter](#) presenter)

Properties

- [RectTransform Root](#) [get]
- [BlackoutComponent Blackout](#) [get]

6.343.1 Detailed Description

Definition at line 20 of file [WindowsSystem.cs](#).

6.343.2 Member Function Documentation

6.343.2.1 CloseWindow()

```
void MergeIt.Core.WindowSystem.WindowsSystem.CloseWindow (
    IWindowPresenter presenter) [inline]
```

Implements [Mergelt.Core.WindowSystem.IWindowSystem](#).

Definition at line 108 of file [WindowsSystem.cs](#).

```
00109     {
00110         ICommandManager commands = RequestCloseWindow(presenter);
00111         AddToQueue(commands);
00112     }
```

6.343.2.2 Dispose()

```
void MergeIt.Core.WindowSystem.WindowsSystem.Dispose () [inline]
```

Definition at line 82 of file [WindowsSystem.cs](#).

```
00083     {
00084         _messageBus.RemoveListener<CloseWindowMessage>(OnCloseWindowMessageHandler);
00085     }
```

6.343.2.3 Initialize()

```
void MergeIt.Core.WindowSystem.WindowsSystem.Initialize () [inline]
```

Implements [Mergelt.SimpleDI.ReservedInterfaces.IInitializable](#).

Definition at line 66 of file [WindowsSystem.cs](#).

```
00067     {
00068         _messageBus.AddListener<CloseWindowMessage>(OnCloseWindowMessageHandler);
00069         _layersConfig = Resources.Load<LayersConfig>(WindowsLayersConfigPath);
00070
00071         var sortedLayers = _layersConfig.Layers.OrderBy(x => x.Order).ToArray();
00072         _layers = new string[sortedLayers.Length];
00073
00074         int index = 0;
00075         foreach (LayerParameters layerParameters in sortedLayers)
00076         {
00077             _layersWindows[layerParameters.Name] = new LinkedList<IWindowPresenter>();
00078             _layers[index++] = layerParameters.Name;
00079         }
00080     }
```

6.343.2.4 OpenWindow< TPresenter >()

```
void MergeIt.Core.WindowSystem.WindowsSystem.OpenWindow< TPresenter > (
    bool closePrevious = false,
    bool enableBlackout = false,
    IWindowArgs args = null ) [inline]
```

Implements [MergeIt.Core.WindowSystem.IWindowSystem](#).

Type Constraints

TPresenter : *class*
TPresenter : *IWindowPresenter*
TPresenter : *new()*

Definition at line 87 of file [WindowsSystem.cs](#).

```
00088     : class, IWindowPresenter, new()
00089     {
00090         WindowCreateInfo createInfo = WindowsStorage.GetWindowInfo<TPresenter>();
00091         TPresenter windowPresenter = _windowFactory.CreateWindow<TPresenter>(createInfo,
00092             Root.transform, args);
00093         if (windowPresenter != null)
00094         {
00095             var openParameters = new WindowOpenParameters
00096             {
00097                 ClosePrevious = closePrevious,
00098                 NeedBlackout = enableBlackout,
00099                 Presenter = windowPresenter
00100             };
00101             _openedWindows[windowPresenter] = openParameters;
00102             ICommandManager commands = RequestOpenWindow(openParameters);
00103             AddToQueue(commands);
00104         }
00105     }
```

6.343.3 Property Documentation

6.343.3.1 Blackout

BlackoutComponent MergeIt.Core.WindowSystem.WindowsSystem.Blackout [get]

Definition at line 53 of file [WindowsSystem.cs](#).

```
00054     {
00055         get
00056         {
00057             if (!_blackout)
00058             {
00059                 _blackout = _windowFactory.GetBlackout(Root);
00060             }
00061             return _blackout;
00062         }
00063     }
```

6.343.3.2 Root

```
RectTransform MergeIt.Core.WindowSystem.WindowsSystem.Root [get]
```

Definition at line 40 of file [WindowsSystem.cs](#).

```
00041     {
00042         get
00043     {
00044         if (!_root)
00045         {
00046             _root = _windowFactory.GetRoot ();
00047         }
00048         return _root;
00049     }
00050 }
```

The documentation for this class was generated from the following file:

- Assets/Mergelt/Source/Core/WindowSystem/WindowsSystem.cs

Chapter 7

File Documentation

7.1 CurrencyComponent.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.IO;
00004 using UnityEditor;
00005 using UnityEditor.UIElements;
00006 using UnityEngine.UIElements;
00007
00008 namespace MergeIt.Editor.Configs
00009 {
00010     public class CurrencyComponent : VisualElement
00011     {
00012         public new class UxmlFactory : UxmlFactory<CurrencyComponent, CurrencyComponentsTraits> { }
00013
00014         public string Title { get; set; }
00015         public string TypeBinding { get; set; }
00016         public string AmountBinding { get; set; }
00017
00018         public CurrencyComponent()
00019         {
00020             RegisterCallback<AttachToPanelEvent>(OnAttachedToPanel);
00021         }
00022
00023         private void OnAttachedToPanel(AttachToPanelEvent evt)
00024         {
00025             Draw();
00026         }
00027
00028         private void Draw()
00029         {
00030             var styleSheet =
00031                 (StyleSheet)EditorGUIUtility.Load(Path.Combine(Constants.InspectorResourcesRoot,
00032                     "ItemsInspectorStyle.uss"));
00033             var visualTreeAsset =
00034                 (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.InspectorResourcesRoot,
00035                     "Configs",
00036                     "CurrencyComponent.uxml"));
00037
00038             visualTreeAsset.CloneTree(this);
00039
00040             styleSheets.Add(styleSheet);
00041
00042             Label titleLabel = this.Q<Label>("Title");
00043             titleLabel.text = Title;
00044
00045             EnumField enumField = this.Q<EnumField>("CurrencyTypeField");
00046             enumField.bindingPath = TypeBinding;
00047
00048             IntegerField integerField = this.Q<IntegerField>("AmountField");
00049             integerField.bindingPath = AmountBinding;
00050         }
00051 }
```

7.2 CurrencyComponentsTraits.cs

```
00001 // Copyright (c) 2024, Awessets
```

```

00002
00003 using UnityEngine.UIElements;
00004
00005 namespace MergeIt.Editor.Configs
00006 {
00007     public class CurrencyComponentsTraits : UxmlTraits
00008     {
00009         private readonly UxmlStringAttributeDescription _nameAttribute = new()
0010             {name = "name"};
0011
0012         private readonly UxmlStringAttributeDescription _titleAttribute = new()
0013             {name = "title", defaultValue = "Currency parameters"};
0014
0015         private readonly UxmlStringAttributeDescription _currencyTypeBindingAttribute = new()
0016             {name = "type-binding"};
0017
0018         private readonly UxmlStringAttributeDescription _currencyAmountBindingAttribute = new()
0019             {name = "amount-binding"};
0020
0021         public override void Init(VisualElement ve, IXmlElementAttributes bag, CreationContext cc)
0022         {
0023             base.Init(ve, bag, cc);
0024
0025             if (ve is CurrencyComponent currencyParameters)
0026             {
0027                 currencyParameters.name = _nameAttribute.GetValueFromBag(bag, cc);
0028                 currencyParameters.Title = _titleAttribute.GetValueFromBag(bag, cc);
0029                 currencyParameters.TypeBinding = _currencyTypeBindingAttribute.GetValueFromBag(bag,
0030 cc);
0031                 currencyParameters.AmountBinding =
0032                     _currencyAmountBindingAttribute.GetValueFromBag(bag, cc);
0033             }
0034         }

```

7.3 BaseSettingsDrawer.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Configs.Elements;
00005 using UnityEngine.UIElements;
00006
00007 namespace MergeIt.Editor.Configs.ItemsDrawers
00008 {
00009     public abstract class BaseSettingsDrawer : IDisposable
0010     {
0011         protected readonly VisualElement Parent;
0012         protected readonly VisualElement Root;
0013         protected readonly ElementConfig Config;
0014         protected Label Arrow;
0015
0016         public BaseSettingsDrawer(VisualElement parent, ElementConfig config)
0017         {
0018             Config = config;
0019             Root = new VisualElement();
0020             Parent = parent;
0021
0022             VisualTreeAsset visualTreeAsset = GetUxml();
0023             visualTreeAsset.CloneTree(Root);
0024
0025             Parent.Add(Root);
0026         }
0027
0028         public abstract void Draw();
0029
0030         public virtual void Hide()
0031         {
0032             Root.style.display = StylesConstants.DisplayNone;
0033         }
0034
0035         public virtual void Show()
0036         {
0037             Root.style.display = StylesConstants.DisplayFlex;
0038         }
0039
0040         protected void Expand(bool expand)
0041         {
0042             OnExpand(expand);
0043
0044             RotateArrow(expand);
0045         }

```

```

00046     protected abstract void OnExpand(bool expand);
00047
00048     protected void RotateArrow(bool rotate)
00049     {
00050         Arrow.style.rotate = rotate ? new StyleRotate(new Rotate(90)) : new StyleRotate(new
00051             Rotate(0));
00052     }
00053
00054     protected abstract VisualTreeAsset GetUxml();
00055
00056     public virtual void Dispose()
00057     {
00058     }
00059 }
00060 }
```

7.4 CommonSettingsDrawer.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.IO;
00004 using MergeIt.Core.Configs.Elements;
00005 using UnityEditor;
00006 using UnityEditor.UIElements;
00007 using UnityEngine;
00008 using UnityEngine.UIElements;
00009
00010 namespace MergeIt.Editor.Configs.ItemsDrawers
00011 {
00012     public class CommonSettingsDrawer : BaseSettingsDrawer
00013     {
00014         private readonly ElementCommonSettings _settings;
00015
00016         private Toggle _consumable;
00017         private Toggle _expanded;
00018         private ObjectField _icon;
00019         private Box _commonSettings;
00020
00021         public CommonSettingsDrawer(VisualElement parent, ElementConfig config) : base(parent, config)
00022         {
00023             _settings = config.CommonSettings;
00024         }
00025
00026         public override void Draw()
00027         {
00028             _icon = Root.Q<ObjectField>("IconContainer");
00029             _icon.objectType = typeof(FieldElementIconComponent);
00030             _icon.RegisterValueChangedCallback(OnImageChanged);
00031
00032             _commonSettings = Root.Q<Box>("CommonSettings");
00033
00034             var sellParams = Root.Q<CurrencyComponent>("SellParameters");
00035             sellParams.Bind(new SerializedObject(Config));
00036
00037             var splitParams = Root.Q<CurrencyComponent>("SplitParameters");
00038             splitParams.Bind(new SerializedObject(Config));
00039
00040             var unlockParams = Root.Q<CurrencyComponent>("UnlockParameters");
00041             unlockParams.Bind(new SerializedObject(Config));
00042
00043             var consumableContainer = Root.Q<VisualElement>("ConsumableContainer");
00044             consumableContainer.style.display =
00045                 _settings.IsConsumable ? StylesConstants.DisplayFlex : StylesConstants.DisplayNone;
00046
00047             _consumable = Root.Q<Toggle>("IsConsumable");
00048             _consumable.RegisterValueChangedCallback(OnConsumableChanged);
00049
00050             _expanded = Root.Q<Toggle>("CommonSettingsFoldout");
00051             _expanded.value = _settings.Expanded;
00052             _expanded.RegisterValueChangedCallback(OnExpandedChanged);
00053
00054             Arrow = Root.Q<Label>("Arrow");
00055
00056             SetImagePlaceholder(Config.CommonSettings.Icon);
00057
00058             Expand(_settings.Expanded);
00059         }
00060
00061         public override void Dispose()
00062         {
00063             base.Dispose();
00064         }
00065 }
```

```

00065         _consumable.UnregisterValueChangedCallback(OnConsumableChanged);
00066         _expanded.UnregisterValueChangedCallback(OnExpandedChanged);
00067         _icon.UnregisterValueChangedCallback(OnImageChanged);
00068
00069     Parent.Remove(Root);
00070 }
00071
00072     private void OnConsumableChanged(ChangeEvent<bool> evt)
00073 {
00074         var container = Root.Q<VisualElement>("ConsumableContainer");
00075         container.style.display = evt.newValue ? StylesConstants.DisplayFlex :
00076             StylesConstants.DisplayNone;
00077     }
00078
00079     private void OnExpandedChanged(ChangeEvent<bool> evt)
00080 {
00081     Expand(evt.newValue);
00082 }
00083
00084     protected override void OnExpand(bool expand)
00085 {
00086     _settings.Expanded = expand;
00087     _commonSettings.style.display = expand ? StylesConstants.DisplayFlex :
00088         StylesConstants.DisplayNone;
00089 }
00090
00091     private void SetImagePlaceholder(FieldElementIconComponent placeholder)
00092 {
00093     if (!placeholder)
00094     {
00095         return;
00096     }
00097
00098     Sprite sprite = null;
00099     UnityEngine.UI.Image image = placeholder.GetImage();
00100     if (image)
00101     {
00102         sprite = image.sprite;
00103     }
00104
00105     var icon = Root.Q<Image>("ItemImage");
00106     icon.image = sprite ? sprite.texture : null;
00107 }
00108
00109     private void OnImageChanged(ChangeEvent<Object> evt)
00110 {
00111     SetImagePlaceholder(evt.newValue as FieldElementIconComponent);
00112 }
00113
00114     protected override VisualTreeAsset GetUxml()
00115 {
00116     return
00117     (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.InspectorResourcesRoot,
00118             "CommonInspector.uxml"));
00119 }
00120 }
00121 }
```

7.5 ElementConfigInspector.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.IO;
00005 using MergeIt.Core.Configs.Elements;
00006 using MergeIt.Core.Configs.Types;
00007 using UnityEditor;
00008 using UnityEditor.UIElements;
00009 using UnityEngine;
00010 using UnityEngine.UIElements;
00011
00012 namespace MergeIt.Editor.Configs.ItemsDrawers
00013 {
00014     [CustomEditor(typeof(ElementConfig))]
00015     public class ElementConfigInspector : UnityEditor.Editor
00016     {
00017         private VisualElement _root;
00018         private ElementConfig _target;
00019         private static CommonSettingsDrawer _commonSettingsDrawer;
00020         private static GeneratorSettingsInspectorDrawer _generatorSettingsDrawer;
00021
00022         public ElementConfig Target
00023     {
```

```
00024         get
00025     {
00026         if (_target == null)
00027         {
00028             _target = target as ElementConfig;
00029         }
00030
00031         return _target;
00032     }
00033 }
00034
00035     public VisualElement Root
00036     {
00037         get => _root;
00038     }
00039
00040     public override VisualElement CreateInspectorGUI()
00041     {
00042         _root = new VisualElement();
00043
00044         GetUxml().CloneTree(Root);
00045
00046         var typeField = Root.Q<EnumField>("ElementType");
00047         typeField.RegisterValueChangedCallback(OnElementTypeChanged);
00048
00049         var stylesheet = GetStylesheet();
00050         if (!Root.styleSheets.Contains(stylesheet))
00051         {
00052             Root.styleSheets.Add(stylesheet);
00053         }
00054
00055         _commonSettingsDrawer?.Dispose();
00056         _commonSettingsDrawer = null;
00057
00058         _generatorSettingsDrawer?.Clear();
00059         _generatorSettingsDrawer = null;
00060
00061         Refresh(Target.Type);
00062
00063         serializedObject.ApplyModifiedProperties();
00064
00065         return Root;
00066     }
00067
00068     private StyleSheet GetStylesheet()
00069     {
00070         return (StyleSheet)EditorGUIUtility.Load(Path.Combine(Constants.InspectorResourcesRoot,
00071             "ItemsInspectorStyle.uss"));
00072     }
00073
00074     private void OnElementTypeChanged(ChangeEvent<Enum> changeEvent)
00075     {
00076         if (changeEvent.newValue == null)
00077         {
00078             return;
00079         }
00080
00081         if (Enum.TryParse(changeEvent.newValue.ToString(), out ElementType type))
00082         {
00083             Debug.Log($"Value changed to ({type}), Target type now is {Target.Type}");
00084
00085             Refresh(type);
00086         }
00087     }
00088
00089     private void Refresh(ElementType type)
00090     {
00091         if (_commonSettingsDrawer == null)
00092         {
00093             _commonSettingsDrawer = new CommonSettingsDrawer(Root, Target);
00094             _commonSettingsDrawer.Draw();
00095         }
00096
00097         switch (type)
00098         {
00099             case ElementType.Regular:
00100                 if (_generatorSettingsDrawer != null)
00101                 {
00102                     _generatorSettingsDrawer.Hide();
00103                 }
00104
00105                 break;
00106
00107             case ElementType.Generator:
00108                 if (_generatorSettingsDrawer != null)
00109                 {
00110                     _generatorSettingsDrawer.Show();
00111                 }
00112         }
00113     }
00114 }
```

```

00111         }
00112         else
00113         {
00114             _generatorSettingsDrawer = new GeneratorSettingsInspectorDrawer(Root, Target);
00115             _generatorSettingsDrawer.Draw();
00116         }
00117     }
00118     break;
00119 }
00120 }
00121
00122 private void OnDisable()
00123 {
00124     EditorUtility.SetDirty(target);
00125     AssetDatabase.SaveAssets();
00126 }
00127
00128 private VisualTreeAsset GetUxml()
00129 {
00130     return
00131     (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.InspectorResourcesRoot,
00132             "ElementInspector.uxml"));
00133 }
00134 }

```

7.6 GeneratorSettingsBaseDrawer.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.IO;
00004 using MergeIt.Core.Configs.Elements;
00005 using UnityEditor;
00006 using UnityEditor.UIElements;
00007 using UnityEngine.UIElements;
00008
00009 namespace MergeIt.Editor.Configs.ItemsDrawers
00010 {
00011     public class GeneratorSettingsBaseDrawer : BaseSettingsDrawer
00012     {
00013         private readonly ElementGeneratorSettings _settings;
00014         private Toggle _expanded;
00015         private Toggle _isUnlimitedToggle;
00016         private Toggle _needOpenToggle;
00017         private IntegerField _openTime;
00018         private Toggle _generateBlocked;
00019         private VisualElement _stackContainer;
00020         private VisualElement _needOpenContainer;
00021         private Box _generatorSettings;
00022
00023         public GeneratorSettingsBaseDrawer(VisualElement parent, ElementConfig config) : base(parent,
00024             config)
00025         {
00026             _settings = config.GeneratorSettings;
00027         }
00028
00029         public override void Draw()
00030         {
00031             _stackContainer = Root.Q<VisualElement>("StackContainer");
00032             _needOpenContainer = Root.Q<VisualElement>("NeedOpenContainer");
00033             _isUnlimitedToggle = Root.Q<Toggle>("IsUnlimited");
00034             _needOpenToggle = Root.Q<Toggle>("NeedOpen");
00035
00036             var skipOpenParams = Root.Q<CurrencyComponent>("SkipOpenParameters");
00037             skipOpenParams.Bind(new SerializedObject(Config));
00038
00039             var skipChargeParams = Root.Q<CurrencyComponent>("SkipChargeParameters");
00040             skipChargeParams.Bind(new SerializedObject(Config));
00041
00042             _generatorSettings = Root.Q<Box>("GeneratorSettings");
00043
00044             _isUnlimitedToggle.RegisterValueChangedCallback(OnUnlimitedChanged);
00045             _needOpenToggle.RegisterValueChangedCallback(OnNeedOpenChanged);
00046
00047             _stackContainer.style.display = _settings.IsUnlimited ? StylesConstants.DisplayNone :
00048                 StylesConstants.DisplayFlex;
00049             _needOpenContainer.style.display = _settings.NeedOpen ? StylesConstants.DisplayFlex :
00050                 StylesConstants.DisplayNone;
00051
00052             _expanded = Root.Q<Toggle>("GeneratorSettingsFoldout");
00053             _expanded.value = _settings.Expanded;
00054             _expanded.RegisterValueChangedCallback(OnExpandedChanged);
00055
00056

```

```

00053         Arrow = Root.Q<Label>("Arrow");
00054
00055         Expand(_settings.Expanded);
00056     }
00057
00058     public void Clear()
00059     {
00060         OnClear();
00061
00062         Parent.Remove(Root);
00063     }
00064
00065     protected virtual void OnClear()
00066     {
00067         _isUnlimitedToggle.UnregisterValueChangedCallback(OnUnlimitedChanged);
00068         _needOpenToggle.UnregisterValueChangedCallback(OnNeedOpenChanged);
00069     }
00070
00071     public override void Dispose()
00072     {
00073         base.Dispose();
00074
00075         Clear();
00076     }
00077
00078     private void OnExpandedChanged(ChangeEvent<bool> evt)
00079     {
00080         Expand(evt.newValue);
00081     }
00082
00083     private void OnUnlimitedChanged(ChangeEvent<bool> evt)
00084     {
00085         _stackContainer.style.display = evt.newValue ? StylesConstants.DisplayNone :
00086         StylesConstants.DisplayFlex;
00087     }
00088
00089     private void OnNeedOpenChanged(ChangeEvent<bool> evt)
00090     {
00091         _needOpenContainer.style.display = evt.newValue ? StylesConstants.DisplayFlex :
00092         StylesConstants.DisplayNone;
00093     }
00094
00095     protected override void OnExpand(bool expand)
00096     {
00097         _settings.Expanded = expand;
00098         _generatorSettings.style.display =
00099             expand ? StylesConstants.DisplayFlex : StylesConstants.DisplayNone;
00100
00101     protected override VisualTreeAsset GetUxml()
00102     {
00103         return
00104         (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.InspectorResourcesRoot,
00105             "GeneratorInspector.uxml"));
00106     }
00106 }

```

7.7 GeneratorSettingsEvolutionDrawer.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Editor.EvolutionsEditor;
00005 using UnityEngine.UIElements;
00006
00007 namespace MergeIt.Editor.Configs.ItemsDrawers
00008 {
00009     public class GeneratorSettingsEvolutionDrawer : GeneratorSettingsBaseDrawer
00010     {
00011         private readonly EvolutionsGraphView _graphView;
00012
00013         private GeneratedItemsCreator _creator;
00014
00015         public GeneratorSettingsEvolutionDrawer(VisualElement parent, ElementConfig config,
00016             EvolutionsGraphView evolutionsGraphView)
00017             : base(parent, config)
00018         {
00019             _graphView = evolutionsGraphView;
00020
00021         public override void Draw()
00022         {

```

```

00023         base.Draw();
00024
00025         var genItemsContainer = Root.Q<VisualElement>("GenItemsContainer");
00026         var genItemsControls = Root.Q<VisualElement>("GeneratedItemsControls");
00027         _creator = new GeneratedItemsCreator(genItemsControls, genItemsContainer, Config,
00028             _graphView.EvolutionsData, _graphView.Configs);
00029         _creator.Generate();
00030     }
00031
00032     protected override void OnClear()
00033     {
00034         base.OnClear();
00035         _creator.Dispose();
00036     }
00037 }
00038 }
```

7.8 GeneratorSettingsInspectorDrawer.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using UnityEngine.UIElements;
00005
00006 namespace MergeIt.Editor.Configs.ItemsDrawers
00007 {
00008     public class GeneratorSettingsInspectorDrawer : GeneratorSettingsBaseDrawer
00009     {
00010         public GeneratorSettingsInspectorDrawer(VisualElement parent, ElementConfig config) :
00011             base(parent, config)
00012         {
00013
00014             public override void Draw()
00015             {
00016                 base.Draw();
00017
00018                 var cantGenerateSection = Root.Q<VisualElement>("CantGenerateSection");
00019                 var helpBox = new HelpBox("Editing this section is available only from the 'Evolutions
editor', " +
00020                                     "that you can find in the menu 'Window/Merge Toolkit/Evolutions
editor'", HelpBoxMessageType.Info);
00021                 cantGenerateSection.Add(helpBox);
00022             }
00023         }
00024     }
00025 }
```

7.9 StylesConstants.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Configs.Types;
00005 using UnityEngine.UIElements;
00006
00007 namespace MergeIt.Editor.Configs
00008 {
00009     public static class StylesConstants
0010     {
0011         public static readonly Dictionary<ElementType, string> NodeStyles = new()
0012         {
0013             {ElementType.Regular, "nodeBorderSimple"},
0014             {ElementType.Generator, "nodeBorderGenerator"}
0015         };
0016
0017         public static StyleEnum<DisplayStyle> DisplayNone = new(DisplayStyle.None);
0018         public static StyleEnum<DisplayStyle> DisplayFlex = new(DisplayStyle.Flex);
0019         public static StyleEnum<Position> AbsolutePosition = new(Position.Absolute);
0020         public static StyleEnum<Position> RelativePosition = new(Position.Relative);
0021         public static StyleLength Length0 = new(0f);
0022         public const string InvalidNode = "nodeInvalid";
0023     }
0024 }
```

7.10 Constants.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Editor
00006 {
00007     public static class Constants
00008     {
00009         public static Vector2 NodeCreationOffset = new(280f, 0f);
00010
00011         public const string LevelEditorResourcesRoot = "MergeLevelEditor";
00012         public const string EvolutionsEditorResourcesRoot = "MergeEvolutionsEditor";
00013         public const string InspectorResourcesRoot = "MergeCustomInspectors";
00014         public const string EvolvedFromPort = "evolved from";
00015         public const string EvolvePort = "evolve";
00016         public const string EvolutionRootPort = "evolution of";
00017         public const string NamePort = "name";
00018         public const string HasNamePort = "has name";
00019         public const string GeneratedByPort = "generated by";
00020         public const string GeneratePort = "generate";
00021     }
00022 }
```

7.11 EvolutionsGraphView.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using System.Linq;
00006 using MergeIt.Core.Configs.Data;
00007 using MergeIt.Core.Configs.Elements;
00008 using MergeIt.Core.Schemes;
00009 using MergeIt.Core.Schemes.Data;
00010 using UnityEditor;
00011 using UnityEditor.Experimental.GraphView;
00012 using UnityEngine;
00013 using UnityEngine.UIElements;
00014
00015 namespace MergeIt.Editor.EvolutionsEditor
00016 {
00017     public class EvolutionsGraphView : GraphView
00018     {
00019         public new class UxmlFactory : UxmlFactory<EvolutionsGraphView, EvolutionsGraphViewTraits>
00020         {
00021         }
00022
00023         protected override bool canCopySelection
00024         {
00025             get => selection.TrueForAll(x => x is IEvolutionNode);
00026         }
00027
00028         protected override bool canDuplicateSelection
00029         {
00030             get => selection.TrueForAll(x => x is IEvolutionNode);
00031         }
00032
00033         private EventBase _currentEvent;
00034         private List<EvolutionData> _evolutionsData;
00035         private List<ElementConfig> _elementConfigs = new();
00036
00037         internal List<EvolutionData> EvolutionsData
00038         {
00039             get => _evolutionsData;
00040         }
00041
00042         internal List<ElementConfig> Configs
00043         {
00044             get => _elementConfigs;
00045         }
00046
00047         public EvolutionsGraphView()
00048         {
00049             serializeGraphElements = OnSerializeGraphElements;
00050             canPasteSerializedData = OnCanPasteSerializedData;
00051             unserializeAndPaste = OnUnserializeAndPaste;
00052             deleteSelection = OnDeleteSelection;
00053         }
00054
00055         public override void BuildContextualMenu(ContextualMenuPopulateEvent evt)
00056         {
```

```

00057         if (evt.target is GraphView)
00058     {
00059         evt.menu.AppendSeparator();
00060         evt.menu.AppendAction("Paste", _ => PasteCallback(),
00061             _ => canPaste ? DropdownMenuAction.Status.Normal :
00062             DropdownMenuAction.Status.Disabled);
00063         evt.menu.AppendAction("Create evolution root", CreateEvolutionRoot);
00064     }
00065
00066         if (evt.target is ISaveNode)
00067     {
00068         evt.menu.AppendSeparator();
00069         evt.menu.AppendAction("Copy", _ => CopySelectionCallback(),
00070             _ => canCopySelection ? DropdownMenuAction.Status.Normal :
00071                 DropdownMenuAction.Status.Disabled);
00072         evt.menu.AppendAction("Duplicate", a => DuplicateSelectionCallback(),
00073             a =>
00074                 canDuplicateSelection ? DropdownMenuAction.Status.Normal :
00075                     DropdownMenuAction.Status.Disabled);
00076         evt.menu.AppendAction("Delete", _ => DeleteSelectionCallback(AskUser.DontAskUser),
00077             _ => canDeleteSelection ? DropdownMenuAction.Status.Normal :
00078                 DropdownMenuAction.Status.Disabled);
00079         evt.menu.AppendSeparator();
00080     }
00081
00082         public void Validate()
00083     {
00084         nodes.ForEach(x =>
00085     {
00086         if (x is IELEMENTNODE elementNode)
00087     {
00088         elementNode.Validate();
00089     });
00090
00091         private void CreateEvolutionRoot(DropdownMenuAction action)
00092     {
00093         var evolutionNodeData = new EvolutionNodeData();
00094         Vector2 position = GetLocalMousePosition(action.eventInfo.mousePosition);
00095         var rect = new Rect(position.x, position.y, 160f, 50f);
00096         evolutionNodeData.Position = rect;
00097
00098         CreateEvolutionNode(evolutionNodeData);
00099     }
00100
00101         private void CreateEvolutionNode(EvolutionNodeData data)
00102     {
00103         var evolutionNode = new EvolutionNode(data);
00104         evolutionNode.SetPosition(data.Position);
00105
00106         AddElement(evolutionNode);
00107     }
00108
00109         private Vector2 GetLocalMousePosition(Vector2 mousePosition)
00110     {
00111         var worldPosition = (mousePosition - (Vector2)viewTransform.position) * (1 / scale);
00112         return worldPosition;
00113     }
00114
00115         public override List<Port> GetCompatiblePorts(Port startPort, NodeAdapter nodeAdapter)
00116     {
00117         var compatibleAnchors = new List<Port>();
00118
00119         foreach (Port candidateAnchor in ports.ToList())
00120     {
00121         if (startPort == candidateAnchor ||
00122             startPort.node == candidateAnchor.node ||
00123             startPort.direction == candidateAnchor.direction ||
00124             startPort.portType != candidateAnchor.portType)
00125     {
00126         continue;
00127     }
00128
00129         compatibleAnchors.Add(candidateAnchor);
00130     }
00131
00132         return compatibleAnchors;
00133     }
00134
00135         public SchemeData Save()
00136     {
00137         var allElements = graphElements.ToList();
00138
00139         var allEdges = allElements.OfType<Edge>();
00140         var allNodes = allElements.OfType<ISaveNode>();

```

```

00140
00141     var elementsToSave = new List<ElementNodeData>();
00142     var evolutionsToSave = new List<EvolutionNodeData>();
00143     var edgesToSave = new List<EdgeData>();
00144
00145     var brokenNodes = new List<ElementNode>();
00146
00147     foreach (ISaveNode saveNode in allNodes)
00148     {
00149         if (saveNode is ElementNode elementNode)
00150         {
00151             if (!elementNode.Validate())
00152             {
00153                 brokenNodes.Add(elementNode);
00154             }
00155         }
00156     }
00157
00158     if (brokenNodes.Count > 0)
00159     {
00160         ClearSelection();
00161
00162         foreach (ElementNode brokenNode in brokenNodes)
00163         {
00164             AddToSelection(brokenNode);
00165         }
00166
00167         EditorWindow.focusedWindow.ShowNotification(new GUIContent($"Can't save scheme with
00168         broken node. Broken nodes was selected."));
00169         return null;
00170     }
00171
00172     foreach (ISaveNode saveNode in allNodes)
00173     {
00174         saveNode.SaveData();
00175         switch (saveNode)
00176         {
00177             case IElementNode elementNode:
00178                 elementsToSave.Add(elementNode.Data as ElementNodeData);
00179                 break;
00180
00181             case IEvolutionNode evolutionNode:
00182                 evolutionsToSave.Add(evolutionNode.Data as EvolutionNodeData);
00183                 break;
00184         }
00185
00186     foreach (Edge edge in allEdges)
00187     {
00188         var edgeData = new EdgeData();
00189         edgeData.InputPortName = edge.input.portName;
00190         edgeData.OutputPortName = edge.output.portName;
00191
00192         if (edge.input.node is ISaveNode saveNode)
00193         {
00194             edgeData.InputNodeID = saveNode.Id;
00195         }
00196
00197         if (edge.output.node is ISaveNode outputNode)
00198         {
00199             edgeData.OutputNodeID = outputNode.Id;
00200         }
00201
00202         edgesToSave.Add(edgeData);
00203     }
00204
00205     List<EvolutionData> evolutionsData = BakeEvolutionsData();
00206     var saveData = new SchemeData
00207     {
00208         ElementsNodesData = elementsToSave,
00209         EvolutionsNodesData = evolutionsToSave,
00210         Edges = edgesToSave,
00211         EvolutionsData = evolutionsData
00212     };
00213
00214     return saveData;
00215 }
00216
00217 public List<EvolutionData> BakeEvolutionsData()
00218 {
00219     var allElements = graphElements.ToList();
00220     var evolutionsNodes = allElements.OfType<IEvolutionNode>();
00221     _evolutionsData = new List<EvolutionData>();
00222
00223     foreach (IEvolutionNode itemNode in evolutionsNodes)
00224     {
00225         Port startPort = itemNode.GetPort(Constants.EvolutionRootPort);

```

```

00226             if (startPort.connected)
00227             {
00228                 var evoData = new EvolutionData(itemNode.Data.Id,
00229                     itemNode.Data.Name,
00230                     itemNode.Data.Description,
00231                     itemNode.Data.Discovered);
00232
00233                 IElementNode firstNode = startPort.connections.First().input.node as IElementNode;
00234                 FillChain(evoData.Chain, firstNode);
00235
00236                 _evolutionsData.Add(evoData);
00237             }
00238         }
00239
00240         ExtractConfigs();
00241
00242         return _evolutionsData;
00243     }
00244
00245     private void FillChain(List<ElementConfig> chain, IElementNode node)
00246     {
00247         chain.Add(node.Config);
00248
00249         Port evolvePort = node.GetPort(Constants.EvolvePort);
00250         if (evolvePort.connected)
00251         {
00252             var nextNode = evolvePort.connections.First().input.node as IElementNode;
00253
00254             FillChain(chain, nextNode);
00255         }
00256     }
00257
00258     public void Restore(SchemeData savedGraph)
00259     {
00260         _evolutionsData = savedGraph.EvolutionsData;
00261
00262         ExtractConfigs();
00263
00264         DeleteElements(nodes.ToList(), edges.ToList());
00265
00266         List<ElementNodeData> elementsNodesData = savedGraph.ElementsNodesData;
00267         List<EvolutionNodeData> evolutionNodesData = savedGraph.EvolutionsNodesData;
00268         List<EdgeData> edgesData = savedGraph.Edges;
00269
00270         if (elementsNodesData != null)
00271         {
00272             CreateNodes(elementsNodesData);
00273         }
00274
00275         if (evolutionNodesData != null)
00276         {
00277             CreateEvolutionsNodes(evolutionNodesData, CreateEvolutionNode);
00278         }
00279
00280         if (edgesData != null)
00281         {
00282             var allNodes = nodes.ToList().OfType<ISaveNode>();
00283
00284             foreach (EdgeData edgeData in edgesData)
00285             {
00286                 ISaveNode inputNode = null;
00287                 ISaveNode outputNode = null;
00288                 foreach (ISaveNode itemNode in allNodes)
00289                 {
00290                     if (itemNode.Id == edgeData.InputNodeID)
00291                     {
00292                         inputNode = itemNode;
00293                     }
00294
00295                     if (itemNode.Id == edgeData.OutputNodeID)
00296                     {
00297                         outputNode = itemNode;
00298                     }
00299
00300                     if (outputNode != null && inputNode != null)
00301                     {
00302                         break;
00303                     }
00304                 }
00305
00306                 if (outputNode != null && inputNode != null)
00307                 {
00308                     var outputPort = outputNode.GetPort(edgeData.OutputPortName);
00309                     var inputPort = inputNode.GetPort(edgeData.InputPortName);
00310
00311                     if (outputPort != null && inputPort != null)
00312                     {
00313
00314
00315
00316
00317
00318
00319
00320
00321
00322
00323
00324
00325
00326
00327
00328
00329
00330
00331
00332
00333
00334
00335
00336
00337
00338
00339
00340
00341
00342
00343
00344
00345
00346
00347
00348
00349
00350
00351
00352
00353
00354
00355
00356
00357
00358
00359
00360
00361
00362
00363
00364
00365
00366
00367
00368
00369
00370
00371
00372
00373
00374
00375
00376
00377
00378
00379
00380
00381
00382
00383
00384
00385
00386
00387
00388
00389
00390
00391
00392
00393
00394
00395
00396
00397
00398
00399
00400
00401
00402
00403
00404
00405
00406
00407
00408
00409
00410
00411
00412
00413
00414
00415
00416
00417
00418
00419
00420
00421
00422
00423
00424
00425
00426
00427
00428
00429
00430
00431
00432
00433
00434
00435
00436
00437
00438
00439
00440
00441
00442
00443
00444
00445
00446
00447
00448
00449
00450
00451
00452
00453
00454
00455
00456
00457
00458
00459
00460
00461
00462
00463
00464
00465
00466
00467
00468
00469
00470
00471
00472
00473
00474
00475
00476
00477
00478
00479
00480
00481
00482
00483
00484
00485
00486
00487
00488
00489
00490
00491
00492
00493
00494
00495
00496
00497
00498
00499
00500
00501
00502
00503
00504
00505
00506
00507
00508
00509
00510
00511
00512
00513
00514
00515
00516
00517
00518
00519
00520
00521
00522
00523
00524
00525
00526
00527
00528
00529
00530
00531
00532
00533
00534
00535
00536
00537
00538
00539
00540
00541
00542
00543
00544
00545
00546
00547
00548
00549
00550
00551
00552
00553
00554
00555
00556
00557
00558
00559
00560
00561
00562
00563
00564
00565
00566
00567
00568
00569
00570
00571
00572
00573
00574
00575
00576
00577
00578
00579
00580
00581
00582
00583
00584
00585
00586
00587
00588
00589
00590
00591
00592
00593
00594
00595
00596
00597
00598
00599
00600
00601
00602
00603
00604
00605
00606
00607
00608
00609
00610
00611
00612
00613
00614
00615
00616
00617
00618
00619
00620
00621
00622
00623
00624
00625
00626
00627
00628
00629
00630
00631
00632
00633
00634
00635
00636
00637
00638
00639
00640
00641
00642
00643
00644
00645
00646
00647
00648
00649
00650
00651
00652
00653
00654
00655
00656
00657
00658
00659
00660
00661
00662
00663
00664
00665
00666
00667
00668
00669
00670
00671
00672
00673
00674
00675
00676
00677
00678
00679
00680
00681
00682
00683
00684
00685
00686
00687
00688
00689
00690
00691
00692
00693
00694
00695
00696
00697
00698
00699
00700
00701
00702
00703
00704
00705
00706
00707
00708
00709
00710
00711
00712
00713
00714
00715
00716
00717
00718
00719
00720
00721
00722
00723
00724
00725
00726
00727
00728
00729
00730
00731
00732
00733
00734
00735
00736
00737
00738
00739
00740
00741
00742
00743
00744
00745
00746
00747
00748
00749
00750
00751
00752
00753
00754
00755
00756
00757
00758
00759
00760
00761
00762
00763
00764
00765
00766
00767
00768
00769
00770
00771
00772
00773
00774
00775
00776
00777
00778
00779
00780
00781
00782
00783
00784
00785
00786
00787
00788
00789
00790
00791
00792
00793
00794
00795
00796
00797
00798
00799
00800
00801
00802
00803
00804
00805
00806
00807
00808
00809
00810
00811
00812
00813
00814
00815
00816
00817
00818
00819
00820
00821
00822
00823
00824
00825
00826
00827
00828
00829
00830
00831
00832
00833
00834
00835
00836
00837
00838
00839
00840
00841
00842
00843
00844
00845
00846
00847
00848
00849
00850
00851
00852
00853
00854
00855
00856
00857
00858
00859
00860
00861
00862
00863
00864
00865
00866
00867
00868
00869
00870
00871
00872
00873
00874
00875
00876
00877
00878
00879
00880
00881
00882
00883
00884
00885
00886
00887
00888
00889
00890
00891
00892
00893
00894
00895
00896
00897
00898
00899
00900
00901
00902
00903
00904
00905
00906
00907
00908
00909
00910
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00930
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00990
00991
00992
00993
00994
00995
00996
00997
00998
00999
01000
01001
01002
01003
01004
01005
01006
01007
01008
01009
01010
01011
01012
01013
01014
01015
01016
01017
01018
01019
01020
01021
01022
01023
01024
01025
01026
01027
01028
01029
01030
01031
01032
01033
01034
01035
01036
01037
01038
01039
01040
01041
01042
01043
01044
01045
01046
01047
01048
01049
01050
01051
01052
01053
01054
01055
01056
01057
01058
01059
01060
01061
01062
01063
01064
01065
01066
01067
01068
01069
01070
01071
01072
01073
01074
01075
01076
01077
01078
01079
01080
01081
01082
01083
01084
01085
01086
01087
01088
01089
01090
01091
01092
01093
01094
01095
01096
01097
01098
01099
01100
01101
01102
01103
01104
01105
01106
01107
01108
01109
01110
01111
01112
01113
01114
01115
01116
01117
01118
01119
01120
01121
01122
01123
01124
01125
01126
01127
01128
01129
01130
01131
01132
01133
01134
01135
01136
01137
01138
01139
01140
01141
01142
01143
01144
01145
01146
01147
01148
01149
01150
01151
01152
01153
01154
01155
01156
01157
01158
01159
01160
01161
01162
01163
01164
01165
01166
01167
01168
01169
01170
01171
01172
01173
01174
01175
01176
01177
01178
01179
01180
01181
01182
01183
01184
01185
01186
01187
01188
01189
01190
01191
01192
01193
01194
01195
01196
01197
01198
01199
01200
01201
01202
01203
01204
01205
01206
01207
01208
01209
01210
01211
01212
01213
01214
01215
01216
01217
01218
01219
01220
01221
01222
01223
01224
01225
01226
01227
01228
01229
01230
01231
01232
01233
01234
01235
01236
01237
01238
01239
01240
01241
01242
01243
01244
01245
01246
01247
01248
01249
01250
01251
01252
01253
01254
01255
01256
01257
01258
01259
01260
01261
01262
01263
01264
01265
01266
01267
01268
01269
01270
01271
01272
01273
01274
01275
01276
01277
01278
01279
01280
01281
01282
01283
01284
01285
01286
01287
01288
01289
01290
01291
01292
01293
01294
01295
01296
01297
01298
01299
01300
01301
01302
01303
01304
01305
01306
01307
01308
01309
01310
01311
01312
01313
01314
01315
01316
01317
01318
01319
01320
01321
01322
01323
01324
01325
01326
01327
01328
01329
01330
01331
01332
01333
01334
01335
01336
01337
01338
01339
01340
01341
01342
01343
01344
01345
01346
01347
01348
01349
01350
01351
01352
01353
01354
01355
01356
01357
01358
01359
01360
01361
01362
01363
01364
01365
01366
01367
01368
01369
01370
01371
01372
01373
01374
01375
01376
01377
01378
01379
01380
01381
01382
01383
01384
01385
01386
01387
01388
01389
01390
01391
01392
01393
01394
01395
01396
01397
01398
01399
01400
01401
01402
01403
01404
01405
01406
01407
01408
01409
01410
01411
01412
01413
01414
01415
01416
01417
01418
01419
01420
01421
01422
01423
01424
01425
01426
01427
01428
01429
01430
01431
01432
01433
01434
01435
01436
01437
01438
01439
01440
01441
01442
01443
01444
01445
01446
01447
01448
01449
01450
01451
01452
01453
01454
01455
01456
01457
01458
01459
01460
01461
01462
01463
01464
01465
01466
01467
01468
01469
01470
01471
01472
01473
01474
01475
01476
01477
01478
01479
01480
01481
01482
01483
01484
01485
01486
01487
01488
01489
01490
01491
01492
01493
01494
01495
01496
01497
01498
01499
01500
01501
01502
01503
01504
01505
01506
01507
01508
01509
01510
01511
01512
01513
01514
01515
01516
01517
01518
01519
01520
01521
01522
01523
01524
01525
01526
01527
01528
01529
01530
01531
01532
01533
01534
01535
01536
01537
01538
01539
01540
01541
01542
01543
01544
01545
01546
01547
01548
01549
01550
01551
01552
01553
01554
01555
01556
01557
01558
01559
01560
01561
01562
01563
01564
01565
01566
01567
01568
01569
01570
01571
01572
01573
01574
01575
01576
01577
01578
01579
01580
01581
01582
01583
01584
01585
01586
01587
01588
01589
01590
01591
01592
01593
01594
01595
01596
01597
01598
01599
01600
01601
01602
01603
01604
01605
01606
01607
01608
01609
01610
01611
01612
01613
01614
01615
01616
01617
01618
01619
01620
01621
01622
01623
01624
01625
01626
01627
01628
01629
01630
01631
01632
01633
01634
01635
01636
01637
01638
01639
01640
01641
01642
01643
01644
01645
01646
01647
01648
01649
01650
01651
01652
01653
01654
01655
01656
01657
01658
01659
01660
01661
01662
01663
01664
01665
01666
01667
01668
01669
01670
01671
01672
01673
01674
01675
01676
01677
01678
01679
01680
01681
01682
01683
01684
01685
01686
01687
01688
01689
01690
01691
01692
01693
01694
01695
01696
01697
01698
01699
01700
01701
01702
01703
01704
01705
01706
01707
01708
01709
01710
01711
01712
01713
01714
01715
01716
01717
01718
01719
01720
01721
01722
01723
01724
01725
01726
01727
01728
01729
01730
01731
01732
01733
01734
01735
01736
01737
01738
01739
01740
01741
01742
01743
01744
01745
01746
01747
01748
01749
01750
01751
01752
01753
01754
01755
01756
01757
01758
01759
01760
01761
01762
01763
01764
01765
01766
01767
01768
01769
01770
01771
01772
01773
01774
01775
01776
01777
01778
01779
01780
01781
01782
01783
01784
01785
01786
01787
01788
01789
01790
01791
01792
01793
01794
01795
01796
01797
01798
01799
01800
01801
01802
01803
01804
01805
01806
01807
01808
01809
01810
01811
01812
01813
01814
01815
01816
01817
01818
01819
01820
01821
01822
01823
01824
01825
01826
01827
01828
01829
01830
01831
01832
01833
01834
01835
01836
01837
01838
01839
01840
01841
01842
01843
01844
01845
01846
01847
01848
01849
01850
01851
01852
01853
01854
01855
01856
01857
01858
01859
01860
01861
01862
01863
01864
01865
01866
01867
01868
01869
01870
01871
01872
01873
01874
01875
01876
01877
01878
01879
01880
01881
01882
01883
01884
01885
01886
01887
01888
01889
01890
01891
01892
01893
01894
01895
01896
01897
01898
01899
01900
01901
01902
01903
01904
01905
01906
01907
01908
01909
01910
01911
01912
01913
01914
01915
01916
01917
01918
01919
01920
01921
01922
01923
01924
01925
01926
01927
01928
01929
01930
01931
01932
01933
01934
01935
01936
01937
01938
01939
01940
01941
01942
01943
01944
01945
01946
01947
01948
01949
01950
01951
01952
01953
01954
01955
01956
01957
01958
01959
01960
01961
01962
01963
01964
01965
01966
01967
01968
01969
01970
01971
01972
01973
01974
01975
01976
01977
01978
01979
01980
01981
01982
01983
01984
01985
01986
01987
01988
01989
01990
01991
01992
01993
01994
01995
01
```

```

00313             Edge newEdge = inputPort.ConnectTo(outputPort);
00314             inputPort.Connect(newEdge);
00315             outputPort.Connect(newEdge);
00316
00317             AddElement(newEdge);
00318         }
00319     }
00320 }
00321 }
00322 }
00323
00324 private void ExtractConfigs()
00325 {
00326     _elementConfigs.Clear();
00327     List<ElementConfig> configs = _evolutionsData.SelectMany(x => x.Chain).ToList();
00328
00329     for (int i = 0; i < configs.Count; i++)
00330     {
00331         var config = configs[i];
00332         if (config)
00333         {
00334             _elementConfigs.Add(config);
00335         }
00336     }
00337 }
00338
00339 private string OnSerializeGraphElements(IEnumerable<GraphElement> elements)
00340 {
00341     var selectedEvolutions = elements.OfType<IEvolutionNode>();
00342     var evolutionsToSave = new List<EvolutionNodeData>();
00343
00344     foreach (IEvolutionNode selectedEvolution in selectedEvolutions)
00345     {
00346         selectedEvolution.SaveData();
00347         evolutionsToSave.Add(selectedEvolution.Data.Copy<EvolutionNodeData>());
00348     }
00349
00350     var saveData = new SerializedElementNodes
00351     {
00352         EvolutionNodesData = evolutionsToSave,
00353     };
00354
00355     string json = JsonUtility.ToJson(saveData, false);
00356
00357     return json;
00358 }
00359
00360 private bool OnCanPasteSerializedData(string data)
00361 {
00362     try
00363     {
00364         JsonUtility.FromJson<SerializedElementNodes>(data);
00365
00366         return true;
00367     }
00368     catch (Exception)
00369     {
00370         return false;
00371     }
00372 }
00373
00374 private void OnUnserializeAndPaste(string operationName, string copyPasteData)
00375 {
00376     var deserializedData = JsonUtility.FromJson<SerializedElementNodes>(copyPasteData);
00377
00378     if (deserializedData != null)
00379     {
00380         if (deserializedData.EvolutionNodesData != null)
00381         {
00382             CreateEvolutionsNodes(deserializedData.EvolutionNodesData, CreateEvolutionNode,
50f);
00383         }
00384     }
00385 }
00386
00387 private void OnDeleteSelection(string operationName, AskUser askUser)
00388 {
00389     var selectedNodes = selection.OfType<Node>().ToList();
00390     var selectedEdges = selection.OfType<Edge>().ToList();
00391
00392     DeleteElements(selectedNodes, selectedEdges);
00393 }
00394
00395 private void DeleteElements(List<Node> nodesToDelete, List<Edge> edgesToDelete)
00396 {
00397     if (nodesToDelete.Count > 0)
00398     {

```

```

00399         var allEdges = edges.ToList();
00400         for (int i = nodesToDelete.Count - 1; i >= 0; i--)
00401     {
00402         Node node = nodesToDelete[i];
00403
00404         var nodeEdgesToDelete = allEdges
00405             .Select(x => x)
00406             .Where(x => x.input.node == node || x.output.node == node)
00407             .ToList();
00408
00409         for (int j = nodeEdgesToDelete.Count - 1; j >= 0; j--)
00410     {
00411         var edge = nodeEdgesToDelete[j];
00412         edge.input.Disconnect(edge);
00413         edge.output.Disconnect(edge);
00414
00415         RemoveElement(edge);
00416     }
00417
00418         if (node is BaseNode baseNode)
00419     {
00420         baseNode.Dispose();
00421     }
00422
00423         RemoveElement(node);
00424     }
00425 }
00426
00427 if (edgesToDelete.Count > 0)
00428 {
00429     for (int i = edgesToDelete.Count - 1; i >= 0; i--)
00430     {
00431         var edge = edgesToDelete[i];
00432         var input = edge.input;
00433         var output = edge.output;
00434
00435         input.Disconnect(edge);
00436         output.Disconnect(edge);
00437
00438         RemoveElement(edge);
00439     }
00440 }
00441 }
00442
00443     private void CreateEvolutionsNodes(List<EvolutionNodeData> elements, Action<EvolutionNodeData>
00444     createCallback, float positionOffset = 0f)
00445     {
00446         foreach (EvolutionNodeData noteData in elements)
00447     {
00448         EvolutionNodeData newData = noteData;
00449         newData.Position =
00450             new Rect(newData.Position.position + new Vector2(positionOffset, positionOffset),
00451             newData.Position.size);
00452
00453         createCallback(newData);
00454     }
00455
00456     private void CreateNodes(List<ElementNodeData> elements, Vector2 positionOffset = default)
00457     {
00458         for (int i = 0; i < elements.Count; i++)
00459     {
00460         var nodeData = elements[i];
00461
00462         CreateNode(nodeData, positionOffset * (i + 1));
00463     }
00464
00465     private void CreateNode(ElementNodeData data, Vector2 positionOffset = default)
00466     {
00467         var newNode = new ElementNode(data, positionOffset, this);
00468
00469         AddElement(newNode);
00470         AddToSelection(newNode);
00471     }
00472
00473     public override void HandleEvent(EventBase evt)
00474     {
00475         base.HandleEvent(evt);
00476
00477         if (_currentEvent is DragPerformEvent)
00478     {
00479         _currentEvent = evt;
00480         return;
00481     }
00482
00483         _currentEvent = evt;

```

```

00484         switch (evt)
00485     {
00486         case DragUpdatedEvent _:
00487             var dragDrop = DragAndDrop.objectReferences;
00488             if (dragDrop.All(x => x is ElementConfig))
00489             {
00490                 DragAndDrop.visualMode = DragAndDropVisualMode.Copy;
00491             }
00492         }
00493     }
00494     break;
00495 
00496     case DragPerformEvent dragPerformEvent:
00497 
00498         var type = dragPerformEvent.target.GetType();
00499         var typeName = type.Name;
00500 
00501         if (typeName == "ObjectFieldDisplay")
00502         {
00503             return;
00504         }
00505 
00506         var mousePosition = GetLocal.mousePosition(dragPerformEvent.localPosition);
00507 
00508         var rect = new Rect(mousePosition.x, mousePosition.y, 100, 100);
00509         var dragDropObjects = DragAndDrop.objectReferences;
00510         if (dragDropObjects.All(x => x is ElementConfig))
00511         {
00512             List<ElementNodeData> elementNodesData =
00513                 dragDropObjects.Where(x => x is ElementConfig).Select(x =>
00514                 {
00515                     var config = x as ElementConfig;
00516                     var nodeData = new ElementNodeData
00517                     {
00518                         ElementConfig = config,
00519                         Position = rect
00520                     };
00521 
00522                     return nodeData;
00523                 }).ToList();
00524 
00525             if (elementNodesData.Count > 0)
00526             {
00527                 CreateNodes(elementNodesData, Constants.NodeCreationOffset);
00528             }
00529         }
00530     }
00531     break;
00532 }
00533 }
00534 
00535 public void ClearNodes()
00536 {
00537     foreach (Node node in nodes)
00538     {
00539         if (node is BaseNode saveNode)
00540         {
00541             saveNode.Dispose();
00542         }
00543     }
00544 }
00545 }
00546 }
```

7.12 EvolutionsGraphViewTraits.cs

```

00001 // Copyright (c) 2024, Awessets
00002 
00003 using UnityEditor.Experimental.GraphView;
00004 using UnityEngine.UIElements;
00005 
00006 namespace MergeIt.Editor.EvolutionsEditor
00007 {
00008     public class EvolutionsGraphViewTraits : GraphView.UxmlTraits
00009     {
00010         private readonly UxmlStringAttributeDescription _nameAttribute = new()
00011             {name = "name"};
00012 
00013         public override void Init(VisualElement ve, IUIAttributes bag, CreationContext cc)
00014         {
00015             base.Init(ve, bag, cc);
00016             if (ve is EvolutionsGraphView evolutionsGraphView)
```

```

00018         {
00019             evolutionsGraphView.name = _nameAttribute.GetValueFromBag(bag, cc);
00020         }
00021     }
00022 }
00023 }
```

7.13 EvolutionsEditorWindow.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.IO;
00005 using MergeIt.Core.Schemes;
00006 using UnityEditor;
00007 using UnityEditor.Experimental.GraphView;
00008 using UnityEditor.UIElements;
00009 using UnityEngine;
00010 using UnityEngine.UIElements;
00011
00012 namespace MergeIt.Editor.EvolutionsEditor
00013 {
00014     public class EvolutionsEditorWindow : EditorWindow
00015     {
00016         private EvolutionsGraphView _graphView;
00017
00018         private StyleSheet _windowStyle;
00019         private VisualTreeAsset _windowTree;
00020         private bool _initialized;
00021         private SchemeObject _currentScheme;
00022
00023         public bool Initialized
00024         {
00025             get => _initialized;
00026         }
00027
00028         [MenuItem("Window/Merge Toolkit/Evolutions", false, 1)]
00029         public static void ShowWindow()
00030         {
00031             var window = GetWindow<EvolutionsEditorWindow>(true, "Evolutions Editor");
00032
00033             if (!window.Initialized)
00034             {
00035                 window.Initialize();
00036             }
00037
00038             window.Show();
00039         }
00040
00041         public void Initialize()
00042         {
00043             if (!_initialized)
00044             {
00045                 _windowStyle =
00046
00047                 (StyleSheet)EditorGUIUtility.Load(Path.Combine(Constants.EvolutionsEditorResourcesRoot,
00048                                         "EvolutionsEditorStyle.uss"));
00049
00050                 _windowTree =
00051
00052                 (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.EvolutionsEditorResourcesRoot,
00053                                         "EvolutionsEditorWindow.uxml"));
00054
00055                 VisualElement root = rootVisualElement;
00056                 VisualElement windowRoot = _windowTree.CloneTree();
00057
00058                 windowRoot.AddToClassList("root");
00059
00060                 var toolbarMenu = windowRoot.Q<ToolbarMenu>("FileMenu");
00061                 toolbarMenu.menu.AppendAction("Open...", _ => OnLoadButtonClicked());
00062                 toolbarMenu.menu.AppendSeparator();
00063                 toolbarMenu.menu.AppendAction("Save", _ => OnSaveButtonClicked(), _ =>
00064                     CanSaveAndExport());
00065                 toolbarMenu.menu.AppendAction("Save As...", _ => OnSaveAsButtonClicked());
00066
00067                 _graphView = windowRoot.Q<EvolutionsGraphView>("EvoGraph");
00068                 _graphView.AddManipulator(new ContentDragger());
00069                 _graphView.AddManipulator(new SelectionDragger());
00070                 _graphView.AddManipulator(new RectangleSelector());
00071                 _graphView.AddManipulator(new ClickSelector());
00072
00073                 _graphView.SetupZoom(0.1f, 3f);
00074
00075                 root.Add(windowRoot);
00076             }
00077         }
00078     }
00079 }
```

```
00072         root.styleSheets.Add(_windowStyle);
00073
00074         _initialized = true;
00075     }
00076 }
00077
00078 public void OnEnable()
00079 {
00080     Initialize();
00081 }
00082
00083 public void OnDisable()
00084 {
00085     _initialized = false;
00086 }
00087
00088 public void OnFocus()
00089 {
00090     _graphView?.Validate();
00091 }
00092
00093 public void OnDestroy()
00094 {
00095     _graphView.ClearNodes();
00096 }
00097
00098 public void Reload(SchemeObject scheme)
00099 {
00100     _currentScheme = scheme;
00101
00102     SchemeData data = scheme.SchemeData;
00103
00104     _graphView.Restore(data);
00105     _graphView.ClearSelection();
00106 }
00107
00108 private void OnLoadButtonClicked()
00109 {
00110     var path = EditorUtility.OpenFilePanelWithFilters("Load scheme", "Assets", new[] {"Saved
schemes", "asset"});
00111     if (path.Length != 0)
00112     {
00113         try
00114         {
00115             string relativePath = null;
00116             if (path.StartsWith(Application.dataPath))
00117             {
00118                 relativePath = "Assets" + path.Substring(Application.dataPath.Length);
00119             }
00120
00121             var currentScheme = AssetDatabase.LoadAssetAtPath<SchemeObject>(relativePath);
00122             Reload(currentScheme);
00123         }
00124         catch (Exception e)
00125         {
00126             Debug.Log($" {e} ");
00127         }
00128     }
00129 }
00130
00131 private void OnSaveAsButtonClicked()
00132 {
00133     SchemeData schemeData = _graphView.Save();
00134
00135     if (schemeData != null)
00136     {
00137         var path = EditorUtility.SaveFilePanelInProject("Save scheme", "New_scheme", "asset",
"Save scheme to file");
00138         if (path.Length != 0)
00139         {
00140             var existedScheme = AssetDatabase.LoadAssetAtPath<SchemeObject>(path);
00141
00142             if (existedScheme != null)
00143             {
00144                 existedScheme.SchemeData = schemeData;
00145                 EditorUtility.SetDirty(existedScheme);
00146
00147                 _currentScheme = existedScheme;
00148             }
00149             else
00150             {
00151                 var newScheme = CreateInstance<SchemeObject>();
00152                 newScheme.SchemeData = schemeData;
00153
00154                 _currentScheme = newScheme;
00155
00156                 AssetDatabase.CreateAsset(newScheme, path);
00157             }
00158         }
00159     }
00160 }
```

```

00157             AssetDatabase.SaveAssets();
00158         }
00159     }
00160     Reload(_currentScheme);
00161   }
00162 }
00163 else
00164 {
00165     Debug.LogWarning("Can't save scheme because of errors.");
00166 }
00167 }
00168
00169 private void OnSaveButtonClicked()
00170 {
00171     SchemeData schemeData = _graphView.Save();
00172
00173     if (schemeData != null)
00174     {
00175         _currentScheme.SchemeData = schemeData;
00176         EditorUtility.SetDirty(_currentScheme);
00177         AssetDatabase.SaveAssets();
00178
00179         Reload(_currentScheme);
00180     }
00181 else
00182 {
00183     Debug.LogWarning("Can't save scheme because of errors.");
00184 }
00185 }
00186
00187 private DropdownMenuAction.Status CanSaveAndExport()
00188 {
00189     if (_currentScheme)
00190     {
00191         return DropdownMenuAction.Status.Normal;
00192     }
00193
00194     return DropdownMenuAction.Status.Disabled;
00195 }
00196 }
00197 }

```

7.14 BaseNode.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Linq;
00005 using UnityEditor.Experimental.GraphView;
00006
00007 namespace MergeIt.Editor.EvolutionsEditor
00008 {
00009     public abstract class BaseNode : Node, ISaveNode, IDisposable
00010     {
00011         public abstract string Id { get; }
00012         public abstract void SaveData();
00013
00014         public Port GetPort(string portName)
00015         {
00016             var port = inputContainer.Children()
00017                 .OfType<Port>()
00018                 .FirstOrDefault(x => x.portName.Equals(portName));
00019
00020             if (port != null)
00021             {
00022                 return port;
00023             }
00024
00025             port = outputContainer.Children()
00026                 .OfType<Port>()
00027                 .FirstOrDefault(x => x.portName.Equals(portName));
00028
00029             return port;
00030         }
00031
00032         public virtual void Dispose()
00033         {
00034         }
00035     }
00036 }

```

7.15 SerializedElementNodes.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using MergeIt.Core.Schemes.Data;
00006
00007 namespace MergeIt.Editor.EvolutionsEditor
00008 {
00009     [Serializable]
00010     public class SerializedElementNodes
00011     {
00012         public List<EvolutionNodeData> EvolutionNodesData;
00013     }
00014 }
```

7.16 ElementNode.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.IO;
00005 using MergeIt.Core.Configs.Elements;
00006 using MergeIt.Core.Configs.Types;
00007 using MergeIt.Core.Schemes.Data;
00008 using MergeIt.Editor.Configs;
00009 using MergeIt.Editor.Configs.ItemsDrawers;
00010 using UnityEditor;
00011 using UnityEditor.Experimental.GraphView;
00012 using UnityEditor.UIElements;
00013 using UnityEngine;
00014 using UnityEngine.UIElements;
00015 using VisualElement = UnityEngine.UIElements.VisualElement;
00016
00017 namespace MergeIt.Editor.EvolutionsEditor
00018 {
00019     public class ElementNode : BaseNode, IElementNode
00020     {
00021         private readonly EvolutionsGraphView _graphView;
00022         private readonly VisualElement _root;
00023         private readonly ElementConfig _config;
00024         private readonly string _configName;
00025
00026         private CommonSettingsDrawer _commonSettingsDrawer;
00027         private GeneratorSettingsEvolutionDrawer _generatorSettingsDrawer;
00028
00029         public override string Id
00030         {
00031             get => Data?.Id;
00032         }
00033
00034         public IElementNodeData Data { get; private set; }
00035
00036         public ElementConfig Config
00037         {
00038             get => _config;
00039         }
00040
00041         public ElementNode(ElementNodeData data, Vector2 positionOffset, EvolutionsGraphView
00042                         graphView)
00042         {
00043             _graphView = graphView;
00044             Data = data;
00045             _config = data.ElementConfig;
00046
00047             if (_config)
00048             {
00049                 var style1 =
00050                     (StyleSheet)EditorGUIUtility.Load(Path.Combine(Constants.EvolutionsEditorResourcesRoot,
00050                                         "GeneratorStyle.uss"));
00051
00052                 var titleLabel = titleContainer.Q<Label>("title-label");
00053                 titleLabel.bindingPath = "_commonSettings._name";
00054                 Port evolvedFromPort = InstantiatePort(Orientation.Horizontal, Direction.Input,
00054                     Port.Capacity.Single,
00055                     typeof(string));
00056
00057                 evolvedFromPort.portName = Constants.EvolvedFromPort;
00058                 inputContainer.Add(evolvedFromPort);
00059
00060                 Port evolvePort = InstantiatePort(Orientation.Horizontal, Direction.Output,
00060                     Port.Capacity.Single,
```

```

00061             typeof(string));
00062             evolvePort.portName = Constants.EvolvePort;
00063             outputContainer.Add(evolvePort);
00064
00065             _root = new VisualElement();
00066
00067             mainContainer.AddToClassList(StylesConstants.NodeStyles[ElementType.Regular]);
00068             mainContainer.AddToClassList(StylesConstants.NodeStyles[ElementType.Generator]);
00069
00070             var typeEnum = new EnumField(_config.Type);
00071             typeEnum.bindingPath = "_type";
00072             typeEnum.RegisterValueChangedCallback(OnElementTypeChanged);
00073
00074             _root.Add(typeEnum);
00075
00076             Refresh(_config.Type);
00077
00078             mainContainer.Add(_root);
00079
00080             Data.Position = new Rect(
00081                 data.Position.x + positionOffset.x,
00082                 data.Position.y + positionOffset.y,
00083                 data.Position.width,
00084                 data.Position.height);
00085
00086             mainContainer.Bind(new SerializedObject(_config));
00087
00088             _configName = _config.name;
00089             _root.styleSheets.Add(style1);
00090         }
00091     else
00092     {
00093         mainContainer.AddToClassList(StylesConstants.InvalidNode);
00094         title = "Config was missing";
00095     }
00096
00097     SetPosition(Data.Position);
00098 }
00099
00100     public override void SaveData()
00101     {
00102         Data.Position = GetPosition();
00103     }
00104
00105     public override void BuildContextualMenu(ContextualMenuPopulateEvent evt)
00106     {
00107         base.BuildContextualMenu(evt);
00108
00109         evt.menu.AppendAction("Select in Project View", SelectAction);
00110         evt.menu.AppendSeparator();
00111     }
00112
00113     public override void Dispose()
00114     {
00115         base.Dispose();
00116
00117         _generatorSettingsDrawer?.Dispose();
00118         _commonSettingsDrawer?.Dispose();
00119
00120         mainContainer.Unbind();
00121     }
00122
00123     public bool Validate()
00124     {
00125         if (Data.ElementConfig)
00126         {
00127             return true;
00128         }
00129
00130         title = $"Lost config: {_configName}";
00131         mainContainer.AddToClassList(StylesConstants.InvalidNode);
00132         mainContainer.EnableInClassList(StylesConstants.NodeStyles[ElementType.Regular], false);
00133         mainContainer.EnableInClassList(StylesConstants.NodeStyles[ElementType.Generator], false);
00134
00135         mainContainer.SetEnabled(false);
00136         return false;
00137     }
00138
00139     private void OnElementTypeChanged(ChangeEvent<Enum> changeEvent)
00140     {
00141         if (changeEvent.newValue == null)
00142         {
00143             return;
00144         }
00145
00146         if (Enum.TryParse(changeEvent.newValue.ToString(), out ElementType type))
00147         {

```

```

00148         Debug.Log($"Value changed to ({type}), Target type now is {_config.Type}");
00149
00150         Refresh(type);
00151     }
00152 }
00153
00154 private void Refresh(ElementType newType)
00155 {
00156     if (_commonSettingsDrawer == null)
00157     {
00158         _commonSettingsDrawer = new CommonSettingsDrawer(_root, _config);
00159         _commonSettingsDrawer.Draw();
00160     }
00161
00162     switch (newType)
00163     {
00164         case ElementType.Regular:
00165             mainContainer.EnableInclassList(StylesConstants.NodeStyles[ElementType.Regular],
00166             true);
00167             mainContainer.EnableInclassList(StylesConstants.NodeStyles[ElementType.Generator],
00168             false);
00169             _generatorSettingsDrawer?.Hide();
00170             break;
00171
00172         case ElementType.Generator:
00173             mainContainer.EnableInclassList(StylesConstants.NodeStyles[ElementType.Regular],
00174             false);
00175             mainContainer.EnableInclassList(StylesConstants.NodeStyles[ElementType.Generator],
00176             true);
00177             if (_generatorSettingsDrawer == null)
00178             {
00179                 _generatorSettingsDrawer = new GeneratorSettingsEvolutionDrawer(_root,
00180                 _config, _graphView);
00181                 _generatorSettingsDrawer.Draw();
00182             }
00183             _generatorSettingsDrawer.Show();
00184             break;
00185     }
00186
00187     private void SelectAction(DropdownMenuAction dropdownMenuAction)
00188     {
00189         if (_config != null)
00190         {
00191             EditorGUIUtility.PingObject (_config);
00192         }
00193     }
00194 }
00195 }

```

7.17 EvolutionNode.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.IO;
00004 using MergeIt.Core.Schemes.Data;
00005 using UnityEditor;
00006 using UnityEditor.Experimental.GraphView;
00007 using UnityEngine.UIElements;
00008
00009 namespace MergeIt.Editor.EvolutionsEditor
00010 {
00011     public class EvolutionNode : BaseNode, IEvolutionNode
00012     {
00013         public string Name { get; }
00014         public string Description { get; }
00015         public bool Discovered { get; }
00016
00017         public override string Id
00018         {
00019             get => Data?.Id;
00020         }
00021
00022         public IEvolutionNodeData Data { get; }
00023
00024         private readonly Port _evolutionRootPort;
00025         private TextField _nameField;
00026         private TextField _descField;
00027         private Toggle _discoveredToggle;

```

```

00028
00029     public EvolutionNode(EvolutionNodeData data)
00030     {
00031         Data = data;
00032
00033         var visualTreeAsset =
00034     (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.EvolutionsEditorResourcesRoot,
00035     "EvolutionNode.uxml"));
00036
00037         mainContainer.AddToClassList("nodeBorderEvolution");
00038
00039         var root = visualTreeAsset.CloneTree();
00040         _nameField = root.Q<TextField>("Name");
00041         _descField = root.Q<TextField>("Description");
00042         _discoveredToggle = root.Q<Toggle>("Discovered");
00043
00044         title = Data.Name;
00045         _nameField.SetValueWithoutNotify(Data.Name);
00046         _descField.SetValueWithoutNotify(Data.Description);
00047         _discoveredToggle.SetValueWithoutNotify(Data.Discovered);
00048
00049         _nameField.RegisterValueChangedCallback(OnEvoNameTextChanged);
00050         _descField.RegisterValueChangedCallback(OnEvoDescriptionTextChanged);
00051         _discoveredToggle.RegisterValueChangedCallback(OnDiscoveredSelected);
00052
00053         mainContainer.Add(root);
00054
00055         topContainer.Remove(inputContainer);
00056         _evolutionRootPort = InstantiatePort(Orientation.Horizontal, Direction.Output,
00057         Port.Capacity.Single, typeof(string));
00058         _evolutionRootPort.portName = Constants.EvolutionRootPort;
00059         outputContainer.Add(_evolutionRootPort);
00060
00061     private void OnEvoNameTextChanged(ChangeEvent<string> evt)
00062     {
00063         Data.Name = evt.newValue;
00064         title = evt.newValue;
00065     }
00066
00067     private void OnEvoDescriptionTextChanged(ChangeEvent<string> evt)
00068     {
00069         Data.Description = evt.newValue;
00070     }
00071
00072     private void OnDiscoveredSelected(ChangeEvent<bool> evt)
00073     {
00074         Data.Discovered = evt.newValue;
00075     }
00076
00077     public override void SaveData()
00078     {
00079         Data.Position = GetPosition();
00080     }
00081
00082     public override void Dispose()
00083     {
00084         base.Dispose();
00085
00086         _nameField.UnregisterValueChangedCallback(OnEvoNameTextChanged);
00087         _descField.UnregisterValueChangedCallback(OnEvoDescriptionTextChanged);
00088         _discoveredToggle.UnregisterValueChangedCallback(OnDiscoveredSelected);
00089     }
00090 }
00091 }
```

7.18 IElementNode.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.Schemes.Data;
00005
00006 namespace MergeIt.Editor.EvolutionsEditor
00007 {
00008     public interface IElementNode : ISaveNode
00009     {
00010         ElementConfig Config { get; }
00011
00012         IElementNodeData Data { get; }
00013 }
```

```
00014         bool Validate();
00015     }
00016 }
```

7.19 IEvolutionNode.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Schemes.Data;
00004
00005 namespace MergeIt.Editor.EvolutionsEditor
00006 {
00007     public interface IEvolutionNode : ISaveNode
00008     {
00009         string Name { get; }
00010         string Description { get; }
00011         bool Discovered { get; }
00012         IEvolutionNodeData Data { get; }
00013     }
00014 }
```

7.20 ISaveNode.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEditor.Experimental.GraphView;
00004
00005 namespace MergeIt.Editor.EvolutionsEditor
00006 {
00007     public interface ISaveNode
00008     {
00009         string Id { get; }
00010         void SaveData();
00011         Port GetPort(string portName);
00012     }
00013 }
```

7.21 SchemeObjectEditor.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Schemes;
00004 using UnityEditor;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Editor.EvolutionsEditor
00008 {
00009     [CustomEditor(typeof(SchemeObject))]
00010     public class SchemeObjectEditor : UnityEditor.Editor
00011     {
00012         public override void OnInspectorGUI()
00013         {
00014             if (GUILayout.Button("Open"))
00015             {
00016                 var window = EditorWindow.GetWindow<EvolutionsEditorWindow>(true, "Evolutions
Editor");
00017
00018                 if (!window.Initialized)
00019                 {
00020                     window.Initialize();
00021                 }
00022
00023                 window.Show();
00024                 window.Reload(target as SchemeObject);
00025             }
00026         }
00027     }
00028 }
```

7.22 GeneratedItemsCreator.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using System.Linq;
00006 using MergeIt.Core.Configs.Data;
00007 using MergeIt.Core.Configs.Elements;
00008 using MergeIt.Editor.Configs;
00009 using UnityEditor;
00010 using UnityEngine.UIElements;
00011 using VisualElement = UnityEngine.UIElements.VisualElement;
00012
00013 namespace MergeIt.Editor
00014 {
00015     public class GeneratedItemsCreator : IDisposable
00016     {
00017         private static Action<ElementConfig, int> _clickAddNewItem;
00018         private static Action<ElementConfig, int, GeneratableElement> _clickRemoveGeneratedItem;
00019         private static Action<ElementConfig> _clickClearAllGeneratedItems;
00020
00021         private readonly VisualElement _itemsContainer;
00022         private readonly ElementGeneratorSettings _settings;
00023         private readonly ElementConfig _config;
00024         private readonly List<EvolutionData> _evolutions;
00025         private readonly List<ElementConfig> _configs;
00026         private readonly List<GeneratedItemUI> _generatedItemUis;
00027         private readonly Button _addButton;
00028         private readonly Button _clearAllButton;
00029
00030         public GeneratedItemsCreator(VisualElement genItemsControls, VisualElement genItemsContainer,
00031             ElementConfig config,
00032             List<EvolutionData> evolutions, List<ElementConfig> configs)
00033         {
00034             _config = config;
00035             _evolutions = evolutions;
00036             _configs = configs;
00037             _itemsContainer = genItemsContainer;
00038             _settings = config.GeneratorSettings;
00039             _generatedItemUis = new List<GeneratedItemUI>();
00040
00041             _addButton = genItemsControls.Q<Button>("CreateGenerateItem");
00042             _addButton.clicked += OnClickAddNewItem;
00043
00044             _clearAllButton = genItemsControls.Q<Button>("ClearGenerateItem");
00045             _clearAllButton.clicked += OnClickClearAllGeneratedItems;
00046
00047             _clickAddNewItem += CreateNewGeneratedElementUI;
00048             _clickRemoveGeneratedItem += RemoveGeneratedItemUI;
00049             _clickClearAllGeneratedItems += ClearAllGeneratedItems;
00050
00051             if (_configs.Count == 0)
00052             {
00053                 var helpBox = new HelpBox("There are no saved evolutions.",
00054                     HelpBoxMessageType.Warning);
00055
00056                 _itemsContainer.Add(helpBox);
00057
00058                 _addButton.style.display = StylesConstants.DisplayNone;
00059                 _clearAllButton.style.display = StylesConstants.DisplayNone;
00060             }
00061         }
00062
00063         public void Generate()
00064         {
00065             var generateItems = _settings.GenerateItems;
00066             if (generateItems?.Count > 0)
00067             {
00068                 for (int i = 0; i < generateItems.Count; i++)
00069                 {
00070                     CreateGeneratedElementUI(i, generateItems[i]);
00071                 }
00072             }
00073         }
00074
00075         private void OnClickAddNewItem()
00076         {
00077             var newItem = new GeneratableElement();
00078
00079             _settings.GenerateItems.Add(newItem);
00080             EditorUtility.SetDirty(_config);
00081
00082             _clickAddNewItem?.Invoke(_config, _settings.GenerateItems.Count - 1);
00083         }
00084
00085         private void OnClickRemoveGeneratedItem(int index, GeneratableElement element)

```

```
00085      {
00086          _settings.GenerateItems.Remove(element);
00087          EditorUtility.SetDirty(_config);
00088      }
00089  }
00090
00091  private void OnClickClearAllGeneratedItems()
00092  {
00093      _settings.GenerateItems?.Clear();
00094      _clickClearAllGeneratedItems?.Invoke(_config);
00095  }
00096
00097
00098  private void CreateGeneratedElementUI(int index, GeneratableElement generatableElement)
00099  {
00100      if (_configs.Count == 1 &&
00101          _configs.First() == _config)
00102      {
00103          var helpBox = new HelpBox($"Evolutions have only this element {_config.name}",
00104              HelpBoxMessageType.Warning);
00105
00106          _itemsContainer.Add(helpBox);
00107          return;
00108      }
00109
00110      if (generatableElement.Element && !_configs.Contains(generatableElement.Element))
00111      {
00112          var helpBox = new HelpBox($"Add {generatableElement.Element.name} to the current
evolution scheme and save it.",
00113              HelpBoxMessageType.Warning);
00114
00115          _itemsContainer.Add(helpBox);
00116          return;
00117      }
00118
00119      var generatableUI = new GeneratedItemUI(generatableElement, _config, index, _evolutions,
00120      _configs);
00121      generatableUI.RemoveGeneratedItemEvent += OnClickRemoveGeneratedItem;
00122
00123      _generatedItemUis.Add(generatableUI);
00124      _itemsContainer.Add(generatableUI);
00125  }
00126
00127  private void CreateNewGeneratedElementUI(ElementConfig config, int index)
00128  {
00129      if (_config == config)
00130      {
00131          CreateGeneratedElementUI(index, _settings.GenerateItems.Last());
00132      }
00133  }
00134
00135  private void RemoveGeneratedItemUI(ElementConfig config, int index, GeneratableElement
element)
00136  {
00137      if (_config == config)
00138      {
00139          _generatedItemUis[index].RemoveGeneratedItemEvent -= OnClickRemoveGeneratedItem;
00140          _generatedItemUis.RemoveAt(index);
00141
00142          _itemsContainer.RemoveAt(index);
00143
00144          for (int i = 0; i < _generatedItemUis.Count; i++)
00145          {
00146              var genItem = _generatedItemUis[i];
00147              genItem.Index = i;
00148          }
00149      }
00150  }
00151
00152  private void ClearAllGeneratedItems(ElementConfig config)
00153  {
00154      if (_config == config)
00155      {
00156          ClearVisualItems();
00157      }
00158  }
00159
00160  private void ClearVisualItems()
00161  {
00162      for (int i = 0; i < _generatedItemUis.Count; i++)
00163      {
00164          _generatedItemUis[i].RemoveGeneratedItemEvent -= OnClickRemoveGeneratedItem;
00165
00166          _generatedItemUis.Clear();
00167          _itemsContainer.Clear();
00168      }
00169  }
```

```

00169     public void Clear()
00170     {
00171         ClearVisualItems();
00172
00173         _clearAllButton.clicked -= OnClickClearAllGeneratedItems;
00174         _addButton.clicked -= OnClickAddNewGeneratedItem;
00175
00176         _clickAddNewGeneratedItem -= CreateNewGeneratedElementUI;
00177         _clickRemoveGeneratedItem -= RemoveGeneratedItemUI;
00178         _clickClearAllGeneratedItems -= ClearAllGeneratedItems;
00179
00180     }
00181
00182     public void Dispose()
00183     {
00184         Clear();
00185     }
00186 }
00187 }
```

7.23 GeneratedItemUI.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using System.IO;
00006 using System.Linq;
00007 using MergeIt.Core.Configs.Data;
00008 using MergeIt.Core.Configs.Elements;
00009 using UnityEditor;
00010 using UnityEditor.UIElements;
00011 using UnityEngine;
00012 using UnityEngine.UIElements;
00013
00014 namespace MergeIt.Editor
00015 {
00016     public class GeneratedItemUI : VisualElement, IDisposable
00017     {
00018         private static Action<GeneratableElement, int> _changedChanceEvent;
00019         private static Action<GeneratableElement, int> _changedIndexEvent;
00020         private static Action<GeneratableElement, ElementConfig> _changedElementEvent;
00021
00022         public Action<int, GeneratableElement> RemoveGeneratedItemEvent { get; set; }
00023
00024         public int Index
00025         {
00026             set => _changedIndexEvent?.Invoke(_target, value);
00027         }
00028
00029         private readonly PopupField<ElementConfig> _elementsConfigPopup;
00030         private readonly ElementConfig _currentConfig;
00031         private readonly GeneratableElement _target;
00032         private readonly Label _indexLabel;
00033         private readonly IntegerField _chanceField;
00034         private readonly Button _removeButton;
00035
00036         private int _index;
00037         private ElementConfig _selectedElement;
00038         private List<ElementConfig> _allConfigs;
00039         private List<EvolutionData> _evolutions;
00040
00041         public GeneratedItemUI(GeneratableElement item,
00042             ElementConfig currentConfig,
00043             int index,
00044             List<EvolutionData> evolutions,
00045             List<ElementConfig> configs)
00046         {
00047             _currentConfig = currentConfig;
00048             _allConfigs = configs;
00049             _evolutions = evolutions;
00050
00051             _target = item;
00052             _index = index;
00053
00054             var template =
00055                 (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.InspectorResourcesRoot,
00056                                         "GeneratedItemTemplate.uxml"));
00057
00058             style.flexDirection = new StyleEnum<FlexDirection>(FlexDirection.Row);
00059             style.paddingTop = new StyleLength(5);
00060
00061             template.CloneTree(this);
00062         }
00063     }
00064 }
```

```
00061     if (item.Element == null && !_allConfigs.Contains(_selectedElement))
00062     {
00063         _selectedElement = _allConfigs.FirstOrDefault(x => x != _currentConfig);
00064         _target.Element = _selectedElement;
00065     }
00066     else
00067     {
00068         _selectedElement = item.Element;
00069     }
00070
00071     var container = this.Q<VisualElement>("ElementsPopup");
00072     if (_selectedElement != null)
00073     {
00074         _elementsConfigPopup = new PopupField<ElementConfig>("", _allConfigs,
00075         _selectedElement);
00076
00077         _elementsConfigPopup.formatListItemCallback = FormatElementCallback;
00078         _elementsConfigPopup.formatSelectedValueCallback = FormatElementCallback;
00079
00080         _elementsConfigPopup.AddToClassList("genItemList");
00081         _elementsConfigPopup.RegisterValueChangedCallback(OnElementChanged);
00082
00083         container.Add(_elementsConfigPopup);
00084
00085         EditorUtility.SetDirty(_currentConfig);
00086     }
00087     else
00088     {
00089         container.Add(new Label("Add more configs!"));
00090     }
00091
00092     _indexLabel = this.Q<Label>("ItemGenIndex");
00093     _indexLabel.text = $"{_index + 1}. ";
00094
00095     _chanceField = this.Q<IntegerField>("ItemChance");
00096     _removeButton = this.Q<Button>("RemoveButton");
00097
00098     _removeButton.clicked += OnRemoveClicked;
00099
00100     _chanceField.value = _target.Possibility;
00101     _chanceField.RegisterValueChangedCallback(OnChanceChanged);
00102
00103     _changedChanceEvent += ChanceChanged;
00104     _changedElementEvent += ElementChanged;
00105     _changedIndexEvent += IndexChanged;
00106 }
00107
00108 private void OnChanceChanged(ChangeEvent<int> evt)
00109 {
00110     _changedChanceEvent?.Invoke(_target, evt.newValue);
00111 }
00112
00113 private void OnRemoveClicked()
00114 {
00115     RemoveGeneratedItemEvent?.Invoke(_index, _target);
00116 }
00117
00118 private void OnElementChanged(ChangeEvent<ElementConfig> evt)
00119 {
00120     _changedElementEvent?.Invoke(_target, evt.newValue);
00121 }
00122
00123 private void IndexChanged(GeneratableElement target, int index)
00124 {
00125     if (target == _target)
00126     {
00127         _index = index;
00128         _indexLabel.text = (_index + 1).ToString();
00129     }
00130 }
00131
00132 private string FormatElementCallback(ElementConfig arg)
00133 {
00134     List<ElementConfig> chain =
00135         _evolutions
00136             .Where(x => x.Chain.Contains(arg))
00137             .Select(x => x.Chain)
00138             .FirstOrDefault();
00139
00140     int index = -1;
00141     if (chain != null)
00142     {
00143         index = chain.IndexOf(arg);
00144     }
00145
00146     string text = index != -1 ? $"{arg.CommonSettings.Name} level {index + 1}" :
```

```

00147     $"{{arg.CommonSettings.Name}}";
00148     }
00149
00150     private void ChanceChanged(GeneratableElement target, int chance)
00151     {
00152         if (target == _target)
00153         {
00154             _target.Possibility = chance;
00155             _chanceField.SetValueWithoutNotify(chance);
00156
00157             EditorUtility.SetDirty(_currentConfig);
00158         }
00159     }
00160
00161     private void ElementChanged(GeneratableElement target, ElementConfig elementConfig)
00162     {
00163         if (target == _target)
00164         {
00165             if (elementConfig == _currentConfig)
00166             {
00167                 EditorWindow.focusedWindow.ShowNotification(new GUIContent("The generator can't
00168                 generate itself."));
00169                 _selectedElement = _allConfigs.FirstOrDefault(x => x != _currentConfig);
00170             }
00171             else
00172             {
00173                 _selectedElement = elementConfig;
00174                 _target.Element = _selectedElement;
00175             }
00176
00177             _elementsConfigPopup.SetValueWithoutNotify(_selectedElement);
00178             EditorUtility.SetDirty(_currentConfig);
00179         }
00180
00181     public void Dispose()
00182     {
00183         _chanceField.UnregisterValueChangedCallback(OnChanceChanged);
00184         _removeButton.clicked -= OnRemoveClicked;
00185
00186         _changedChanceEvent -= ChanceChanged;
00187         _changedElementEvent -= ElementChanged;
00188         _changedIndexEvent -= IndexChanged;
00189     }
00190 }
00191 }
```

7.24 ActionCommandBase.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Editor.LevelEditor.Commands
00004 {
00005     public abstract class ActionCommandBase : IActionCommand
00006     {
00007         protected LevelEditorWindow Window;
00008
00009         public ActionCommandBase(LevelEditorWindow window)
00010         {
00011             Window = window;
00012         }
00013
00014         public abstract void Execute();
00015
00016         public abstract void Undo();
00017     }
00018 }
```

7.25 ActionCommandManager.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Editor.Core.LevelEditor.Commands;
00004
00005 namespace MergeIt.Editor.LevelEditor.Commands
00006 {
00007     public class ActionCommandManager : IActionCommandManager
00008     {
```

```

00009     private readonly LimitedStack<IActionCommand> _undoStack = new(10);
00010     private readonly LimitedStack<IActionCommand> _redoStack = new(10);
00011
00012     public void ExecuteCommand(IActionCommand command)
00013     {
00014         command.Execute();
00015         _undoStack.Push(command);
00016         _redoStack.Clear();
00017     }
00018
00019     public void Undo()
00020     {
00021         if (_undoStack.Any())
00022         {
00023             var command = _undoStack.Pop();
00024             command.Undo();
00025
00026             _redoStack.Push(command);
00027         }
00028     }
00029
00030     public void Redo()
00031     {
00032         if (_redoStack.Any())
00033         {
00034             var command = _redoStack.Pop();
00035             command.Execute();
00036
00037             _undoStack.Push(command);
00038         }
00039     }
00040 }
00041 }
```

7.26 ApplyCellCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Data;
00004
00005 namespace MergeIt.Editor.LevelEditor.Commands
00006 {
00007     public class ApplyCellCommand : ActionCommandBase
00008     {
00009         private readonly LevelElementData _newData;
00010         public ApplyCellCommand(LevelEditorWindow window, LevelElementData newData)
00011             : base(window)
00012         {
00013             _newData = newData;
00014         }
00015
00016         public override void Execute()
00017         {
00018             Window.ApplyCell(_newData, false);
00019         }
00020
00021         public override void Undo()
00022         {
00023             Window.UndoApplyCell(null, _newData);
00024         }
00025     }
00026 }
```

7.27 ChangeCellCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Data;
00004
00005 namespace MergeIt.Editor.LevelEditor.Commands
00006 {
00007     public class ChangeCellCommand : ActionCommandBase
00008     {
00009         private readonly LevelElementData _previousData;
00010         private readonly LevelElementData _newData;
00011
00012         public ChangeCellCommand(LevelEditorWindow window, LevelElementData previousData,
00013             LevelElementData newData)
00014             : base(window)
```

```

00014     {
00015         _previousData = previousData.GetClone();
00016         _newData = newData.GetClone();
00017     }
00018
00019     public override void Execute()
00020     {
00021         Window.ApplyCell(_newData, true);
00022     }
00023
00024     public override void Undo()
00025     {
00026         Window.UndoApplyCell(_previousData, _newData);
00027     }
00028 }
00029 }
```

7.28 ClearCellCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Data;
00004
00005 namespace MergeIt.Editor.LevelEditor.Commands
00006 {
00007     public class ClearCellCommand : ActionCommandBase
00008     {
00009         private readonly LevelElementData _previousData;
00010
00011         public ClearCellCommand(LevelEditorWindow window, LevelElementData previousData)
00012             : base(window)
00013         {
00014             _previousData = previousData;
00015         }
00016
00017         public override void Undo()
00018         {
00019             EvolutionSelection.UpdateCell(_previousData);
00020             Window.UndoClearCell(_previousData);
00021         }
00022
00023         public override void Execute()
00024         {
00025             Window.ClearCell(_previousData.Position);
00026         }
00027     }
00028 }
```

7.29 IActionCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Editor.LevelEditor.Commands
00004 {
00005     public interface IActionCommand
00006     {
00007         void Execute();
00008         void Undo();
00009     }
00010 }
```

7.30 IActionCommandManager.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Editor.LevelEditor.Commands
00004 {
00005     public interface IActionCommandManager
00006     {
00007         void ExecuteCommand(IActionCommand command);
00008         void Undo();
00009         void Redo();
00010     }
00011 }
```

7.31 LimitedStack.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using System.Linq;
00005
00006 namespace MergeIt.Editor.Core.LevelEditor.Commands
00007 {
00008     public class LimitedStack<T>
00009     {
00010         private readonly LinkedList<T> _list = new();
00011         private readonly int _maxSize;
00012
00013         public int Count => _list.Count;
00014
00015         public LimitedStack(int maxSize)
00016         {
00017             _maxSize = maxSize;
00018         }
00019
00020         public void Push(T item)
00021         {
00022             if (_list.Count >= _maxSize)
00023             {
00024                 _list.RemoveLast();
00025             }
00026             _list.AddFirst(item);
00027         }
00028
00029         public T Pop()
00030         {
00031             if (_list.Count == 0)
00032                 return default;
00033
00034             var value = _list.First.Value;
00035             _list.RemoveFirst();
00036             return value;
00037         }
00038
00039         public T Peek()
00040         {
00041             return _list.Count > 0 ? _list.First.Value : default;
00042         }
00043
00044         public bool Any()
00045         {
00046             return _list.Any();
00047         }
00048
00049         public void Clear()
00050         {
00051             _list.Clear();
00052         }
00053     }
00054 }
```

7.32 EvolutionSelection.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using System.Linq;
00005 using MergeIt.Core.Configs.Data;
00006 using MergeIt.Core.Configs.Elements;
00007 using MergeIt.Core.FieldElements;
00008 using UnityEngine;
00009
00010 namespace MergeIt.Editor.LevelEditor
00011 {
00012     public class EvolutionSelection
00013     {
00014         public static EvolutionData Evolution => _evolution;
00015         public static ElementConfig EvoItem => _elementData.Element;
00016         public static LevelElementData ElementData => _elementData;
00017
00018         private static LevelElementData _elementData;
00019         private static List<EvolutionData> _evolutions;
00020         private static EvolutionData _evolution;
00021
00022         public static void Init(List<EvolutionData> evolutions)
00023         {
00024             _evolutions = evolutions;
```

```

00025         var firstEvo = _evolutions.First();
00026         var firstEvoItem = firstEvo.Chain.First();
00027
00028         _evolution = firstEvo;
00029
00030         _elementData = new LevelElementData
00031     {
00032         EvolutionId = firstEvo.Id,
00033         Element = firstEvoItem,
00034     };
00035 }
00036 }
00037
00038     public static void SelectCell(int row, int column, LevelElementData itemData = null)
00039     {
00040         UpdateCell(itemData, GridPoint.Create(row, column));
00041     }
00042
00043     public static void LockCell(bool isLocked)
00044     {
00045         _elementData.IsBlocked = isLocked;
00046     }
00047
00048     public static void UpdateCell(LevelElementData itemData, GridPoint newPosition = default, bool
copyPosition = true)
00049     {
00050         if (itemData != null)
00051         {
00052             _elementData.CopyFrom(itemData, copyPosition);
00053             _evolution = _evolutions.FirstOrDefault(x => x.Id == itemData.EvolutionId);
00054         }
00055         else
00056         {
00057             _elementData.Position = newPosition;
00058         }
00059     }
00060
00061     public static void SelectEvolution(EvolutionData evolutionData)
00062     {
00063         _evolution = evolutionData;
00064
00065         _elementData.EvolutionId = _evolution.Id;
00066
00067         if (!_evolution.Chain.Contains(_elementData.Element))
00068         {
00069             _elementData.Element = _evolution.Chain.FirstOrDefault();
00070         }
00071     }
00072
00073     public static void SelectElement(ElementConfig evolutionElement)
00074     {
00075         var evolution = Evolution;
00076
00077         if (evolution.Chain.Contains(evolutionElement))
00078         {
00079             _elementData.Element = evolutionElement;
00080         }
00081         else
00082         {
00083             Debug.LogWarning($"Evolution doesn't contain item with id: ({evolutionElement})!");
00084         }
00085     }
00086
00087     public static void ClearSelection(bool force = false)
00088     {
00089         if (force)
00090         {
00091             _elementData = null;
00092         }
00093     }
00094 }
00095 }
```

7.33 LevelConfigEditor.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs;
00004 using UnityEditor;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Editor.LevelEditor
00008 {
```

```
00009     [CustomEditor(typeof(LevelConfig))]
0010     public class LevelConfigEditor : UnityEditor.Editor
0011     {
0012         public override void OnInspectorGUI()
0013         {
0014             if (GUILayout.Button("Open"))
0015             {
0016                 LevelEditorWindow.Show(target as LevelConfig);
0017             }
0018         }
0019     }
0020 }
```

7.34 LevelEditorWindow.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using System.IO;
00006 using System.Linq;
00007 using MergeIt.Core.Configs;
00008 using MergeIt.Core.Configs.Data;
00009 using MergeIt.Core.Configs.Elements;
00010 using MergeIt.Core.FieldElements;
00011 using MergeIt.Core.Schemes;
00012 using MergeIt.Editor.Configs;
00013 using MergeIt.Editor.LevelEditor.Commands;
00014 using UnityEditor;
00015 using UnityEditor.UIElements;
00016 using UnityEngine;
00017 using UnityEngine.UIElements;
00018 using Object = UnityEngine.Object;
00019 using PopupWindow = UnityEngine.UIElements.PopupWindow;
00020 using Task = System.Threading.Tasks.Task;
00021 using VisualElement = UnityEngine.UIElements.VisualElement;
00022
00023 namespace MergeIt.Editor.LevelEditor
00024 {
00025     public class LevelEditorWindow : EditorWindow
00026     {
00027         private static LevelEditorWindow _window;
00028
00029         private Button _saveButton;
00030         private Button _loadButton;
00031
00032         private VisualElement _grid;
00033         private VisualElement _itemFrameSettings;
00034         private VisualElement _itemFrameEvolutions;
00035         private StyleSheet _globalStyle;
00036         private ToolbarToggle[,] _toggles;
00037         private ToolbarToggle _selectedToggle;
00038         private GridPoint _selectedCell;
00039         private Toggle _isLockedToggle;
00040         private Button _applyButton;
00041         private Button _clearButton;
00042         private Image _itemImage;
00043         private Button _copyButton;
00044         private IntegerField _copyRow;
00045         private IntegerField _copyColumn;
00046
00047         private Button _createNewButton;
00048         private IntegerField _createFieldWidth;
00049         private IntegerField _createFieldHeight;
00050
00051         private Label _warnLabel;
00052
00053         private SchemeObject _loadedScheme;
00054         private PopupField<EvolutionData> _evoPopup;
00055         private PopupField<ElementConfig> _evolutionChainElementsPopup;
00056
00057         private Texture _lockTexture;
00058
00059         private int _fieldWidth = 0;
00060         private int _fieldHeight = 0;
00061
00062         private ObjectField _evolutionsField;
00063         private LevelConfig _levelConfig;
00064         private PopupWindow _createLevelPopup;
00065         private Dictionary<GridPoint, LevelElementData> _fieldElements = new();
00066         private ToolbarButton _createButton;
00067         private Rect _popupRect;
00068         private Button _createLevelButton;
```

```

00069
00070     private GridPoint _copiedCell = GridPoint.Default;
00071
00072     private IActionCommandManager _commandManager;
00073
00074     [MenuItem("Window/Merge Toolkit/Level (Field)", false, 2)]
00075     public static void ShowWindow()
00076     {
00077         Show(null);
00078     }
00079
00080     public static void Show(LevelConfig levelConfig)
00081     {
00082         _window = GetWindow<LevelEditorWindow>(true);
00083         _window.minSize = new Vector2(800, 600);
00084         _window.titleContent = new GUIContent("New level");
00085
00086         if (levelConfig != null)
00087         {
00088             _window.Reload(levelConfig);
00089         }
00090     }
00091
00092     public void CreateGUI()
00093     {
00094         _window = this;
00095         _commandManager = new ActionCommandManager();
00096
00097         VisualElement root = rootVisualElement;
00098         root.focusable = true;
00099         root.RegisterCallback<KeyDownEvent>(evt =>
00100         {
00101             if (evt.commandKey || evt.ctrlKey)
00102             {
00103                 switch (evt.keyCode)
00104                 {
00105                     case KeyCode.Z:
00106                         _commandManager.Undo();
00107                         evt.StopPropagation();
00108                         break;
00109
00110                     case KeyCode.Y:
00111                         _commandManager.Redo();
00112                         evt.StopPropagation();
00113                         break;
00114                 }
00115             });
00116
00117         _lockTexture =
00118         AssetDatabase.LoadAssetAtPath<Texture>("Assets/MergeIt/Content/Images/Common/lock.png");
00119
00120         var visualTree =
00121
00122             (VisualTreeAsset)EditorGUIUtility.Load(Path.Combine(Constants.LevelEditorResourcesRoot,
00123                                                 "LevelEditorWindow.uxml"));
00124         VisualElement windowRoot = visualTree.CloneTree();
00125         root.Add(windowRoot);
00126
00127         _globalStyle =
00128             (StyleSheet)EditorGUIUtility.Load(Path.Combine(Constants.LevelEditorResourcesRoot,
00129                                                 "LevelEditorWindow.uss"));
00130         root.styleSheets.Add(_globalStyle);
00131         root.RegisterCallback<KeyUpEvent>(evt =>
00132         {
00133             if (evt.keyCode == KeyCode.S && (evt.commandKey || evt.ctrlKey))
00134             {
00135                 SaveButtonClicked();
00136             }
00137         });
00138
00139         _evolutionsField = root.Q<ObjectField>("EvolutionsField");
00140         _evolutionsField.objectType = typeof(SchemeObject);
00141         _evolutionsField.RegisterValueChangedCallback(OnEvolutionsConfigChanged);
00142
00143         ToolbarMenu toolbarMenu = windowRoot.Q<ToolbarMenu>("FileMenu");
00144         toolbarMenu.menu.AppendAction("Open...", _ => LoadButtonClicked());
00145         toolbarMenu.menu.AppendAction("Save", _ => SaveButtonClicked());
00146         toolbarMenu.menu.AppendAction("Save As...", _ => SaveAsButtonClicked());
00147
00148         _createButton = root.Q<ToolbarButton>("CreateButton");
00149         _createButton.clicked += CreateButtonClicked;
00150
00151         _createLevelButton = root.Q<Button>("CreateLevelButton");
00152         _createLevelButton.clicked += CreateLevelButtonClicked;
00153
00154         _createLevelPopup = root.Q<PopupWindow>("CreateLevelPopup");

```

```
00154         _createLevelPopup.visible = false;
00155
00156         _createFieldHeight = root.Q<IntegerField>("FieldHeight");
00157         _createFieldWidth = root.Q<IntegerField>("FieldWidth");
00158
00159         _createFieldHeight.RegisterValueChangedCallback(OnFieldSizeChanged);
00160         _createFieldWidth.RegisterValueChangedCallback(OnFieldSizeChanged);
00161
00162         _warnLabel = root.Q<Label>("WarningLabel");
00163         _warnLabel.visible = false;
00164
00165         _grid = root.Q<VisualElement>("ItemsGrid");
00166         _grid.focusable = true;
00167         _grid.RegisterCallback<KeyUpEvent>(evt =>
00168     {
00169         if (evt.keyCode is KeyCode.Delete or KeyCode.Backspace)
00170         {
00171             if (_selectedToggle != null)
00172             {
00173                 ClearButtonClicked();
00174             }
00175         }
00176         else if (evt.commandKey || evt.ctrlKey)
00177         {
00178             switch (evt.keyCode)
00179             {
00180                 case KeyCode.C:
00181                     if (_selectedCell != GridPoint.Default)
00182                     {
00183                         _copiedCell = _selectedCell;
00184                     }
00185
00186                     break;
00187
00188                 case KeyCode.V:
00189                     if (_selectedToggle != null &&
00190                         _copiedCell != GridPoint.Default &&
00191                         _selectedCell != _copiedCell)
00192                     {
00193                         CopyCell(_copiedCell.X + 1, _copiedCell.Y + 1);
00194                     }
00195
00196                     break;
00197                 }
00198             });
00199
00200
00201         _itemFrameSettings = root.Q<VisualElement>("ItemFrameSettings");
00202         SwitchSideBarVisibility(false);
00203         _isLockedToggle = _itemFrameSettings.Q<Toggle>("IsLocked");
00204         _isLockedToggle.RegisterValueChangedCallback(OnIsLockedChanged);
00205
00206         _itemFrameEvolutions = _itemFrameSettings.Q<VisualElement>("EvoPanel");
00207
00208         _itemImage = _itemFrameSettings.Q<Image>("ItemIcon");
00209
00210         _applyButton = _itemFrameSettings.Q<Button>("ApplyButton");
00211         _applyButton.clicked += ApplyButtonClicked;
00212
00213         _clearButton = _itemFrameSettings.Q<Button>("ClearButton");
00214         _clearButton.clicked += ClearButtonClicked;
00215
00216         _copyButton = _itemFrameSettings.Q<Button>("CopyButton");
00217         _copyRow = _itemFrameSettings.Q<IntegerField>("CopyRow");
00218         _copyColumn = _itemFrameSettings.Q<IntegerField>("CopyColumn");
00219
00220         _copyButton.clicked += CopyButtonClicked;
00221
00222         CheckCreateButton();
00223     }
00224
00225     public void ApplyCell(LevelElementData newData, bool wasChanged)
00226     {
00227         SetupElementCell(newData);
00228         CellSelected(true, newData.Position.X, newData.Position.Y);
00229         DrawElementCell(newData, _selectedToggle);
00230
00231         if (!wasChanged)
00232         {
00233             SetItemData();
00234         }
00235     }
00236
00237     public void UndoApplyCell(LevelElementData previousData, LevelElementData newData)
00238     {
00239         if (previousData != null)
00240         {
```

```

00241             CellSelected(true, previousData.Position.X, previousData.Position.Y);
00242             ApplyCell(previousData, true);
00243         }
00244         else
00245         {
00246             ClearCell(newData.Position);
00247         }
00248     }
00249
00250     public void ClearCell(GridPoint point)
00251     {
00252         CellSelected(true, point.X, point.Y);
00253         _fieldElements.Remove(point);
00254         _selectedToggle.style.backgroundImage = null;
00255
00256         ChangeVisualLock(false, _selectedToggle);
00257     }
00258
00259     public void UndoClearCell(LevelElementData previousData)
00260     {
00261         SetupElementCell(previousData);
00262         CellSelected(true, previousData.Position.X, previousData.Position.Y);
00263         DrawElementCell(previousData, _selectedToggle);
00264     }
00265
00266     private void OnDestroy()
00267     {
00268         rootVisualElement.UnregisterCallback<MouseUpEvent>(OnElementClicked);
00269
00270         _isLockedToggle.UnregisterValueChangedCallback(OnIsLockedChanged);
00271         _createFieldHeight.UnregisterValueChangedCallback(OnFieldSizeChanged);
00272         _createFieldWidth.UnregisterValueChangedCallback(OnFieldSizeChanged);
00273         _createLevelButton.clicked -= CreateLevelButtonClicked;
00274         _createButton.clicked -= CreateButtonClicked;
00275         _copyButton.clicked -= CopyButtonClicked;
00276         _clearButton.clicked -= ClearButtonClicked;
00277         _applyButton.clicked -= ApplyButtonClicked;
00278     }
00279
00280     private void OnFieldSizeChanged(ChangeEvent<int> evt)
00281     {
00282         CheckCreateButton();
00283     }
00284
00285     private void CheckCreateButton()
00286     {
00287         if (_createFieldHeight.value <= 0 || _createFieldWidth.value <= 0)
00288         {
00289             _createLevelButton.SetEnabled(false);
00290         }
00291         else
00292         {
00293             _createLevelButton.SetEnabled(true);
00294         }
00295     }
00296
00297     private void OnElementClicked(MouseUpEvent evt)
00298     {
00299         if (_createLevelPopup.visible && !_popupRect.Contains(evt.mousePosition))
00300         {
00301             ToggleCreatePopup();
00302             rootVisualElement.UnregisterCallback<MouseUpEvent>(OnElementClicked);
00303         }
00304     }
00305
00306     private void SaveButtonClicked()
00307     {
00308         PackLevel(_levelConfig);
00309
00310         EditorUtility.SetDirty(_levelConfig);
00311
00312         Debug.Log($"Level saved successfully: {AssetDatabase.GetAssetPath(_levelConfig)}");
00313     }
00314
00315     private void SaveAsButtonClicked()
00316     {
00317         var path = EditorUtility.SaveFilePanelInProject("Save level", "New_level", "asset", "Save
00318         level to file");
00319         if (path.Length != 0)
00320         {
00321             var levelConfig = CreateInstance<LevelConfig>();
00322
00323             PackLevel(levelConfig);
00324
00325             AssetDatabase.CreateAsset(levelConfig, path);
00326
00327             Debug.Log($"New level saved successfully: {path}");
00328     }

```

```
00327             _levelConfig = levelConfig;
00328             _window.titleContent = new GUIContent(path);
00329         }
00330     }
00331 }
00332
00333 private void LoadButtonClicked()
00334 {
00335     var path = EditorUtility.OpenFilePanelWithFilters("Load level", "Assets",
00336         new[] { "Saved levels", "asset" });
00337     if (path.Length != 0)
00338     {
00339         try
00340         {
00341             string relativePath = null;
00342             if (path.StartsWith(Application.dataPath))
00343             {
00344                 relativePath = "Assets" + path.Substring(Application.dataPath.Length);
00345             }
00346
00347             var levelConfig = AssetDatabase.LoadAssetAtPath<LevelConfig>(relativePath);
00348             Debug.Log($"Start loading config with elements:
{levelConfig?.FieldElementsData?.Count}");
00349
00350             Reload(levelConfig);
00351
00352             Debug.Log($"Level loaded successfully: {relativePath}");
00353         }
00354         catch (Exception e)
00355         {
00356             Debug.Log($"{e}");
00357         }
00358     }
00359 }
00360
00361 private async void Reload(LevelConfig levelConfig)
00362 {
00363     UnpackLevel(levelConfig);
00364
00365     while (EvolutionSelection.ElementData == null)
00366     {
00367         await Task.Yield();
00368     }
00369
00370     DrawLevel();
00371
00372     var path = AssetDatabase.GetAssetPath(levelConfig);
00373
00374     _window.titleContent = new GUIContent(path);
00375 }
00376
00377 private void CreateButtonClicked()
00378 {
00379     ToggleCreatePopup();
00380     rootVisualElement.RegisterCallback<MouseUpEvent>(OnElementClicked,
TrickleDown.TrickleDown);
00381 }
00382
00383 private void CreateLevelButtonClicked()
00384 {
00385     var path = EditorUtility.SaveFilePanelInProject("Create level", "New_level", "asset",
"Create new level");
00386     if (path.Length != 0)
00387     {
00388         _fieldWidth = _createFieldWidth.value;
00389         _fieldHeight = _createFieldHeight.value;
00390
00391         _levelConfig = CreateInstance<LevelConfig>();
00392         _levelConfig.FieldWidth = _fieldWidth;
00393         _levelConfig.FieldHeight = _fieldHeight;
00394         _levelConfig.EvolutionsScheme = _evolutionsField.value as SchemeObject;
00395
00396         AssetDatabase.CreateAsset(_levelConfig, path);
00397
00398         Debug.Log($"Level created successfully: {path}");
00399
00400         _fieldElements = new Dictionary<GridPoint, LevelElementData>();
00401
00402         DrawLevel();
00403
00404         _window.titleContent = new GUIContent(path);
00405         _createLevelPopup.style.display = StylesConstants.DisplayNone;
00406     }
00407 }
00408
00409 private void ToggleCreatePopup()
00410 {
```

```

00411         if (_createLevelPopup.visible)
00412         {
00413             _createLevelPopup.visible = false;
00414         }
00415     else
00416     {
00417         _createLevelPopup.visible = true;
00418         Rect worldBound = _createButton.worldBound;
00419         var buttonPosition = new Vector2(worldBound.xMin, worldBound.yMax);
00420         _createLevelPopup.style.top = buttonPosition.y;
00421         _createLevelPopup.style.left = buttonPosition.x;
00422         _popupRect = new Rect(buttonPosition.x, buttonPosition.y,
00423             _createLevelPopup.worldBound.width,
00424             _createLevelPopup.worldBound.height);
00425     }
00426 }
00427 private void PackLevel(LevelConfig levelConfig)
00428 {
00429     List<LevelElementData> elements = _fieldElements.Values.ToList();
00430     levelConfig.EvolutionsScheme = _evolutionsField.value as SchemeObject;
00431     levelConfig.FieldElementsData = elements;
00432     levelConfig.FieldHeight = _fieldHeight;
00433     levelConfig.FieldWidth = _fieldWidth;
00434 }
00435
00436 private void UnpackLevel(LevelConfig levelConfig)
00437 {
00438     _levelConfig = levelConfig;
00439     if (levelConfig.FieldElementsData?.Count != 0)
00440     {
00441         _fieldElements = new();
00442         if (levelConfig.FieldElementsData != null)
00443         {
00444             foreach (var levelElementData in levelConfig.FieldElementsData)
00445             {
00446                 _fieldElements[levelElementData.Position] = levelElementData.GetClone();
00447             }
00448         }
00449     }
00450     else
00451     {
00452         _fieldElements = new Dictionary<GridPoint, LevelElementData>();
00453     }
00454
00455     _fieldHeight = levelConfig.FieldHeight;
00456     _fieldWidth = levelConfig.FieldWidth;
00457
00458     var evolution = _levelConfig.EvolutionsScheme;
00459     _evolutionsField.SetValueWithoutNotify(evolution);
00460     SetupEvolution(evolution);
00461 }
00462
00463 private void CopyButtonClicked()
00464 {
00465     int row = _copyRow.value;
00466     int column = _copyColumn.value;
00467
00468     CopyCell(row, column);
00469 }
00470
00471 private void CopyCell(int row, int column)
00472 {
00473     if (row < 1 || row > _fieldHeight)
00474     {
00475         Debug.Log($"Row number {row} is out of bounds");
00476         return;
00477     }
00478
00479     if (column < 1 || column > _fieldWidth)
00480     {
00481         Debug.Log($"Column number {row} is out of bounds");
00482         return;
00483     }
00484
00485     _fieldElements.TryGetValue(GridPoint.Create(row - 1, column - 1), out LevelElementData
00486     data);
00487
00488     if (data != null)
00489     {
00490         EvolutionSelection.UpdateCell(data, copyPosition: false);
00491
00492         PerformApplyCellCommand();
00493     }
00494     else
00495     {
00496         PerformClearCellCommand(EvolutionSelection.ElementData.Position);
00497     }
00498 }

```

```
00496         }
00497     }
00498 
00499     private void ApplyButtonClicked()
00500     {
00501         PerformApplyCellCommand();
00502     }
00503 
00504     private void ClearButtonClicked()
00505     {
00506         PerformClearCellCommand(EvolutionSelection.ElementData.Position);
00507     }
00508 
00509     private void PerformApplyCellCommand()
00510     {
00511         var newData = EvolutionSelection.ElementData.GetClone();
00512 
00513         if (_fieldElements.TryGetValue(newData.Position, out var data))
00514         {
00515             var changeCommand = new ChangeCellCommand(this, data, newData);
00516             _commandManager.ExecuteCommand(changeCommand);
00517         }
00518         else
00519         {
00520             var applyCommand = new ApplyCellCommand(this, newData);
00521             _commandManager.ExecuteCommand(applyCommand);
00522         }
00523     }
00524 
00525     private void PerformClearCellCommand(GridPoint point)
00526     {
00527         var existingData = EvolutionSelection.ElementData.GetClone();
00528         var command = new ClearCellCommand(this, existingData);
00529         _commandManager.ExecuteCommand(command);
00530     }
00531 
00532     private void OnIsLockedChanged(ChangeEvent<bool> evt)
00533     {
00534         SetLock(evt.newValue);
00535     }
00536 
00537     private void OnEvolutionConfigChanged(ChangeEvent<Object> evt)
00538     {
00539         ClearAll();
00540 
00541         SetupEvolution(evt.newValue as SchemeObject);
00542     }
00543 
00544     private void SetupEvolution(SchemeObject config)
00545     {
00546         if (config)
00547         {
00548             _loadedScheme = config;
00549 
00550             if (_loadedScheme != null)
00551             {
00552                 Debug.Log($"Evolution config changed to: {_loadedScheme.name}");
00553 
00554                 EvolutionSelection.Init(_loadedScheme.Evolution);
00555                 SwitchFieldActiveState(true);
00556             }
00557         }
00558     }
00559 
00560     private void SwitchFieldActiveState(bool active)
00561     {
00562         _warnLabel.visible = !active;
00563 
00564         if (_toggles != null)
00565         {
00566             for (int i = 0; i < _toggles.GetLength(0); i++)
00567             {
00568                 for (int j = 0; j < _toggles.GetLength(1); j++)
00569                 {
00570                     _toggles[i, j].SetEnabled(active);
00571                 }
00572             }
00573         }
00574     }
00575 
00576     private void SwitchSideBarVisibility(bool active)
00577     {
00578         _itemFrameSettings.visible = active;
00579     }
00580 
00581     private void ClearAll()
00582     {
```

```

00583     _selectedToggle = null;
00584     _selectedCell = GridPoint.Default;
00585
00586     SwitchFieldActiveState(false);
00587     SwitchSideBarVisibility(false);
00588     ClearField();
00589     EvolutionSelection.ClearSelection();
00590 }
00591
00592     private void ClearField()
00593 {
00594     foreach (KeyValuePair<GridPoint, LevelElementData> fieldElement in _fieldElements)
00595     {
00596         GridPoint point = fieldElement.Key;
00597         ToolbarToggle toggle = _toggles[point.X, point.Y];
00598         toggle.style.backgroundImage = null;
00599         ChangeVisualLock(false, toggle);
00600         UnselectToggle(toggle, true);
00601     }
00602
00603     _fieldElements.Clear();
00604 }
00605
00606     private void DrawLevel()
00607 {
00608     _grid.Clear();
00609     _toggles = new ToolbarToggle[_fieldHeight, _fieldWidth];
00610
00611     for (int i = 0; i < _fieldHeight; i++)
00612     {
00613         if (i == 0)
00614         {
00615             var visualElementHeader = new VisualElement();
00616             visualElementHeader.AddToClassList("itemsGridRowHeader");
00617             for (int j = 0; j < _fieldWidth; j++)
00618             {
00619                 var labelColumn = new Label($"{j + 1}");
00620                 labelColumn.AddToClassList("gridHeaderLabel");
00621                 visualElementHeader.Add(labelColumn);
00622             }
00623
00624             _grid.Add(visualElementHeader);
00625         }
00626
00627         var visualElement = new VisualElement();
00628         visualElement.AddToClassList("itemsGridRow");
00629
00630         var labelRow = new Label($"{i + 1}");
00631         labelRow.AddToClassList("gridLabelColumn");
00632
00633         visualElement.Add(labelRow);
00634
00635         for (int j = 0; j < _fieldWidth; j++)
00636         {
00637             var button = new ToolbarToggle();
00638             var elementPosition = GridPoint.Create(i, j);
00639             if (_fieldElements.TryGetValue(elementPosition, out LevelElementData data))
00640             {
00641                 DrawElementCell(data, button);
00642             }
00643
00644             _toggles[i, j] = button;
00645
00646             int row = i;
00647             int column = j;
00648
00649             button.RegisterCallback<MouseUpEvent>(evt =>
00650             {
00651                 if (button.value)
00652                 {
00653                     SelectToggle(button, false);
00654                 }
00655                 else
00656                 {
00657                     UnselectToggle(button, false);
00658                 }
00659
00660                 CellSelected(button.value, row, column);
00661                 OnElementClicked(evt);
00662             });
00663
00664             button.AddToClassList("gridButton");
00665             visualElement.Add(button);
00666         }
00667
00668         _grid.Add(visualElement);
00669     }

```

```
00670
00671         if (_loadedScheme == null)
00672     {
00673         SwitchFieldActiveState(false);
00674     }
00675 }
00676
00677 private void CellSelected(bool selected, int row, int column)
00678 {
00679     if (selected)
00680     {
00681         for (int i = 0; i < _fieldHeight; i++)
00682         {
00683             for (int j = 0; j < _fieldWidth; j++)
00684             {
00685                 var toggle = _toggles[i, j];
00686
00687                 if (i == row && j == column)
00688                 {
00689                     GridPoint elementPosition = GridPoint.Create(row, column);
00690
00691                     _fieldElements.TryGetValue(elementPosition, out LevelElementData
00692             elementData);
00693
00694                     _selectedToggle = toggle;
00695                     _selectedCell = new GridPoint(row, column);
00696                     _selectedToggle.SetValueWithoutNotify(true);
00697
00698                     SelectToggle(_selectedToggle, true);
00699
00700                     SwitchSideBarVisibility(true);
00701
00702                     EvolutionSelection.SelectCell(row, column, elementData);
00703
00704                     SetItemData();
00705
00706                     continue;
00707
00708                 }
00709             }
00710         }
00711     }
00712     else
00713     {
00714         EvolutionSelection.ClearSelection();
00715         SwitchSideBarVisibility(false);
00716     }
00717 }
00718
00719 private void SetLock(bool isLocked)
00720 {
00721     EvolutionSelection.LockCell(isLocked);
00722 }
00723
00724 private void ChangeVisualLock(bool isLocked, Toggle toggle)
00725 {
00726     var locker = toggle.Q<Image>("Locker");
00727     if (isLocked)
00728     {
00729         if (locker == null)
00730         {
00731             locker = CreateLocker();
00732             toggle.Add(locker);
00733         }
00734     }
00735     else
00736     {
00737         if (locker != null)
00738         {
00739             toggle.Remove(locker);
00740         }
00741     }
00742 }
00743
00744 private void SetupElementCell(LevelElementData itemData)
00745 {
00746     if (_fieldElements.TryGetValue(itemData.Position, out LevelElementData fieldElement))
00747     {
00748         fieldElement.CopyFrom(itemData);
00749     }
00750     else
00751     {
00752         _fieldElements[itemData.Position] = itemData;
00753     }
00754 }
00755
00756 private void DrawElementCell(LevelElementData elementData, Toggle toggle)
```

```

00756      {
00757          ElementConfig elementConfig = elementData.Element;
00758
00759          Sprite icon = GetIcon(elementConfig.CommonSettings.Icon);
00760
00761          toggle.style.backgroundImage = icon != null ? icon.texture : null;
00762
00763          ChangeVisualLock(elementData.IsBlocked, toggle);
00764      }
00765
00766  private Sprite GetIcon(FieldElementIconComponent placeholder)
00767  {
00768      Sprite sprite = null;
00769      if (placeholder)
00770      {
00771          sprite = placeholder.GetImage()?.sprite;
00772      }
00773
00774      return sprite;
00775  }
00776
00777  private Image CreateLocker()
00778  {
00779      var locker = new Image
00780      {
00781          name = "Locker",
00782          image = _lockTexture,
00783          style =
00784          {
00785              left = StylesConstants.Length0,
00786              right = StylesConstants.Length0,
00787              top = StylesConstants.Length0,
00788              bottom = StylesConstants.Length0,
00789              position = StylesConstants.AbsolutePosition,
00790          }
00791      };
00792
00793      return locker;
00794  }
00795
00796  private void SelectToggle(ToolbarToggle toggle, bool setValue)
00797  {
00798      if (setValue)
00799      {
00800          toggle.SetValueWithoutNotify(true);
00801      }
00802      toggle.AddToList("gridSelectedButton");
00803      toggle.RemoveFromList("gridButton");
00804  }
00805
00806  private void UnselectToggle(ToolbarToggle toggle, bool setValue)
00807  {
00808      if (setValue)
00809      {
00810          toggle.SetValueWithoutNotify(false);
00811      }
00812      toggle.AddToList("gridButton");
00813      toggle.RemoveFromList("gridSelectedButton");
00814  }
00815
00816  private void SetItemData()
00817  {
00818      if (_evoPopup == null)
00819      {
00820          _evoPopup = new PopupField<EvolutionData>("Evolution:", _loadedScheme.Evolution,
00821                                         EvolutionSelection.Evolution);
00822
00823          _evoPopup.formatListItemCallback = FormatEvolutionCallback;
00824          _evoPopup.formatSelectedValueCallback = FormatEvolutionCallback;
00825
00826          _evoPopup.AddToList("evoList");
00827          _evoPopup.RegisterValueChangedCallback(OnEvolutionChanged);
00828          _itemFrameEvolutions.Add(_evoPopup);
00829      }
00830
00831      int evoIndex = _loadedScheme.Evolution.IndexOf(EvolutionSelection.Evolution);
00832
00833      if (evoIndex == _evoPopup.index)
00834      {
00835          SendChangedEvent(_evoPopup, _evoPopup.value, EvolutionSelection.Evolution);
00836      }
00837      else
00838      {
00839          _evoPopup.index = _loadedScheme.Evolution.IndexOf(EvolutionSelection.Evolution);
00840      }
00841
00842      _isLockedToggle.SetValueWithoutNotify(EvolutionSelection.ElementData.IsBlocked);

```

```

00843         }
00844
00845     private void OnEvolutionChanged(ChangeEvent<EvolutionData> evt)
00846     {
00847         EvolutionSelection.SelectEvolution(evt.newValue);
00848
00849         _evoPopup.index = _loadedScheme.Evolution.IndexOf(EvolutionSelection.Evolution);
00850
00851         if (_evolutionChainElementsPopup != null)
00852         {
00853
00854             _evolutionChainElementsPopup.UnregisterValueChangedCallback(OnEvolutionChainElementChanged);
00855             _itemFrameEvolutions.Remove(_evolutionChainElementsPopup);
00856
00857             List<ElementConfig> evolutionItems = EvolutionSelection.Evolution.Chain;
00858
00859             _evolutionChainElementsPopup =
00860                 new PopupField<ElementConfig>("Element:", evolutionItems, EvolutionSelection.EvoItem);
00861
00862             _evolutionChainElementsPopup.RegisterValueChangedCallback(OnEvolutionChainElementChanged);
00863
00864             _evolutionChainElementsPopup.formatSelectedValueCallback =
00865                 OnFormatEvolutionChainItemCallback;
00866             _evolutionChainElementsPopup.formatListItemSelectedCallback = OnFormatEvolutionChainItemCallback;
00867
00868             _evolutionChainElementsPopup.AddToList("evoList");
00869             _itemFrameEvolutions.Add(_evolutionChainElementsPopup);
00870
00871             SendChangedEvent(_evolutionChainElementsPopup, _evolutionChainElementsPopup.value,
00872                             EvolutionSelection.EvoItem);
00873         }
00874
00875     private void OnEvolutionChainElementChanged(ChangeEvent<ElementConfig> evt)
00876     {
00877         EvolutionSelection.SelectElement(evt.newValue);
00878
00879         var elementConfig = evt.newValue;
00880         Sprite icon = GetIcon(elementConfig.CommonSettings.Icon);
00881
00882         _itemImage.image = icon != null ? icon.texture : null;
00883
00884     private void SendChangedEvent<T>(PopupField<T> field, T prevValue, T newValue)
00885     {
00886         var changeEvent = ChangeEvent<T>.GetPooled(prevValue, newValue);
00887         changeEvent.target = field;
00888         field.SendEvent(changeEvent);
00889     }
00890
00891     private string FormatEvolutionCallback(EvolutionData arg)
00892     {
00893         return arg.Name;
00894     }
00895
00896     private string OnFormatEvolutionChainItemCallback(ElementConfig item)
00897     {
00898         var evoIndex = _evoPopup.index;
00899         var evolution = _loadedScheme.Evolution[evoIndex];
00900         var itemIndex = evolution.Chain.IndexOf(item);
00901
00902         var elementConfig = item;
00903         string itemName = $"Level {itemIndex + 1}: {elementConfig.CommonSettings.Name}";
00904
00905         return itemName;
00906     }
00907 }
00908 }
```

7.35 ConfigsPostprocessor.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.IO;
00005 using System.Reflection;
00006 using System.Threading.Tasks;
00007 using MergeIt.Core.Configs.Elements;
00008 using UnityEditor;
00009
0010 namespace MergeIt.Editor.Helpers
0011 {
0012     public class ConfigsPostprocessor : AssetModificationProcessor

```

```

00013     {
00014         private async static void OnWillCreateAsset(string assetName)
00015     {
00016         string ext = Path.GetExtension(assetName);
00017         if (ext == ".asset")
00018         {
00019             await WaitCreation(assetName);
00020         }
00021     }
00022
00023     private async static Task WaitCreation(string path)
00024     {
00025         var configType = AssetDatabase.GetMainAssetTypeAtPath(path);
00026         if (configType != null)
00027         {
00028             CheckConfig(configType, path);
00029         }
00030         else
00031         {
00032             await Task.Yield();
00033
00034             configType = AssetDatabase.GetMainAssetTypeAtPath(path);
00035             CheckConfig(configType, path);
00036         }
00037     }
00038
00039     private static void CheckConfig(Type configType, string path)
00040     {
00041         if (configType != null & configType == typeof(ElementConfig))
00042         {
00043             var config = AssetDatabase.LoadAssetAtPath<ElementConfig>(path);
00044
00045             MethodInfo methodInfo = configType.GetMethod("GenerateGuid", BindingFlags.NonPublic |
00046             BindingFlags.Instance);
00047
00048             methodInfo?.Invoke(config, null);
00049
00050             EditorUtility.SetDirty(config);
00051             AssetDatabase.SaveAssets();
00052             AssetDatabase.Refresh();
00053         }
00054     }
00055 }

```

7.36 SlicedFilledImageEditor.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Helpers;
00004 using UnityEditor;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Editor.Helpers
00008 {
00009     [CustomEditor( typeof( SlicedFilledImage ) ), CanEditMultipleObjects]
00010     public class SlicedFilledImageEditor : UnityEditor.Editor
00011     {
00012         private SerializedProperty _spriteProp, _colorProp;
00013         private GUIContent _spriteLabel;
00014
00015         private void OnEnable()
00016         {
00017             _spriteProp = serializedObject.FindProperty( "m_Sprite" );
00018             _colorProp = serializedObject.FindProperty( "m_Color" );
00019             _spriteLabel = new GUIContent( "Source Image" );
00020         }
00021
00022         public override void OnInspectorGUI()
00023         {
00024             serializedObject.Update();
00025
00026             EditorGUILayout.PropertyField( _spriteProp, _spriteLabel );
00027             EditorGUILayout.PropertyField( _colorProp );
00028             DrawPropertiesExcluding( serializedObject, "m_Script", "m_Sprite", "m_Color",
00029             "m_OnCullStateChanged" );
00030
00031             serializedObject.ApplyModifiedProperties();
00032         }
00033     }

```

7.37 AnimationControllerBase.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Core.Animations
00007 {
00008     public abstract class AnimationControllerBase : MonoBehaviour, IAnimationController
00009     {
00010         public abstract void Initialize(IAnimationListener listener);
00011
00012         public virtual void SetState(string state)
00013         {
00014             throw new NotImplementedException();
00015         }
00016
00017         public virtual void SetState(int state)
00018         {
00019             throw new NotImplementedException();
00020         }
00021
00022         public virtual void SetState<T>(T state) where T : Enum
00023         {
00024             throw new NotImplementedException();
00025         }
00026     }
00027 }
```

7.38 AnimationListener.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Animations
00004 {
00005     public class AnimationListener : IAnimationListener
00006     {
00007         public virtual void ResetAnimationState()
00008         {
00009
00010         }
00011     }
00012 }
```

7.39 AnimationWindowStates.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Animations
00006 {
00007     public static class AnimationWindowStates
00008     {
00009         public static readonly int Open = Animator.StringToHash("Open");
00010         public static readonly int Close = Animator.StringToHash("Close");
00011     }
00012 }
```

7.40 AnimatorStateMachineBehaviour.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Core.Animations
00007 {
00008     [SharedBetweenAnimators]
00009     public class AnimatorStateMachineBehaviour : StateMachineBehaviour
00010     {
00011         private static readonly Dictionary<Animator, bool> ActiveAnimators = new();
00012     }
00013 }
```

```

00013     public override void OnStateEnter(Animator animator, AnimatorStateInfo stateInfo, int
00014         layerIndex)
00015     {
00016         base.OnStateEnter(animator, stateInfo, layerIndex);
00017         ActiveAnimators[animator] = true;
00018     }
00019
00020     public override void OnStateUpdate(Animator animator, AnimatorStateInfo stateInfo, int
00021         layerIndex)
00022     {
00023         base.OnStateUpdate(animator, stateInfo, layerIndex);
00024         if (!ActiveAnimators[animator])
00025         {
00026             return;
00027         }
00028
00029         if (stateInfo.normalizedTime >= 1f &&
00030             !animator.IsInTransition(layerIndex))
00031         {
00032             ActiveAnimators[animator] = false;
00033             CheckForEnd(animator, stateInfo.shortNameHash);
00034         }
00035     }
00036
00037     private void CheckForEnd(Animator animator, int stateHash)
00038     {
00039         var listener = animator.gameObject.GetComponent<IWindowAnimationController>();
00040         if (listener != null)
00041         {
00042             if (stateHash == AnimationWindowStates.Open)
00043             {
00044                 listener.OnOpenEnd();
00045             }
00046             else if (stateHash == AnimationWindowStates.Close)
00047             {
00048                 listener.OnCloseEnd();
00049             }
00050         }
00051     }
00052 }
00053
00054
00055 }
```

7.41 IAnimationController.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004
00005 namespace MergeIt.Core.Animations
00006 {
00007     public interface IAnimationController
00008     {
00009         void Initialize(IAnimationListener listener);
00010
00011         void SetState(string state);
00012         void SetState(int state);
00013         void SetState<T>(T state) where T : Enum;
00014     }
00015 }
```

7.42 IAnimationListener.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Animations
00004 {
00005     public interface IAnimationListener
00006     {
00007         void ResetAnimationState();
00008     }
00009 }
```

7.43 IWindowAnimationController.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Animations
00004 {
00005     public interface IWindowAnimationController
00006     {
00007         void Initialize(IWindowAnimationListener listener);
00008
00009         void OpenWindow();
00010         void CloseWindow();
00011         void OnOpenEnd();
00012         void OnCloseEnd();
00013     }
00014 }
```

7.44 IWindowAnimationListener.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Animations
00004 {
00005     public interface IWindowAnimationListener
00006     {
00007         void OnOpenStarted();
00008         void OnOpenFinished();
00009         void OnCloseStarted();
00010         void OnCloseFinished();
00011     }
00012 }
```

7.45 MecanimController.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Animations
00006 {
00007     [RequireComponent(typeof(Animator))]
00008     public class MecanimController : AnimationControllerBase
00009     {
00010         private Animator _animator;
00011         private IAnimationListener _listener;
00012
00013         private void Awake()
00014         {
00015             _animator = GetComponent<Animator>();
00016         }
00017
00018         public override void Initialize(IAnimationListener listener)
00019         {
00020             _listener = listener;
00021         }
00022
00023         public override void SetState(string state)
00024         {
00025             _animator.Play(state);
00026         }
00027
00028         public override void SetState(int state)
00029         {
00030             _animator.Play(state);
00031         }
00032
00033         public override void SetState<T>(T state)
00034         {
00035             _animator.Play(state.ToString());
00036         }
00037     }
00038 }
00039 }
```

7.46 WindowDefaultAnimationController.cs

```
00001 // Copyright (c) 2024, Awessets
```

```

00002
00003 namespace MergeIt.Core.Animations
00004 {
00005     public class WindowDefaultAnimationController : IWindowAnimationController
00006     {
00007         private IWindowAnimationListener _listener;
00008
00009         public void Initialize(IWindowAnimationListener listener)
0010         {
0011             _listener = listener;
0012         }
0013
0014         public void OpenWindow()
0015         {
0016             _listener.OnOpenStarted();
0017             OnOpenEnd();
0018         }
0019
0020         public void CloseWindow()
0021         {
0022             _listener.OnCloseStarted();
0023             OnCloseEnd();
0024         }
0025
0026         public void OnOpenEnd()
0027         {
0028             _listener.OnOpenFinished();
0029         }
0030
0031         public void OnCloseEnd()
0032         {
0033             _listener.OnCloseFinished();
0034         }
0035     }
0036 }
```

7.47 WindowMecanimAnimationController.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Animations
00006 {
00007     [RequireComponent(typeof(Animator))]
00008     public class WindowMecanimAnimationController : MonoBehaviour, IWindowAnimationController
00009     {
0010         [SerializeField]
0011         private Animator _animator;
0012
0013         [SerializeField]
0014         private string _openTrigger;
0015
0016         [SerializeField]
0017         private string _closeTrigger;
0018
0019         private int _openHash;
0020         private int _closeHash;
0021
0022         private void Awake()
0023         {
0024             _animator = GetComponent<Animator>();
0025             _openHash = Animator.StringToHash(_openTrigger);
0026             _closeHash = Animator.StringToHash(_closeTrigger);
0027         }
0028
0029         private IWindowAnimationListener _listener;
0030
0031         public void Initialize(IWindowAnimationListener listener)
0032         {
0033             _listener = listener;
0034         }
0035
0036         public void OpenWindow()
0037         {
0038             _listener.OnOpenStarted();
0039             _animator.SetTrigger(_openHash);
0040         }
0041
0042         public void CloseWindow()
0043         {
0044             _listener.OnCloseStarted();
0045             _animator.SetTrigger(_closeHash);
0046         }
0047     }
0048 }
```

```

00046         }
00047
00048     public void OnOpenEnd()
00049     {
00050         _listener.OnOpenFinished();
00051     }
00052
00053     public void OnCloseEnd()
00054     {
00055         _listener.OnCloseFinished();
00056     }
00057 }
00058
00059 }
```

7.48 Command.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using Cysharp.Threading.Tasks;
00005
00006 namespace MergeIt.Core.Commands
00007 {
00008     public abstract class Command : ICommand, IDisposable
00009     {
00010         public event Action<ICommand> Finished;
00011
00012         public virtual void Execute()
00013         {
00014
00015     }
00016
00017         public async virtual UniTask ExecuteAsync()
00018         {
00019             await UniTask.RunOnThreadPool(Execute);
00020         }
00021
00022         public void Dispose()
00023         {
00024             OnDispose();
00025         }
00026
00027         protected virtual void Finish()
00028         {
00029             Finished?.Invoke(this);
00030         }
00031
00032         protected virtual void OnDispose()
00033         {
00034
00035     }
00036 }
00037 }
```

7.49 CommandManager.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using System.Threading.Tasks;
00006 using Cysharp.Threading.Tasks;
00007 using UnityEngine;
00008
00009 namespace MergeIt.Core.Commands
00010 {
00011     public class CommandManager : ICommandManager
00012     {
00013         public event Action<ICommandManager> Finished;
00014         public bool Executing { get; private set; }
00015
00016         private readonly Queue<ICommand> _commands = new Queue<ICommand>();
00017         private ICommand _currentCommand;
00018
00019         public void Run()
00020         {
00021             Executing = true;
00022             RunNext();
```

```
00023     }
00024
00025     public void RunSimultaneously()
00026     {
00027         Executing = true;
00028         if (_commands.Count > 0)
00029         {
00030             while (_commands.Count > 0)
00031             {
00032                 var currentCommand = _commands.Dequeue();
00033                 currentCommand.Finished += OnSimultaneousCommandFinished;
00034                 currentCommand.Execute();
00035             }
00036         }
00037         else
00038         {
00039             Finish();
00040         }
00041     }
00042
00043     public async UniTask RunAsync()
00044     {
00045         Executing = true;
00046         foreach ( ICommand command in _commands)
00047         {
00048             _currentCommand = command;
00049             await _currentCommand.ExecuteAsync();
00050         }
00051
00052         Finish();
00053     }
00054
00055     public void Add(ICommand command)
00056     {
00057         _commands.Enqueue(command);
00058     }
00059
00060     private void OnCommandFinished(ICommand command)
00061     {
00062         command.Finished -= OnCommandFinished;
00063
00064         if (_commands.Count == 0)
00065         {
00066             Finish();
00067         }
00068         else
00069         {
00070             RunNext();
00071         }
00072     }
00073
00074     private void OnSimultaneousCommandFinished(ICommand command)
00075     {
00076         command.Finished -= OnSimultaneousCommandFinished;
00077
00078         if (_commands.Count == 0)
00079         {
00080             Finish();
00081         }
00082     }
00083
00084     private void RunNext()
00085     {
00086         if (_currentCommand != null)
00087         {
00088             Debug.Log($"Command manager is busy. Running command {_currentCommand.GetType()}");
00089             return;
00090         }
00091
00092         if (_commands.Count > 0)
00093         {
00094             _currentCommand = _commands.Dequeue();
00095             _currentCommand.Finished += OnCommandFinished;
00096             _currentCommand.Execute();
00097         }
00098     }
00099
00100    private void Finish()
00101    {
00102        Executing = false;
00103        Finished?.Invoke(this);
00104    }
00105 }
00106 }
```

7.50 ICommand.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using Cysharp.Threading.Tasks;
00005
00006 namespace MergeIt.Core.Commands
00007 {
00008     public interface ICommand
00009     {
00010         event Action<ICommand> Finished;
00011         void Execute();
00012         UniTask ExecuteAsync();
00013     }
00014 }
```

7.51 ICommandManager.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004
00005 namespace MergeIt.Core.Commands
00006 {
00007     public interface ICommandManager
00008     {
00009         event Action<ICommandManager> Finished;
00010         bool Executing { get; }
00011         void Run();
00012         void RunSimultaneously();
00013         void Add(ICommand command);
00014     }
00015 }
```

7.52 ConfigsConstants.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Configs
00004 {
00005     public static class ConfigsConstants
00006     {
00007         public const string LevelConfigsPath = "Configs/Levels";
00008         public const string ConfigsPath = "Configs/ConfigsContainer";
00009     }
00010 }
```

7.53 EvolutionData.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using MergeIt.Core.Configs.Elements;
00006 using UnityEngine;
00007
00008 namespace MergeIt.Core.Configs.Data
00009 {
00010     [Serializable]
00011     public class EvolutionData : IEquatable<EvolutionData>
00012     {
00013         [SerializeField] private string _guid;
00014         [SerializeField] private string _name;
00015         [SerializeField] private string _description;
00016         [SerializeField] private bool _discovered;
00017         [SerializeField] private List<ElementConfig> _chain;
00018         private IEquatable<EvolutionData> _equatableImplementation;
00019
00020         public EvolutionData(string guid, string name, string description, bool discovered)
00021         {
00022             _guid = guid;
```

```

00023         _name = name;
00024         _description = description;
00025         _discovered = discovered;
00026         _chain = new List<ElementConfig>();
00027     }
00028
00029     public string Name
00030     {
00031         get => _name;
00032     }
00033
00034     public string Description
00035     {
00036         get => _description;
00037     }
00038
00039     public bool Discovered
00040     {
00041         get => _discovered;
00042     }
00043
00044     public List<ElementConfig> Chain
00045     {
00046         get => _chain;
00047     }
00048
00049     public string Id
00050     {
00051         get => _guid;
00052     }
00053
00054     public bool Equals(EvolutionData other)
00055     {
00056         if (ReferenceEquals(null, other)) return false;
00057         if (ReferenceEquals(this, other)) return true;
00058         return _guid == other._guid;
00059     }
00060
00061     public override bool Equals(object obj)
00062     {
00063         if (ReferenceEquals(null, obj)) return false;
00064         if (ReferenceEquals(this, obj)) return true;
00065         if (obj.GetType() != this.GetType()) return false;
00066         return Equals((EvolutionData) obj);
00067     }
00068
00069     public override int GetHashCode()
00070     {
00071         return (_guid != null ? _guid.GetHashCode() : 0);
00072     }
00073 }
00074 }
```

7.54 LevelElementData.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Diagnostics;
00005 using MergeIt.Core.Configs.Elements;
00006 using MergeIt.Core.FieldElements;
00007
00008 namespace MergeIt.Core.Configs.Data
00009 {
00010     [Serializable, DebuggerDisplay("Position={Position}, Element={Element?.Id},
00011     Evo={Evolution?.Name}")]
00011     public class LevelElementData
00012     {
00013         public string EvolutionId;
00014         public ElementConfig Element;
00015         public bool IsBlocked;
00016         public GridPoint Position;
00017
00018         public void CopyFrom(LevelElementData itemData, bool copyPosition = true)
00019         {
00020             EvolutionId = itemData.EvolutionId;
00021             Element = itemData.Element;
00022             IsBlocked = itemData.IsBlocked;
00023
00024             if (copyPosition)
00025             {
00026                 Position = itemData.Position.Copy();
00027             }
00028         }
00029     }
00030 }
```

```
00028     }
00029
00030     public LevelElementData GetClone(bool copyPosition = true)
00031     {
00032         var fieldCellData = new LevelElementData();
00033         fieldCellData.CopyFrom(this, copyPosition);
00034
00035         return fieldCellData;
00036     }
00037 }
00038 }
```

7.55 EffectConfig.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Configs.Effects
00006 {
00007     public class EffectConfig : ScriptableObject
00008     {
00009         public string Name;
00010         public float Duration;
00011     }
00012 }
```

7.56 ElementConfig.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Configs.Types;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Core.Configs.Elements
00008 {
00009     [CreateAssetMenu(fileName = "ElementConfig", menuName = "Merge Toolkit/Element")]
00010     public class ElementConfig : ScriptableObject
00011     {
00012         [SerializeField]
00013         private ElementType _type;
00014
00015         [SerializeField]
00016         private ElementCommonSettings _commonSettings;
00017
00018         [SerializeField]
00019         private ElementGeneratorSettings _generatorSettings;
00020
00021         [SerializeField]
00022         private string _id;
00023
00024         public string Id
00025         {
00026             get
00027             {
00028                 if (string.IsNullOrEmpty(_id))
00029                 {
00030                     _id = Guid.NewGuid().ToString();
00031                 }
00032
00033                 return _id;
00034             }
00035         }
00036
00037         public ElementType Type
00038         {
00039             get => _type;
00040         }
00041
00042         public ElementCommonSettings CommonSettings
00043         {
00044             get => _commonSettings;
00045             set => _commonSettings = value;
00046         }
00047
00048         public ElementGeneratorSettings GeneratorSettings
00049         {
00050             get => _generatorSettings;
```

```

00051         set => _generatorSettings = value;
00052     }
00053
00054     public FieldElementIconComponent GetIconComponent()
00055     {
00056         return _commonSettings?.Icon;
00057     }
00058
00059     protected void GenerateGuid()
00060     {
00061         _id = Guid.NewGuid().ToString();
00062     }
00063 }
00064 }
```

7.57 FieldElementIconComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Helpers;
00004 using UnityEngine;
00005 using UnityEngine.UI;
00006
00007 namespace MergeIt.Core.Configs.Elements
00008 {
00009     public class FieldElementIconComponent : MonoBehaviour
00010     {
00011         [SerializeField]
00012         private Material _blockedMaterial;
00013
00014         [SerializeField]
00015         private Image _image;
00016
00017         [SerializeField]
00018         private RectTransform _rectTransform;
00019
00020         private static Material _defaultMaterial;
00021
00022         public RectTransform RectTransform
00023         {
00024             get => _rectTransform;
00025         }
00026
00027         private void Awake()
00028         {
00029             _rectTransform = GetComponent<RectTransform>();
00030
00031             if (!_defaultMaterial)
00032             {
00033                 _defaultMaterial = new(Shader.Find("UI/Default"));
00034             }
00035         }
00036
00037         private void Start()
00038         {
00039             _rectTransform.Stretch();
00040         }
00041
00042         public void SetBlocked(bool locked)
00043         {
00044             if (locked)
00045             {
00046                 _image.material = _blockedMaterial;
00047             }
00048             else
00049             {
00050                 _image.material = _defaultMaterial;
00051             }
00052         }
00053
00054         public Image GetImage()
00055         {
00056             return _image;
00057         }
00058     }
00059 }
```

7.58 ConsumableSettings.cs

```
00001 // Copyright (c) 2024, Awessets
```

```
00002
00003 using System;
00004 using MergeIt.Core.Configs.Types;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Core.Configs.Elements
00008 {
00009     [Serializable]
0010     public struct ConsumableSettings
0011     {
0012         [SerializeField] private ConsumableTargetType _target;
0013         [SerializeField] private ConsumableOperationType _operation;
0014         [SerializeField] private int _value;
0015
0016         public ConsumableTargetType Target
0017         {
0018             get => _target;
0019         }
0020
0021         public ConsumableOperationType Operation
0022         {
0023             get => _operation;
0024         }
0025
0026         public int Value
0027         {
0028             get => _value;
0029         }
0030     }
0031 }
0032 }
```

7.59 CurrencySettings.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Configs.Types;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Core.Configs.Elements
00008 {
00009     [Serializable]
0010     public struct CurrencySettings
0011     {
0012         [SerializeField] private CurrencyType _currencyType;
0013         [SerializeField] private int _amount;
0014
0015         public CurrencyType Currency
0016         {
0017             get => _currencyType;
0018         }
0019
0020         public int Amount
0021         {
0022             get => _amount;
0023         }
0024     }
0025 }
```

7.60 ElementCommonSettings.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Core.Configs.Elements
00007 {
00008     [Serializable]
00009     public class ElementCommonSettings
0010     {
0011         [SerializeField, HideInInspector]
0012         private bool _expanded;
0013
0014         [SerializeField] private string _description;
0015         [SerializeField] private CurrencySettings _splitCostSettings;
0016         [SerializeField] private CurrencySettings _sellCostSettings;
0017         [SerializeField] private CurrencySettings _unlockCostSettings;
```

```

00018     [SerializeField] private string _name;
00019     [SerializeField] private FieldElementIconComponent _icon;
00020     [SerializeField] private bool _isConsumable;
00021     [SerializeField] private ConsumableSettings _consumableSettings;
00022
00023     public bool Expanded
00024     {
00025         get => _expanded;
00026         set => _expanded = value;
00027     }
00028
00029     public string Description
00030     {
00031         get => _description;
00032     }
00033
00034     public CurrencySettings SplitCostSettings
00035     {
00036         get => _splitCostSettings;
00037     }
00038
00039     public CurrencySettings SellCostSettings
00040     {
00041         get => _sellCostSettings;
00042     }
00043
00044     public CurrencySettings UnlockCostSettings
00045     {
00046         get => _unlockCostSettings;
00047     }
00048
00049     public ConsumableSettings ConsumableSettings
00050     {
00051         get => _consumableSettings;
00052     }
00053
00054     public bool IsConsumable
00055     {
00056         get => _isConsumable;
00057     }
00058
00059     public string Name
00060     {
00061         get => _name;
00062     }
00063
00064     public FieldElementIconComponent Icon
00065     {
00066         get => _icon;
00067     }
00068 }
00069 }
```

7.61 ElementGeneratorSettings.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Core.Configs.Elements
00008 {
00009     [Serializable]
00010     public class ElementGeneratorSettings
00011     {
00012         [SerializeField, HideInInspector]
00013         private bool _expanded;
00014
00015         [SerializeField] private int _stack;
00016         [SerializeField] private int _minDrop;
00017         [SerializeField] private int _maxDrop;
00018         [SerializeField] private int _chargeTime;
00019         [SerializeField] private int _dropCount;
00020         [SerializeField] private CurrencySettings _skipChargeCostSettings;
00021         [SerializeField] private bool _adSkip;
00022         [SerializeField] private bool _charged;
00023         [SerializeField] private int _energyCost;
00024         [SerializeField] private bool _isUnlimited;
00025         [SerializeField] private bool _needOpen;
00026         [SerializeField] private int _openTime;
00027         [SerializeField] private CurrencySettings _skipOpenCostSettings;
00028         [SerializeField] private bool _generateBlocked;
```

```
00029     [SerializeField] private List<GeneratableElement> _generateItems;
00030
00031     public bool Expanded
00032     {
00033         get => _expanded;
00034         set => _expanded = value;
00035     }
00036
00037     public int Stack
00038     {
00039         get => _stack;
00040     }
00041
00042     public int MinDrop
00043     {
00044         get => _minDrop;
00045     }
00046
00047     public int MaxDrop
00048     {
00049         get => _maxDrop;
00050     }
00051
00052     public int ChargeTime
00053     {
00054         get => _chargeTime;
00055     }
00056
00057     public int DropCount
00058     {
00059         get => _dropCount;
00060     }
00061
00062     public CurrencySettings SkipChargeCostSettings
00063     {
00064         get => _skipChargeCostSettings;
00065     }
00066
00067     public bool AdSkip
00068     {
00069         get => _adSkip;
00070     }
00071
00072     public bool Charged
00073     {
00074         get => _charged;
00075     }
00076
00077     public int EnergyCost
00078     {
00079         get => _energyCost;
00080     }
00081
00082     public bool IsUnlimited
00083     {
00084         get => _isUnlimited;
00085     }
00086
00087     public bool NeedOpen
00088     {
00089         get => _needOpen;
00090     }
00091
00092     public int OpenTime
00093     {
00094         get => _openTime;
00095     }
00096
00097     public CurrencySettings SkipOpenCostSettings
00098     {
00099         get => _skipOpenCostSettings;
00100     }
00101
00102     public bool GenerateBlocked
00103     {
00104         get => _generateBlocked;
00105     }
00106
00107     public List<GeneratableElement> GenerateItems
00108     {
00109         get => _generateItems;
00110     }
00111 }
00112 }
```

7.62 GeneratableElement.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004
00005 namespace MergeIt.Core.Configs.Elements
00006 {
00007     [Serializable]
00008     public class GeneratableElement
00009     {
00010         public ElementConfig Element;
00011         public int Possibility;
00012     }
00013 }
```

7.63 GameConfig.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Configs
00006 {
00007     [CreateAssetMenu(fileName = "GameConfig", menuName = "Merge Toolkit/Game config")]
00008     public class GameConfig : ScriptableObject
00009     {
00010         [Tooltip("Max count of rechargeable energy.")]
00011         public int EnergyCap;
00012
00013         [Tooltip("Required time to restore 1 energy point.")]
00014         public int EnergyRestoreTime;
00015         public int InitialSoftCurrency;
00016         public int InitialHardCurrency;
00017         public int InitialSplittersCount;
00018
00019         [Tooltip("Name of element's generation level, that would be displayed.")]
00020         public string LevelString;
00021         [Tooltip("Used in case you use a saving strategy that performs game save every n seconds")]
00022         public float SaveInterval = 10;
00023     }
00024 }
```

7.64 HintsConfig.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Configs.Hints
00006 {
00007     [CreateAssetMenu(fileName = "HintsConfig", menuName = "Merge Toolkit/Hints Config")]
00008     public class HintsConfig : ScriptableObject
00009     {
00010         [SerializeField]
00011         private float _interval;
00012
00013         public float Interval
00014         {
00015             get => _interval;
00016         }
00017     }
00018 }
```

7.65 GameIconsConfig.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Configs.Icons
00006 {
00007     [CreateAssetMenu(fileName = "GameIconsConfig", menuName = "Merge Toolkit/Game Icons Config")]
00008     public class GameIconsConfig : ScriptableObject
00009     {
```

```
00010     [SerializeField]
00011     private IconParameters[] _infoPanelCurrencyIcons;
00012
00013     public IconParameters[] InfoPanelCurrencyIcons
00014     {
00015         get => _infoPanelCurrencyIcons;
00016     }
00017 }
00018 }
```

7.66 IconParameters.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Configs.Types;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Core.Configs.Icons
00008 {
00009     [Serializable]
00010     public class IconParameters
00011     {
00012         [SerializeField]
00013         private CurrencyType _currencyType;
00014
00015         [SerializeField]
00016         private Sprite _currencyIcon;
00017
00018         public CurrencyType CurrencyType
00019         {
00020             get => _currencyType;
00021         }
00022
00023         public Sprite CurrencyIcon
00024         {
00025             get => _currencyIcon;
00026         }
00027     }
00028 }
```

7.67 InventoryConfig.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Core.Configs.Inventory
00007 {
00008     [CreateAssetMenu(fileName = "InventoryConfig", menuName = "Merge Toolkit/Inventory config")]
00009     public class InventoryConfig : ScriptableObject
00010     {
00011         public int InitialCapacity;
00012
00013         public CurrencySettings[] PaidCells;
00014     }
00015 }
```

7.68 LevelConfig.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Configs.Data;
00005 using MergeIt.Core.Schemes;
00006 using UnityEngine;
00007
00008 namespace MergeIt.Core.Configs
00009 {
00010     [CreateAssetMenu(fileName = "LevelConfig", menuName = "Merge Toolkit/Level config")]
00011     public class LevelConfig : ScriptableObject
00012     {
00013         public int FieldWidth;
00014         public int FieldHeight;
```

```

00015     public SchemeObject EvolutionsScheme;
00016     public List<LevelElementData> FieldElementsData;
00017 }
00018 }
```

7.69 LevelUpConfig.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Configs.LevelUp
00006 {
00007     [CreateAssetMenu(fileName = "LevelUpConfig", menuName = "Merge Toolkit/Level Up Config")]
00008     public class LevelUpConfig : ScriptableObject
00009     {
00010         public LevelUpParameters[] LevelUp;
00011     }
00012 }
00013 }
```

7.70 LevelUpParameters.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Configs.Elements;
00005
00006 namespace MergeIt.Core.Configs.LevelUp
00007 {
00008     [Serializable]
00009     public class LevelUpParameters
00010     {
00011         public int Experience;
00012         public ElementConfig[] Bonuses;
00013     }
00014 }
```

7.71 ConsumableOperationType.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Configs.Types
00004 {
00005     public enum ConsumableOperationType
00006     {
00007         Multiply,
00008         Divide,
00009         Add,
00010         Subtract,
00011     }
00012 }
```

7.72 ConsumableTargetType.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Configs.Types
00004 {
00005     public enum ConsumableTargetType
00006     {
00007         SoftCurrency,
00008         HardCurrency,
00009         Time,
00010         Energy,
00011         Experience,
00012         Splitter
00013     }
00014 }
```

7.73 CurrencyType.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Configs.Types
00004 {
00005     public enum CurrencyType
00006     {
00007         Soft,
00008         Hard,
00009         Splitter,
00010         Energy
00011     }
00012 }
```

7.74 ElementType.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Configs.Types
00004 {
00005     public enum ElementType
00006     {
00007         Regular,
00008         Generator
00009     }
00010 }
```

7.75 LayerParameters.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Core.Configs.Windows
00007 {
00008     [Serializable]
00009     public class LayerParameters : IEquatable<LayerParameters>, IComparable<LayerParameters>
00010     {
00011         [SerializeField]
00012         private string _name;
00013
00014         [SerializeField]
00015         private int _order;
00016
00017         public string Name
00018         {
00019             get => _name;
00020         }
00021
00022         public int Order
00023         {
00024             get => _order;
00025         }
00026
00027         public bool Equals(LayerParameters other)
00028         {
00029             if (ReferenceEquals(null, other)) return false;
00030             if (ReferenceEquals(this, other)) return true;
00031             return _name == other._name && _order == other._order;
00032         }
00033         public override bool Equals(object obj)
00034         {
00035             if (ReferenceEquals(null, obj)) return false;
00036             if (ReferenceEquals(this, obj)) return true;
00037             if (obj.GetType() != this.GetType()) return false;
00038             return Equals((LayerParameters)obj);
00039         }
00040
00041         public override int GetHashCode()
00042         {
00043             return HashCode.Combine(_name, _order);
00044         }
00045
00046         public int CompareTo(LayerParameters other)
00047         {
00048             if (ReferenceEquals(this, other)) return 0;
00049             if (ReferenceEquals(null, other)) return 1;
```

```

00050         return _order.CompareTo(other._order);
00051     }
00052 }
00053 }
```

7.76 LayersConfig.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Configs.Windows
00006 {
00007     [CreateAssetMenu(fileName = "LayersConfig", menuName = "Merge Toolkit/Windows/Layers config")]
00008     public class LayersConfig : ScriptableObject
00009     {
00010         [SerializeField]
00011         private LayerParameters[] _layers;
00012
00013         public LayerParameters[] Layers
00014         {
00015             get => _layers;
00016         }
00017     }
00018 }
```

7.77 IElementsStockComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004
00005 namespace MergeIt.Core.ElementsStock
00006 {
00007     public interface IElementsStockComponent
00008     {
00009         void SetupElement(ElementConfig elementConfig);
00010         void PopElement();
00011     }
00012 }
```

7.78 IElementsStockData.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Saves;
00004
00005 namespace MergeIt.Core.ElementsStock
00006 {
00007     public interface IElementsStockData : ISavable
00008     {
00009         string[] Elements { get; set; }
00010     }
00011 }
```

7.79 EvolutionProgressData.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using Newtonsoft.Json;
00005
00006 namespace MergeIt.Core.Evolutions
00007 {
00008     [Serializable]
00009     public class EvolutionProgressData
00010     {
00011         [JsonProperty("id")]
00012         public string EvolutionId { get; set; }
00013
00014         [JsonProperty("p")]
00015         public int Progress { get; set; }
00016     }
00017 }
```

7.80 IEvolutionsProgressData.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Saves;
00005
00006 namespace MergeIt.Core.Evolutions
00007 {
00008     public interface IEvolutionsProgressData : ISavable
00009     {
00010         List<EvolutionProgressData> EvolutionsProgress { get; }
00011     }
00012 }
```

7.81 FieldElementState.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.FieldElements
00004 {
00005     public enum FieldElementState
00006     {
00007         Idle,
00008         Hint
00009     }
00010 }
```

7.82 FieldElement.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Diagnostics;
00004
00005 namespace MergeIt.Core.FieldElements
00006 {
00007     [DebuggerDisplay("{ToString()}")]
00008     public class FieldElement : IFieldElement
00009     {
00010         public IConfigParameters ConfigParameters { get; set; }
00011         public IInfoParameters InfoParameters { get; set; }
00012         public IGeneratorParameters GeneratorParameters { get; set; }
00013         public IGeneratorOpenParameters GeneratorOpenParameters { get; set; }
00014         public IProduceParameters ProduceParameters { get; set; }
00015         public IProducedByParameters ProducedByParameters { get; set; }
00016
00017         public override string ToString()
00018         {
00019             return $"Type: {InfoParameters.Type}, Pos: {InfoParameters.LogicPosition}, Blocked: {InfoParameters.IsBlocked}";
00020         }
00021     }
00022 }
```

7.83 FieldElementData.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using Newtonsoft.Json;
00005
00006 namespace MergeIt.Core.FieldElements
00007 {
00008     [Serializable]
00009     public class FieldElementData
00010     {
00011         [JsonProperty("cp")]
00012         public SavedConfigParameters ConfigParameters;
00013         [JsonProperty("ip")]
00014         public SavedInfoParameters InfoParameters;
00015         [JsonProperty("gp")]
00016         public SavedGeneratorParameters GeneratorParameters;
00017         [JsonProperty("go")]
00018         public SavedGeneratorOpenParameters GeneratorOpenParameters;
00019     }
00020 }
```

7.84 GeneratableFieldElement.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004
00005 namespace MergeIt.Core.FieldElements
00006 {
00007     public class GeneratableFieldElement
00008     {
00009         public ElementConfig Config { get; set; }
00010         public int Possibility { get; set; }
00011     }
00012 }
```

7.85 GridPoint.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Diagnostics;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Core.FieldElements
00008 {
00009     [Serializable, DebuggerDisplay("ToString()")]
00010     public struct GridPoint : IEquatable<GridPoint>, IComparable<GridPoint>
00011     {
00012         public static bool operator ==(GridPoint p1, GridPoint p2)
00013         {
00014             return p1.Equals(p2);
00015         }
00016
00017         public static bool operator !=(GridPoint p1, GridPoint p2)
00018         {
00019             return !p1.Equals(p2);
00020         }
00021
00022         public static readonly GridPoint Default = new GridPoint(-1, -1);
00023
00024         [SerializeField]
00025         private int _x;
00026
00027         [SerializeField]
00028         private int _y;
00029
00030         public readonly int X
00031         {
00032             get => _x;
00033         }
00034
00035         public readonly int Y
00036         {
00037             get => _y;
00038         }
00039
00040         public GridPoint(int x, int y)
00041         {
00042             _x = x;
00043             _y = y;
00044         }
00045
00046         public GridPoint Copy()
00047         {
00048             return new GridPoint(_x, _y);
00049         }
00050
00051         public static GridPoint Create(int row, int column)
00052         {
00053             return new GridPoint(row, column);
00054         }
00055
00056         public bool Equals(GridPoint other)
00057         {
00058             return _x == other._x && _y == other._y;
00059         }
00060
00061         public int CompareTo(GridPoint other)
00062         {
00063             int xComparison = _x.CompareTo(other._x);
00064             if (xComparison != 0) return xComparison;
00065             return _y.CompareTo(other._y);
00066         }
}
```

```

00067     public override bool Equals(object obj)
00068     {
00069         return obj is GridPoint other && Equals(other);
00070     }
00071
00072     public override int GetHashCode()
00073     {
00074         unchecked
00075         {
00076             return _x * 397 ^ _y;
00077         }
00078     }
00079
00080     public override string ToString()
00081     {
00082         return $"(Row: {_x}, Column: {_y})";
00083     }
00084 }
00085 }
00086 }
```

7.86 IFieldElement.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.FieldElements
00004 {
00005     public interface IFieldElement
00006     {
00007         IConfigParameters ConfigParameters { get; set; }
00008         IInfoParameters InfoParameters { get; set; }
00009         IGeneratorParameters GeneratorParameters { get; set; }
00010         IGeneratorOpenParameters GeneratorOpenParameters { get; set; }
00011         IProduceParameters ProduceParameters { get; set; }
00012         IProducedByParameters ProducedByParameters { get; set; }
00013     }
00014 }
```

7.87 IFieldElementModel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.FieldElements
00004 {
00005     public interface IFieldElementModel
00006     {
00007         bool IsBusy { get; set; }
00008         bool IsLocked { get; set; }
00009         bool Selected { get; set; }
00010         int ClicksCount { get; set; }
00011         GridPoint Point { get; set; }
00012     }
00013 }
```

7.88 IFieldElementPresenter.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.MVP;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Core.FieldElements
00007 {
00008     public interface IFieldElementPresenter : IPresenter
00009     {
00010         FieldElementState State { get; }
00011         Transform Transform { get; }
00012         RectTransform RectTransform { get; }
00013         Canvas Canvas { get; }
00014         bool IsAvailable { get; }
00015         bool IsBusy { get; }
00016         bool IsLocked { get; }
00017
00018         void Update(IFieldElement fieldElement);
00019         void Activate(bool isActive);
```

```

00020     void Release();
00021     void SetBusy(bool isBusy);
00022     // void UpdateInitialPosition();
00023     void Select(bool isSelected);
00024     void StartDrag();
00025     void EndDrag();
00026     void SetParent(Transform parent, bool resetPosition);
00027     void SetPoint(GridPoint point);
00028     void SetLock(bool block);
00029     void Remove();
00030     void ResetPosition();
00031     IFIELD_ELEMENTVIEW GetView();
00032     IFIELD_ELEMENTMODEL GetModel();
00033     void SetState(FieldElementState state);
00034 }
00035 }
```

7.89 IFIELD_ELEMENTVIEW.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.MVP;
00004 using UnityEngine;
00005 using UnityEngine.UI;
00006
00007 namespace MergeIt.Core.FieldElements
00008 {
00009     public interface IFIELD_ELEMENTVIEW : IView
00010     {
00011         RectTransform RectTransform { get; }
00012         Canvas Canvas { get; }
00013         GraphicRaycaster GraphicRaycaster { get; }
00014         void Lock(bool isLocked);
00015         void ResetState();
00016     }
00017 }
```

7.90 ConfigParameters.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Data;
00004 using MergeIt.Core.Configs.Elements;
00005
00006 namespace MergeIt.Core.FieldElements
00007 {
00008     public class ConfigParameters : IConfigParameters
00009     {
00010         public ElementConfig ElementConfig { get; set; }
00011         public EvolutionData EvolutionData { get; set; }
00012     }
00013 }
```

7.91 GeneratorOpenParameters.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Utils;
00004
00005 namespace MergeIt.Core.FieldElements
00006 {
00007     public class GeneratorOpenParameters : IGeneratorOpenParameters
00008     {
00009         public bool IsOpening => StartOpeningTime != 0;
00010         public long StartOpeningTime { get; set; }
00011         public Bindable<float> RemainingTime { get; set; } = new();
00012
00013         public void CopyFrom(SavedGeneratorOpenParameters other)
00014         {
00015             StartOpeningTime = other.StartOpeningTime;
00016         }
00017     }
00018 }
```

7.92 GeneratorParameters.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Utils;
00004
00005 namespace MergeIt.Core.FieldElements
00006 {
00007     public class GeneratorParameters : IGeneratorParameters
00008     {
00009         public bool Charging => StartChargingTime != 0;
00010
00011         public int AvailableToDrop { get; set; }
00012         public int ChargedCount { get; set; }
00013         public long StartChargingTime { get; set; }
00014         public float RemainChargeTime { get; set; }
00015         public float MinDropFullChargeTime { get; set; }
00016         public int DroppedElements { get; set; }
00017
00018         public Bindable<float> MinDropChargeTime { get; set; } = new();
00019
00020         public void CopyFrom(SavedGeneratorParameters other)
00021         {
00022             AvailableToDrop = other.AvailableToDrop;
00023             StartChargingTime = other.StartChargingTime;
00024             DroppedElements = other.DroppedElements;
00025             ChargedCount = other.ChargedCount;
00026         }
00027     }
00028 }
```

7.93 IConfigParameters.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Data;
00004 using MergeIt.Core.Configs.Elements;
00005
00006 namespace MergeIt.Core.FieldElements
00007 {
00008     public interface IConfigParameters
00009     {
00010         ElementConfig ElementConfig { get; set; }
00011         EvolutionData EvolutionData { get; set; }
00012     }
00013 }
```

7.94 IGeneratorOpenParameters.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Utils;
00004
00005 namespace MergeIt.Core.FieldElements
00006 {
00007     public interface IGeneratorOpenParameters
00008     {
00009         bool IsOpening { get; }
00010         long StartOpeningTime { get; set; }
00011         Bindable<float> RemainingTime { get; set; }
00012
00013         void CopyFrom(SavedGeneratorOpenParameters other);
00014     }
00015 }
```

7.95 IGeneratorParameters.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Utils;
00004
00005 namespace MergeIt.Core.FieldElements
00006 {
00007     public interface IGeneratorParameters
00008     {
```

```

00009     bool Charging { get; }
00010
00011     int AvailableToDrop { get; set; }
00012     int ChargedCount { get; set; }
00013     long StartChargingTime { get; set; }
00014     Bindable<float> MinDropChargeTime { get; set; }
00015     float MinDropFullChargeTime { get; set; }
00016     float RemainChargeTime { get; set; }
00017     int DroppedElements { get; set; }
00018
00019     void CopyFrom(SavedGeneratorParameters other);
00020 }
00021
00022 }
```

7.96 IInfoParameters.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Types;
00004
00005 namespace MergeIt.Core.FieldElements
00006 {
00007     public interface IInfoParameters
00008     {
00009         GridPoint LogicPosition { get; set; }
00010         bool IsBlocked { get; set; }
00011         public ElementType Type { get; set; }
00012         public string Name { get; set; }
00013         public string Description { get; set; }
00014     }
00015 }
```

7.97 InfoParameters.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Types;
00004
00005 namespace MergeIt.Core.FieldElements
00006 {
00007     public class InfoParameters : IInfoParameters
00008     {
00009         public GridPoint LogicPosition { get; set; }
00010         public bool IsBlocked { get; set; }
00011         public ElementType Type { get; set; }
00012         public string Name { get; set; }
00013         public string Description { get; set; }
00014     }
00015 }
```

7.98 IProducedByParameters.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Configs.Elements;
00005
00006 namespace MergeIt.Core.FieldElements
00007 {
00008     public interface IProducedByParameters
00009     {
00010         List<ElementConfig> Elements { get; set; }
00011     }
00012
00013 }
```

7.99 IProduceParameters.cs

```

00001 // Copyright (c) 2024, Awessets
00002
```

```
00003 using System.Collections.Generic;
00004
00005 namespace MergeIt.Core.FieldElements
00006 {
00007     public interface IProduceParameters
00008     {
00009         List<GeneratableFieldElement> Elements { get; set; }
00010     }
00011 }
```

7.100 ProducedByParameters.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Configs.Elements;
00005
00006 namespace MergeIt.Core.FieldElements
00007 {
00008     public class ProducedByParameters : IProducedByParameters
00009     {
00010         public List<ElementConfig> Elements { get; set; }
00011     }
00012 }
```

7.101 ProduceParameters.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004
00005 namespace MergeIt.Core.FieldElements
00006 {
00007     public class ProduceParameters : IProduceParameters
00008     {
00009         public List<GeneratableFieldElement> Elements { get; set; }
00010     }
00011 }
```

7.102 SavedConfigParameters.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using Newtonsoft.Json;
00005
00006 namespace MergeIt.Core.FieldElements
00007 {
00008     [Serializable]
00009     public class SavedConfigParameters
00010     {
00011         [JsonProperty("id")]
00012         public string ElementId { get; set; }
00013         [JsonProperty("eid")]
00014         public string EvolutionId { get; set; }
00015     }
00016 }
```

7.103 SavedGeneratorOpenParameters.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using Newtonsoft.Json;
00005
00006 namespace MergeIt.Core.FieldElements
00007 {
00008     [Serializable]
00009     public class SavedGeneratorOpenParameters
00010     {
00011         [JsonProperty("s")]
00012     }
00013 }
```

```

00012     public long StartOpeningTime { get; set; }
00013
00014     public void CopyFrom(IGeneratorOpenParameters other)
00015     {
00016         StartOpeningTime = other.StartOpeningTime;
00017     }
00018 }
00019 }
```

7.104 SavedGeneratorParameters.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using Newtonsoft.Json;
00005
00006 namespace MergeIt.Core.FieldElements
00007 {
00008     [Serializable]
00009     public class SavedGeneratorParameters
00010     {
00011         [JsonProperty("a")]
00012         public int AvailableToDrop { get; set; }
00013         [JsonProperty("s")]
00014         public long StartChargingTime { get; set; }
00015         [JsonProperty("d")]
00016         public int DroppedElements { get; set; }
00017         [JsonProperty("c")]
00018         public int ChargedCount { get; set; }
00019
00020         public void CopyFrom(IGeneratorParameters other)
00021         {
00022             AvailableToDrop = other.AvailableToDrop;
00023             StartChargingTime = other.StartChargingTime;
00024             DroppedElements = other.DroppedElements;
00025             ChargedCount = other.ChargedCount;
00026         }
00027     }
00028 }
```

7.105 SavedInfoParameters.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using Newtonsoft.Json;
00005
00006 namespace MergeIt.Core.FieldElements
00007 {
00008     [Serializable]
00009     public class SavedInfoParameters
00010     {
00011         [JsonProperty("p")]
00012         public GridPoint LogicPosition { get; set; }
00013         [JsonProperty("b")]
00014         public bool IsBlocked { get; set; }
00015     }
00016 }
```

7.106 CurvesHelper.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Helpers
00006 {
00007     public static class CurvesHelper
00008     {
00009         public static AnimationCurve CopyCurveFrom(AnimationCurve curve, float originValue)
00010         {
00011             var newCurve = new AnimationCurve();
00012             for (int i = 0; i < curve.keys.Length; i++)
00013             {
00014                 Keyframe kf = curve.keys[i];
```

```

00015         kf.value += originValue;
00016         newCurve.AddKey(kf);
00017     }
00018
00019     return newCurve;
00020 }
00021
00022 public static AnimationCurve CopyCurveFrom(AnimationCurve curve, float originValue, float
lastValue)
00023 {
00024     var newCurve = new AnimationCurve();
00025     for (int i = 0; i < curve.keys.Length; i++)
00026     {
00027         Keyframe kf = curve.keys[i];
00028         if (i == curve.keys.Length - 1)
00029         {
00030             kf.value += lastValue;
00031         }
00032         else
00033         {
00034             kf.value += originValue;
00035         }
00036
00037         newCurve.AddKey(kf);
00038     }
00039
00040     return newCurve;
00041 }
00042 }
00043 }
```

7.107 ListExtensions.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004
00005 namespace MergeIt.Core.Helpers
00006 {
00007     public static class ListExtensions
00008     {
00009         public static IList<int> GenerateShuffledArray(int size)
0010     {
0011         IList<int> array = new int[size];
0012
0013         for (int i = 0; i < size; i++)
0014         {
0015             array[i] = i;
0016         }
0017
0018         array.Shuffle();
0019
0020         return array;
0021     }
0022
0023     public static void Shuffle<TSource>(this IList<TSource> list)
0024     {
0025         for (int i = list.Count - 1; i > 0; i--)
0026         {
0027             int j = UnityEngine.Random.Range(0, i + 1);
0028
0029             (list[i], list[j]) = (list[j], list[i]);
0030         }
0031     }
0032 }
0033 }
```

7.108 NumericExtensions.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Types;
00004 using MergeIt.Core.Utils;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Core.Helpers
00008 {
00009     public static class NumericExtensions
0010     {
```

```

00011     public static void ApplyOperation(this Bindable<int> target, ConsumableOperationType opType,
00012         int value, bool holdValue)
00013     {
00014         if (holdValue)
00015         {
00016             target.Hold();
00017         }
00018         switch (opType)
00019         {
00020             case ConsumableOperationType.Multiply:
00021                 target.Value *= value;
00022                 break;
00023
00024             case ConsumableOperationType.Divide:
00025                 target.Value /= value;
00026                 break;
00027
00028             case ConsumableOperationType.Add:
00029                 target.Value += value;
00030                 break;
00031
00032             case ConsumableOperationType.Subtract:
00033                 int targetValue = target.Value;
00034                 targetValue = Mathf.Clamp(targetValue - value, 0, int.MaxValue);
00035                 target.Value = targetValue;
00036                 break;
00037         }
00038     }
00039 }
00040 }
```

7.109 RectTransformExtensions.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Helpers
00006 {
00007     public static class RectTransformExtensions
00008     {
00009         public static void Stretch(this RectTransform rectTransform)
00010         {
00011             rectTransform.anchorMin = new Vector2(0, 0);
00012             rectTransform.anchorMax = new Vector2(1, 1);
00013             rectTransform.offsetMin = new Vector2(0, 0);
00014             rectTransform.offsetMax = new Vector2(0, 0);
00015         }
00016     }
00017 }
```

7.110 SavesHelper.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.IO;
00004 using UnityEditor;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Editor.Helpers
00008 {
00009     public class SavesHelper : UnityEditor.Editor
00010     {
00011         [MenuItem("Tools/Merge Toolkit/Clear saves", false, 4)]
00012         public static void ClearSaves()
00013         {
00014             string path = Path.Combine(Application.streamingAssetsPath, "Saves");
00015
00016             if (Directory.Exists(path))
00017             {
00018                 Directory.Delete(path, true);
00019                 string meta = Path.ChangeExtension(path, "meta");
00020
00021                 if (File.Exists(meta))
00022                 {
00023                     File.Delete(meta);
00024                 }
00025             }
00026         }
00027     }
00028 }
```

```

00026             AssetDatabase.Refresh();
00027         }
00028     }
00029 }
00030 }
```

7.111 SavesHelper.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using System.Linq;
00006 using System.Reflection;
00007 using MergeIt.Core.Saves;
00008
00009 namespace MergeIt.Core.Helpers
00010 {
00011     public static class SavesHelper
00012     {
00013         public static readonly Dictionary<Type, (string Name, string Extension)> SavableData = new();
00014
00015         static SavesHelper()
00016         {
00017             Type interfaceType = typeof(ISavable);
00018             IEnumerable<Type> allTypes = AppDomain.CurrentDomain.GetAssemblies()
00019                 .SelectMany(assembly => assembly.GetTypes())
00020                 .Where(type => !type.IsAbstract && interfaceType.IsAssignableFrom(type));
00021
00022             foreach (Type type in allTypes)
00023             {
00024                 var savable = type.GetCustomAttribute<SavableAttribute>();
00025                 if (savable != null)
00026                 {
00027                     SavableData[type] = (savable.Name, savable.Extension);
00028                 }
00029             }
00030         }
00031
00032         public static string GetFileName<T>() where T : class, ISavable
00033         {
00034             Type type = typeof(T);
00035
00036             if (SavableData.TryGetValue(type, out var data))
00037             {
00038                 return data.Name;
00039             }
00040
00041             return string.Empty;
00042         }
00043
00044         public static string GetExtension<T>() where T : class, ISavable
00045         {
00046             Type type = typeof(T);
00047
00048             if (SavableData.TryGetValue(type, out var data))
00049             {
00050                 return data.Extension;
00051             }
00052
00053             return string.Empty;
00054         }
00055
00056         public static string GetFileNameWithExtension<T>() where T : class, ISavable
00057         {
00058             Type type = typeof(T);
00059
00060             if (SavableData.TryGetValue(type, out var data))
00061             {
00062                 return $"{data.Name}.{data.Extension}";
00063             }
00064
00065             return string.Empty;
00066         }
00067     }
00068 }
```

7.112 TimeExtensions.cs

```
00001 // Copyright (c) 2024, Awessets
```

```

00002
00003 using System;
00004 using System.Text;
00005
00006 namespace MergeIt.Core.Helpers
00007 {
00008     public static class TimeExtensions
00009     {
00010         private static readonly DateTimeOffset UnixEpoch = new(1970, 1, 1, 0, 0, 0, TimeSpan.Zero);
00011
00012         public static DateTimeOffset FromUnixTimeMilliseconds(long milliseconds)
00013         {
00014             return UnixEpoch.AddMilliseconds(milliseconds);
00015         }
00016
00017         public static string FormatTime(this long value)
00018         {
00019             TimeSpan timeSpan = TimeSpan.FromSeconds(value);
00020
00021             if (timeSpan.Hours > 0)
00022             {
00023                 return $"{timeSpan:hh\\:mm\\:ss}";
00024             }
00025
00026             return $"{timeSpan:mm\\:ss}";
00027         }
00028
00029         public static string FormatTime(this float value)
00030         {
00031             TimeSpan timeSpan = TimeSpan.FromSeconds(value);
00032
00033             if (timeSpan.Hours > 0)
00034             {
00035                 return $"{timeSpan:hh\\:mm\\:ss}";
00036             }
00037
00038             return $"{timeSpan:mm\\:ss}";
00039         }
00040
00041         public static string FormatTime(this int value)
00042         {
00043             TimeSpan timeSpan = TimeSpan.FromSeconds(value);
00044             var sb = new StringBuilder();
00045
00046             if (timeSpan.Hours > 0)
00047             {
00048                 sb.Append($"{timeSpan.Hours}h");
00049             }
00050
00051             if (timeSpan.Minutes > 0)
00052             {
00053                 sb.Append($"{timeSpan.Minutes}m");
00054             }
00055
00056             if (timeSpan.Seconds > 0)
00057             {
00058                 sb.Append($"{timeSpan.Seconds}s");
00059             }
00060
00061             return sb.ToString();
00062         }
00063     }
00064 }

```

7.113 IInventoryData.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Saves;
00005
00006 namespace MergeIt.Core.Inventory
00007 {
00008     public interface IInventoryData : ISavable
00009     {
00010         int InventorySize { get; set; }
00011         FieldElementData[] InventoryElements { get; set; }
00012     }
00013 }

```

7.114 IMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Messages
00004 {
00005     public interface IMessage
00006     {
00007
00008     }
00009 }
```

7.115 IMessageBus.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004
00005 namespace MergeIt.Core.Messages
00006 {
00007     public interface IMessageBus
00008     {
00009         void DefineMessage<T>() where T : IMessage;
00010         void AddListener<T>(Action<T> callback) where T : IMessage;
00011         void RemoveListener<T>(Action<T> callback) where T : IMessage;
00012         void Fire<T>(T messageData) where T : IMessage;
00013         void Fire<T>() where T : IMessage, new();
00014     }
00015 }
```

7.116 MessageBus.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using System.Linq;
00006 using MergeIt.SimpleDI.ReservedInterfaces;
00007 using UnityEngine;
00008
00009 namespace MergeIt.Core.Messages
00010 {
00011     public class MessageBus : IMessageBus, IInitializable
00012     {
00013         private static readonly Dictionary<Type, List<object>> Messages = new Dictionary<Type,
List<object>>();
00014
00015         public void Initialize()
00016         {
00017             Type interfaceType = typeof(IMessage);
00018             IEnumerable<Type> allTypes = AppDomain.CurrentDomain.GetAssemblies()
00019                 .SelectMany(assembly => assembly.GetTypes())
00020                 .Where(type => !type.IsAbstract && interfaceType.IsAssignableFrom(type));
00021
00022             foreach (Type type in allTypes)
00023             {
00024                 Messages[type] = new List<object>();
00025             }
00026         }
00027
00028         public void DefineMessage<T>() where T : IMessage
00029         {
00030             Type type = typeof(T);
00031
00032             if (!Messages.ContainsKey(type))
00033             {
00034                 Messages.Add(type, new List<object>());
00035             }
00036         }
00037
00038         public void AddListener<T>(Action<T> callback) where T : IMessage
00039         {
00040             Type type = typeof(T);
00041
00042             if (!Messages.ContainsKey(type))
00043             {
00044                 Debug.LogWarning($"Message with type {type} was not defined and will added to
cache.");
00045             }
00046         }
00047     }
00048 }
```

```

00046             Messages[type] = new List<object>();
00047         }
00048
00049         Messages[type].Add(callback);
00050     }
00051
00052     public void RemoveListener<T>(Action<T> callback) where T : IMessage
00053     {
00054         if (Messages.TryGetValue(typeof(T), out List<object> callbacks))
00055         {
00056             callbacks.Remove(callback);
00057         }
00058     }
00059
00060     public void Fire<T>(T messageData) where T : IMessage
00061     {
00062         Type type = typeof(T);
00063
00064         if (Messages.TryGetValue(type, out var message))
00065         {
00066             for (var index = 0; index < message.Count;)
00067             {
00068                 var callback = message[index];
00069                 Action<T> action = callback as Action<T>;
00070
00071                 if (action == null)
00072                 {
00073                     message.RemoveAt(index);
00074
00075                     continue;
00076                 }
00077
00078                 action(messageData);
00079
00080                 index++;
00081             }
00082         }
00083     }
00084
00085     public void Fire<T>() where T : IMessage, new()
00086     {
00087         Type type = typeof(T);
00088
00089         if (Messages.TryGetValue(type, out var message))
00090         {
00091             for (var index = 0; index < message.Count;)
00092             {
00093                 var callback = message[index];
00094                 Action<T> action = callback as Action<T>;
00095
00096                 if (action == null)
00097                 {
00098                     message.RemoveAt(index);
00099
00100                     continue;
00101                 }
00102
00103                 action(new T());
00104
00105                 index++;
00106             }
00107         }
00108     }
00109 }
00110 }
```

7.117 IMModel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.MVP
00004 {
00005     public interface IMModel
00006     {
00007
00008     }
00009
00010 }
```

7.118 IPresenter.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004
00005 namespace MergeIt.Core.MVP
00006 {
00007     public interface IPresenter : IDisposable
00008     {
00009         void Initialize(IView view);
00010     }
00011 }
```

7.119 IView.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Core.MVP
00007 {
00008     public interface IView
00009     {
00010         event Action InitializeEvent;
00011         event Action DestroyEvent;
00012
00013         GameObject GameObject { get; }
00014
00015         void Initialize();
00016     }
00017 }
```

7.120 Model.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.MVP
00004 {
00005     public class Model : IModel
00006     {
00007     }
00008 }
```

7.121 Presenter.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.MVP
00006 {
00007     public abstract class Presenter<TView, TModel> : IPresenter
00008         where TView : Component, IView
00009         where TModel : IModel, new()
00010     {
00011         public TView View { get; private set; }
00012         public TModel Model { get; } = new();
00013
00014         public void Initialize(IView view)
00015         {
00016             View = view as TView;
00017             View.DestroyEvent += Dispose;
00018
00019             OnInitialize(View);
00020         }
00021
00022         public void Dispose()
00023         {
00024             if (View)
00025             {
00026                 View.DestroyEvent -= Dispose;
00027             }
00028         }
00029 }
```

```

00029         OnDispose();
00030     }
00031
00032     protected virtual void OnInitialize(TView view)
00033     {
00034
00035     }
00036
00037     protected virtual void OnDispose()
00038     {
00039
00040     }
00041 }
00042 }
```

7.122 View.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Core.MVP
00007 {
00008     public abstract class View : MonoBehaviour, IView
00009     {
00010         public event Action InitializeEvent;
00011         public event Action DestroyEvent;
00012
00013         public GameObject GameObject
00014         {
00015             get => gameObject;
00016         }
00017
00018         public virtual void Initialize()
00019         {
00020             InitializeEvent?.Invoke();
00021         }
00022
00023         protected virtual void OnDestroy()
00024         {
00025             DestroyEvent?.Invoke();
00026         }
00027     }
00028 }
```

7.123 GameSaveType.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004
00005 namespace MergeIt.Core.Saves
00006 {
00007     [Flags]
00008     public enum GameSaveType
00009     {
00010         Field = 1 << 0,
00011         Inventory = 1 << 1,
00012         User = 1 << 2,
00013         Stock = 1 << 3,
00014         EvolutionsProgress = 1 << 4,
00015         All = Field | Inventory | User | Stock | EvolutionsProgress
00016     }
00017 }
```

7.124 ISavable.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Saves
00004 {
00005     public interface ISavable
00006     {
00007     }
00008 }
```

7.125 SavableAttribute.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004
00005 namespace MergeIt.Core.Saves
00006 {
00007     public class SavableAttribute : Attribute
00008     {
00009         public string Name { get; }
00010         public string Extension { get; }
00011
00012         public SavableAttribute(string name, string extension)
00013         {
00014             Name = name;
00015             Extension = extension;
00016         }
00017     }
00018 }
```

7.126 EdgeData.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004
00005 namespace MergeIt.Core.Schemes.Data
00006 {
00007     [Serializable]
00008     public class EdgeData
00009     {
00010         public string OutputNodeID;
00011         public string InputNodeID;
00012         public string OutputPortName;
00013         public string InputPortName;
00014     }
00015 }
```

7.127 ElementNodeData.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Configs.Elements;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Core.Schemes.Data
00008 {
00009     [Serializable]
00010     public class ElementNodeData : IELEMENTNODEDATA
00011     {
00012         [SerializeField] private string _id = Guid.NewGuid().ToString();
00013         [SerializeField] private ElementConfig _elementConfig;
00014         [SerializeField] private Rect _position;
00015
00016         public string Id
00017         {
00018             get => _id;
00019             set => _id = value;
00020         }
00021
00022         public ElementConfig ElementConfig
00023         {
00024             get => _elementConfig;
00025             set => _elementConfig = value;
00026         }
00027
00028         public Rect Position
00029         {
00030             get => _position;
00031             set => _position = value;
00032         }
00033
00034         public T Copy<T>() where T : IELEMENTNODEDATA, new()
00035         {
00036             var copiedData = new T
00037             {
00038                 Position = Position,
```

```
00039         ElementConfig = ElementConfig
00040     };
00041
00042     return copiedData;
00043 }
00044
00045
00046 }
```

7.128 EvolutionNodeData.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Core.Schemes.Data
00007 {
00008     [Serializable]
00009     public class EvolutionNodeData : IEvolutionNodeData
00010     {
00011         [SerializeField] private string _id;
00012         [SerializeField] private string _name;
00013         [SerializeField] private string _description;
00014         [SerializeField] private bool _discovered;
00015         [SerializeField] private Rect _position;
00016
00017         public string Id
00018     {
00019         get
00020         {
00021             if (string.IsNullOrEmpty(_id))
00022             {
00023                 _id = Guid.NewGuid().ToString();
00024             }
00025
00026             return _id;
00027         }
00028         set => _id = value;
00029     }
00030
00031         public bool Discovered
00032     {
00033         get => _discovered;
00034         set => _discovered = value;
00035     }
00036
00037         public string Name
00038     {
00039         get => _name;
00040         set => _name = value;
00041     }
00042
00043         public string Description
00044     {
00045         get => _description;
00046         set => _description = value;
00047     }
00048
00049         public Rect Position
00050     {
00051         get => _position;
00052         set => _position = value;
00053     }
00054
00055         public T Copy<T>() where T : IEvolutionNodeData, new()
00056     {
00057         var copiedData = new T
00058     {
00059         Position = Position,
00060         Discovered = Discovered,
00061         Name = Name,
00062         Description = Description
00063     };
00064
00065         return copiedData;
00066     }
00067 }
00068 }
```

7.129 IElementNodeData.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Core.Schemes.Data
00007 {
00008     public interface IElementNodeData
00009     {
00010         string Id { get; set; }
00011         ElementConfig ElementConfig { get; set; }
00012         Rect Position { get; set; }
00013
00014         T Copy<T>() where T : IElementNodeData, new();
00015     }
00016 }
```

7.130 IEvolutionNodeData.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Schemes.Data
00006 {
00007     public interface IEvolutionNodeData
00008     {
00009         string Id { get; set; }
00010         public string Name { get; set; }
00011         public string Description { get; set; }
00012         public bool Discovered { get; set; }
00013
00014         Rect Position { get; set; }
00015
00016         T Copy<T>() where T : IEvolutionNodeData, new();
00017     }
00018 }
```

7.131 SchemeData.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using MergeIt.Core.Configs.Data;
00006 using MergeIt.Core.Schemes.Data;
00007
00008 namespace MergeIt.Core.Schemes
00009 {
00010     [Serializable]
00011     public class SchemeData
00012     {
00013         public List<ElementNodeData> ElementsNodesData;
00014         public List<EvolutionNodeData> EvolutionsNodesData;
00015         public List<EdgeData> Edges;
00016
00017         public List<EvolutionData> EvolutionsData;
00018     }
00019 }
```

7.132 SchemeObject.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Configs.Data;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Core.Schemes
00008 {
00009     public class SchemeObject : ScriptableObject
00010     {
00011         public SchemeData SchemeData;
```

```

00012     public List<EvolutionData> Evolution
00013     {
00014         get => SchemeData?.EvolutionsData;
00015     }
00016 }
00017 }
00018 }
```

7.133 IConfigsService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using Cysharp.Threading.Tasks;
00006 using MergeIt.Core.Configs;
00007 using MergeIt.Core.Configs.Data;
00008 using MergeIt.Core.Configs.Effects;
00009 using MergeIt.Core.Configs.Elements;
00010 using MergeIt.Core.Configs.Hints;
00011 using MergeIt.Core.Configs.Inventory;
00012 using MergeIt.Core.Configs.LevelUp;
00013 using MergeIt.Core.Configs.Types;
00014 using MergeIt.Core.FieldElements;
00015 using UnityEngine;
00016
00017 namespace MergeIt.Core.Services
00018 {
00019     public interface IConfigsService
00020     {
00021         UniTask Load();
00022         LevelConfig LevelConfig { get; }
00023         GameConfig GameConfig { get; }
00024         InventoryConfig InventoryConfig { get; }
00025         HintsConfig HintsConfig { get; }
00026         Sprite GetCurrencyIcon(CurrencyType type);
00027         EvolutionData GetEvolutionData(string evolutionId);
00028         string GetEvolutionIdByElement(ElementConfig element);
00029         EvolutionData GetEvolutionByElement(ElementConfig element);
00030         T GetEffectConfig<T>(string type) where T : EffectConfig;
00031         LevelUpParameters GetLevelUpData(int currentLevel);
00032         IFIELDLEMENTVIEW GetElementPrefab(ElementType type);
00033         IEnumerable<ElementConfig> GetConfigs(Func<ElementConfig, bool> predicate = null);
00034         ElementConfig GetConfig(string id);
00035     }
00036 }
```

7.134 ICurrencyService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004
00005 namespace MergeIt.Core.Services
00006 {
00007     public interface ICurrencyService
00008     {
00009         bool TryPay(CurrencySettings currencySettings);
00010         void Sell(CurrencySettings currencySettings);
00011     }
00012 }
```

7.135 IELEMENTSERVICE.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004
00005 namespace MergeIt.Core.Services
00006 {
00007     public interface IELEMENTSERVICE
00008     {
00009         void TrySell(IFIELDELEMENT fieldElement);
00010         void TryUnlock(IFIELDELEMENT fieldElement);
00011         void TrySplit(IFIELDELEMENT fieldElement);
00012     }
00013 }
```

7.136 IElementsStockService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.ElementsStock;
00005
00006 namespace MergeIt.Core.Services
00007 {
00008     public interface IElementsStockService
00009     {
00010         void CreateStock();
00011         void SetupStock(IElementsStockData stockData);
00012         void Add(ElementConfig elementConfig);
00013         bool Remove();
00014         ElementConfig GetNext();
00015         ElementConfig GetCurrent();
00016         IElementsStockData GetData();
00017     }
00018 }
```

7.137 IEnergyService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Services
00004 {
00005     public interface IEnergyService
00006     {
00007     }
00008 }
```

7.138 IEvolutionsService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Configs.Data;
00005 using MergeIt.Core.Configs.Elements;
00006 using MergeIt.Core.Evolutions;
00007
00008 namespace MergeIt.Core.Services
00009 {
00010     public interface IEvolutionsService
00011     {
00012         List<ElementConfig> GetEvolutionChain(EvolutionData evolutionData);
00013         List<ElementConfig> GetGeneratedBy(ElementConfig config);
00014         List<ElementConfig> GetGenerates(ElementConfig config);
00015
00016         void SetupEvolutionsProgress(IEvolutionsProgressData data);
00017         IEvolutionsProgressData GetData();
00018         void CreateEvolutionsProgress();
00019         void UpdateProgress(ElementConfig elementId);
00020         int GetEvolutionProgress(string id);
00021     }
00022 }
```

7.139 IGameFieldActionsService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Services
00004 {
00005     public interface IGameFieldActionsService
00006     {
00007     }
00008 }
```

7.140 IGameFieldService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.FieldElements;
00005
00006 namespace MergeIt.Core.Services
00007 {
00008     public interface IGameFieldService
00009     {
00010         GridPoint? GetFreeCell();
00011         IFIELDElement CreateNewElement(ElementConfig config, GridPoint point, bool blocked = false);
00012     }
00013 }
```

7.141 IGameLoadService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Saves;
00004
00005 namespace MergeIt.Core.Services
00006 {
00007     public interface IGameLoadService
00008     {
00009         T Load<T>() where T : class, ISavable;
00010     }
00011 }
```

7.142 IGameSaveService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using Cysharp.Threading.Tasks;
00004 using MergeIt.Core.Saves;
00005
00006 namespace MergeIt.Core.Services
00007 {
00008     public interface IGameSaveService
00009     {
00010         UniTask Save(GameSaveType saveType);
00011     }
00012 }
```

7.143 IGameService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Services
00004 {
00005     public interface IGameService
00006     {
00007     }
00008 }
```

7.144 IGeneratorsService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004
00005 namespace MergeIt.Core.Services
00006 {
00007     public interface IGeneratorsService
00008     {
00009         void TryOpen(IFIELDElement generator);
00010         void TrySkipOpening(IFIELDElement generator);
00011         void TrySkipCharging(IFIELDElement generator);
00012     }
00013 }
```

7.145 IInfoPanelService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Services
00004 {
00005     public interface IInfoPanelService
00006     {
00007     }
00008
00009 }
```

7.146 IInventoryService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Inventory;
00005
00006 namespace MergeIt.Core.Services
00007 {
00008     public interface IInventoryService
00009     {
00010         void CreateInventory();
00011         void SetupInventory(IInventoryData inventoryData);
00012         IInventoryData GetData();
00013         void Add(IFieldElement element);
00014         bool Remove(IFieldElement fieldElement);
00015         void OpenWindow();
00016     }
00017 }
```

7.147 IResourcesLoaderService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.Services
00006 {
00007     public interface IResourcesLoaderService
00008     {
00009         T GetObject<T>(string path) where T : Object;
00010     }
00011 }
```

7.148 IUserProgressService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004
00005 namespace MergeIt.Core.Services
00006 {
00007     public interface IUserProgressService
00008     {
00009         int GetCurrentLevelMaxExp();
00010         bool CanLevelUp();
00011         ElementConfig[] GetLevelUpPrizes();
00012     }
00013 }
```

7.149 IUserService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.User;
00004
00005 namespace MergeIt.Core.Services
00006 {
```

```

00007     public interface IUserService
00008     {
00009         void CreateUser();
0010         void SetupUser(IUserData userData);
0011     }
0012 }
```

7.150 IUserData.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Saves;
00004
00005 namespace MergeIt.Core.User
00006 {
00007     public interface IUserData : ISavable
00008     {
00009         string Name { get; set; }
0010         int Energy { get; set; }
0011         int SoftCurrency { get; set; }
0012         int HardCurrency { get; set; }
0013         int Splitters { get; set; }
0014         int Level { get; set; }
0015         int Experience { get; set; }
0016         long EnergyRestoringStartTime { get; set; }
0017     }
0018 }
```

7.151 BlackoutComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Core.WindowSystem
00006 {
00007     public class BlackoutComponent : MonoBehaviour
00008     {
00009         private Canvas _canvas;
0010
0011         private void Awake()
0012         {
0013             _canvas = GetComponent<Canvas>();
0014         }
0015
0016         public void SetActive(bool active)
0017         {
0018             gameObject.SetActive(active);
0019         }
0020
0021         public void SetLayer(string layer = null)
0022         {
0023             string layerName = string.IsNullOrEmpty(layer) ? Configs.Windows.SortingLayers.Default :
0024             layer;
0025             _canvas.overrideSorting = true;
0026             _canvas.sortingLayerName = layerName;
0027         }
0028     }
0029 }
```

7.152 WindowCloseCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Commands;
00004 using MergeIt.Core.Helpers;
00005 using MergeIt.Core.WindowSystem.Windows;
00006
00007 namespace MergeIt.Core.WindowSystem.Commands
00008 {
00009     public sealed class WindowCloseCommand : Command, IMonoUpdateHandler
0010     {
0011         private IWindowPresenter _presenter;
0012     }
0013 }
```

```
00013     public WindowCloseCommand(IWindowPresenter presenter)
00014     {
00015         MonoEventsListener.Instance.SubscribeOnUpdate(this);
00016
00017         _presenter = presenter;
00018         if (_presenter.State != WindowState.Closing &&
00019             _presenter.State != WindowState.Closed)
00020         {
00021             _presenter.Close();
00022         }
00023         else
00024         {
00025             Finish();
00026         }
00027     }
00028
00029     public void Update()
00030     {
00031         if (_presenter is {State: WindowState.Closed})
00032         {
00033             _presenter = null;
00034             Finish();
00035         }
00036     }
00037
00038     protected override void OnDispose()
00039     {
00040         base.OnDispose();
00041
00042         MonoEventsListener.Instance.UnsubscribeFromUpdate(this);
00043     }
00044 }
00045 }
```

7.153 WindowHideCommand.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Commands;
00004 using MergeIt.Core.Helpers;
00005 using MergeIt.Core.WindowSystem.Windows;
00006
00007 namespace MergeIt.Core.WindowSystem.Commands
00008 {
00009     public sealed class WindowHideCommand : Command, IMonoUpdateHandler
00010     {
00011         private IWindowPresenter _presenter;
00012
00013         public WindowHideCommand(IWindowPresenter presenter)
00014         {
00015             MonoEventsListener.Instance.SubscribeOnUpdate(this);
00016
00017             _presenter = presenter;
00018             if (_presenter.State != WindowState.Hiding &&
00019                 _presenter.State != WindowState.Hidden)
00020             {
00021                 _presenter.Hide();
00022             }
00023             else
00024             {
00025                 Finish();
00026             }
00027         }
00028
00029         public void Update()
00030         {
00031             if (_presenter is {State: WindowState.Hidden})
00032             {
00033                 _presenter = null;
00034                 Finish();
00035             }
00036         }
00037
00038         protected override void OnDispose()
00039         {
00040             base.OnDispose();
00041
00042             MonoEventsListener.Instance.UnsubscribeFromUpdate(this);
00043         }
00044     }
00045 }
```

7.154 WindowOpenCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Commands;
00004 using MergeIt.Core.Helpers;
00005 using MergeIt.Core.WindowSystem.Windows;
00006
00007 namespace MergeIt.Core.WindowSystem.Commands
00008 {
00009     public sealed class WindowOpenCommand : Command, IMonoUpdateHandler
00010     {
00011         private IWindowPresenter _presenter;
00012
00013         public WindowOpenCommand(IWindowPresenter presenter)
00014         {
00015             MonoEventsListener.Instance.SubscribeOnUpdate(this);
00016
00017             _presenter = presenter;
00018             if (_presenter.State != WindowState.Opening &&
00019                 _presenter.State != WindowState.Opened)
00020             {
00021                 _presenter.Show();
00022             }
00023             else
00024             {
00025                 Finish();
00026             }
00027         }
00028
00029         public void Update()
00030         {
00031             if (_presenter is {State: WindowState.Opened})
00032             {
00033                 _presenter = null;
00034                 Finish();
00035             }
00036         }
00037
00038         protected override void OnDispose()
00039         {
00040             base.OnDispose();
00041
00042             MonoEventsListener.Instance.UnsubscribeFromUpdate(this);
00043         }
00044     }
00045 }
```

7.155 IWindowArgs.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.WindowSystem.Data
00004 {
00005     public interface IWindowArgs
00006     {
00007     }
00008 }
```

7.156 IWindowOpenParameters.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.WindowSystem.Windows;
00004
00005 namespace MergeIt.Core.WindowSystem.Data
00006 {
00007     public interface IWindowOpenParameters
00008     {
00009         bool ClosePrevious { get; set; }
00010         bool NeedBlackout { get; set; }
00011         IWindowPresenter Presenter { get; set; }
00012     }
00013 }
```

7.157 WindowArgs.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.WindowSystem.Data
00004 {
00005     public abstract class WindowArgs : IWindowArgs
00006     {
00007
00008     }
00009 }
```

7.158 WindowOpenParameters.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.WindowSystem.Windows;
00004
00005 namespace MergeIt.Core.WindowSystem.Data
00006 {
00007     public class WindowOpenParameters : IWindowOpenParameters
00008     {
00009         public bool ClosePrevious { get; set; }
00010         public bool NeedBlackout { get; set; }
00011         public IWindowPresenter Presenter { get; set; }
00012     }
00013 }
```

7.159 IWindowFactory.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.WindowSystem.Data;
00004 using MergeIt.Core.WindowSystem.Windows;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Core.WindowSystem.Factory
00008 {
00009     public interface IWindowFactory
00010     {
00011         RectTransform GetRoot();
00012         BlackoutComponent GetBlackout(RectTransform parent);
00013         TPresenter CreateWindow<TPresenter>(WindowCreateInfo prefabPath, Transform parent, IWindowArgs
00014             windowArgs = null)
00015             where TPresenter : class, IWindowPresenter, new();
00016     }
00017 }
```

7.160 WindowFactory.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Helpers;
00004 using MergeIt.Core.WindowSystem.Data;
00005 using MergeIt.Core.WindowSystem.Windows;
00006 using UnityEngine;
00007
00008 namespace MergeIt.Core.WindowSystem.Factory
00009 {
00010     public class WindowFactory : IWindowFactory
00011     {
00012         private const string RootPath = "Prefabs/Windows/WindowsRoot";
00013         private const string Blackout = "Prefabs/Windows/Blackout";
00014
00015         public RectTransform GetRoot()
00016         {
00017             var root = Resources.Load<Canvas>(RootPath);
00018
00019             if (root)
00020             {
00021                 root.worldCamera = Camera.main;
00022                 GameObject gameObject = Object.Instantiate(root.gameObject);
00023
00024                 return gameObject.GetComponent<RectTransform>();
00025             }
00026         }
00027     }
00028 }
```

```

00026
00027         return null;
00028     }
00029
00030     public BlackoutComponent GetBlackout(RectTransform parent)
00031     {
00032         var blackoutObject = Resources.Load<GameObject>(Blackout);
00033
00034         if (blackoutObject)
00035         {
00036             GameObject gameObject = Object.Instantiate(blackoutObject, parent);
00037
00038             if (gameObject)
00039             {
00040                 if (gameObject.TryGetComponent(out RectTransform rectTransform))
00041                 {
00042                     rectTransform.Stretch();
00043                     rectTransform.SetAsFirstSibling();
00044                 }
00045             }
00046
00047             gameObject.TryGetComponent(out BlackoutComponent blackoutComponent);
00048
00049             return blackoutComponent;
00050         }
00051
00052         return null;
00053     }
00054
00055     public TPresenter CreateWindow<TPresenter>(WindowCreateInfo createInfo, Transform parent,
00056     IWindowArgs windowArgs)
00057     where TPresenter : class, IWindowPresenter, new()
00058     {
00059         if (!string.IsNullOrEmpty(createInfo.PrefabPath))
00060         {
00061             var presenter = new TPresenter();
00062             GameObject windowPrototype = Resources.Load<GameObject>(createInfo.PrefabPath);
00063             GameObject windowObject = Object.Instantiate(windowPrototype, parent);
00064
00065             if ((windowObject ? windowObject.GetComponent(createInfo.Type) : null) is IWindow
00066             window)
00067             {
00068                 window.Initialize();
00069
00070                 presenter.Initialize(window, createInfo.LayerName, windowArgs);
00071                 presenter.SetWindowLayer();
00072                 presenter.SetWindowActive(false);
00073             }
00074
00075             return presenter;
00076         }
00077
00078         return null;
00079     }
00080 }

```

7.161 IWindowSystem.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.WindowSystem.Data;
00004 using MergeIt.Core.WindowSystem.Windows;
00005
00006 namespace MergeIt.Core.WindowSystem
00007 {
00008     public interface IWindowSystem
00009     {
00010         void OpenWindow<TPresenter>(bool closePrevious = false, bool enableBlackout = false,
00011         IWindowArgs args = null)
00012         where TPresenter : class, IWindowPresenter, new();
00013
00014         void CloseWindow(IWindowPresenter presenter);
00015     }
00016 }

```

7.162 CloseWindowMessage.cs

```
00001 // Copyright (c) 2024, Awessets
```

```
00002
00003 using MergeIt.Core.Messages;
00004 using MergeIt.Core.WindowSystem.Windows;
00005
00006 namespace MergeIt.Core.WindowSystem.Messages
00007 {
00008     public class CloseWindowMessage : IMessage
00009     {
0010         public IWindowPresenter Presenter { get; set; }
0011     }
0012 }
```

7.163 SortingLayers.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Configs.Windows
00004 {
00005     public static partial class SortingLayers
00006     {
00007         public const string Default = "Default";
00008     }
00009 }
```

7.164 SortingLayers.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.WindowSystem
00004 {
00005     public static partial class SortingLayers
00006     {
00007         public const string Main = "Main";
00008         public const string Popup = "Popup";
00009         public const string Modal = "Modal";
0010     }
0011 }
```

7.165 WindowName.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.WindowSystem
00004 {
00005     public static class WindowName
00006     {
00007         public const string Inventory = "Inventory/InventoryWindow";
00008         public const string UserInfo = "UserProgress/UserInfoWindow";
00009         public const string ElementInfo = "ElementInfo/ElementInfoWindow";
0010     }
0011 }
```

7.166 IWindow.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.MVP;
00005
00006 namespace MergeIt.Core.WindowSystem.Windows
00007 {
00008     public interface IWindow : IView
00009     {
0010         event Action InitiateCloseEvent;
0011
0012         event Action ShowStartEvent;
0013         event Action ShowEndEvent;
0014         event Action CloseStartEvent;
0015         event Action CloseEndEvent;
0016
0017         void Show();
0018         void Close();
0019         void SetLayer(string layer);
0020     }
0021 }
```

7.167 IWindowPresenter.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.MVP;
00004 using MergeIt.Core.WindowSystem.Data;
00005
00006 namespace MergeIt.Core.WindowSystem.Windows
00007 {
00008     public interface IWindowPresenter : IPresenter
00009     {
00010         string Layer { get; }
00011         WindowState State { get; }
00012
00013         void Initialize(IView view, string layer, IWindowArgs windowArgs = null);
00014         void Show();
00015         void Hide();
00016         void Close();
00017         void SetWindowActive(bool active);
00018         void SetWindowLayer();
00019     }
00020 }

```

7.168 WindowBase.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Animations;
00005 using MergeIt.Core.MVP;
00006 using UnityEngine;
00007 using UnityEngine.UI;
00008
00009 namespace MergeIt.Core.WindowSystem.Windows
00010 {
00011     public abstract class WindowBase : View, IWindow, IWindowAnimationListener
00012     {
00013         public event Action InitiateCloseEvent;
00014
00015         public event Action ShowStartEvent;
00016         public event Action ShowEndEvent;
00017         public event Action CloseStartEvent;
00018         public event Action CloseEndEvent;
00019
00020         [SerializeField]
00021         protected Button closeButton;
00022
00023         [SerializeField]
00024         protected Button ClickOutsideArea;
00025
00026         public Canvas Canvas
00027         {
00028             get
00029             {
00030                 if (!_canvas)
00031                 {
00032                     _canvas = GetComponent<Canvas>();
00033                 }
00034
00035                 return _canvas;
00036             }
00037         }
00038
00039         public IWindowAnimationController AnimationController
00040         {
00041             get
00042             {
00043                 if (_animationController == null)
00044                 {
00045                     _animationController = GetComponent<IWindowAnimationController>() ?? new
00046                     WindowDefaultAnimationController();
00047                     _animationController.Initialize(this);
00048                 }
00049
00050                 return _animationController;
00051             }
00052         }
00053
00054         private Canvas _canvas;
00055         private IWindowAnimationController _animationController;
00056
00057         public virtual void Show()
00058         {

```

```
00058     SubscribeOnClose();
00059     AnimationController.OpenWindow();
00060 }
00061
00062     public virtual void Close()
00063 {
00064     UnsubscribeFromClose();
00065     AnimationController.CloseWindow();
00066 }
00067
00068     public virtual void SetLayer(string layer)
00069 {
00070     if (!string.IsNullOrEmpty(layer))
00071     {
00072         Canvas.overrideSorting = true;
00073         Canvas.sortingLayerName = layer;
00074     }
00075     else
00076     {
00077         Canvas.overrideSorting = false;
00078         Canvas.sortingLayerName = Configs.Windows.SortingLayers.Default;
00079     }
00080 }
00081
00082     public virtual void OnOpenStarted()
00083 {
00084     ShowStartEvent?.Invoke();
00085 }
00086
00087     public virtual void OnOpenFinished()
00088 {
00089     ShowEndEvent?.Invoke();
00090 }
00091
00092     public virtual void OnCloseStarted()
00093 {
00094     CloseStartEvent?.Invoke();
00095 }
00096
00097     public virtual void OnCloseFinished()
00098 {
00099     CloseEndEvent?.Invoke();
00100 }
00101
00102     protected override void OnDestroy()
00103 {
00104     base.OnDestroy();
00105
00106     UnsubscribeFromClose();
00107 }
00108
00109     private void SubscribeOnClose()
00110 {
00111     CloseButton.onClick.AddListener(OnCloseButtonClick);
00112
00113     if (ClickOutsideArea)
00114     {
00115         ClickOutsideArea.onClick.AddListener(OnCloseButtonClick);
00116     }
00117 }
00118
00119     private void UnsubscribeFromClose()
00120 {
00121     CloseButton.onClick.RemoveListener(OnCloseButtonClick);
00122
00123     if (ClickOutsideArea)
00124     {
00125         ClickOutsideArea.onClick.RemoveListener(OnCloseButtonClick);
00126     }
00127 }
00128
00129     private void OnCloseButtonClick()
00130 {
00131     InitiateCloseEvent?.Invoke();
00132 }
00133 }
00134 }
```

7.169 WindowCreateInfo.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
```

```

00004
00005 namespace MergeIt.Core.WindowSystem.Windows
00006 {
00007     public struct WindowCreateInfo : IEquatable<WindowCreateInfo>
00008     {
00009         public string LayerName { get; set; }
00010         public string PrefabPath { get; set; }
00011         public Type Type { get; set; }
00012
00013         public bool Equals(WindowCreateInfo other)
00014         {
00015             return LayerName == other.LayerName && PrefabPath == other.PrefabPath && Type ==
00016             other.Type;
00017         }
00018
00019         public override bool Equals(object obj)
00020         {
00021             return obj is WindowCreateInfo other && Equals(other);
00022         }
00023
00024         public override int GetHashCode()
00025         {
00026             return HashCode.Combine(LayerName, PrefabPath, Type);
00027         }
00028     }

```

7.170 WindowModel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.MVP;
00004
00005 namespace MergeIt.Core.WindowSystem.Windows
00006 {
00007     public abstract class WindowModel : IModel
00008     {
00009
00010     }
00011 }

```

7.171 WindowPresenter.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004 using MergeIt.Core.MVP;
00005 using MergeIt.Core.WindowSystem.Data;
00006 using MergeIt.Core.WindowSystem.Messages;
00007 using MergeIt.SimpleDI;
00008 using UnityEngine;
00009
00010 namespace MergeIt.Core.WindowSystem.Windows
00011 {
00012     public abstract class WindowPresenter<TWindow, TModel> : Presenter<TWindow, TModel>,
00013     IWindowPresenter
00014     where TWindow : Component, IWindow
00015     where TModel : WindowModel, new()
00016     {
00017         protected IMessageBus MessageBus;
00018         protected IWindowSystem WindowSystem;
00019
00020         private WindowState _state;
00021
00022         public string Layer { get; private set; }
00023
00024         public WindowState State
00025         {
00026             get => _state;
00027         }
00028
00029         public void Initialize(IView view, string layer, IWindowArgs windowArgs = null)
00030         {
00031             base.Initialize(view);
00032
00033             MessageBus = DiContainer.Get<IMessageBus>();
00034             WindowSystem = DiContainer.Get<IWindowSystem>();
00035             Layer = layer;

```

```
00036         OnInitialize(windowArgs);
00037     }
00038 }
00039 
00040     public void Show()
00041 {
00042     _state = WindowState.Opening;
00043     View.ShowStartEvent += OnWindowStartShowing;
00044     View.ShowEndEvent += OnWindowShown;
00045     View.Show();
00046 }
00047 
00048     public void Hide()
00049 {
00050     _state = WindowState.Hiding;
00051     InitiateClosing();
00052 }
00053 
00054     public void Close()
00055 {
00056     _state = WindowState.Closing;
00057     InitiateClosing();
00058 }
00059 
00060     public void SetWindowActive(bool active)
00061 {
00062     View.gameObject.SetActive(active);
00063 }
00064 
00065     public void SetWindowLayer()
00066 {
00067     View.SetLayer(Layer);
00068 }
00069 
00070     public void DestroyWindow()
00071 {
00072     OnDestroyWindow();
00073 
00074     View.InitiateCloseEvent -= OnInitiateCloseEvent;
00075 
00076     Object.Destroy(View.gameObject);
00077 }
00078 
00079     protected virtual void OnInitialize(IWindowArgs args = null)
00080 {
00081 }
00082 
00083     protected virtual void OnDestroyWindow()
00084 {
00085 }
00086 
00087     protected override void OnDispose()
00088 {
00089     base.OnDispose();
00090 
00091     if (View)
00092     {
00093         View.InitiateCloseEvent -= OnInitiateCloseEvent;
00094     }
00095 }
00096 
00097     private void InitiateClosing()
00098 {
00099     View.CloseStartEvent += OnWindowStartClosing;
00100     View.CloseEndEvent += OnWindowClosed;
00101     View.Close();
00102 }
00103 
00104     private void OnInitiateCloseEvent()
00105 {
00106     var message = new CloseWindowMessage {Presenter = this};
00107     MessageBus.Fire(message);
00108 }
00109 
00110     protected virtual void OnWindowStartShowing()
00111 {
00112     SetWindowActive(true);
00113 
00114     View.ShowStartEvent -= OnWindowStartShowing;
00115 }
00116 
00117     protected virtual void OnWindowShown()
00118 {
00119     View.ShowEndEvent -= OnWindowShown;
00120     View.InitiateCloseEvent += OnInitiateCloseEvent;
00121 
00122     _state = WindowState.Opened;
```

```

00123      }
00124
00125      protected virtual void OnWindowStartClosing()
00126      {
00127          View.CloseStartEvent -= OnWindowStartClosing;
00128          View.InitiateCloseEvent -= OnInitiateCloseEvent;
00129      }
00130
00131      protected virtual void OnWindowClosed()
00132      {
00133          View.CloseEndEvent -= OnWindowClosed;
00134
00135          if (_state == WindowState.Closing)
00136          {
00137              _state = WindowState.Closed;
00138              SetWindowActive(false);
00139              DestroyWindow();
00140          }
00141          else if (_state == WindowState.Hiding)
00142          {
00143              _state = WindowState.Hidden;
00144          }
00145      }
00146  }
00147 }
00148 }
```

7.172WindowState.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.WindowSystem.Windows
00004 {
00005     public enum WindowState
00006     {
00007         None,
00008         Opening,
00009         Opened,
00010         Hiding,
00011         Hidden,
00012         Closing,
00013         Closed,
00014     }
00015 }
```

7.173WindowsStorage.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using System.IO;
00006 using MergeIt.Core.WindowSystem.Windows;
00007
00008 namespace MergeIt.Core.WindowSystem
00009 {
00010     public static class WindowsStorage
00011     {
00012         private const string WindowsbasePath = "Prefabs/Windows";
00013
00014         private static readonly Dictionary<Type, WindowCreateInfo> ControllersLayers = new();
00015
00016         public static void Register<TPresenter, TWindow>(string prefabName, string layerName)
00017             where TPresenter : IWindowPresenter
00018             where TWindow : WindowBase
00019         {
00020             ControllersLayers[typeof(TPresenter)] = new WindowCreateInfo
00021             {
00022                 Type = typeof(TWindow),
00023                 LayerName = layerName,
00024                 PrefabPath = Path.Combine(WindowsbasePath, prefabName)
00025             };
00026         }
00027
00028         public static WindowCreateInfo GetWindowInfo<TPresenter>()
00029             where TPresenter : IWindowPresenter
00030         {
00031             Type type = typeof(TPresenter);
00032         }
00033     }
00034 }
```

```
00033         return GetWindowInfo(type);
00034     }
00035
00036     public static WindowCreateInfo GetWindowInfo(Type type)
00037     {
00038         ControllersLayers.TryGetValue(type, out WindowCreateInfo windowInfo);
00039
00040         return windowInfo;
00041     }
00042 }
00043
00044 }
```

7.174 WindowsSystem.cs

```
00069     _layersConfig = Resources.Load<LayersConfig>(WindowsLayersConfigPath);
00070
00071     var sortedLayers = _layersConfig.Layers.OrderBy(x => x.Order).ToArray();
00072     _layers = new string[sortedLayers.Length];
00073
00074     int index = 0;
00075     foreach (LayerParameters layerParameters in sortedLayers)
00076     {
00077         _layersWindows[layerParameters.Name] = new LinkedList<IWindowPresenter>();
00078         _layers[index++] = layerParameters.Name;
00079     }
00080 }
00081
00082 public void Dispose()
00083 {
00084     _messageBus.RemoveListener<CloseWindowMessage>(OnCloseWindowMessageHandler);
00085 }
00086
00087 public void OpenWindow<TPresenter>(bool closePrevious = false, bool enableBlackout = false,
00088 IWindowArgs args = null)
00089     where TPresenter : class, IWindowPresenter, new()
00090 {
00091     WindowCreateInfo createInfo = WindowsStorage.GetWindowInfo<TPresenter>();
00092     TPresenter windowPresenter = _windowFactory.CreateWindow<TPresenter>(createInfo,
00093     Root.transform, args);
00094     if (windowPresenter != null)
00095     {
00096         var openParameters = new WindowOpenParameters
00097         {
00098             ClosePrevious = closePrevious,
00099             NeedBlackout = enableBlackout,
00100             Presenter = windowPresenter
00101         };
00102
00103         _openedWindows>windowPresenter] = openParameters;
00104
00105         ICommandManager commands = RequestOpenWindow(openParameters);
00106         AddToQueue(commands);
00107     }
00108 }
00109
00110 public void CloseWindow(IWindowPresenter presenter)
00111 {
00112     ICommandManager commands = RequestCloseWindow(presenter);
00113     AddToQueue(commands);
00114 }
00115
00116 private ICommandManager RequestOpenWindow(WindowOpenParameters parameters)
00117 {
00118     var commandManager = new CommandManager();
00119     string layer = parameters.Presenter.Layer;
00120
00121     int index = Array.IndexOf(_layers, layer);
00122     if (index == -1)
00123     {
00124         Debug.LogError($"Couldn't find layer with name ({layer})");
00125
00126         return null;
00127     }
00128
00129     for (int i = index; i < _layers.Length; i++)
00130     {
00131         string layerName = _layers[i];
00132         LinkedList<IWindowPresenter> list = _layersWindows[layerName];
00133
00134         var node = list.Last;
00135
00136         while (node != null)
00137         {
00138             WindowState state = node.Value.State;
00139             LinkedListNode<IWindowPresenter> prev = node.Previous;
00140
00141             switch (state)
00142             {
00143                 case WindowState.Opened:
00144                     if (parameters.ClosePrevious)
00145                     {
00146                         commandManager.Add(new WindowCloseCommand(node.Value));
00147                         Remove(list, node);
00148                     }
00149                     else
00150                     {
00151                         commandManager.Add(new WindowHideCommand(node.Value));
00152                     }
00153                     break;
00154
00155                 case WindowState.None:
```

```
00154         case WindowState.Hidden:
00155         case WindowState.Closed:
00156             if (parameters.ClosePrevious)
00157             {
00158                 Remove(list, node);
00159             }
00160             break;
00161         }
00162         node = prev;
00163     }
00164 }
00165
00166 if (parameters.NeedBlackout)
00167 {
00168     Blackout.SetActive(true);
00169     Blackout.SetLayer(layer);
00170 }
00171
00172 commandManager.Add(new WindowOpenCommand(parameters.Presenter));
00173 _layersWindows[layer].AddLast(parameters.Presenter);
00174
00175 return commandManager;
00176 }
00177
00178 private ICommandManager RequestCloseWindow(IWindowPresenter presenter)
00179 {
00180     CommandManager commandManager = new CommandManager();
00181     string layer = presenter.Layer;
00182
00183     int index = Array.IndexOf(_layers, layer);
00184     if (index == -1)
00185     {
00186         Debug.LogError($"Couldn't find layer with name ({layer})");
00187
00188         return null;
00189     }
00190
00191     for (int i = index; i < _layers.Length; i++)
00192     {
00193         string layerName = _layers[i];
00194         LinkedList<IWindowPresenter> list = _layersWindows[layerName];
00195
00196         var node = list.Last;
00197
00198         if (i == index)
00199         {
00200             IWindowPresenter value = node?.Value;
00201
00202             if (value == presenter)
00203             {
00204                 commandManager.Add(new WindowCloseCommand(value));
00205                 IWindowPresenter prevWindow = node.Previous?.Value;
00206
00207                 if (prevWindow != null)
00208                 {
00209                     commandManager.Add(new WindowOpenCommand(prevWindow));
00210                 }
00211
00212                 Remove(list, node);
00213             }
00214         }
00215         else
00216         {
00217             while (node != null)
00218             {
00219                 WindowState state = node.Value.State;
00220                 switch (state)
00221                 {
00222                     case WindowState.Opened:
00223                         commandManager.Add(new WindowCloseCommand(node.Value));
00224                         break;
00225
00226                     case WindowState.Closed:
00227                         Remove(list, node);
00228
00229                     case WindowState.Hidden:
00230                         Remove(list, node);
00231
00232                     node = prev;
00233                 }
00234             }
00235         }
00236
00237     return commandManager;
00238 }
00239
00240 private void Remove(LinkedList<IWindowPresenter> list, LinkedListNode<IWindowPresenter> node)
```

```

00241      {
00242          list.Remove(node);
00243          _openedWindows.Remove(node.Value);
00244      }
00245
00246      private void AddToQueue(ICommandManager commands)
00247      {
00248          if (commands != null)
00249          {
00250              commands.Finished += OnBatchCommandFinished;
00251              _commandsQueue.Enqueue(commands);
00252              UpdateQueue();
00253          }
00254      }
00255
00256      private void UpdateQueue()
00257      {
00258          if (_currentCommands is not { Executing: true } && _commandsQueue.Any())
00259          {
00260              _currentCommands = _commandsQueue.Dequeue();
00261              _currentCommands.RunSimultaneously();
00262          }
00263          else if (!_commandsQueue.Any())
00264          {
00265              bool needShowBlackout = false;
00266              IWindowPresenter lastPresenter = null;
00267
00268              foreach (var openedWindow in _openedWindows)
00269              {
00270                  IWindowOpenParameters parameters = openedWindow.Value;
00271                  if (parameters.NeedBlackout)
00272                  {
00273                      needShowBlackout = true;
00274                      lastPresenter = openedWindow.Key;
00275                  }
00276              }
00277
00278              Blackout.SetActive(needShowBlackout);
00279              Blackout.SetLayer(lastPresenter?.Layer);
00280          }
00281      }
00282
00283      private void OnBatchCommandFinished(ICommandManager commandManager)
00284      {
00285          commandManager.Finished -= OnBatchCommandFinished;
00286          UpdateQueue();
00287      }
00288
00289      private void OnCloseWindowMessageHandler(CloseWindowMessage message)
00290      {
00291          CloseWindow(message.Presenter);
00292      }
00293  }
00294 }
```

7.175 CheckEvolutionsProgressCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Commands;
00005 using MergeIt.Core.Configs.Elements;
00006 using MergeIt.Core.FieldElements;
00007 using MergeIt.Core.Saves;
00008 using MergeIt.Core.Services;
00009 using MergeIt.Game.Evolutions;
00010 using MergeIt.Game.Field;
00011 using MergeIt.SimpleDI;
00012
00013 namespace MergeIt.Game.Commands
00014 {
00015     public class CheckEvolutionsProgressCommand : Command
00016     {
00017         private readonly IGameLoadService _gameLoadService = DiContainer.Get<IGameLoadService>();
00018         private readonly IGameSaveService _gameSaveService = DiContainer.Get<IGameSaveService>();
00019         private readonly IEvolutionsService _evolutionsService =
00020             DiContainer.Get<IEvolutionsService>();
00021         private readonly FieldLogicModel _fieldLogicModel = DiContainer.Get<FieldLogicModel>();
00022
00023         public override void Execute()
00024         {
00025             var evolutionsProgressData = _gameLoadService.Load<EvolutionsProgressData>();
```

```

00026         if (evolutionsProgressData == null)
00027         {
00028             _evolutionsService.CreateEvolutionsProgress();
00029         }
00030         else
00031         {
00032             _evolutionsService.SetupEvolutionsProgress(evolutionsProgressData);
00033         }
00034     }
00035     UpdateProgress();
00036 }
00037
00038     private void UpdateProgress()
00039     {
00040         Dictionary<GridPoint, IFieldElement> fieldElements = _fieldLogicModel.FieldElements;
00041         foreach (KeyValuePair<GridPoint, IFieldElement> fieldElementPair in fieldElements)
00042         {
00043             ElementConfig elementConfig = fieldElementPair.Value.ConfigParameters.ElementConfig;
00044             _evolutionsService.UpdateProgress(elementConfig);
00045         }
00046
00047         _gameSaveService.Save(GameSaveType.EvolutionsProgress);
00048     }
00049 }
00050 }
```

7.176 CheckOpeningGeneratorsCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Commands;
00004 using MergeIt.Game.Field;
00005 using MergeIt.SimpleDI;
00006
00007 namespace MergeIt.Game.Commands
00008 {
00009     public class CheckOpeningGeneratorsCommand : Command
00010     {
00011         private readonly FieldLogicModel _fieldLogicModel = DiContainer.Get<FieldLogicModel>();
00012
00013         public override void Execute()
00014         {
00015
00016     }
00017 }
00018 }
```

7.177 EffectCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Commands;
00004 using MergeIt.Game.Effects.Controllers;
00005
00006 namespace MergeIt.Game.Commands
00007 {
00008     public class EffectCommand : Command
00009     {
00010         private IEFFECT _effect;
00011
00012         public EffectCommand(IEFFECT effect)
00013         {
00014             _effect = effect;
00015         }
00016
00017         public override void Execute()
00018         {
00019             base.Execute();
00020
00021         }
00022     }
00023 }
```

7.178 LoadConfigsCommand.cs

```
00001 // Copyright (c) 2024, Awessets
```

```

00002
00003 using Cysharp.Threading.Tasks;
00004 using MergeIt.Core.Commands;
00005 using MergeIt.Core.Services;
00006 using MergeIt.SimpleDI;
00007
00008 namespace MergeIt.Game.Commands
00009 {
0010     public class LoadConfigsCommand : Command
0011     {
0012         private readonly IConfigsService _configsService = DiContainer.Get<IConfigsService>();
0013
0014         public async override UniTask ExecuteAsync()
0015         {
0016             await _configsService.Load();
0017         }
0018     }
0019 }

```

7.179 PrepareEnergyCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Commands;
00005 using MergeIt.Core.Configs;
00006 using MergeIt.Core.Services;
00007 using MergeIt.Game.Services;
00008 using MergeIt.SimpleDI;
00009
00010 namespace MergeIt.Game.Commands
00011 {
0012     public class PrepareEnergyCommand : Command
0013     {
0014         private readonly UserServiceModel _userServiceModel = DiContainer.Get<UserServiceModel>();
0015         private readonly IConfigsService _configsServiceModel = DiContainer.Get<IConfigsService>();
0016
0017         public override void Execute()
0018         {
0019             CheckEnergy();
0020         }
0021
0022         private void CheckEnergy()
0023         {
0024             GameConfig config = _configsServiceModel.GameConfig;
0025             int currentEnergy = _userServiceModel.Energy.Value;
0026
0027             if (currentEnergy >= config.EnergyCap)
0028             {
0029                 _userServiceModel.EnergyRestoringStartTime = -1;
0030             }
0031             else
0032             {
0033                 long currentTime = DateTimeOffset.UtcNow.ToUnixTimeSeconds();
0034                 long elapsedTime = currentTime - _userServiceModel.EnergyRestoringStartTime;
0035                 long restoredEnergy = elapsedTime / config.EnergyRestoreTime;
0036
0037                 currentEnergy += (int)restoredEnergy;
0038                 int actualEnergy = Math.Min(currentEnergy, config.EnergyCap);
0039                 _userServiceModel.Energy.Value = actualEnergy;
0040
0041                 if (actualEnergy < config.EnergyCap)
0042                 {
0043                     _userServiceModel.EnergyRestoringStartTime += config.EnergyRestoreTime *
0044                         restoredEnergy;
0045                 }
0046                 else
0047                 {
0048                     _userServiceModel.EnergyRestoringStartTime = -1;
0049                 }
0050             }
0051         }
0052     }

```

7.180 PrepareFieldCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002

```

```

00003 using System.Collections.Generic;
00004 using MergeIt.Core.Commands;
00005 using MergeIt.Core.Configs;
00006 using MergeIt.Core.Configs.Data;
00007 using MergeIt.Core.FieldElements;
00008 using MergeIt.Core.Saves;
00009 using MergeIt.Core.Services;
00010 using MergeIt.Game.Converters;
00011 using MergeIt.Game.Factories.FieldElement;
00012 using MergeIt.Game.Field;
00013 using MergeIt.Game.Services;
00014 using MergeIt.SimpleDI;
00015
00016 namespace MergeIt.Game.Commands
00017 {
00018     public class PrepareFieldCommand : Command
00019     {
00020         private readonly IGameLoadService _gameLoadService = DiContainer.Get<IGameLoadService>();
00021         private readonly IFieldElementFactory _fieldElementFactory =
00022             DiContainer.Get<IFieldElementFactory>();
00023         private readonly IGameSaveService _saveService = DiContainer.Get<IGameSaveService>();
00024         private readonly IConfigProcessor _configProcessor = DiContainer.Get<IConfigProcessor>();
00025         private readonly GameServiceModel _gameServiceModel = DiContainer.Get<GameServiceModel>();
00026         private readonly FieldLogicModel _fieldLogicModel = DiContainer.Get<FieldLogicModel>();
00027         private readonly IConfigsService _configsService = DiContainer.Get<IConfigsService>();
00028
00029         public override void Execute()
00030         {
00031             var loadedLevel = _gameLoadService.Load<FieldData>();
00032
00033             if (loadedLevel != null)
00034             {
00035                 _fieldLogicModel.FieldHeight = loadedLevel.FieldHeight;
00036                 _fieldLogicModel.FieldWidth = loadedLevel.FieldWidth;
00037
00038                 Dictionary<GridPoint, IFieldElement> fieldElements = _fieldLogicModel.FieldElements;
00039
00040                 foreach (FieldElementData fieldElementData in loadedLevel.SavedElementsData)
00041                 {
00042                     IFieldElement fieldElement =
00043                         _fieldElementFactory.CreateFieldElement(fieldElementData);
00044
00045                     fieldElements.Add(fieldElement.InfoParameters.LogicPosition, fieldElement);
00046                 }
00047             }
00048             else
00049             {
00050                 LevelConfig levelConfig = _configsService.LevelConfig;
00051                 _fieldLogicModel.FieldHeight = levelConfig.FieldHeight;
00052                 _fieldLogicModel.FieldWidth = levelConfig.FieldWidth;
00053
00054                 Dictionary<GridPoint, IFieldElement> fieldElements = _fieldLogicModel.FieldElements;
00055
00056                 foreach (LevelElementData fieldElementData in levelConfig.FieldElementsData)
00057                 {
00058                     IFieldElement fieldElement =
00059                         _fieldElementFactory.CreateFieldElement(fieldElementData);
00060
00061                     fieldElements.Add(fieldElement.InfoParameters.LogicPosition, fieldElement);
00062                 }
00063
00064                 loadedLevel = _configProcessor.BuildLevel();
00065
00066                 _gameServiceModel.LoadedLevel = loadedLevel;
00067                 _saveService.Save(GameSaveType.Field);
00068             }
00069         }
00070     }

```

7.181 PrepareInventoryCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Commands;
00004 using MergeIt.Core.Services;
00005 using MergeIt.Game.Factories.Inventory;
00006 using MergeIt.SimpleDI;
00007
00008 namespace MergeIt.Game.Commands
00009 {
00010     public class PrepareInventoryCommand : Command
00011     {

```

```

00012     private readonly IGameLoadService _gameLoadService = DiContainer.Get<IGameLoadService>();
00013     private readonly IIInventoryService _inventoryService = DiContainer.Get<IIInventoryService>();
00014
00015     public override void Execute()
00016     {
00017         var inventoryData = _gameLoadService.Load<InventoryData>();
00018
00019         if (inventoryData == null)
00020         {
00021             _inventoryService.CreateInventory();
00022         }
00023         else
00024         {
00025             _inventoryService.SetupInventory(inventoryData);
00026         }
00027     }
00028 }
00029 }
```

7.182 PrepareStockCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Commands;
00004 using MergeIt.Core.Services;
00005 using MergeIt.Game.ElementsStock;
00006 using MergeIt.SimpleDI;
00007
00008 namespace MergeIt.Game.Commands
00009 {
00010     public class PrepareStockCommand : Command
00011     {
00012         private readonly IGameLoadService _gameLoadService = DiContainer.Get<IGameLoadService>();
00013         private readonly IElementsStockService _stockService =
00014             DiContainer.Get<IElementsStockService>();
00015
00016         public override void Execute()
00017         {
00018             var stockData = _gameLoadService.Load<ElementsStockData>();
00019
00020             if (stockData == null)
00021             {
00022                 _stockService.CreateStock();
00023             }
00024             else
00025             {
00026                 _stockService.SetupStock(stockData);
00027             }
00028         }
00029 }
```

7.183 PrepareUserCommand.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Commands;
00004 using MergeIt.Core.Services;
00005 using MergeIt.Game.User;
00006 using MergeIt.SimpleDI;
00007
00008 namespace MergeIt.Game.Commands
00009 {
00010     public class PrepareUserCommand : Command
00011     {
00012         private readonly IGameLoadService _gameLoadService = DiContainer.Get<IGameLoadService>();
00013         private readonly IUserService _userService = DiContainer.Get<IUserService>();
00014
00015         public override void Execute()
00016         {
00017             var userData = _gameLoadService.Load<UserData>();
00018
00019             if (userData == null)
00020             {
00021                 _userService.CreateUser();
00022             }
00023             else
00024             {
00025                 _userService.SetupUser(userData);
00026             }
00027         }
00028     }
00029 }
```

```
00026         }
00027     }
00028 }
00029 }
```

7.184 ConfigsContainer.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs;
00004 using MergeIt.Core.Configs.Effects;
00005 using MergeIt.Core.Configs.Hints;
00006 using MergeIt.Core.Configs.Icons;
00007 using MergeIt.Core.Configs.Inventory;
00008 using MergeIt.Core.Configs.LevelUp;
00009 using UnityEngine;
00010
00011 namespace MergeIt.Game.Configs
00012 {
00013     [CreateAssetMenu(fileName = "ConfigsContainer", menuName = "Merge Toolkit/Configs container")]
00014     public class ConfigsContainer : ScriptableObject
00015     {
00016         [SerializeField]
00017         private ElementPrefab[] _prefabs;
00018
00019         [SerializeField]
00020         private GameConfig _gameConfig;
00021
00022         [SerializeField]
00023         private GameIconsConfig _iconsConfig;
00024
00025         [SerializeField]
00026         private InventoryConfig _inventoryConfig;
00027
00028         [SerializeField]
00029         private LevelConfig _levelConfig;
00030
00031         [SerializeField]
00032         private LevelUpConfig _levelUpConfig;
00033
00034         [SerializeField]
00035         private HintsConfig _hintsConfig;
00036
00037         [SerializeField]
00038         private EffectConfig[] _effectsConfigs;
00039
00040         public ElementPrefab[] Prefabs
00041         {
00042             get => _prefabs;
00043         }
00044
00045         public GameConfig GameConfig
00046         {
00047             get => _gameConfig;
00048         }
00049
00050         public GameIconsConfig IconsConfig
00051         {
00052             get => _iconsConfig;
00053         }
00054
00055         public InventoryConfig InventoryConfig
00056         {
00057             get => _inventoryConfig;
00058         }
00059
00060         public LevelConfig LevelConfig
00061         {
00062             get => _levelConfig;
00063         }
00064
00065         public LevelUpConfig LevelUpConfig
00066         {
00067             get => _levelUpConfig;
00068         }
00069
00070         public EffectConfig[] EffectConfigs
00071         {
00072             get => _effectsConfigs;
00073         }
00074
00075         public HintsConfig HintsConfig
00076         {
```

```

00077         get => _hintsConfig;
00078     }
00079 }
00080 }
```

7.185 ElementPrefab.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Configs.Types;
00005 using MergeIt.Game.Field.Elements;
00006 using UnityEngine;
00007
00008 namespace MergeIt.Game.Configs
00009 {
00010     [Serializable]
00011     public class ElementPrefab
00012     {
00013         [SerializeField]
00014         private ElementType _type;
00015
00016         [SerializeField]
00017         private FieldElementView _prefab;
00018
00019         public ElementType Type
00020         {
00021             get => _type;
00022         }
00023
00024         public FieldElementView Prefab
00025         {
00026             get => _prefab;
00027         }
00028     }
00029 }
```

7.186 ConfigsService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using System.Linq;
00006 using Cysharp.Threading.Tasks;
00007 using MergeIt.Core.Configs;
00008 using MergeIt.Core.Configs.Data;
00009 using MergeIt.Core.Configs.Effects;
00010 using MergeIt.Core.Configs.Elements;
00011 using MergeIt.Core.Configs.Hints;
00012 using MergeIt.Core.Configs.Inventory;
00013 using MergeIt.Core.Configs.LevelUp;
00014 using MergeIt.Core.Configs.Types;
00015 using MergeIt.Core.FieldElements;
00016 using MergeIt.Core.Services;
00017 using MergeIt.Game.Field.Elements;
00018 using UnityEngine;
00019
00020 namespace MergeIt.Game.Configs.Services
00021 {
00022     public class ConfigsService : IConfigsService
00023     {
00024         private readonly ConfigsServiceModel _model = new();
00025
00026         private ConfigsContainer _configsContainer;
00027
00028         public LevelConfig LevelConfig
00029         {
00030             get => _model.LevelConfig;
00031         }
00032
00033         public InventoryConfig InventoryConfig
00034         {
00035             get => _model.InventoryConfig;
00036         }
00037
00038         public GameConfig GameConfig
00039         {
00040             get => _model.GameConfig;
00041         }
00042     }
00043 }
```

```
00041         }
00042
00043     public HintsConfig HintsConfig
00044     {
00045         get => _model.HintsConfig;
00046     }
00047
00048     public async UniTask Load()
00049     {
00050         var requestResult = await
Resources.LoadAsync<ConfigsContainer>(ConfigsConstants.ConfigsPath);
00051
00052         _configsContainer = requestResult as ConfigsContainer;
00053
00054         if (_configsContainer)
00055         {
00056             _model.StoreElementsPrefabs(_configsContainer.Prefabs);
00057             _model.StoreGameConfig(_configsContainer.GameConfig);
00058             _model.StoreIconsConfig(_configsContainer.IconsConfig);
00059             _model.StoreInventoryConfig(_configsContainer.InventoryConfig);
00060             _model.StoreLevelConfig(_configsContainer.LevelConfig);
00061             _model.StoreLevelUpConfig(_configsContainer.LevelUpConfig);
00062             _model.StoreEffectsConfigs(_configsContainer.EffectConfigs);
00063             _model.StoreHintsConfig(_configsContainer.HintsConfig);
00064         }
00065         else
00066         {
00067             Debug.Log("Something went wrong while loading configs container.");
00068         }
00069     }
00070
00071
00072     public Sprite GetCurrencyIcon(CurrencyType type)
00073     {
00074         Sprite icon = _model.IconsConfig.InfoPanelCurrencyIcons.FirstOrDefault(x => x.CurrencyType
== type)?.CurrencyIcon;
00075
00076         return icon;
00077     }
00078
00079     public EvolutionData GetEvolutionData(string evolutionId)
00080     {
00081         return _model.LevelConfig.EvolutionsScheme.Evolution.FirstOrDefault(x => x.Id ==
evolutionId);
00082     }
00083
00084     public string GetEvolutionIdByElement(ElementConfig element)
00085     {
00086         return GetEvolutionByElement(element)?.Id;
00087     }
00088
00089     public EvolutionData GetEvolutionByElement(ElementConfig element)
00090     {
00091         return _model.LevelConfig.EvolutionsScheme.Evolution.FirstOrDefault(x =>
x.Chain.Contains(element));
00092     }
00093
00094     public T GetEffectConfig<T>(string type) where T : EffectConfig
00095     {
00096         _model.EffectConfigs.TryGetValue(type, out EffectConfig effectConfig);
00097
00098         return effectConfig as T;
00099     }
00100
00101     public LevelUpParameters GetLevelUpData(int currentLevel)
00102     {
00103         return _model.LevelUpConfig.LevelUp[currentLevel - 1];
00104     }
00105
00106     public IFIELDElementView GetElementPrefab(ElementType type)
00107     {
00108         _model.Prefabs.TryGetValue(type, out FieldElementView view);
00109
00110         return view;
00111     }
00112
00113     public IEnumerable<ElementConfig> GetConfigs(Func<ElementConfig, bool> predicate = null)
00114     {
00115         if (predicate != null)
00116         {
00117             return _model.ElementConfigs.Where(predicate);
00118         }
00119
00120         return _model.ElementConfigs;
00121     }
00122
00123     public ElementConfig GetConfig(string id)
```

```

00124         {
00125             return _model.ElementConfigs.FirstOrDefault(x => x.Id == id);
00126         }
00127     }
00128 }
```

7.187 ConfigsServiceModel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using System.Linq;
00005 using MergeIt.Core.Configs;
00006 using MergeIt.Core.Configs.Data;
00007 using MergeIt.Core.Configs.Effects;
00008 using MergeIt.Core.Configs.Elements;
00009 using MergeIt.Core.Configs.Hints;
00010 using MergeIt.Core.Configs.Icons;
00011 using MergeIt.Core.Configs.Inventory;
00012 using MergeIt.Core.Configs.LevelUp;
00013 using MergeIt.Core.Configs.Types;
00014 using MergeIt.Game.Field.Elements;
00015
00016 namespace MergeIt.Game.Configs.Services
00017 {
00018     public class ConfigsServiceModel
00019     {
00020         private GameConfig _gameConfig;
00021         private GameIconsConfig _iconsConfig;
00022         private InventoryConfig _inventoryConfig;
00023         private LevelConfig _levelConfig;
00024         private LevelUpConfig _levelUpConfig;
00025         private Dictionary<string, EffectConfig> _effectConfigs;
00026         private Dictionary<ElementType, FieldElementView> _elementsPrefabs;
00027         private List<ElementConfig> _elementConfigs = new();
00028         private HintsConfig _hintsConfig;
00029
00030         public GameConfig GameConfig
00031         {
00032             get => _gameConfig;
00033         }
00034
00035         public GameIconsConfig IconsConfig
00036         {
00037             get => _iconsConfig;
00038         }
00039
00040         public InventoryConfig InventoryConfig
00041         {
00042             get => _inventoryConfig;
00043         }
00044
00045         public LevelConfig LevelConfig
00046         {
00047             get => _levelConfig;
00048         }
00049
00050         public LevelUpConfig LevelUpConfig
00051         {
00052             get => _levelUpConfig;
00053         }
00054
00055         public Dictionary<string, EffectConfig> EffectConfigs
00056         {
00057             get => _effectConfigs;
00058         }
00059
00060         public List<ElementConfig> ElementConfigs
00061         {
00062             get => _elementConfigs;
00063         }
00064
00065         public Dictionary<ElementType, FieldElementView> Prefabs
00066         {
00067             get => _elementsPrefabs;
00068         }
00069
00070         public HintsConfig HintsConfig
00071         {
00072             get => _hintsConfig;
00073         }
00074
00075         public void StoreElementsPrefabs(ElementPrefab[] prefabs)
```

```

00076      {
00077          _elementsPrefabs = prefabs.ToDictionary(key => key.Type, value => value.Prefab);
00078      }
00079
00080      public void StoreGameConfig(GameConfig config)
00081      {
00082          _gameConfig = config;
00083      }
00084
00085      public void StoreIconsConfig(GameIconsConfig config)
00086      {
00087          _iconsConfig = config;
00088      }
00089
00090      public void StoreInventoryConfig(InventoryConfig config)
00091      {
00092          _inventoryConfig = config;
00093      }
00094
00095      public void StoreLevelConfig(LevelConfig config)
00096      {
00097          _levelConfig = config;
00098
00099          ExtractElementsConfigs(_levelConfig);
00100     }
00101
00102      public void StoreLevelUpConfig(LevelUpConfig config)
00103      {
00104          _levelUpConfig = config;
00105      }
00106
00107      public void StoreEffectsConfigs(EffectConfig[] configs)
00108      {
00109          _effectConfigs = configs.ToDictionary(config => config.Name, config => config);
00110      }
00111
00112      private void ExtractElementsConfigs(LevelConfig levelConfig)
00113      {
00114          List<EvolutionData> evolutionScheme = levelConfig.EvolutionsScheme.Evolution;
00115
00116          for (int i = 0; i < evolutionScheme.Count; i++)
00117          {
00118              EvolutionData evolutionData = evolutionScheme[i];
00119
00120              foreach (ElementConfig elementConfig in evolutionData.Chain)
00121              {
00122                  _elementConfigs.Add(elementConfig);
00123              }
00124          }
00125      }
00126
00127      public void StoreHintsConfig(HintsConfig hintsConfig)
00128      {
00129          _hintsConfig = hintsConfig;
00130      }
00131  }
00132 }

```

7.188 ConfigProcessor.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using MergeIt.Core.Configs;
00006 using MergeIt.Core.Configs.Data;
00007 using MergeIt.Core.Configs.Elements;
00008 using MergeIt.Core.Configs.Types;
00009 using MergeIt.Core.FieldElements;
00010 using MergeIt.Core.Services;
00011 using MergeIt.Game.Field;
00012 using MergeIt.Game.Helpers;
00013 using MergeIt.SimpleDI;
00014
00015 namespace MergeIt.Game.Converters
00016 {
00017     public class ConfigProcessor : IConfigProcessor
00018     {
00019         [Introduce]
00020         private IConfigsService _configsService;
00021
00022         [Introduce]
00023         private FieldLogicModel _fieldLogicModel;

```

```

00024
00025     public FieldData Convert(LevelConfig levelConfig)
00026     {
00027         var saveFieldData = new FieldData
00028         {
00029             FieldHeight = levelConfig.FieldHeight,
00030             FieldWidth = levelConfig.FieldWidth,
00031             SavedElementsData = new List<FieldElementData>()
00032         };
00033
00034         foreach (LevelElementData levelElementData in levelConfig.FieldElementsData)
00035         {
00036             FieldElementData savedElementData = ConvertToFieldElementData(levelElementData);
00037             saveFieldData.SavedElementsData.Add(savedElementData);
00038         }
00039
00040         return saveFieldData;
00041     }
00042
00043     public FieldData BuildLevel()
00044     {
00045         var levelConfig = _configsService.LevelConfig;
00046         var elements = _fieldLogicModel.FieldElements;
00047
00048         var saveFieldData = new FieldData
00049         {
00050             FieldHeight = levelConfig.FieldHeight,
00051             FieldWidth = levelConfig.FieldWidth,
00052             SavedElementsData = new List<FieldElementData>()
00053         };
00054
00055         foreach (IFieldElement fieldElement in elements.Values)
00056         {
00057             var savedElementData = ConvertToFieldElementData(fieldElement);
00058             saveFieldData.SavedElementsData.Add(savedElementData);
00059         }
00060
00061         return saveFieldData;
00062     }
00063
00064     public FieldElementData ConvertToFieldElementData(LevelElementData levelElementData)
00065     {
00066         ElementConfig elementConfig = levelElementData.Element;
00067         FieldElementData fieldElementData = null;
00068
00069         if (elementConfig != null)
00070         {
00071             fieldElementData = new FieldElementData
00072             {
00073                 InfoParameters = new SavedInfoParameters
00074                 {
00075                     LogicPosition = levelElementData.Position,
00076                     IsBlocked = levelElementData.IsBlocked
00077                 },
00078                 ConfigParameters = new SavedConfigParameters
00079                 {
00080                     ElementId = levelElementData.Element.Id,
00081                     EvolutionId = levelElementData.EvolutionId
00082                 }
00083             };
00084
00085             switch (elementConfig.Type)
00086             {
00087                 case ElementType.Generator:
00088                     var generatorParameters = new SavedGeneratorParameters
00089                     {
00090                         DroppedElements = 0
00091                     };
00092
00093                     if (elementConfig.GeneratorSettings.Charged)
00094                     {
00095                         generatorParameters.AvailableToDrop =
00096                         elementConfig.GeneratorSettings.MaxDrop;
00097                     }
00098                     else
00099                     {
00100                         generatorParameters.StartChargingTime =
00101                         DateTimeOffset.UtcNow.ToUnixTimeSeconds();
00102                     }
00103
00104                     fieldElementData.GeneratorParameters = generatorParameters;
00105
00106                     if (elementConfig.GeneratorSettings.NeedOpen)
00107                     {
00108                         fieldElementData.GeneratorOpenParameters = new
00109                         SavedGeneratorOpenParameters();
00110                     }
00111
00112             }
00113         }
00114     }

```

```
00108                     break;
00109                 }
00110             }
00111         }
00112     }
00113     return fieldElementData;
00114 }
00115
00116     public FieldElementData ConvertToFieldElementData(IFieldElement fieldElement)
00117     {
00118         var fieldElementData = new FieldElementData
00119         {
00120             ConfigParameters = new SavedConfigParameters
00121             {
00122                 ElementId = fieldElement.ConfigParameters.ElementConfig.Id,
00123                 EvolutionId = fieldElement.ConfigParameters.EvolutionData.Id
00124             },
00125
00126             InfoParameters = new SavedInfoParameters
00127             {
00128                 LogicPosition = fieldElement.InfoParameters.LogicPosition,
00129                 IsBlocked = fieldElement.InfoParameters.IsBlocked
00130             }
00131         };
00132
00133         if (fieldElement.GeneratorParameters != null)
00134         {
00135             var generatorParameters = new SavedGeneratorParameters
00136             {
00137                 AvailableToDrop = fieldElement.GeneratorParameters.AvailableToDrop,
00138                 StartChargingTime = fieldElement.GeneratorParameters.StartChargingTime,
00139                 DroppedElements = fieldElement.GeneratorParameters.DroppedElements,
00140                 ChargedCount = fieldElement.GeneratorParameters.ChargedCount,
00141             };
00142
00143             fieldElementData.GeneratorParameters = generatorParameters;
00144
00145             if (fieldElement.GeneratorOpenParameters != null)
00146             {
00147                 fieldElementData.GeneratorOpenParameters = new SavedGeneratorOpenParameters
00148                 {
00149                     StartOpeningTime = fieldElement.GeneratorOpenParameters.StartOpeningTime
00150                 };
00151             }
00152         }
00153
00154         return fieldElementData;
00155     }
00156
00157     public IFIELDElement ConvertToFieldElement(FieldElementData data)
00158     {
00159         ElementConfig elementConfig = _configsService.GetConfig(data.ConfigParameters.ElementId);
00160
00161         IFIELDElement fieldElement = new FieldElement();
00162
00163         fieldElement.ConfigParameters = CreateConfigParameters(data.ConfigParameters);
00164         fieldElement.InfoParameters = CreateInfoParameters(data.InfoParameters, elementConfig);
00165         fieldElement.ProducedByParameters = CreateProducedByParameters(elementConfig);
00166
00167         switch (elementConfig.Type)
00168         {
00169             case ElementType.Generator:
00170                 ElementGeneratorSettings generatorSettings = elementConfig.GeneratorSettings;
00171                 fieldElement.GeneratorParameters =
00172                     CreateGeneratorParameters(data.GeneratorParameters);
00173                 fieldElement.ProduceParameters = CreateProduceParameters(generatorSettings);
00174
00175                 if (data.GeneratorOpenParameters != null &&
00176                     generatorSettings.NeedOpen)
00177                 {
00178                     fieldElement.GeneratorOpenParameters =
00179                         CreateGeneratorOpenParameters(data.GeneratorOpenParameters);
00180
00181                     fieldElement.UpdateGenerator();
00182                     break;
00183                 }
00184
00185             return fieldElement;
00186         }
00187
00188         private IConfigParameters CreateConfigParameters(SavedConfigParameters parameters)
00189         {
00190             ElementConfig elementConfig = _configsService.GetConfig(parameters.ElementId);
00191             EvolutionData evolutionData = _configsService.GetEvolutionData(parameters.EvolutionId);
00192
00193             return new ConfigParameters
```

```

00193         {
00194             ElementConfig = elementConfig,
00195             EvolutionData = evolutionData
00196         };
00197     }
00198
00199     private IGeneratorParameters CreateGeneratorParameters(SavedGeneratorParameters data)
00200     {
00201         var generatorParameters = new GeneratorParameters();
00202
00203         generatorParameters.CopyFrom(data);
00204
00205         return generatorParameters;
00206     }
00207
00208     private IInfoParameters CreateInfoParameters(SavedInfoParameters parameters, ElementConfig
config)
00209     {
00210         return new InfoParameters
00211         {
00212             LogicPosition = parameters.LogicPosition,
00213             IsBlocked = parameters.IsBlocked,
00214             Name = config.CommonSettings.Name,
00215             Description = config.GetDescription(),
00216             Type = config.Type
00217         };
00218     }
00219
00220     private IProduceParameters CreateProduceParameters(ElementGeneratorSettings generatorSettings)
00221     {
00222         IProduceParameters produceParameters = new ProduceParameters();
00223         produceParameters.Elements = new List<GeneratableFieldElement>();
00224
00225         foreach (GeneratableElement generatableElement in generatorSettings.GenerateItems)
00226         {
00227             ElementConfig generatableConfig = generatableElement.Element;
00228             produceParameters.Elements.Add(new GeneratableFieldElement
00229             {
00230                 Config = generatableConfig,
00231                 Possibility = generatableElement.Possibility
00232             });
00233         }
00234
00235         return produceParameters;
00236     }
00237
00238     private IGeneratorOpenParameters CreateGeneratorOpenParameters(SavedGeneratorOpenParameters
data)
00239     {
00240         IGeneratorOpenParameters generatorOpenParameters = new GeneratorOpenParameters();
00241
00242         generatorOpenParameters.CopyFrom(data);
00243
00244         return generatorOpenParameters;
00245     }
00246
00247     private IProducedByParameters CreateProducedByParameters(ElementConfig element)
00248     {
00249         IProducedByParameters producedByParameters = null;
00250         foreach (ElementConfig elementConfig in _configsService.GetConfigs())
00251         {
00252             if (elementConfig.Type != ElementType.Generator ||
00253                 elementConfig == element)
00254             {
00255                 continue;
00256             }
00257
00258             ElementGeneratorSettings generatorParameters = elementConfig.GeneratorSettings;
00259             if (generatorParameters.GenerateItems.Exists(x => x.Element == elementConfig))
00260             {
00261                 if (producedByParameters == null)
00262                 {
00263                     producedByParameters = new ProducedByParameters();
00264                     producedByParameters.Elements = new List<ElementConfig>();
00265                 }
00266
00267                 producedByParameters.Elements.Add(elementConfig);
00268             }
00269         }
00270
00271         return producedByParameters;
00272     }
00273 }
00274 }
```

7.189 IConfigProcessor.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs;
00004 using MergeIt.Core.Configs.Data;
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Game.Field;
00007
00008 namespace MergeIt.Game.Converters
00009 {
00010     public interface IConfigProcessor
00011     {
00012         FieldData Convert(LevelConfig levelConfig);
00013         FieldElementData ConvertToFieldElementData(IFieldElement fieldElement);
00014         FieldElementData ConvertToFieldElementData(LevelElementData levelElementData);
00015         FieldData BuildLevel();
00016         IFIELDElement ConvertToFieldElement(FieldElementData data);
00017     }
00018 }
```

7.190 JumpEffectConfig.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Effects;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Effects.Configs
00007 {
00008     [CreateAssetMenu(fileName = "JumpEffectConfig", menuName = "Merge Toolkit/Effects/Jump Effect
Config")]
00009     public class JumpEffectConfig : EffectConfig
00010     {
00011         public AnimationCurve CurveY;
00012         public AnimationCurve ScaleCurve;
00013     }
00014 }
```

7.191 MoveEffectConfig.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Effects;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Effects.Configs
00007 {
00008     [CreateAssetMenu(fileName = "MoveEffectConfig", menuName = "Merge Toolkit/Effects/Move Effect
Config")]
00009     public class MoveEffectConfig : EffectConfig
00010     {
00011         public AnimationCurve PositionXCurve;
00012         public AnimationCurve PositionYCurve;
00013         public AnimationCurve ScaleCurve;
00014     }
00015
00016 }
```

7.192 MoveToUiEffectConfig.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Effects;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Effects.Configs
00007 {
00008     [CreateAssetMenu(fileName = "MoveToUiEffectConfig", menuName = "Merge Toolkit/Effects/Move To UI
Effect Config")]
00009     public class MoveToUiEffectConfig : EffectConfig
00010     {
00011         public AnimationCurve PositionXCurve;
00012         public AnimationCurve PositionYCurve;
00013         public AnimationCurve ScaleCurve;
00014     }
00015 }
```

7.193 BaseEffect.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Services;
00005 using MergeIt.Game.Effects.Parameters;
00006 using MergeIt.SimpleDI;
00007 using UnityEngine;
00008
00009 namespace MergeIt.Game.Effects.Controllers
00010 {
00011     public abstract class BaseEffect : IEffector
00012     {
00013         protected Transform Target;
00014         protected Animator Animator;
00015         protected Action FinishedCallback;
00016
00017         protected readonly IConfigsService ConfigsService = DiContainer.Get<IConfigsService>();
00018
00019         public bool Started { get; set; }
00020
00021         public virtual void Setup(Transform target, IEffectorParameters effectParameters = null, Action
00022             finishedCallback = null)
00023         {
00024             Target = target;
00025             FinishedCallback = finishedCallback;
00026         }
00027
00028         protected void Finish()
00029         {
00030             Started = false;
00031             FinishedCallback?.Invoke();
00032             FinishedCallback = null;
00033
00034             if (Animator)
00035             {
00036                 Animator.enabled = true;
00037             }
00038
00039             OnFinished();
00040         }
00041
00042         protected virtual void OnStarted()
00043         {
00044         }
00045
00046         protected virtual void OnFinished()
00047         {
00048         }
00049
00050
00051         public virtual void Start()
00052         {
00053             Started = true;
00054
00055             Target.TryGetComponent(out Animator);
00056
00057             if (Animator)
00058             {
00059                 Animator.enabled = false;
00060             }
00061
00062             OnStarted();
00063         }
00064
00065         public abstract void Update();
00066     }
00067 }
00068 }
```

7.194 BaseEffectWithPresenter.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Game.Effects.Parameters;
00006
00007 namespace MergeIt.Game.Effects.Controllers
00008 {
00009     public abstract class BaseEffectWithPresenter : BaseEffect, IEffectorWithPresenter
```

```

00010     {
00011         protected IFIELDELEMENTPRESENTER Presenter;
00012
00013         public virtual void Setup(IFIELDELEMENTPRESENTER presenter, IEFFECTPARAMETERS effectParameters
00014 = null, Action finishedCallback = null)
00015         {
00016             base.Setup(presenter.RectTransform, effectParameters, finishedCallback);
00017             Presenter = presenter;
00018         }
00019
00020         public IFIELDELEMENTPRESENTER GetPresenter()
00021         {
00022             return Presenter;
00023         }
00024
00025         protected virtual void FinishCallbackHandler()
00026         {
00027             Presenter.Canvas.sortingOrder = 1;
00028             // Presenter.UpdateInitialPosition();
00029             Presenter.SetBusy(false);
00030
00031             Finish();
00032         }
00033     }
00034 }
```

7.195 IEffect.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Game.Effects.Parameters;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Game.Effects.Controllers
00008 {
00009     public interface IEFFECT
00010     {
00011         void Start();
00012         void Update();
00013         void Setup(Transform target, IEFFECTPARAMETERS effectParameters = null, Action
00014 finishedCallback = null);
00015 }
```

7.196 IEFFECTWITHPRESENTER.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Game.Effects.Parameters;
00006
00007 namespace MergeIt.Game.Effects.Controllers
00008 {
00009     public interface IEFFECTWITHPRESENTER
00010     {
00011         void Setup(IFIELDELEMENTPRESENTER presenter, IEFFECTPARAMETERS effectParameters = null, Action
00012 finishedCallback = null);
00013         IFIELDELEMENTPRESENTER GetPresenter();
00014     }
00015 }
```

7.197 JumpEffect.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Helpers;
00005 using MergeIt.Game.Effects.Configs;
00006 using MergeIt.Game.Effects.Parameters;
00007 using UnityEngine;
00008
```

```

00009 namespace MergeIt.Game.Effects.Controllers
00010 {
00011     public class JumpEffect : BaseEffect
00012     {
00013         private readonly JumpEffectConfig _effectConfig;
00014
00015         private float _progress;
00016         private Vector2 _initialPosition;
00017
00018         private AnimationCurve _jumpCurve;
00019         private AnimationCurve _scaleCurve;
00020         private Vector3 _initialScale;
00021
00022         public JumpEffect()
00023         {
00024             _effectConfig = ConfigsService.GetEffectConfig<JumpEffectConfig>(EffectName.JumpElement);
00025         }
00026
00027         public override void Update()
00028         {
00029             if (Started)
00030             {
00031                 _progress += Time.deltaTime / _effectConfig.Duration;
00032
00033                 float posY = _jumpCurve.Evaluate(_progress);
00034                 float scale = _scaleCurve.Evaluate(_progress);
00035
00036                 Vector3 position = Target.position;
00037                 Vector3 localScale = Target.localScale;
00038                 position.y = posY;
00039                 localScale.x = scale;
00040                 localScale.y = scale;
00041
00042                 Target.position = position;
00043                 Target.localScale = localScale;
00044
00045                 if (_progress >= 1f)
00046                 {
00047                     Target.position = _initialPosition;
00048                     Target.localScale = _initialScale;
00049
00050                     Finish();
00051                 }
00052             }
00053         }
00054
00055         public override void Setup(Transform target, IEffectParameters effectParameters = null, Action
finishedCallback = null)
00056         {
00057             base.Setup(target, effectParameters, finishedCallback);
00058
00059             _initialPosition = target.position;
00060             _initialScale = target.localScale;
00061
00062             _jumpCurve = CurvesHelper.CopyCurveFrom(_effectConfig.CurveY, _initialPosition.y);
00063             _scaleCurve = CurvesHelper.CopyCurveFrom(_effectConfig.CurveY, _initialScale.x);
00064         }
00065     }
00066 }

```

7.198 JumpEffectWithPresenter.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Game.Effects.Parameters;
00006
00007 namespace MergeIt.Game.Effects.Controllers
00008 {
00009     public class JumpEffectWithPresenter : BaseEffectWithPresenter
00010     {
00011         private readonly JumpEffect _effect = new();
00012
00013         public override void Start()
00014         {
00015             base.Start();
00016
00017             _effect.Start();
00018         }
00019
00020         public override void Setup(IFieldElementPresenter presenter, IEffectParameters
effectParameters = null, Action finishedCallback = null)

```

```

00021      {
00022          base.Setup(presenter, effectParameters, finishedCallback);
00023
00024          _effect.Setup(presenter.RectTransform, effectParameters, FinishCallbackHandler);
00025      }
00026
00027      public override void Update()
00028      {
00029          if (_effect.Started)
00030          {
00031              _effect.Update();
00032          }
00033      }
00034
00035      protected override void OnStarted()
00036      {
00037          base.OnStarted();
00038
00039          Presenter.Canvas.sortingOrder = 2;
00040          Presenter.SetBusy(true);
00041      }
00042  }
00043 }
```

7.199 MoveEffect.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Helpers;
00005 using MergeIt.Game.Effects.Configs;
00006 using MergeIt.Game.Effects.Parameters;
00007 using MergeIt.Game.Enums;
00008 using MergeIt.Game.Helpers;
00009 using UnityEngine;
00010
00011 namespace MergeIt.Game.Effects.Controllers
00012 {
00013     public class MoveEffect : BaseEffect
00014     {
00015         private readonly MoveEffectConfig _effectConfig;
00016
00017         private float _progress;
00018         private Vector3 _moveFrom;
00019         private Vector3 _moveTo;
00020
00021         private AnimationCurve _scaleCurve;
00022
00023         public MoveEffect()
00024         {
00025             _effectConfig = ConfigsService.GetEffectConfig<MoveEffectConfig>(EffectName.MoveElement);
00026         }
00027
00028         public override void Setup(Transform target, IEFFECTPARAMETERS effectParameters = null, Action
finishedCallback = null)
00029         {
00030             base.Setup(target, effectParameters, finishedCallback);
00031
00032             if (effectParameters is MoveEffectParameters moveEffectParameters)
00033             {
00034                 Setup(moveEffectParameters.FromPosition, moveEffectParameters.ToPosition,
moveEffectParameters.TargetInHud);
00035             }
00036         }
00037
00038         public void Setup(Vector3 moveFrom, Vector3 moveTo, HudPanelType targetInHud)
00039         {
00040             _moveFrom = moveFrom;
00041             _moveTo = moveTo;
00042
00043             Target.position = _moveFrom;
00044
00045             if (targetInHud != 0)
00046             {
00047                 _moveTo = HudTargets.GetTarget(targetInHud).position;
00048             }
00049
00050             _scaleCurve = CurvesHelper.CopyCurveFrom(_effectConfig.ScaleCurve, Target.localScale.x);
00051         }
00052
00053         public override void Update()
00054         {
00055             if (Started)
```

```

00056         {
00057             _progress += Time.deltaTime / _effectConfig.Duration;
00058
00059             float posX = _effectConfig.PositionXCurve.Evaluate(_progress);
00060             float posY = _effectConfig.PositionYCurve.Evaluate(_progress);
00061             float scale = _scaleCurve.Evaluate(_progress);
00062
00063             Vector3 position = Vector3.Lerp(_moveFrom, _moveTo, _progress);
00064             position.x += posX;
00065             position.y += posY;
00066
00067             Target.position = position;
00068             Target.localScale = new Vector3(scale, scale, 1f);
00069
00070             if (_progress >= 1f)
00071             {
00072                 Target.position = _moveTo;
00073                 Target.localScale = Vector3.one;
00074
00075                 Finish();
00076             }
00077         }
00078     }
00079 }
00080 }
```

7.200 MoveEffectWithPresenter.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Game.Effects.Parameters;
00006
00007 namespace MergeIt.Game.Effects.Controllers
00008 {
00009     public class MoveEffectWithPresenter : BaseEffectWithPresenter
00010     {
00011         private readonly MoveEffect _effect = new();
00012
00013         public override void Start()
00014         {
00015             base.Start();
00016
00017             _effect.Start();
00018         }
00019
00020         public override void Setup(IFieldElementPresenter presenter, IEFFECTParameters
00021             effectParameters = null, Action finishedCallback = null)
00022         {
00023             base.Setup(presenter, effectParameters, finishedCallback);
00024
00025             _effect.Setup(presenter.RectTransform, effectParameters, FinishCallbackHandler);
00026         }
00027
00028         public override void Update()
00029         {
00030             if (_effect.Started)
00031             {
00032                 _effect.Update();
00033             }
00034
00035             protected override void OnStarted()
00036             {
00037                 base.OnStarted();
00038
00039                 Presenter.Canvas.sortingOrder = 2;
00040                 Presenter.SetBusy(true);
00041             }
00042
00043             protected override void OnFinished()
00044             {
00045                 base.OnFinished();
00046
00047                 Presenter.Canvas.sortingOrder = 1;
00048                 // _presenter.UpdateInitialPosition();
00049                 Presenter.SetBusy(false);
00050             }
00051         }
00052 }
```

7.201 MoveToUiEffect.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Helpers;
00005 using MergeIt.Game.Effects.Configs;
00006 using MergeIt.Game.Effects.Parameters;
00007 using MergeIt.Game.Helpers;
00008 using UnityEngine;
00009
00010 namespace MergeIt.Game.Effects.Controllers
00011 {
00012     public class MoveToUiEffect : BaseEffect
00013     {
00014         private MoveToUiEffectConfig _effectConfig;
00015
00016         private float _progress;
00017         private Vector3 _moveFrom;
00018         private Vector3 _moveTo;
00019
00020         private AnimationCurve _scaleCurve;
00021
00022         public override void Setup(Transform target, IEffectParameters effectParameters = null, Action
finishedCallback = null)
00023         {
00024             base.Setup(target, effectParameters, finishedCallback);
00025
00026             if (effectParameters?.OverriddenName != null)
00027             {
00028                 _effectConfig =
ConfigsService.GetEffectConfig<MoveToUiEffectConfig>(effectParameters.OverriddenName);
00029             }
00030             else
00031             {
00032                 _effectConfig =
ConfigsService.GetEffectConfig<MoveToUiEffectConfig>(EffectName.MoveToUiElement);
00033             }
00034
00035             if (effectParameters is MoveEffectParameters moveEffectParameters)
00036             {
00037                 _moveTo = HudTargets.GetTarget(moveEffectParameters.TargetInHud).position;
00038                 _moveFrom = moveEffectParameters.FromPosition;
00039             }
00040
00041             _scaleCurve = CurvesHelper.CopyCurveFrom(_effectConfig.ScaleCurve, Target.localScale.x);
00042         }
00043
00044         public override void Update()
00045         {
00046             if (Started)
00047             {
00048                 _progress += Time.deltaTime / _effectConfig.Duration;
00049
00050                 float posX = _effectConfig.PositionXCurve.Evaluate(_progress);
00051                 float posY = _effectConfig.PositionYCurve.Evaluate(_progress);
00052                 float scale = _scaleCurve.Evaluate(_progress);
00053
00054                 Vector3 position = Vector3.Lerp(_moveFrom, _moveTo, _progress);
00055                 position.x += posX;
00056                 position.y += posY;
00057
00058                 Target.position = position;
00059                 Target.localScale = new Vector3(scale, scale, 1f);
00060
00061                 if (_progress >= 1f)
00062                 {
00063                     Finish();
00064                 }
00065             }
00066         }
00067     }
00068 }
```

7.202 MoveToUiEffectWithPresenter.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Game.Effects.Parameters;
00006
00007 namespace MergeIt.Game.Effects.Controllers
```

```

00008 {
00009     public class MoveToUiEffectWithPresenter : BaseEffectWithPresenter
00010     {
00011         private readonly MoveToUiEffect _effect = new();
00012
00013         public override void Start()
00014         {
00015             base.Start();
00016
00017             _effect.Start();
00018         }
00019
00020         public override void Setup(IFieldElementPresenter presenter, IEffectParameters
00021             effectParameters = null, Action finishedCallback = null)
00022         {
00023             base.Setup(presenter, effectParameters, finishedCallback);
00024
00025             _effect.Setup(presenter.RectTransform, effectParameters, FinishCallbackHandler);
00026         }
00027
00028         public override void Update()
00029         {
00030             if (_effect.Started)
00031             {
00032                 _effect.Update();
00033             }
00034
00035         protected override void OnStarted()
00036         {
00037             base.OnStarted();
00038
00039             Presenter.Canvas.sortingOrder = 2;
00040             Presenter.SetBusy(true);
00041         }
00042
00043         // protected void OnFinishedSubEffect()
00044         // {
00045         //     Presenter.Canvas.sortingOrder = 1;
00046         //     // _presenter.UpdateInitialPosition();
00047         //     // Presenter.SetBusy(false);
00048         //     //
00049         //     // Finish();
00050         // }
00051     }
00052 }

```

7.203 EffectName.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Game.Effects
00004 {
00005     public static class EffectName
00006     {
00007         public const string MoveElement = "Move Element";
00008         public const string JumpElement = "Jump Element";
00009         public const string MoveToStock = "Move To Stock";
00010         public const string MoveToUiElement = "Move To UI";
00011     }
00012 }

```

7.204 EffectsFactory.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Game.Effects.Controllers;
00006 using MergeIt.Game.Effects.Parameters;
00007 using MergeIt.SimpleDI;
00008 using UnityEngine;
00009
00010 namespace MergeIt.Game.Effects
00011 {
00012     public class EffectsFactory : IEffectorFactory
00013     {
00014         [Introduce]
00015         private IEffectorManager _effectorManager;

```

```

00016     public void CreateEffect<T>(IFieldElementPresenter target, IEffectParameters effectParameters
00017     = null, Action finishedCallback = null)
00018         where T : IEffectWithPresenter, IEffect, new()
00019     {
00020         var effectController = new T();
00021         effectController.Setup(target, effectParameters, finishedCallback);
00022         _effectsManager.AddEffect(effectController);
00023     }
00024
00025
00026     public void CreateEffect<T>(RectTransform target, IEffectParameters effectParameters = null,
00027         Action finishedCallback = null)
00028         where T : IEffect, new()
00029     {
00030         var effectController = new T();
00031         effectController.Setup(target, effectParameters, finishedCallback);
00032         _effectsManager.AddEffect(effectController);
00033     }
00034 }
00035 }
```

7.205 EffectsManager.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Messages;
00005 using MergeIt.Core.Services;
00006 using MergeIt.Game.Effects.Controllers;
00007 using MergeIt.SimpleDI;
00008 using MergeIt.SimpleDI.ReservedInterfaces;
00009
00010 namespace MergeIt.Game.Effects
00011 {
00012     public class EffectsManager : IEffectsManager, IUpdatable
00013     {
00014         private readonly List<IEffect> _effectControllers = new();
00015
00016         [Introduce]
00017         private IMessageBus _messageBus;
00018
00019         [Introduce]
00020         private IConfigsService _configsService;
00021
00022         public void AddEffect(IEffect effectController)
00023         {
00024             _effectControllers.Add(effectController);
00025             effectController.Start();
00026         }
00027
00028         public void Update()
00029         {
00030             for (int i = 0; i < _effectControllers.Count; i++)
00031             {
00032                 IEffect effect = _effectControllers[i];
00033                 effect.Update();
00034             }
00035         }
00036
00037         private void OnEffectFinished(IEffect effectController)
00038         {
00039             _effectControllers.Remove(effectController);
00040         }
00041     }
00042 }
```

7.206 IEffectsFactory.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Game.Effects.Controllers;
00006 using MergeIt.Game.Effects.Parameters;
00007 using UnityEngine;
00008
00009 namespace MergeIt.Game.Effects
00010 {
```

```

00011     public interface IEffectsFactory
00012     {
00013         void CreateEffect<T>(IFieldElementPresenter target, IEffectParameters effectParameters = null,
00014             Action finishedCallback = null)
00015             where T : IEffectWithPresenter, IEffect, new();
00016         void CreateEffect<T>(RectTransform target, IEffectParameters effectParameters = null, Action
00017             finishedCallback = null)
00018             where T : IEffect, new();
00019     }

```

7.207 IEffectsManager.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Game.Effects.Controllers;
00004
00005 namespace MergeIt.Game.Effects
00006 {
00007     public interface IEffectsManager
00008     {
00009         void AddEffect(IEffect effectController);
00010     }
00011 }

```

7.208 IEffectParameters.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Game.Effects.Parameters
00004 {
00005     public interface IEffectParameters
00006     {
00007         string OverriddenName { get; }
00008     }
00009 }

```

7.209 MoveEffectParameters.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Game.Enums;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Effects.Parameters
00007 {
00008     public class MoveEffectParameters : IEffectParameters
00009     {
00010         private readonly Vector3 _fromPosition;
00011         private readonly Vector3 _toPosition;
00012         private readonly HudPanelType _targetInHud;
00013
00014         public string OverriddenName { get; }
00015
00016         public Vector3 FromPosition
00017         {
00018             get => _fromPosition;
00019         }
00020
00021         public Vector3 ToPosition
00022         {
00023             get => _toPosition;
00024         }
00025
00026         public HudPanelType TargetInHud
00027         {
00028             get => _targetInHud;
00029         }
00030
00031         public MoveEffectParameters(Vector3 fromPosition, Vector3 toPosition = default, HudPanelType
00032             targetInHud = 0, string overriddenName = null)
00033         {
00034             OverriddenName = overriddenName;
00035         }
00036     }
00037 }

```

```
00035         _fromPosition = fromPosition;
00036         _toPosition = toPosition;
00037         _targetInHud = targetInHud;
00038     }
00039 }
00040 }
```

7.210 ElementsStockComponent.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.ElementsStock;
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Core.Messages;
00007 using MergeIt.Core.Services;
00008 using MergeIt.Game.Messages;
00009 using MergeIt.SimpleDI;
00010 using UnityEngine;
00011 using UnityEngine.UI;
00012
00013 namespace MergeIt.Game.ElementsStock
00014 {
00015     public class ElementsStockComponent : MonoBehaviour, IElementsStockComponent
00016     {
00017         [SerializeField]
00018         private Button _elementButton;
00019
00020         [SerializeField]
00021         private Transform _elementContainer;
00022
00023         private IMessageBus _messageBus;
00024         private IGameFieldService _fieldService;
00025         private IElementsStockService _stockService;
00026
00027         private ElementConfig _currentElement;
00028         private GameObject _iconPrefab;
00029
00030         public void Initialize()
00031     {
00032         _elementButton.onClick.AddListener(OnElementClick);
00033
00034         _messageBus = DiContainer.Get<IMessageBus>();
00035         _fieldService = DiContainer.Get<IGameFieldService>();
00036         _stockService = DiContainer.Get<IElementsStockService>();
00037
00038         _messageBus.AddListener<UpdateStockMessage>(UpdateStockMessageHandler);
00039
00040         SetupElement(_stockService.GetNext());
00041     }
00042
00043     public void OnDestroy()
00044     {
00045         _elementButton.onClick.RemoveListener(OnElementClick);
00046         _messageBus.RemoveListener<UpdateStockMessage>(UpdateStockMessageHandler);
00047     }
00048
00049     public void SetupElement(ElementConfig elementConfig)
00050     {
00051         if (elementConfig != null)
00052         {
00053             gameObject.SetActive(true);
00054             _currentElement = elementConfig;
00055
00056             UpdateView();
00057         }
00058         else
00059         {
00060             Hide();
00061         }
00062     }
00063
00064     public void PopElement()
00065     {
00066         if (_currentElement != null)
00067         {
00068             GridPoint? pointContainer = _fieldService.GetFreeCell();
00069
00070             if (pointContainer != null)
00071             {
00072                 GridPoint point = pointContainer.Value;
00073
00074                 IFieldElement newElement = _fieldService.CreateNewElement(_currentElement, point);
```

```

00075         var message = new CreateElementMessage
00076         {
00077             NewElement = newElement,
00078             FromPosition = _elementContainer.position,
00079             ToPoint = point,
00080         };
00081         _messageBus.Fire(message);
00082
00083         if (_iconPrefab)
00084         {
00085             Destroy(_iconPrefab);
00086         }
00087
00088         if (_stockService.Remove())
00089         {
00090             SetupElement(_stockService.GetNext());
00091         }
00092         else
00093         {
00094             Hide();
00095         }
00096     }
00097 }
00098 }
00099 }
00100
00101     private void OnElementClick()
00102     {
00103         PopElement();
00104     }
00105
00106     private void Hide()
00107     {
00108         _currentElement = null;
00109         gameObject.SetActive(false);
00110     }
00111
00112     private void UpdateView()
00113     {
00114         if (_currentElement != null)
00115         {
00116             _iconPrefab = Instantiate(_currentElement.CommonSettings.Icon.gameObject);
00117             if (_iconPrefab)
00118             {
00119                 _iconPrefab.transform.SetParent(_elementContainer);
00120                 _iconPrefab.transform.localScale = Vector3.one;
00121             }
00122         }
00123     }
00124
00125     private void UpdateStockMessageHandler(UpdateStockMessage _)
00126     {
00127         if (!_currentElement)
00128         {
00129             SetupElement(_stockService.GetCurrent());
00130         }
00131     }
00132 }
00133 }
```

7.211 ElementsStockData.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.ElementsStock;
00005 using MergeIt.Core.Saves;
00006 using Newtonsoft.Json;
00007
00008 namespace MergeIt.Game.ElementsStock
00009 {
00010     [Serializable, Savable("stock", "dat")]
00011     public class ElementsStockData : IElementsStockData
00012     {
00013         [JsonProperty("e")]
00014         public string[] Elements { get; set; }
00015     }
00016 }
```

7.212 HudPanelType.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Game.Enums
00004 {
00005     public enum HudPanelType
00006     {
00007         None,
00008         UserProgress,
00009         SoftCurrency,
00010         HardCurrency,
00011         Energy,
00012         ElementStock
00013     }
00014 }
```

7.213 EvolutionsProgressData.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using MergeIt.Core.Evolutions;
00006 using MergeIt.Core.Saves;
00007 using Newtonsoft.Json;
00008
00009 namespace MergeIt.Game.Evolutions
00010 {
00011     [Serializable, Savable("evoprogress", "dat")]
00012     public class EvolutionsProgressData : IEvolutionsProgressData
00013     {
00014         [JsonProperty("ep")]
00015         public List<EvolutionProgressData> EvolutionsProgress { get; } = new();
00016     }
00017 }
```

7.214 ElementInfoFactory.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Configs.Elements;
00005 using MergeIt.Game.Windows.ElementInfo;
00006 using UnityEngine;
00007
00008 namespace MergeIt.Game.Factories.ElementInfo
00009 {
00010     public class ElementInfoFactory : IElementInfoFactory
00011     {
00012         private static Dictionary<ElementInfoType, string> PrefabsLocations = new()
00013         {
00014             {ElementInfoType.InfoWindow, "Prefabs/Windows/ElementInfo/ElementInfoWindowItem"},
00015             {ElementInfoType.UserProgressWindow,
00016             "Prefabs/Windows/UserProgress/UserProgressWindowItem"}
00017         };
00018         public ElementInfoItemComponent CreateElementWindowItem(ElementConfig elementConfig,
00019             ElementInfoType infoType = ElementInfoType.InfoWindow, bool isLocked = false)
00020         {
00021             if (PrefabsLocations.TryGetValue(infoType, out string itemInfoPath))
00022             {
00023                 GameObject itemPanelObject = Resources.Load<GameObject>(itemInfoPath);
00024
00025                 if (itemPanelObject)
00026                 {
00027                     var panelItemObject = Object.Instantiate(itemPanelObject);
00028                     if (panelItemObject.TryGetComponent(out ElementInfoItemComponent resultComponent))
00029                     {
00030                         FieldElementIconComponent iconPrototype = elementConfig.GetIconComponent();
00031
00032                         var icon = Object.Instantiate(iconPrototype, panelItemObject.transform);
00033                         var iconRectTransform = icon.GetComponent<RectTransform>();
00034
00035                         iconPrototype.SetBlocked(isLocked);
00036
00037                         resultComponent.SetIcon(iconRectTransform, isLocked);
00038                     }
00039                 }
00040             }
00041         }
00042     }
00043 }
```

```

00040             return resultComponent;
00041         }
00042     }
00043
00044     return null;
00045 }
00046
00047 public ElementInfoItemComponent CreateUnknownElementWindowItem(ElementInfoType infoType =
    ElementInfoType.InfoWindow)
00048 {
00049     if (PrefabsLocations.TryGetValue(infoType, out string itemInfoPath))
00050     {
00051         GameObject itemPanelObject = Resources.Load<GameObject>(itemInfoPath);
00052
00053         if (itemPanelObject)
00054         {
00055             var panelItemObject = Object.Instantiate(itemPanelObject);
00056             if (panelItemObject.TryGetComponent(out ElementInfoItemComponent resultComponent))
00057             {
00058                 resultComponent.SetUnknown();
00059             }
00060
00061             return resultComponent;
00062         }
00063     }
00064
00065     return null;
00066 }
00067 }
00068 }
```

7.215 ElementInfoType.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Game.Factories.ElementInfo
00004 {
00005     public enum ElementInfoType
00006     {
00007         InfoWindow,
00008         UserProgressWindow
00009     }
00010 }
```

7.216 IElementInfoFactory.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Game.Windows.ElementInfo;
00005
00006 namespace MergeIt.Game.Factories.ElementInfo
00007 {
00008     public interface IElementInfoFactory
00009     {
00010         ElementInfoItemComponent CreateElementWindowItem(ElementConfig elementConfig, ElementInfoType
00011             infoType = ElementInfoType.InfoWindow, bool isLocked = false);
00012         ElementInfoItemComponent CreateUnknownElementWindowItem(ElementInfoType infoType =
00013             ElementInfoType.InfoWindow);
00014     }
00015 }
```

7.217 FieldFactory.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Services;
00005 using MergeIt.Game.Field;
00006 using MergeIt.SimpleDI;
00007 using UnityEngine;
00008
00009 namespace MergeIt.Game.Factories.Field
00010 {
```

```

00011     public class FieldFactory : IFieldFactory
00012     {
00013         public const string FieldGameObjectPath = "Prefabs/Field/FieldContainer";
00014         public const string FieldCellObjectOddPath = "Prefabs/Field/FieldCellOdd";
00015         public const string FieldCellObjectEvenPath = "Prefabs/Field/FieldCellEven";
00016
00017         [Introduce]
00018         private IResourcesLoaderService _resourcesLoaderService;
00019
00020         public FieldPresenter CreateField(Transform parent)
00021         {
00022             var fieldViewPrefab = _resourcesLoaderService.GetObject<FieldView>(FieldGameObjectPath);
00023             var fieldViewObject = Object.Instantiate(fieldViewPrefab.gameObject, parent);
00024
00025             fieldViewObject.TryGetComponent(out FieldView fieldView);
00026
00027             if (fieldView)
00028             {
00029                 var presenter = new FieldPresenter();
00030                 presenter.Initialize(fieldView);
00031
00032                 fieldView.Initialize();
00033
00034                 return presenter;
00035             }
00036
00037             return null;
00038         }
00039
00040         public FieldCellComponent CreateFieldCell(Transform parent, GridPoint gridPoint)
00041         {
00042             int sum = (gridPoint.X + gridPoint.Y) % 2;
00043             var fieldCellPrefab = sum == 0 ?
00044                 _resourcesLoaderService.GetObject<FieldCellComponent>(FieldCellObjectEvenPath) :
00045                 _resourcesLoaderService.GetObject<FieldCellComponent>(FieldCellObjectOddPath);
00046             var fieldCellObject = Object.Instantiate(fieldCellPrefab.gameObject, parent);
00047
00048             fieldCellObject.TryGetComponent(out FieldCellComponent fieldCell);
00049
00050             if (fieldCell)
00051             {
00052                 fieldCell.Initialize(gridPoint);
00053             }
00054
00055             return fieldCell;
00056         }
00057     }
00058 }
```

7.218 IFieldFactory.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Game.Field;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Game.Factories.Field
00008 {
00009     public interface IFieldFactory
00010     {
00011         FieldPresenter CreateField(Transform parent);
00012         FieldCellComponent CreateFieldCell(Transform parent, GridPoint gridPoint);
00013     }
00014 }
```

7.219 FieldElementFactory.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Configs.Data;
00005 using MergeIt.Core.Configs.Elements;
00006 using MergeIt.Core.Configs.Types;
00007 using MergeIt.Core.FieldElements;
00008 using MergeIt.Core.Services;
00009 using MergeIt.Game.Converters;
00010 using MergeIt.SimpleDI;
00011
```

```
00012 namespace MergeIt.Game.Factories.FieldElement
00013 {
00014     public class FieldElementFactory : IFieldElementFactory
00015     {
00016         [Introduce]
00017         private IConfigsService _configsService;
00018
00019         [Introduce]
00020         private IConfigProcessor _configProcessor;
00021
00022
00023         public IFieldElement CreateFieldElement(FieldElementData data)
00024         {
00025             return _configProcessor.ConvertToFieldElement(data);
00026         }
00027
00028         public IFieldElement CreateFieldElement(LevelElementData data)
00029         {
00030             FieldElementData fieldElementData = _configProcessor.ConvertToFieldElementData(data);
00031
00032             return _configProcessor.ConvertToFieldElement(fieldElementData);
00033         }
00034
00035         public IFieldElement CreateFieldElement(ElementConfig elementConfig, GridPoint point, bool
00036         isBlocked = false)
00037         {
00038             FieldElementData fieldElementData = CreateFieldElementData(elementConfig, point,
00039             isBlocked);
00040
00041             return _configProcessor.ConvertToFieldElement(fieldElementData);
00042
00043         private FieldElementData CreateFieldElementData(ElementConfig elementConfig, GridPoint point,
00044             bool isBlocked)
00045         {
00046             FieldElementData fieldElementData = new FieldElementData();
00047             ElementConfig config = elementConfig;
00048             string evolutionId = _configsService.GetEvolutionIdByElement(elementConfig);
00049
00050             fieldElementData.ConfigParameters = new SavedConfigParameters
00051             {
00052                 ElementId = elementConfig.Id,
00053                 EvolutionId = evolutionId
00054             };
00055
00056             fieldElementData.InfoParameters = new SavedInfoParameters
00057             {
00058                 LogicPosition = point,
00059                 IsBlocked = isBlocked
00060             };
00061
00062             switch (config.Type)
00063             {
00064                 case ElementType.Generator:
00065                     var generatorParameters = new SavedGeneratorParameters();
00066
00067                     if (config.GeneratorSettings.Charged)
00068                     {
00069                         generatorParameters.AvailableToDrop = config.GeneratorSettings.MaxDrop;
00070                     }
00071                     else
00072                     {
00073                         generatorParameters.StartChargingTime =
00074                             DateTimeOffset.UtcNow.ToUnixTimeSeconds();
00075                     }
00076
00077                     fieldElementData.GeneratorParameters = generatorParameters;
00078
00079                     if (config.GeneratorSettings.NeedOpen)
00080                     {
00081                         var generatorOpenParameters = new SavedGeneratorOpenParameters
00082                         {
00083                             StartOpeningTime = 0
00084                         };
00085
00086                         fieldElementData.GeneratorOpenParameters = generatorOpenParameters;
00087                     }
00088
00089                     break;
00090
00091             }
00092         }
00093 }
```

7.220 FieldElementVisualFactory.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using System.ComponentModel;
00005 using MergeIt.Core.Configs.Elements;
00006 using MergeIt.Core.Configs.Types;
00007 using MergeIt.Core.FieldElements;
00008 using MergeIt.Core.Services;
00009 using MergeIt.Game.Factories.Icons;
00010 using MergeIt.Game.Field.Elements;
00011 using MergeIt.Game.Field.Elements.Generator;
00012 using MergeIt.SimpleDI;
00013 using UnityEngine;
00014 using UnityEngine.Pool;
00015 using Object = UnityEngine.Object;
00016
00017 namespace MergeIt.Game.Factories.FieldElement
00018 {
00019     public class FieldElementVisualFactory : IFieldElementVisualFactory
00020     {
00021         [Introduce]
00022         private IConfigsService _configsService;
00023
00024         [Introduce]
00025         private IIIconFactory _iconFactory;
00026
00027         private readonly Dictionary<ElementConfig, IObjectPool<IFieldElementPresenter>> _pools = new();
00028
00029         private IFIELDElement _actualElement;
00030
00031         public IFIELDElementPresenter CreateFIELDElement(IFIELDElement element)
00032         {
00033             ElementConfig config = element.ConfigParameters.ElementConfig;
00034
00035             _actualElement = element;
00036
00037             if (!_pools.TryGetValue(config, out IObjectPool<IFieldElementPresenter> pool))
00038             {
00039                 pool = new ObjectPool<IFieldElementPresenter>(
00040                     () =>
00041                     {
00042                         ElementType type = _actualElement.InfoParameters.Type;
00043                         IFIELDElementView objectPrototype = _configsService.GetElementPrefab(type);
00044                         GameObject fieldObject = Object.Instantiate(objectPrototype.GameObject);
00045
00046                         _iconFactory.CreateIcon(config, fieldObject.transform);
00047
00048                         var view = fieldObject.GetComponent<IFIELDElementView>();
00049
00050                         IFIELDElementPresenter elementPresenter = null;
00051                         switch (type)
00052                         {
00053                             case ElementType.Regular:
00054                                 elementPresenter = new FieldElementPresenter();
00055                                 break;
00056
00057                             case ElementType.Generator:
00058                                 elementPresenter = new FieldElementGeneratorPresenter();
00059                                 break;
00060
00061                             default:
00062                                 throw new InvalidEnumArgumentException($"Unknown item type: {type}");
00063                         }
00064
00065                         if (elementPresenter != null)
00066                         {
00067                             elementPresenter.Initialize(view);
00068                             elementPresenter.Update(_actualElement);
00069
00070                             if (view != null)
00071                             {
00072                                 view.Initialize();
00073                             }
00074
00075
00076                             return elementPresenter;
00077                         },
00078                         pooledElement =>
00079                         {
00080                             pooledElement.Update(_actualElement);
00081                             pooledElement.Activate(true);
00082                         },
00083                         pooledElement =>
00084                         {
00085                             pooledElement.Release();
00086                         };
00087
00088             }
00089         }
00090     }
00091 }
```

```

00086             pooledElement.Activate(false);
00087         },
00088         pooledElement =>
00089     {
00090         pooledElement.Remove();
00091     },
00092     defaultCapacity: 2,
00093     maxSize: 20);
00094
00095     _pools[config] = pool;
00096 }
00097
00098     IFieldElementPresenter presenter = SetupElement(pool);
00099     return presenter;
00100 }
00101
00102     private IFieldElementPresenter SetupElement(IObjectPool<IFieldElementPresenter> pool)
00103     {
00104         IFieldElementPresenter presenter = pool.Get();
00105
00106         return presenter;
00107     }
00108 }
00109 }
```

7.221 IFieldElementFactory.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Data;
00004 using MergeIt.Core.Configs.Elements;
00005 using MergeIt.Core.FieldElements;
00006
00007 namespace MergeIt.Game.Factories.FieldElement
00008 {
00009     public interface IFieldElementFactory
0010     {
0011         IFieldElement CreateFieldElement(FieldElementData data);
0012         IFieldElement CreateFieldElement(LevelElementData data);
0013         IFieldElement CreateFieldElement(ElementConfig elementConfig, GridPoint point, bool isBlocked
0014 = false);
0015     }
0016 }
```

7.222 IFieldElementVisualFactory.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004
00005 namespace MergeIt.Game.Factories.FieldElement
00006 {
00007     public interface IFieldElementVisualFactory
00008     {
00009         IFieldElementPresenter CreateFieldElement(IFieldElement element);
0010     }
0011 }
```

7.223 IconFactory.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Configs.Elements;
00005 using UnityEngine;
00006 using UnityEngine.Pool;
00007
00008 namespace MergeIt.Game.Factories.Icons
00009 {
0010     public class IconFactory : IIconFactory
0011     {
0012         private readonly Dictionary<ElementConfig, IObjectPool<FieldElementIconComponent>> _icons =
0013         new();
0014         public FieldElementIconComponent CreateIcon(ElementConfig config, Transform parent = null)
```

```

00015     {
00016         FieldElementIconComponent iconComponent;
00017         if (!_icons.TryGetValue(config, out IObjectPool<FieldElementIconComponent> iconsPool))
00018     {
00019         iconsPool = new ObjectPool<FieldElementIconComponent>(
00020             () =>
00021             {
00022                 FieldElementIconComponent iconPrototype = config.GetIconComponent();
00023                 GameObject iconContainer = Object.Instantiate(iconPrototype.gameObject);
00024                 iconContainer.TryGetComponent(out FieldElementIconComponent icon);
00025
00026                 return icon;
00027             },
00028             OnGetIcon,
00029             OnReleaseIcon,
00030             OnDestroyIcon,
00031             defaultCapacity: 2,
00032             maxSize: 20);
00033
00034         _icons[config] = iconsPool;
00035     }
00036
00037         iconComponent = GetIconComponent(iconsPool, parent);
00038
00039         return iconComponent;
00040     }
00041
00042     private FieldElementIconComponent GetIconComponent(IObjectPool<FieldElementIconComponent>
00043         iconsPool, Transform parent)
00044     {
00045         var iconComponent = iconsPool.Get();
00046
00047         iconComponent.transform.SetParent(parent);
00048         iconComponent.transform.localScale = Vector3.one;
00049         iconComponent.RectTransform.SetAsFirstSibling();
00050
00051         return iconComponent;
00052     }
00053
00054     private void OnGetIcon(FieldElementIconComponent obj)
00055     {
00056         obj.gameObject.SetActive(true);
00057     }
00058
00059     private void OnReleaseIcon(FieldElementIconComponent obj)
00060     {
00061         obj.gameObject.SetActive(false);
00062     }
00063
00064     private void OnDestroyIcon(FieldElementIconComponent obj)
00065     {
00066         Object.Destroy(obj.gameObject);
00067     }
00068 }
00069 }
```

7.224 IIIconFactory.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Factories.Icons
00007 {
00008     public interface IIIconFactory
00009     {
00010         FieldElementIconComponent CreateIcon(ElementConfig config, Transform parent = null);
00011     }
00012 }
```

7.225 IIInventoryFactory.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Game.UI.InventoryPanel;
00005 using MergeIt.Game.Windows.Inventory;
```

```

00006 using UnityEngine;
00007
00008 namespace MergeIt.Game.Factories.Inventory
00009 {
0010     public interface IInventoryFactory
0011     {
0012         InventoryPanelItemComponent CreateInventoryPanelItem(IFieldElement fieldElement);
0013         InventoryWindowItemComponent CreateInventoryWindowItem(IFieldElement fieldElement);
0014         InventoryWindowPaidCellComponent CreateWindowPaidCell();
0015         GameObject CreateWindowEmptyCell();
0016     }
0017 }

```

7.226 InventoryData.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Core.Inventory;
00006 using MergeIt.Core.Saves;
00007 using Newtonsoft.Json;
00008
00009 namespace MergeIt.Game.Factories.Inventory
0010 {
0011     [Serializable, Savable("inventory", "dat")]
0012     public class InventoryData : IInventoryData
0013     {
0014         [JsonProperty("s")]
0015         public int InventorySize { get; set; }
0016         [JsonProperty("e")]
0017         public FieldElementData[] InventoryElements { get; set; }
0018     }
0019 }

```

7.227 InventoryFactory.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Game.UI.InventoryPanel;
00006 using MergeIt.Game.Windows.Inventory;
00007 using UnityEngine;
00008
00009 namespace MergeIt.Game.Factories.Inventory
0010 {
0011     public class InventoryFactory : IInventoryFactory
0012     {
0013         private const string PanelItemPath = "Prefabs/InventoryPanel/InventoryPanelItem";
0014         private const string WindowItemPath = "Prefabs/Windows/Inventory/InventoryWindowItem";
0015         private const string WindowEmptyItemPath =
0016             "Prefabs/Windows/Inventory/InventoryWindowEmptyCell";
0017         private const string WindowPaidCellPath = "Prefabs/Windows/Inventory/InventoryWindowPaidCell";
0018
0019         public InventoryPanelItemComponent CreateInventoryPanelItem(IFieldElement fieldElement)
0020         {
0021             GameObject itemPanelObject = Resources.Load<GameObject>(PanelItemPath);
0022
0023             if (itemPanelObject)
0024             {
0025                 GameObject panelItemObject = Object.Instantiate(itemPanelObject);
0026                 FieldElementIconComponent iconPrototype =
0027                     fieldElement.ConfigParameters.ElementConfig.GetIconComponent();
0028
0029                 var icon = Object.Instantiate(iconPrototype, panelItemObject.transform);
0030                 var iconRectTransform = icon.GetComponent<RectTransform>();
0031                 iconRectTransform.SetParent(panelItemObject.transform);
0032
0033                 var resultComponent = panelItemObject.GetComponent<InventoryPanelItemComponent>();
0034                 return resultComponent;
0035             }
0036
0037             return null;
0038         }
0039
0040         public InventoryWindowItemComponent CreateInventoryWindowItem(IFieldElement fieldElement)
0041         {
0042             GameObject itemPanelObject = Resources.Load<GameObject>(WindowItemPath);
0043
0044             if (itemPanelObject)
0045             {
0046                 GameObject panelItemObject = Object.Instantiate(itemPanelObject);
0047                 FieldElementIconComponent iconPrototype =
0048                     fieldElement.ConfigParameters.ElementConfig.GetIconComponent();
0049
0050                 var icon = Object.Instantiate(iconPrototype, panelItemObject.transform);
0051                 var iconRectTransform = icon.GetComponent<RectTransform>();
0052                 iconRectTransform.SetParent(panelItemObject.transform);
0053
0054                 var resultComponent = panelItemObject.GetComponent<InventoryWindowItemComponent>();
0055                 return resultComponent;
0056             }
0057
0058             return null;
0059         }
0060
0061         public GameObject CreateWindowEmptyCell()
0062         {
0063             return null;
0064         }
0065
0066         public GameObject CreateWindowPaidCell()
0067         {
0068             return null;
0069         }
0070
0071         public void Save()
0072         {
0073             var inventoryData = new InventoryData();
0074             inventoryData.InventorySize = InventorySize;
0075             inventoryData.InventoryElements = InventoryElements;
0076
0077             var saveData = new SaveData();
0078             saveData.InventoryData = inventoryData;
0079
0080             var json = JsonConvert.SerializeObject(saveData);
0081             var bytes = Encoding.UTF8.GetBytes(json);
0082
0083             var savePath = Path.Combine(Application.persistentDataPath, "Inventory.dat");
0084             File.WriteAllBytes(savePath, bytes);
0085         }
0086
0087         public void Load()
0088         {
0089             var savePath = Path.Combine(Application.persistentDataPath, "Inventory.dat");
0090
0091             if (!File.Exists(savePath))
0092             {
0093                 return;
0094             }
0095
0096             var bytes = File.ReadAllBytes(savePath);
0097             var json = Encoding.UTF8.GetString(bytes);
0098
0099             var saveData = JsonConvert.DeserializeObject<SaveData>(json);
0100             var inventoryData = saveData.InventoryData;
0101
0102             if (inventoryData != null)
0103             {
0104                 InventorySize = inventoryData.InventorySize;
0105                 InventoryElements = inventoryData.InventoryElements;
0106             }
0107         }
0108
0109         public void Clear()
0110         {
0111             InventorySize = 0;
0112             InventoryElements = null;
0113         }
0114
0115         public void AddElement(IFieldElement element)
0116         {
0117             if (InventoryElements == null)
0118             {
0119                 InventoryElements = new FieldElementData[1];
0120             }
0121
0122             var index = InventoryElements.Length;
0123             var elementData = new FieldElementData();
0124             elementData.Element = element;
0125             elementData.Icon = element.Icon;
0126
0127             InventoryElements = Array.Resize<FieldElementData>(InventoryElements, index + 1);
0128             InventoryElements[index] = elementData;
0129         }
0130
0131         public void RemoveElement(IFieldElement element)
0132         {
0133             if (InventoryElements == null)
0134             {
0135                 return;
0136             }
0137
0138             var index = InventoryElements.Length;
0139             var elementData = new FieldElementData();
0140             elementData.Element = element;
0141             elementData.Icon = element.Icon;
0142
0143             for (int i = 0; i < index; i++)
0144             {
0145                 if (InventoryElements[i].Element == element)
0146                 {
0147                     InventoryElements = Array.Resize<FieldElementData>(InventoryElements, index - 1);
0148                     InventoryElements[i] = elementData;
0149                     break;
0150                 }
0151             }
0152         }
0153
0154         public void SetElement(IFieldElement element, int index)
0155         {
0156             if (InventoryElements == null)
0157             {
0158                 return;
0159             }
0160
0161             var elementData = new FieldElementData();
0162             elementData.Element = element;
0163             elementData.Icon = element.Icon;
0164
0165             InventoryElements[index] = elementData;
0166         }
0167
0168         public void SetIcon(IFieldElement element, int index)
0169         {
0170             if (InventoryElements == null)
0171             {
0172                 return;
0173             }
0174
0175             var elementData = new FieldElementData();
0176             elementData.Element = element;
0177             elementData.Icon = element.Icon;
0178
0179             InventoryElements[index] = elementData;
0180         }
0181
0182         public void SetElement(IFieldElement element)
0183         {
0184             if (InventoryElements == null)
0185             {
0186                 return;
0187             }
0188
0189             var elementData = new FieldElementData();
0190             elementData.Element = element;
0191             elementData.Icon = element.Icon;
0192
0193             InventoryElements[0] = elementData;
0194         }
0195
0196         public void SetIcon(IFieldElement element)
0197         {
0198             if (InventoryElements == null)
0199             {
0200                 return;
0201             }
0202
0203             var elementData = new FieldElementData();
0204             elementData.Element = element;
0205             elementData.Icon = element.Icon;
0206
0207             InventoryElements[0] = elementData;
0208         }
0209
0210         public void SetElement(IFieldElement element, FieldElementIconComponent icon)
0211         {
0212             if (InventoryElements == null)
0213             {
0214                 return;
0215             }
0216
0217             var elementData = new FieldElementData();
0218             elementData.Element = element;
0219             elementData.Icon = icon;
0220
0221             InventoryElements[0] = elementData;
0222         }
0223
0224         public void SetIcon(IFieldElement element, FieldElementIconComponent icon)
0225         {
0226             if (InventoryElements == null)
0227             {
0228                 return;
0229             }
0230
0231             var elementData = new FieldElementData();
0232             elementData.Element = element;
0233             elementData.Icon = icon;
0234
0235             InventoryElements[0] = elementData;
0236         }
0237
0238         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale)
0239         {
0240             if (InventoryElements == null)
0241             {
0242                 return;
0243             }
0244
0245             var elementData = new FieldElementData();
0246             elementData.Element = element;
0247             elementData.Icon = icon;
0248             elementData.Scale = scale;
0249
0250             InventoryElements[0] = elementData;
0251         }
0252
0253         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale)
0254         {
0255             if (InventoryElements == null)
0256             {
0257                 return;
0258             }
0259
0260             var elementData = new FieldElementData();
0261             elementData.Element = element;
0262             elementData.Icon = icon;
0263             elementData.Scale = scale;
0264
0265             InventoryElements[0] = elementData;
0266         }
0267
0268         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position)
0269         {
0270             if (InventoryElements == null)
0271             {
0272                 return;
0273             }
0274
0275             var elementData = new FieldElementData();
0276             elementData.Element = element;
0277             elementData.Icon = icon;
0278             elementData.Scale = scale;
0279             elementData.Position = position;
0280
0281             InventoryElements[0] = elementData;
0282         }
0283
0284         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position)
0285         {
0286             if (InventoryElements == null)
0287             {
0288                 return;
0289             }
0290
0291             var elementData = new FieldElementData();
0292             elementData.Element = element;
0293             elementData.Icon = icon;
0294             elementData.Scale = scale;
0295             elementData.Position = position;
0296
0297             InventoryElements[0] = elementData;
0298         }
0299
0300         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation)
0301         {
0302             if (InventoryElements == null)
0303             {
0304                 return;
0305             }
0306
0307             var elementData = new FieldElementData();
0308             elementData.Element = element;
0309             elementData.Icon = icon;
0310             elementData.Scale = scale;
0311             elementData.Position = position;
0312             elementData.Rotation = rotation;
0313
0314             InventoryElements[0] = elementData;
0315         }
0316
0317         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation)
0318         {
0319             if (InventoryElements == null)
0320             {
0321                 return;
0322             }
0323
0324             var elementData = new FieldElementData();
0325             elementData.Element = element;
0326             elementData.Icon = icon;
0327             elementData.Scale = scale;
0328             elementData.Position = position;
0329             elementData.Rotation = rotation;
0330
0331             InventoryElements[0] = elementData;
0332         }
0333
0334         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor)
0335         {
0336             if (InventoryElements == null)
0337             {
0338                 return;
0339             }
0340
0341             var elementData = new FieldElementData();
0342             elementData.Element = element;
0343             elementData.Icon = icon;
0344             elementData.Scale = scale;
0345             elementData.Position = position;
0346             elementData.Rotation = rotation;
0347             elementData.Anchor = anchor;
0348
0349             InventoryElements[0] = elementData;
0350         }
0351
0352         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor)
0353         {
0354             if (InventoryElements == null)
0355             {
0356                 return;
0357             }
0358
0359             var elementData = new FieldElementData();
0360             elementData.Element = element;
0361             elementData.Icon = icon;
0362             elementData.Scale = scale;
0363             elementData.Position = position;
0364             elementData.Rotation = rotation;
0365             elementData.Anchor = anchor;
0366
0367             InventoryElements[0] = elementData;
0368         }
0369
0370         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset)
0371         {
0372             if (InventoryElements == null)
0373             {
0374                 return;
0375             }
0376
0377             var elementData = new FieldElementData();
0378             elementData.Element = element;
0379             elementData.Icon = icon;
0380             elementData.Scale = scale;
0381             elementData.Position = position;
0382             elementData.Rotation = rotation;
0383             elementData.Anchor = anchor;
0384             elementData.Offset = offset;
0385
0386             InventoryElements[0] = elementData;
0387         }
0388
0389         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset)
0390         {
0391             if (InventoryElements == null)
0392             {
0393                 return;
0394             }
0395
0396             var elementData = new FieldElementData();
0397             elementData.Element = element;
0398             elementData.Icon = icon;
0399             elementData.Scale = scale;
0400             elementData.Position = position;
0401             elementData.Rotation = rotation;
0402             elementData.Anchor = anchor;
0403             elementData.Offset = offset;
0404
0405             InventoryElements[0] = elementData;
0406         }
0407
0408         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot)
0409         {
0410             if (InventoryElements == null)
0411             {
0412                 return;
0413             }
0414
0415             var elementData = new FieldElementData();
0416             elementData.Element = element;
0417             elementData.Icon = icon;
0418             elementData.Scale = scale;
0419             elementData.Position = position;
0420             elementData.Rotation = rotation;
0421             elementData.Anchor = anchor;
0422             elementData.Offset = offset;
0423             elementData.Pivot = pivot;
0424
0425             InventoryElements[0] = elementData;
0426         }
0427
0428         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot)
0429         {
0430             if (InventoryElements == null)
0431             {
0432                 return;
0433             }
0434
0435             var elementData = new FieldElementData();
0436             elementData.Element = element;
0437             elementData.Icon = icon;
0438             elementData.Scale = scale;
0439             elementData.Position = position;
0440             elementData.Rotation = rotation;
0441             elementData.Anchor = anchor;
0442             elementData.Offset = offset;
0443             elementData.Pivot = pivot;
0444
0445             InventoryElements[0] = elementData;
0446         }
0447
0448         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition)
0449         {
0450             if (InventoryElements == null)
0451             {
0452                 return;
0453             }
0454
0455             var elementData = new FieldElementData();
0456             elementData.Element = element;
0457             elementData.Icon = icon;
0458             elementData.Scale = scale;
0459             elementData.Position = position;
0460             elementData.Rotation = rotation;
0461             elementData.Anchor = anchor;
0462             elementData.Offset = offset;
0463             elementData.Pivot = pivot;
0464             elementData.WorldPosition = worldPosition;
0465
0466             InventoryElements[0] = elementData;
0467         }
0468
0469         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition)
0470         {
0471             if (InventoryElements == null)
0472             {
0473                 return;
0474             }
0475
0476             var elementData = new FieldElementData();
0477             elementData.Element = element;
0478             elementData.Icon = icon;
0479             elementData.Scale = scale;
0480             elementData.Position = position;
0481             elementData.Rotation = rotation;
0482             elementData.Anchor = anchor;
0483             elementData.Offset = offset;
0484             elementData.Pivot = pivot;
0485             elementData.WorldPosition = worldPosition;
0486
0487             InventoryElements[0] = elementData;
0488         }
0489
0490         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation)
0491         {
0492             if (InventoryElements == null)
0493             {
0494                 return;
0495             }
0496
0497             var elementData = new FieldElementData();
0498             elementData.Element = element;
0499             elementData.Icon = icon;
0500             elementData.Scale = scale;
0501             elementData.Position = position;
0502             elementData.Rotation = rotation;
0503             elementData.Anchor = anchor;
0504             elementData.Offset = offset;
0505             elementData.Pivot = pivot;
0506             elementData.WorldPosition = worldPosition;
0507             elementData.WorldRotation = worldRotation;
0508
0509             InventoryElements[0] = elementData;
0510         }
0511
0512         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation)
0513         {
0514             if (InventoryElements == null)
0515             {
0516                 return;
0517             }
0518
0519             var elementData = new FieldElementData();
0520             elementData.Element = element;
0521             elementData.Icon = icon;
0522             elementData.Scale = scale;
0523             elementData.Position = position;
0524             elementData.Rotation = rotation;
0525             elementData.Anchor = anchor;
0526             elementData.Offset = offset;
0527             elementData.Pivot = pivot;
0528             elementData.WorldPosition = worldPosition;
0529             elementData.WorldRotation = worldRotation;
0530
0531             InventoryElements[0] = elementData;
0532         }
0533
0534         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale)
0535         {
0536             if (InventoryElements == null)
0537             {
0538                 return;
0539             }
0540
0541             var elementData = new FieldElementData();
0542             elementData.Element = element;
0543             elementData.Icon = icon;
0544             elementData.Scale = scale;
0545             elementData.Position = position;
0546             elementData.Rotation = rotation;
0547             elementData.Anchor = anchor;
0548             elementData.Offset = offset;
0549             elementData.Pivot = pivot;
0550             elementData.WorldPosition = worldPosition;
0551             elementData.WorldRotation = worldRotation;
0552             elementData.WorldScale = worldScale;
0553
0554             InventoryElements[0] = elementData;
0555         }
0556
0557         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale)
0558         {
0559             if (InventoryElements == null)
0560             {
0561                 return;
0562             }
0563
0564             var elementData = new FieldElementData();
0565             elementData.Element = element;
0566             elementData.Icon = icon;
0567             elementData.Scale = scale;
0568             elementData.Position = position;
0569             elementData.Rotation = rotation;
0570             elementData.Anchor = anchor;
0571             elementData.Offset = offset;
0572             elementData.Pivot = pivot;
0573             elementData.WorldPosition = worldPosition;
0574             elementData.WorldRotation = worldRotation;
0575             elementData.WorldScale = worldScale;
0576
0577             InventoryElements[0] = elementData;
0578         }
0579
0580         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor)
0581         {
0582             if (InventoryElements == null)
0583             {
0584                 return;
0585             }
0586
0587             var elementData = new FieldElementData();
0588             elementData.Element = element;
0589             elementData.Icon = icon;
0590             elementData.Scale = scale;
0591             elementData.Position = position;
0592             elementData.Rotation = rotation;
0593             elementData.Anchor = anchor;
0594             elementData.Offset = offset;
0595             elementData.Pivot = pivot;
0596             elementData.WorldPosition = worldPosition;
0597             elementData.WorldRotation = worldRotation;
0598             elementData.WorldScale = worldScale;
0599             elementData.WorldAnchor = worldAnchor;
0600
0601             InventoryElements[0] = elementData;
0602         }
0603
0604         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor)
0605         {
0606             if (InventoryElements == null)
0607             {
0608                 return;
0609             }
0610
0611             var elementData = new FieldElementData();
0612             elementData.Element = element;
0613             elementData.Icon = icon;
0614             elementData.Scale = scale;
0615             elementData.Position = position;
0616             elementData.Rotation = rotation;
0617             elementData.Anchor = anchor;
0618             elementData.Offset = offset;
0619             elementData.Pivot = pivot;
0620             elementData.WorldPosition = worldPosition;
0621             elementData.WorldRotation = worldRotation;
0622             elementData.WorldScale = worldScale;
0623             elementData.WorldAnchor = worldAnchor;
0624
0625             InventoryElements[0] = elementData;
0626         }
0627
0628         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor, Vector3 worldPivot)
0629         {
0630             if (InventoryElements == null)
0631             {
0632                 return;
0633             }
0634
0635             var elementData = new FieldElementData();
0636             elementData.Element = element;
0637             elementData.Icon = icon;
0638             elementData.Scale = scale;
0639             elementData.Position = position;
0640             elementData.Rotation = rotation;
0641             elementData.Anchor = anchor;
0642             elementData.Offset = offset;
0643             elementData.Pivot = pivot;
0644             elementData.WorldPosition = worldPosition;
0645             elementData.WorldRotation = worldRotation;
0646             elementData.WorldScale = worldScale;
0647             elementData.WorldAnchor = worldAnchor;
0648             elementData.WorldPivot = worldPivot;
0649
0650             InventoryElements[0] = elementData;
0651         }
0652
0653         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor, Vector3 worldPivot)
0654         {
0655             if (InventoryElements == null)
0656             {
0657                 return;
0658             }
0659
0660             var elementData = new FieldElementData();
0661             elementData.Element = element;
0662             elementData.Icon = icon;
0663             elementData.Scale = scale;
0664             elementData.Position = position;
0665             elementData.Rotation = rotation;
0666             elementData.Anchor = anchor;
0667             elementData.Offset = offset;
0668             elementData.Pivot = pivot;
0669             elementData.WorldPosition = worldPosition;
0670             elementData.WorldRotation = worldRotation;
0671             elementData.WorldScale = worldScale;
0672             elementData.WorldAnchor = worldAnchor;
0673             elementData.WorldPivot = worldPivot;
0674
0675             InventoryElements[0] = elementData;
0676         }
0677
0678         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor, Vector3 worldPivot, Vector3 worldEuler)
0679         {
0680             if (InventoryElements == null)
0681             {
0682                 return;
0683             }
0684
0685             var elementData = new FieldElementData();
0686             elementData.Element = element;
0687             elementData.Icon = icon;
0688             elementData.Scale = scale;
0689             elementData.Position = position;
0690             elementData.Rotation = rotation;
0691             elementData.Anchor = anchor;
0692             elementData.Offset = offset;
0693             elementData.Pivot = pivot;
0694             elementData.WorldPosition = worldPosition;
0695             elementData.WorldRotation = worldRotation;
0696             elementData.WorldScale = worldScale;
0697             elementData.WorldAnchor = worldAnchor;
0698             elementData.WorldPivot = worldPivot;
0699             elementData.WorldEuler = worldEuler;
0700
0701             InventoryElements[0] = elementData;
0702         }
0703
0704         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor, Vector3 worldPivot, Vector3 worldEuler)
0705         {
0706             if (InventoryElements == null)
0707             {
0708                 return;
0709             }
0710
0711             var elementData = new FieldElementData();
0712             elementData.Element = element;
0713             elementData.Icon = icon;
0714             elementData.Scale = scale;
0715             elementData.Position = position;
0716             elementData.Rotation = rotation;
0717             elementData.Anchor = anchor;
0718             elementData.Offset = offset;
0719             elementData.Pivot = pivot;
0720             elementData.WorldPosition = worldPosition;
0721             elementData.WorldRotation = worldRotation;
0722             elementData.WorldScale = worldScale;
0723             elementData.WorldAnchor = worldAnchor;
0724             elementData.WorldPivot = worldPivot;
0725             elementData.WorldEuler = worldEuler;
0726
0727             InventoryElements[0] = elementData;
0728         }
0729
0730         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor, Vector3 worldPivot, Vector3 worldEuler, Vector3 worldScale)
0731         {
0732             if (InventoryElements == null)
0733             {
0734                 return;
0735             }
0736
0737             var elementData = new FieldElementData();
0738             elementData.Element = element;
0739             elementData.Icon = icon;
0740             elementData.Scale = scale;
0741             elementData.Position = position;
0742             elementData.Rotation = rotation;
0743             elementData.Anchor = anchor;
0744             elementData.Offset = offset;
0745             elementData.Pivot = pivot;
0746             elementData.WorldPosition = worldPosition;
0747             elementData.WorldRotation = worldRotation;
0748             elementData.WorldScale = worldScale;
0749             elementData.WorldAnchor = worldAnchor;
0750             elementData.WorldPivot = worldPivot;
0751             elementData.WorldEuler = worldEuler;
0752
0753             InventoryElements[0] = elementData;
0754         }
0755
0756         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor, Vector3 worldPivot, Vector3 worldEuler, Vector3 worldScale)
0757         {
0758             if (InventoryElements == null)
0759             {
0760                 return;
0761             }
0762
0763             var elementData = new FieldElementData();
0764             elementData.Element = element;
0765             elementData.Icon = icon;
0766             elementData.Scale = scale;
0767             elementData.Position = position;
0768             elementData.Rotation = rotation;
0769             elementData.Anchor = anchor;
0770             elementData.Offset = offset;
0771             elementData.Pivot = pivot;
0772             elementData.WorldPosition = worldPosition;
0773             elementData.WorldRotation = worldRotation;
0774             elementData.WorldScale = worldScale;
0775             elementData.WorldAnchor = worldAnchor;
0776             elementData.WorldPivot = worldPivot;
0777             elementData.WorldEuler = worldEuler;
0778
0779             InventoryElements[0] = elementData;
0780         }
0781
0782         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor, Vector3 worldPivot, Vector3 worldEuler, Vector3 worldScale, Vector3 worldScale)
0783         {
0784             if (InventoryElements == null)
0785             {
0786                 return;
0787             }
0788
0789             var elementData = new FieldElementData();
0790             elementData.Element = element;
0791             elementData.Icon = icon;
0792             elementData.Scale = scale;
0793             elementData.Position = position;
0794             elementData.Rotation = rotation;
0795             elementData.Anchor = anchor;
0796             elementData.Offset = offset;
0797             elementData.Pivot = pivot;
0798             elementData.WorldPosition = worldPosition;
0799             elementData.WorldRotation = worldRotation;
0800             elementData.WorldScale = worldScale;
0801             elementData.WorldAnchor = worldAnchor;
0802             elementData.WorldPivot = worldPivot;
0803             elementData.WorldEuler = worldEuler;
0804             elementData.WorldScale = worldScale;
0805
0806             InventoryElements[0] = elementData;
0807         }
0808
0809         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor, Vector3 worldPivot, Vector3 worldEuler, Vector3 worldScale, Vector3 worldScale)
0810         {
0811             if (InventoryElements == null)
0812             {
0813                 return;
0814             }
0815
0816             var elementData = new FieldElementData();
0817             elementData.Element = element;
0818             elementData.Icon = icon;
0819             elementData.Scale = scale;
0820             elementData.Position = position;
0821             elementData.Rotation = rotation;
0822             elementData.Anchor = anchor;
0823             elementData.Offset = offset;
0824             elementData.Pivot = pivot;
0825             elementData.WorldPosition = worldPosition;
0826             elementData.WorldRotation = worldRotation;
0827             elementData.WorldScale = worldScale;
0828             elementData.WorldAnchor = worldAnchor;
0829             elementData.WorldPivot = worldPivot;
0830             elementData.WorldEuler = worldEuler;
0831             elementData.WorldScale = worldScale;
0832
0833             InventoryElements[0] = elementData;
0834         }
0835
0836         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor, Vector3 worldPivot, Vector3 worldEuler, Vector3 worldScale, Vector3 worldScale, Vector3 worldScale)
0837         {
0838             if (InventoryElements == null)
0839             {
0840                 return;
0841             }
0842
0843             var elementData = new FieldElementData();
0844             elementData.Element = element;
0845             elementData.Icon = icon;
0846             elementData.Scale = scale;
0847             elementData.Position = position;
0848             elementData.Rotation = rotation;
0849             elementData.Anchor = anchor;
0850             elementData.Offset = offset;
0851             elementData.Pivot = pivot;
0852             elementData.WorldPosition = worldPosition;
0853             elementData.WorldRotation = worldRotation;
0854             elementData.WorldScale = worldScale;
0855             elementData.WorldAnchor = worldAnchor;
0856             elementData.WorldPivot = worldPivot;
0857             elementData.WorldEuler = worldEuler;
0858             elementData.WorldScale = worldScale;
0859             elementData.WorldScale = worldScale;
0860
0861             InventoryElements[0] = elementData;
0862         }
0863
0864         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor, Vector3 worldPivot, Vector3 worldEuler, Vector3 worldScale, Vector3 worldScale, Vector3 worldScale)
0865         {
0866             if (InventoryElements == null)
0867             {
0868                 return;
0869             }
0870
0871             var elementData = new FieldElementData();
0872             elementData.Element = element;
0873             elementData.Icon = icon;
0874             elementData.Scale = scale;
0875             elementData.Position = position;
0876             elementData.Rotation = rotation;
0877             elementData.Anchor = anchor;
0878             elementData.Offset = offset;
0879             elementData.Pivot = pivot;
0880             elementData.WorldPosition = worldPosition;
0881             elementData.WorldRotation = worldRotation;
0882             elementData.WorldScale = worldScale;
0883             elementData.WorldAnchor = worldAnchor;
0884             elementData.WorldPivot = worldPivot;
0885             elementData.WorldEuler = worldEuler;
0886             elementData.WorldScale = worldScale;
0887             elementData.WorldScale = worldScale;
0888
0889             InventoryElements[0] = elementData;
0890         }
0891
0892         public void SetElement(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor, Vector3 worldPivot, Vector3 worldEuler, Vector3 worldScale, Vector3 worldScale, Vector3 worldScale, Vector3 worldScale)
0893         {
0894             if (InventoryElements == null)
0895             {
0896                 return;
0897             }
0898
0899             var elementData = new FieldElementData();
0900             elementData.Element = element;
0901             elementData.Icon = icon;
0902             elementData.Scale = scale;
0903             elementData.Position = position;
0904             elementData.Rotation = rotation;
0905             elementData.Anchor = anchor;
0906             elementData.Offset = offset;
0907             elementData.Pivot = pivot;
0908             elementData.WorldPosition = worldPosition;
0909             elementData.WorldRotation = worldRotation;
0910             elementData.WorldScale = worldScale;
0911             elementData.WorldAnchor = worldAnchor;
0912             elementData.WorldPivot = worldPivot;
0913             elementData.WorldEuler = worldEuler;
0914             elementData.WorldScale = worldScale;
0915             elementData.WorldScale = worldScale;
0916             elementData.WorldScale = worldScale;
0917
0918             InventoryElements[0] = elementData;
0919         }
0920
0921         public void SetIcon(IFieldElement element, FieldElementIconComponent icon, float scale, Vector3 position, Vector3 rotation, Vector3 anchor, Vector3 offset, Vector3 pivot, Vector3 worldPosition, Vector3 worldRotation, Vector3 worldScale, Vector3 worldAnchor, Vector3 worldPivot, Vector3 worldEuler, Vector3 worldScale, Vector3 worldScale, Vector3 worldScale, Vector3 worldScale)
0922         {
0923             if (InventoryElements == null)
0924             {
0925                 return;
0926             }
0927
0928             var elementData = new FieldElementData();
0929             elementData.Element = element;
0930             element
```

```

00041
00042     if (itemPanelObject)
00043     {
00044         var panelItemObject = Object.Instantiate(itemPanelObject);
00045         if (panelItemObject.TryGetComponent(out InventoryWindowItemComponent resultComponent))
00046         {
00047             FieldElementIconComponent iconPrototype =
00048                 fieldElement.ConfigParameters.ElementConfig.GetIconComponent();
00049
00050             var icon = Object.Instantiate(iconPrototype, panelItemObject.transform);
00051             var iconRectTransform = icon.GetComponent<RectTransform>();
00052
00053             resultComponent.SetIcon(iconRectTransform);
00054         }
00055
00056         return resultComponent;
00057     }
00058
00059     return null;
00060 }
00061
00062     public InventoryWindowPaidCellComponent CreateWindowPaidCell()
00063     {
00064         GameObject itemPanelObject = Resources.Load<GameObject>(WindowPaidCellPath);
00065
00066         if (itemPanelObject)
00067         {
00068             var panelItemObject = Object.Instantiate(itemPanelObject);
00069
00070             var resultComponent =
00071                 panelItemObject.GetComponent<InventoryWindowPaidCellComponent>();
00072             return resultComponent;
00073         }
00074
00075         return null;
00076     }
00077
00078     public GameObject CreateWindowEmptyCell()
00079     {
00080         GameObject itemPanelObject = Resources.Load<GameObject>(WindowEmptyItemPath);
00081
00082         if (itemPanelObject)
00083         {
00084             var panelItemObject = Object.Instantiate(itemPanelObject);
00085             return panelItemObject.gameObject;
00086         }
00087
00088     }
00089 }
00090 }
```

7.228 FieldActionProcessorBase.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005 using MergeIt.Core.Services;
00006 using MergeIt.SimpleDI;
00007 using UnityEngine;
00008
00009 namespace MergeIt.Game.Field.Actions
00010 {
00011     public abstract class FieldActionProcessorBase : IFieldActionProcessor
00012     {
00013         protected readonly IMessageBus MessageBus = DiContainer.Get<IMessageBus>();
00014         protected readonly IGameSaveService SaveService = DiContainer.Get<IGameSaveService>();
00015         protected readonly FieldLogicModel FieldLogicModel = DiContainer.Get<FieldLogicModel>();
00016
00017         public virtual void ProcessClick(FieldCellComponent cellComponent)
00018         {
00019
00020     }
00021
00022         public virtual void ProcessEndDrag(GridPoint fromPoint, GameObject toGameObject)
00023         {
00024
00025     }
00026 }
00027 }
```

7.229 FieldConsumableProcessor.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.Configs.Types;
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Core.Helpers;
00007 using MergeIt.Core.Saves;
00008 using MergeIt.Core.Services;
00009 using MergeIt.Game.Effects;
00010 using MergeIt.Game.Effects.Controllers;
00011 using MergeIt.Game.Effects.Parameters;
00012 using MergeIt.Game.Enums;
00013 using MergeIt.Game.Messages;
00014 using MergeIt.Game.Services;
00015 using MergeIt.SimpleDI;
00016 using UnityEngine;
00017
00018 namespace MergeIt.Game.Field.Actions
00019 {
00020     public class FieldConsumableProcessor : FieldActionProcessorBase
00021     {
00022         private readonly UserServiceModel _userServiceModel = DiContainer.Get<UserServiceModel>();
00023         private readonly IEffectorFactory _effectorFactory = DiContainer.Get<IEffectorFactory>();
00024         private readonly IGameSaveService _saveService = DiContainer.Get<IGameSaveService>();
00025
00026         public override void ProcessClick(FieldCellComponent cellComponent)
00027         {
00028             base.ProcessClick(cellComponent);
00029
00030             if (cellComponent.FieldElementPresenter?.GetModel().ClicksCount < 2)
00031             {
00032                 return;
00033             }
00034
00035             if (!FieldLogicModel.FieldElements.TryGetValue(cellComponent.Point, out IFIELDElement
00036                 fieldElement))
00037             {
00038                 return;
00039             }
00040
00041             if (fieldElement.InfoParameters.IsBlocked)
00042             {
00043                 return;
00044             }
00045
00046             ElementCommonSettings settings =
00047                 fieldElement.ConfigParameters.ElementConfig.CommonSettings;
00048
00049             if (!settings.IsConsumable)
00050             {
00051                 return;
00052             }
00053
00054             ConsumableSettings consumableSettings = settings.ConsumableSettings;
00055             Transform transform = cellComponent.FieldElementPresenter.Transform;
00056             MoveEffectParameters parameters = null;
00057
00058             MessageBus.Fire<ResetHintsMessage>();
00059
00060             switch (consumableSettings.Target)
00061             {
00062                 case ConsumableTargetType.SoftCurrency:
00063                     _userServiceModel.SoftCurrency.ApplyOperation(consumableSettings.Operation,
00064                         consumableSettings.Value, false);
00065                     _saveService.Save(GameSaveType.User);
00066                     parameters = new MoveEffectParameters(transform.position, Vector3.zero,
00067                         HudPanelType.SoftCurrency);
00068
00069                     _effectorFactory.CreateEffect<MoveToUiEffectWithPresenter>(cellComponent.FieldElementPresenter,
00070                         parameters, OnEffectFinished);
00071                     break;
00072
00073                 case ConsumableTargetType.HardCurrency:
00074                     _userServiceModel.HardCurrency.ApplyOperation(consumableSettings.Operation,
00075                         consumableSettings.Value, false);
00076                     _saveService.Save(GameSaveType.User);
00077                     parameters = new MoveEffectParameters(transform.position, Vector3.zero,
00078                         HudPanelType.HardCurrency);
00079
00080                     _effectorFactory.CreateEffect<MoveToUiEffectWithPresenter>(cellComponent.FieldElementPresenter,
00081                         parameters, OnEffectFinished);
00082                     break;
00083
00084                 case ConsumableTargetType.Time:
00085                     RemoveItem(cellComponent);
00086             }
00087         }
00088     }
00089 }

```

```

00076             MessageBus.Fire(new SkipTimeMessage {Seconds = consumableSettings.Value});
00077             break;
00078
00079         case ConsumableTargetType.Energy:
00080             _userServiceModel.Energy.ApplyOperation(consumableSettings.Operation,
00081             consumableSettings.Value, false);
00082             _saveService.Save(GameSaveType.User);
00083             parameters = new MoveEffectParameters(transform.position, Vector3.zero,
00084             HudPanelType.Energy);
00085
00086         _effectsFactory.CreateEffect<MoveToUiEffectWithPresenter>(cellComponent.FieldElementPresenter,
00087             parameters, OnEffectFinished);
00088             break;
00089
00090         case ConsumableTargetType.Experience:
00091             _userServiceModel.Experience.ApplyOperation(consumableSettings.Operation,
00092             consumableSettings.Value, false);
00093             _saveService.Save(GameSaveType.User);
00094             parameters = new MoveEffectParameters(transform.position, Vector3.zero,
00095             HudPanelType.UserProgress);
00096
00097         _effectsFactory.CreateEffect<MoveToUiEffectWithPresenter>(cellComponent.FieldElementPresenter,
00098             parameters, OnEffectFinished);
00099             break;
00100
00101     void OnEffectFinished()
00102     {
00103         RemoveItem(cellComponent);
00104     }
00105 }
00106
00107 private void RemoveItem(FieldCellComponent cellComponent)
00108 {
00109     var removeMessage = new RemoveElementMessage
00110     {
00111         RemoveAtPoint = cellComponent.Point
00112     };
00113
00114     MessageBus.Fire(removeMessage);
00115
00116     SendResetMessage();
00117 }
00118
00119 private void SendResetMessage()
00120 {
00121     var resetMessage = new ResetSelectionMessage();
00122
00123     MessageBus.Fire(resetMessage);
00124 }
00125 }
00126 }
```

7.230 FieldGenerationProcessor.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Core.Saves;
00006 using MergeIt.Core.Services;
00007 using MergeIt.Game.Helpers;
00008 using MergeIt.Game.Messages;
00009 using MergeIt.Game.Services;
00010 using MergeIt.SimpleDI;
00011 using UnityEngine;
00012
00013 namespace MergeIt.Game.Field.Actions
00014 {
00015     public class FieldGenerationProcessor : FieldActionProcessorBase
00016     {
00017         private readonly UserServiceModel _userServiceModel = DiContainer.Get<UserServiceModel>();
```

```

00018     private readonly IGameFieldService _gameFieldService = DiContainer.Get<IGameFieldService>();
00019     private readonly FieldLogicModel _fieldLogicModel = DiContainer.Get<FieldLogicModel>();
00020
00021     public override void ProcessClick(FieldCellComponent cellComponent)
00022     {
00023         base.ProcessClick(cellComponent);
00024
00025         if (cellComponent.FieldElementPresenter?.GetModel()?.ClicksCount < 2)
00026         {
00027             return;
00028         }
00029
00030         if (!FieldLogicModel.FieldElements.TryGetValue(cellComponent.Point, out IFIELDElement
00031             fieldElement))
00032         {
00033             return;
00034         }
00035
00036         IGeneratorParameters generatorParameters = fieldElement.GeneratorParameters;
00037         if (generatorParameters == null)
00038         {
00039             return;
00040         }
00041
00042         IGeneratorOpenParameters generatorOpenParameters = fieldElement.GeneratorOpenParameters;
00043         if (generatorOpenParameters != null)
00044         {
00045             Debug.Log("Generator should be open or opening.");
00046             return;
00047         }
00048
00049         if (_userServiceModel.Energy.Value == 0)
00050         {
00051             Debug.Log("Not enough energy.");
00052             return;
00053         }
00054
00055         GridPoint? freePoint = _gameFieldService.GetFreeCell();
00056
00057         if (!freePoint.HasValue)
00058         {
00059             SendFieldFull();
00060             return;
00061         }
00062
00063         ElementGeneratorSettings generatorSettings =
00064             fieldElement.ConfigParameters.ElementConfig.GeneratorSettings;
00065
00066         if (generatorParameters.AvailableToDrop > 0)
00067         {
00068             GeneratableElement candidate = fieldElement.GetCandidateToGenerate();
00069
00070             if (candidate == null)
00071             {
00072                 SendGeneratorBusy(fieldElement.InfoParameters.LogicPosition);
00073                 return;
00074             }
00075
00076             IFIELDElement generatedElement =
00077                 _gameFieldService.CreateNewElement(candidate.Element, freePoint.Value,
00078                 generatorSettings.GenerateBlocked);
00079
00080             var position = fieldElement.InfoParameters.LogicPosition;
00081
00082             var generateResult = fieldElement.TryGenerate();
00083             if (generateResult.HasValue)
00084             {
00085                 if (!generateResult.Value)
00086                 {
00087                     SendRemoveElement(position);
00088                 }
00089             }
00090             else
00091             {
00092                 SendCheckGenerator(position);
00093             }
00094
00095             _userServiceModel.Energy.Value--;
00096
00097             SendCreateElement(generatedElement, position, freePoint.Value);
00098             SaveService.Save(GameSaveType.Field);
00099
00100         }
00101         else

```

```

00102         {
00103             SendGeneratorBusy(fieldElement.InfoParameters.LogicPosition);
00104         }
00105     }
00106 
00107     private void SendCreateElement(IFieldElement newElement, GridPoint fromPoint, GridPoint
00108     toPoint)
00109     {
00110         var message = new CreateElementMessage
00111         {
00112             FromPosition = _fieldLogicModel.GetPosition(fromPoint),
00113             ToPoint = toPoint,
00114             NewElement = newElement
00115         };
00116 
00117         MessageBus.Fire(message);
00118     }
00119 
00120     private void SendRemoveElement(GridPoint elementPosition)
00121     {
00122         var message = new RemoveElementMessage
00123         {
00124             RemoveAtPoint = elementPosition
00125         };
00126 
00127         MessageBus.Fire(message);
00128     }
00129 
00130     private void SendCheckGenerator(GridPoint generatorPosition)
00131     {
00132         var message = new CheckGeneratorMessage
00133         {
00134             GeneratorPoint = generatorPosition
00135         };
00136 
00137         MessageBus.Fire(message);
00138     }
00139 
00140     private void SendFieldFull()
00141     {
00142         Debug.Log($"Generation impossible. Field is full.");
00143 
00144         var message = new FieldFullMessage();
00145 
00146         MessageBus.Fire(message);
00147     }
00148 
00149     private void SendGeneratorBusy(GridPoint point)
00150     {
00151         Debug.Log($"Generation impossible. Generator charging.");
00152 
00153         var message = new GeneratorBusyMessage
00154         {
00155             Point = point,
00156         };
00157 
00158         MessageBus.Fire(message);
00159     }
00160 }

```

7.231 FieldMergeProcessor.cs

```

00001 // Copyright (c) 2024, Awessets
00002 
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.Configs.Types;
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Core.Saves;
00007 using MergeIt.Core.Services;
00008 using MergeIt.Game.Factories.FieldElement;
00009 using MergeIt.Game.Helpers;
00010 using MergeIt.Game.Messages;
00011 using MergeIt.SimpleDI;
00012 using UnityEngine;
00013 
00014 namespace MergeIt.Game.Field.Actions
00015 {
00016     public class FieldMergeProcessor : FieldActionProcessorBase
00017     {
00018         private readonly IFIELDElementFactory _fieldElementFactory =
00019             DiContainer.Get<IFIELDElementFactory>();
00020         private readonly IEvolutionsService _evolutionsService =
00021             DiContainer.Get<IEvolutionsService>();

```

```

00020     private readonly FieldLogicModel _fieldLogicModel = DiContainer.Get<FieldLogicModel>();
00021
00022     public override void ProcessClick(FieldCellComponent cellComponent)
00023     {
00024         base.ProcessClick(cellComponent);
00025
00026         var presenter = cellComponent.FieldElementPresenter;
00027         if (presenter != null)
00028         {
00029             presenter.GetModel().ClicksCount++;
00030
00031             if (!presenter.GetModel().Selected)
00032             {
00033                 presenter.Select(true);
00034             }
00035         }
00036     }
00037
00038     public override void ProcessEndDrag(GridPoint fromPoint, GameObject toGameObject)
00039     {
00040         base.ProcessEndDrag(fromPoint, toGameObject);
00041
00042         GridPoint toPoint = GridPoint.Default;
00043
00044         if (toGameObject)
00045         {
00046             if (toGameObject.TryGetComponent(out FieldCellComponent toCell))
00047             {
00048                 toPoint = toCell.Point;
00049
00050                 if (fromPoint.Equals(toPoint))
00051                 {
00052                     SendResetPositionMessage(fromPoint);
00053
00054                     return;
00055                 }
00056
00057                 IFieldElementPresenter toPresenter = toCell.FieldElementPresenter;
00058
00059                 if (toPresenter != null)
00060                 {
00061                     var mergedResult = TryMerge(fromPoint, toPoint);
00062                     if (mergedResult != null)
00063                     {
00064                         SendRemoveElement(fromPoint);
00065                         SendRemoveElement(toPoint);
00066                         SendMergeMessage(fromPoint, mergedResult);
00067
00068                         SaveService.Save(GameSaveType.Field);
00069                     }
00070                     else if (!toPresenter.IsLocked)
00071                     {
00072                         SendSwapMessage(fromPoint, toPoint);
00073                         SaveService.Save(GameSaveType.Field);
00074                     }
00075                     else
00076                     {
00077                         SendResetPositionMessage(fromPoint);
00078                     }
00079                 }
00080             }
00081         }
00082         SendSwapMessage(fromPoint, toPoint);
00083         SaveService.Save(GameSaveType.Field);
00084     }
00085
00086     else
00087     {
00088         SendResetPositionMessage(fromPoint);
00089     }
00090 }
00091 else
00092 {
00093     SendResetPositionMessage(fromPoint);
00094 }
00095 }

00096 private IFIELDElement TryMerge(GridPoint fromPoint, GridPoint toPoint)
00097 {
00098     IFIELDElement firstElement = FieldLogicModel.FieldElements[fromPoint];
00099     IFIELDElement secondElement = FieldLogicModel.FieldElements[toPoint];
00100
00101     ElementConfig firstId = firstElement.ConfigParameters.ElementConfig;
00102     ElementConfig secondId = secondElement.ConfigParameters.ElementConfig;
00103
00104     if (firstId == secondId)
00105     {

```

```
00107     ElementConfig nextInEvolution = firstElement.GetNextInEvolution();
00108     if (nextInEvolution)
00109     {
00110         IFieldElement newElement =
00111             _fieldElementFactory.CreateFieldElement(nextInEvolution, toPoint);
00112         _evolutionsService.UpdateProgress(newElement.ConfigParameters.ElementConfig);
00113
00114         if (firstElement.InfoParameters.Type == ElementType.Generator &&
00115             newElement.InfoParameters.Type == ElementType.Generator)
00116         {
00117             SetupNewGenerator(firstElement, secondElement, newElement);
00118         }
00119
00120         return newElement;
00121     }
00122
00123     return null;
00124 }
00125
00126 private void SendResetPositionMessage(GridPoint fromPoint)
00127 {
00128     var resetMessage = new ResetPositionMessage
00129     {
00130         From = fromPoint
00131     };
00132
00133     MessageBus.Fire(resetMessage);
00134 }
00135
00136 private void SendMergeMessage(GridPoint fromPoint, IFieldElement newElement)
00137 {
00138     var resetMessage = new MergeElementsMessage
00139     {
00140         From = fromPoint,
00141         NewElement = newElement
00142     };
00143
00144     MessageBus.Fire(resetMessage);
00145 }
00146
00147 private void SendSwapMessage(GridPoint fromPoint, GridPoint toPoint)
00148 {
00149     var resetMessage = new SwapElementsMessage
00150     {
00151         From = fromPoint,
00152         To = toPoint
00153     };
00154
00155     MessageBus.Fire(resetMessage);
00156 }
00157
00158 private void SendRemoveElement(GridPoint elementPosition)
00159 {
00160     var message = new RemoveElementMessage
00161     {
00162         RemoveAtPoint = elementPosition
00163     };
00164
00165     MessageBus.Fire(message);
00166 }
00167
00168 private void SetupNewGenerator(IFieldElement generator1, IFieldElement generator2,
00169     IFieldElement newGenerator)
00170 {
00171     if (!newGenerator.ConfigParameters.ElementConfig.GeneratorSettings.Charged)
00172     {
00173         var parameters1 = generator1.GeneratorParameters;
00174         var parameters2 = generator2.GeneratorParameters;
00175
00176         int chargedCount = parameters1.ChargedCount + parameters2.ChargedCount;
00177         int availableCount = parameters1.AvailableToDrop + parameters2.AvailableToDrop;
00178         int allCount = chargedCount + availableCount;
00179
00180         newGenerator.GeneratorParameters.ChargedCount = allCount;
00181         newGenerator.UpdateGenerator();
00182     }
00183 }
00184 }
00185 }
00186 }
```

7.232 IFIELDACTIONPROCESSOR.CS

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Field.Actions
00007 {
00008     public interface IFIELDACTIONPROCESSOR
00009     {
00010         void ProcessClick(FieldCellComponent cellComponent);
00011         void ProcessEndDrag(GridPoint fromPoint, GameObject toGameObject);
00012     }
00013 }
```

7.233 FIELDELEMENTANIMATIONCONTROLLER.CS

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Animations;
00005 using MergeIt.Core.FieldElements;
00006 using UnityEngine;
00007
00008 namespace MergeIt.Game.Field.Elements.Animations
00009 {
00010     [RequireComponent(typeof(Animator))]
00011     public class FieldElementAnimationController : AnimationControllerBase
00012     {
00013         private IAnimationListener _listener;
00014         private Animator _animator;
00015
00016         private static readonly Dictionary<FieldElementState, int> StatesHash = new();
00017
00018         static FieldElementAnimationController()
00019         {
00020             StatesHash[FieldElementState.Idle] =
00021                 Animator.StringToHash(FieldElementState.Idle.ToString());
00022             StatesHash[FieldElementState.Hint] =
00023                 Animator.StringToHash(FieldElementState.Hint.ToString());
00024         }
00025
00026         public static int GetFieldElementState(FieldElementState state)
00027         {
00028             StatesHash.TryGetValue(state, out int hash);
00029             return hash;
00030         }
00031
00032         private void Awake()
00033         {
00034             _animator = GetComponent<Animator>();
00035         }
00036
00037         public override void Initialize(IAnimationListener listener)
00038         {
00039             _listener = listener;
00040         }
00041
00042         public override void SetState<T>(T state)
00043         {
00044             var concreteState = (FieldElementState)(object)state;
00045
00046             if (StatesHash.TryGetValue(concreteState, out int hash))
00047             {
00048                 switch (concreteState)
00049                 {
00050                     case FieldElementState.Idle:
00051                         _animator.Play(hash);
00052                         break;
00053
00054                     case FieldElementState.Hint:
00055                         _animator.SetTrigger(hash);
00056                         break;
00057                 }
00058             }
00059
00060         private void OnDisable()
00061         {
00062             _listener?.ResetAnimationState();
00063         }
00064 }
```

```
00064      }
00065 }
```

7.234 FieldElementStateMachineBehaviour.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Field.Elements.Animations
00007 {
00008     [SharedBetweenAnimators]
00009     public class FieldElementStateMachineBehaviour : StateMachineBehaviour
00010     {
00011         public override void OnStateExit(Animator animator, AnimatorStateInfo stateInfo, int
layerIndex)
00012         {
00013             base.OnStateExit(animator, stateInfo, layerIndex);
00014             var view = animator.gameObject.GetComponent<IFieldElementView>();
00015             int stateHash = stateInfo.shortNameHash;
00016
00017             if (view != null)
00018             {
00019                 if (stateHash ==
FieldElementAnimationController.GetFieldElementState(FieldElementState.Hint))
00020                 {
00021                     view.ResetState();
00022                 }
00023             }
00024         }
00025     }
00026 }
```

7.235 FieldElementModel.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005 using MergeIt.Core.MVP;
00006 using MergeIt.Game.Messages;
00007 using MergeIt.SimpleDI;
00008 using UnityEngine;
00009
00010 namespace MergeIt.Game.Field.Elements
00011 {
00012
00013     public class FieldElementModel : IModel, IFIELDElementModel
00014     {
00015         private readonly IMessageBus _messageBus = DiContainer.Get<IMessageBus>();
00016
00017         private bool _selected;
00018         private Vector3 _initialPosition;
00019         private GridPoint _point;
00020         private int _clicksCount;
00021
00022         public bool IsBusy { get; set; }
00023         public bool IsLocked { get; set; }
00024
00025         public bool Selected
00026         {
00027             get => _selected;
00028             set
00029             {
00030                 if (value != _selected)
00031                 {
00032                     _selected = value;
00033
00034                     if (_selected)
00035                     {
00036                         SendSelected();
00037                     }
00038                     else
00039                     {
00040                         _clicksCount = 0;
00041                     }
00042                 }
00043             }
00044 }
```

```

00044         }
00045
00046     public int ClicksCount
00047     {
00048         get => _clicksCount;
00049         set => _clicksCount = value;
00050     }
00051
00052     public GridPoint Point
00053     {
00054         get => _point;
00055         set => _point = value;
00056     }
00057
00058     private void SendSelected()
00059     {
00060         var message = new SelectedElementMessage
00061         {
00062             Point = Point
00063         };
00064
00065         _messageBus.Fire(message);
00066     }
00067 }
00068
00069 }
```

7.236 FieldElementPresenter.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Game.Field.Elements
00004 {
00005     public class FieldElementPresenter : FieldElementPresenterBase<FieldElementView,
    FieldElementModel>
00006     {
00007
00008     }
00009 }
```

7.237 FieldElementPresenterBase.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Animations;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Core.Helpers;
00006 using MergeIt.Core.MVP;
00007 using UnityEngine;
00008 using Object = UnityEngine.Object;
00009
00010 namespace MergeIt.Game.Field.Elements
00011 {
00012     public class FieldElementPresenterBase<TView, TModel> : Presenter<TView, TModel>,
    IFieldElementPresenter, IAnimationListener
00013     where TView : FieldElementView
00014     where TModel : FieldElementModel, new()
00015     {
00016         protected IFIELDElement FieldElement;
00017         private Transform _transform;
00018         private RectTransform _rectTransform;
00019         private Canvas _canvas;
00020         private FieldElementState _state;
00021
00022         public FieldElementState State
00023         {
00024             get => _state;
00025         }
00026
00027         public Transform Transform
00028         {
00029             get
00030             {
00031                 if (!_transform)
00032                 {
00033                     _transform = View.transform;
00034                 }
00035
00036             return _transform;
00037         }
00038     }
00039 }
```

```
00037         }
00038     }
00039
00040     public bool IsAvailable
00041     {
00042         get => !Model.IsBusy && !Model.IsLocked;
00043     }
00044
00045     public bool IsBusy
00046     {
00047         get => Model.IsBusy;
00048     }
00049
00050     public bool IsLocked
00051     {
00052         get => Model.IsLocked;
00053     }
00054
00055     public RectTransform RectTransform
00056     {
00057         get
00058         {
00059             if (!_rectTransform)
00060             {
00061                 _rectTransform = View.RectTransform;
00062             }
00063
00064             return _rectTransform;
00065         }
00066     }
00067
00068     public Canvas Canvas
00069     {
00070         get
00071         {
00072             if (!_canvas)
00073             {
00074                 _canvas = View.Canvas;
00075             }
00076
00077             return _canvas;
00078         }
00079     }
00080
00081     public virtual void Update(IFieldElement fieldElement)
00082     {
00083         FieldElement = fieldElement;
00084         Model.Point = FieldElement.InfoParameters.LogicPosition;
00085         Model.IsLocked = FieldElement.InfoParameters.IsBlocked;
00086
00087         View.GameObject.name = $"[{Model.Point.X}, {Model.Point.Y}]"
00088         {FieldElement.InfoParameters.Name}";
00089
00090         View.Lock(Model.IsLocked);
00091         View.ResetState();
00092     }
00093
00094     public virtual void Activate(bool isActive)
00095     {
00096         Model.Selected = false;
00097         View.GameObject.SetActive(isActive);
00098     }
00099
00100     public virtual void Release()
00101     {
00102     }
00103
00104     public virtual void Remove()
00105     {
00106         Model.Selected = false;
00107         View.ResetStateEvent -= OnResetState;
00108
00109         Object.Destroy(View.GameObject);
00110     }
00111
00112     public virtual void SetParent(Transform parent, bool resetPosition = true)
00113     {
00114         Transform.SetParent(parent);
00115         RectTransform.Stretch();
00116
00117         if (resetPosition)
00118         {
00119             RectTransform.anchoredPosition = Vector2.zero;
00120         }
00121
00122         Transform.localScale = Vector3.one;
```

```

00123     }
00124
00125     public virtual void ResetPosition()
00126     {
00127         RectTransform.anchoredPosition = Vector2.zero;
00128     }
00129
00130     public virtual void SetPoint(GridPoint point)
00131     {
00132         Model.Point = point;
00133     }
00134
00135     public virtual void SetLock(bool block)
00136     {
00137         Model.IsLocked = block;
00138         View.Lock(block);
00139     }
00140
00141     public virtual void Select(bool select)
00142     {
00143         Model.Selected = select;
00144     }
00145
00146     public virtual void StartDrag()
00147     {
00148         View.Canvas.sortingOrder = 2;
00149         View.GraphicRaycaster.enabled = false;
00150     }
00151
00152     public virtual void EndDrag()
00153     {
00154         View.Canvas.sortingOrder = 1;
00155         View.GraphicRaycaster.enabled = true;
00156     }
00157
00158     public virtual void SetBusy(bool isBusy)
00159     {
00160         Model.IsBusy = isBusy;
00161     }
00162
00163     public virtual IFIELDELEMENTVIEW GetView()
00164     {
00165         return View;
00166     }
00167
00168     public virtual IFIELDELEMENTMODEL GetModel()
00169     {
00170         return Model;
00171     }
00172
00173     public virtual void SetState(FieldElementState state)
00174     {
00175         _state = state;
00176         View.AnimationController.SetState(state);
00177     }
00178
00179     public virtual void ResetAnimationState()
00180     {
00181         View.ResetState();
00182     }
00183
00184     protected override void OnInitialize(TView view)
00185     {
00186         base.OnInitialize(view);
00187
00188         View.ResetStateEvent += OnResetState;
00189
00190         View.AnimationController.Initialize(this);
00191     }
00192
00193     private void OnResetState()
00194     {
00195         _state = FieldElementState.Idle;
00196     }
00197 }
00198 }
```

7.238 FieldElementView.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Animations;
00004 using MergeIt.Core.Configs.Elements;
```

```
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Core.MVP;
00007 using UnityEngine;
00008 using UnityEngine.Events;
00009 using UnityEngine.UI;
00010
00011 namespace MergeIt.Game.Field.Elements
00012 {
00013     public class FieldElementView : View, IFieldElementView
00014     {
00015         public UnityAction ResetStateEvent;
00016
00017         [SerializeField]
00018         private GameObject _blocker;
00019
00020         private IAnimationController _animationController;
00021
00022         public GraphicRaycaster GraphicRaycaster => GetComponent<GraphicRaycaster>();
00023
00024         public RectTransform RectTransform
00025         {
00026             get
00027             {
00028                 if (!_rectTransform)
00029                 {
00030                     _rectTransform = GetComponent<RectTransform>();
00031                 }
00032
00033                 return _rectTransform;
00034             }
00035         }
00036
00037         public Canvas Canvas
00038         {
00039             get
00040             {
00041                 if (!_canvas)
00042                 {
00043                     _canvas = GetComponent<Canvas>();
00044                 }
00045
00046                 return _canvas;
00047             }
00048         }
00049
00050         public IAnimationController AnimationController
00051         {
00052             get { return _animationController ??= GetComponent<IAnimationController>(); }
00053         }
00054
00055         private RectTransform _rectTransform;
00056         private Canvas _canvas;
00057         private FieldElementModel _model;
00058         private FieldElementIconComponent _iconComponent;
00059
00060         // private void Start()
00061         // {
00062         //     RectTransform rectTransform = GetComponent<RectTransform>();
00063         //     rectTransform.Stretch();
00064         // }
00065
00066         public void Lock(bool isLocked)
00067         {
00068             if (!_iconComponent)
00069             {
00070                 _iconComponent = GetComponentInChildren<FieldElementIconComponent>();
00071             }
00072
00073             _iconComponent.SetBlocked(isLocked);
00074             _blocker.SetActive(isLocked);
00075         }
00076
00077         public void ResetState()
00078         {
00079             ResetStateEvent?.Invoke();
00080         }
00081     }
00082 }
```

7.239 FieldElementGeneratorModel.cs

```
00001 // Copyright (c) 2024, Awessets
00002
```

```
00003 namespace MergeIt.Game.Field.Elements.Generator
00004 {
00005     public class FieldElementGeneratorModel : FieldElementModel
00006     {
00007     }
00008 }
00009 }
```

7.240 FieldElementGeneratorPresenter.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005 using MergeIt.Game.Messages;
00006 using MergeIt.SimpleDI;
00007
00008 namespace MergeIt.Game.Field.Elements.Generator
00009 {
00010     public class FieldElementGeneratorPresenter : FieldElementPresenterBase<FieldElementGeneratorView,
00011         FieldElementGeneratorModel>
00012     {
00013         private readonly IMessageBus _messageBus;
00014         private IGeneratorParameters _generatorParameters;
00015         private IGeneratorOpenParameters _generatorOpenParameters;
00016
00017         public FieldElementGeneratorPresenter()
00018         {
00019             _messageBus = DiContainer.Get<IMessageBus>();
00020             _messageBus.AddListener<CheckGeneratorMessage>(CheckGeneratorMessageHandler);
00021             _messageBus.AddListener<GeneratorOpenStartMessage>(GeneratorOpenStartMessageHandler);
00022         }
00023
00024         public override void Update(IFieldElement fieldElement)
00025         {
00026             base.Update(fieldElement);
00027
00028             _generatorParameters = FieldElement.GeneratorParameters;
00029             _generatorOpenParameters = FieldElement.GeneratorOpenParameters;
00030
00031             Check();
00032             CheckOpening();
00033         }
00034
00035         public override void Release()
00036         {
00037             base.Release();
00038
00039             View.HideTimer();
00040         }
00041
00042         protected override void OnDispose()
00043         {
00044             base.OnDispose();
00045
00046             _messageBus.RemoveListener<CheckGeneratorMessage>(CheckGeneratorMessageHandler);
00047             _messageBus.RemoveListener<GeneratorOpenStartMessage>(GeneratorOpenStartMessageHandler);
00048         }
00049
00050         private void CheckGeneratorMessageHandler(CheckGeneratorMessage message)
00051         {
00052             if (Model.Point == message.GeneratorPoint)
00053             {
00054                 Check();
00055             }
00056         }
00057
00058         private void GeneratorOpenStartMessageHandler(GeneratorOpenStartMessage message)
00059         {
00060             if (Model.Point == message.GeneratorPoint)
00061             {
00062                 CheckOpening();
00063             }
00064         }
00065
00066         private void Check()
00067         {
00068             if (_generatorParameters.AvailableToDrop == 0)
00069             {
00070                 View.SetTimer(_generatorParameters.MinDropChargeTime,
00071                             _generatorParameters.MinDropFullChargeTime);
00072             }
00073         }
00074     }
00075 }
```

```

00072     private void CheckOpening()
00073     {
00074         if (_generatorOpenParameters?.IsOpening == true)
00075         {
00076             var fullOpenTime =
00077                 FieldElement.ConfigParameters.ElementConfig.GeneratorSettings.OpenTime;
00078             View.SetTimer(_generatorOpenParameters.RemainingTime, fullOpenTime);
00079         }
00080     }
00081 }
00082 }
```

7.241 FieldElementGeneratorView.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Utils;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Field.Elements.Generator
00007 {
00008     public class FieldElementGeneratorView : FieldElementView
00009     {
00010         [SerializeField]
00011         private GeneratorTimerComponent _timer;
00012
00013         public void SetTimer(Bindable<float> remainChargeTime, float fullRemainTime)
00014         {
00015             _timer.StartTimer(remainChargeTime, fullRemainTime);
00016         }
00017
00018         public void HideTimer()
00019         {
00020             _timer.gameObject.SetActive(false);
00021         }
00022     }
00023 }
```

7.242 GeneratorTimerComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Utils;
00004 using UnityEngine;
00005 using UnityEngine.UI;
00006
00007 namespace MergeIt.Game.Field.Elements.Generator
00008 {
00009     public class GeneratorTimerComponent : MonoBehaviour
00010     {
00011         [SerializeField]
00012         private Image _fill;
00013
00014         private float _remainFullTime;
00015         private Bindable<float> _remainChargeTime;
00016
00017         public void StartTimer(Bindable<float> remainChargeTime, float fullRemainTime)
00018         {
00019             if (remainChargeTime.Value > 0f)
00020             {
00021                 gameObject.SetActive(true);
00022
00023                 _remainFullTime = fullRemainTime;
00024
00025                 _remainChargeTime?.Unsubscribe(OnRemainTimeChanged);
00026
00027                 _remainChargeTime = remainChargeTime;
00028                 _remainChargeTime.Subscribe(OnRemainTimeChanged, true);
00029             }
00030         }
00031
00032         private void OnRemainTimeChanged(float newValue)
00033         {
00034             if (newValue <= 0f)
00035             {
00036                 gameObject.SetActive(false);
00037             }
00038             else
```

```

00039         {
00040             var fill = newValue / _remainFullTime;
00041             _fill.fillAmount = 1f - fill;
00042         }
00043     }
00044
00045     private void OnDisable()
00046     {
00047         _remainChargeTime.Unsubscribe(OnRemainTimeChanged);
00048         _remainChargeTime = null;
00049         _remainFullTime = 0f;
00050     }
00051 }
00052 }
```

7.243 FieldCellComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005 using MergeIt.Game.Messages;
00006 using MergeIt.Game.Services;
00007 using MergeIt.SimpleDI;
00008 using UnityEngine;
00009 using UnityEngine.Events;
00010
00011 namespace MergeIt.Game.Field
00012 {
00013     public class FieldCellComponent : MonoBehaviour,
00014         IDragHandler,
00015         IPointerDownHandler,
00016         IPointerUpHandler,
00017         IBeginDragHandler,
00018         IEndDragHandler
00019     {
00020         private readonly float _pointerClickTime = 0.2f;
00021
00022         [SerializeField]
00023         private GameObject _selector;
00024
00025         [SerializeField]
00026         private Transform _objectContainer;
00027
00028         private RectTransform _rectTransform;
00029         private IFIELDElementPresenter _fieldElementPresenter;
00030         private GridPoint _gridPoint;
00031
00032         private GameServiceModel _gameServiceModel;
00033         private IMessageBus _messageBus;
00034
00035         private Vector3 _offset;
00036         private Camera _mainCamera;
00037         private Transform _transform;
00038
00039         private float _pointerDownTime;
00040
00041         public IFIELDElementPresenter FieldElementPresenter => _fieldElementPresenter;
00042
00043         public RectTransform RectTransform
00044         {
00045             get
00046             {
00047                 if (!_rectTransform)
00048                 {
00049                     _rectTransform = GetComponent<RectTransform>();
00050                 }
00051
00052                 return _rectTransform;
00053             }
00054         }
00055
00056
00057         public Transform ObjectContainer
00058         {
00059             get => _objectContainer;
00060         }
00061
00062         public GridPoint Point
00063         {
00064             get => _gridPoint;
00065         }
00066 }
```

```
00067     public void Initialize(GridPoint gridPoint)
00068     {
00069         _gridPoint = gridPoint;
00070         _messageBus = DiContainer.Get<IMessageBus>();
00071         _gameServiceModel = DiContainer.Get<GameServiceModel>();
00072
00073         _messageBus.AddListener<ResetPositionMessage>(OnResetPositionMessageHandler);
00074     }
00075
00076     public void ToggleSelector(bool activate)
00077     {
00078         _selector.SetActive(activate);
00079
00080         if (activate)
00081         {
00082             _messageBus.Fire<ResetHintsMessage>();
00083         }
00084
00085         if (!activate && _fieldElementPresenter != null)
00086         {
00087             _fieldElementPresenter.GetModel().Selected = false;
00088         }
00089     }
00090
00091     public void AttachObject(IFieldElementPresenter presenter, bool resetPosition = true)
00092     {
00093         if (presenter != null)
00094         {
00095             _fieldElementPresenter = presenter;
00096             _fieldElementPresenter.SetParent(ObjectContainer, resetPosition);
00097             _fieldElementPresenter.SetPoint(Point);
00098         }
00099         else
00100         {
00101             ToggleSelector(false);
00102         }
00103     }
00104
00105     public void DetachObject()
00106     {
00107         _fieldElementPresenter = null;
00108     }
00109
00110     public void OnPointerDown(PointerEventData eventData)
00111     {
00112         if (_fieldElementPresenter?.IsBusy == false)
00113         {
00114             _offset = _fieldElementPresenter.Transform.position -
00115             _gameServiceModel.MainCamera.ScreenToWorldPoint(eventData.position);
00116             _pointerDownTime = Time.realtimeSinceStartup;
00117         }
00118     }
00119
00120     public void OnPointerUp(PointerEventData eventData)
00121     {
00122         if (_fieldElementPresenter?.IsBusy == false)
00123         {
00124             var pointerUpTime = Time.realtimeSinceStartup;
00125             var diff = pointerUpTime - _pointerDownTime;
00126
00127             if (diff <= _pointerClickTime)
00128             {
00129                 SendClick();
00130             }
00131         }
00132     }
00133
00134     public void OnDrag(PointerEventData eventData)
00135     {
00136         if (_fieldElementPresenter?.IsAvailable == true)
00137         {
00138             _fieldElementPresenter.Transform.position =
00139             _gameServiceModel.MainCamera.ScreenToWorldPoint(eventData.position) + _offset;
00140         }
00141
00142     public void OnBeginDrag(PointerEventData eventData)
00143     {
00144         if (_fieldElementPresenter?.IsAvailable == true)
00145         {
00146             _fieldElementPresenter.Select(true);
00147             _fieldElementPresenter.StartDrag();
00148
00149             _messageBus.Fire(new ActivateHintsMessage { Active = false });
00150
00151             SetVisualSelection(false);
00151         }
00151     }
```

```

00152         }
00153
00154     public void OnEndDrag(PointerEventData eventData)
00155     {
00156         if (_fieldElementPresenter?.IsAvailable == true)
00157         {
00158             _fieldElementPresenter.EndDrag();
00159             SendEndDrag(eventData);
00160
00161             _messageBus.Fire(new ActivateHintsMessage { Active = true });
00162         }
00163     }
00164
00165     private void SetVisualSelection(bool activate)
00166     {
00167         ToggleSelector(activate);
00168     }
00169
00170     private void OnResetPositionMessageHandler(ResetPositionMessage message)
00171     {
00172         if (message.From == Point)
00173         {
00174             SetVisualSelection(true);
00175         }
00176     }
00177
00178     private void SendClick()
00179     {
00180         var clickMessage = new ClickElementMessage
00181         {
00182             Cell = this
00183         };
00184
00185         _messageBus.Fire(clickMessage);
00186     }
00187
00188     private void SendEndDrag(PointerEventData eventData)
00189     {
00190         var raycastGameObject = eventData.pointerCurrentRaycast.gameObject;
00191
00192         var endDragMessage = new EndDragElementMessage
00193         {
00194             FromPoint,
00195             ToGameObject = raycastGameObject,
00196             Position = eventData.position
00197         };
00198
00199         _messageBus.Fire(endDragMessage);
00200     }
00201
00202     private void OnDestroy()
00203     {
00204         _messageBus.RemoveListener<ResetPositionMessage>(OnResetPositionMessageHandler);
00205     }
00206 }
00207 }
```

7.244 FieldData.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Core.Saves;
00007 using Newtonsoft.Json;
00008
00009 namespace MergeIt.Game.Field
00010 {
00011     [Serializable, Savable("field", "dat")]
00012     public class FieldData : ISavable
00013     {
00014         [JsonProperty("w")]
00015         public int FieldWidth;
00016         [JsonProperty("h")]
00017         public int FieldHeight;
00018         [JsonProperty("e")]
00019         public List<FieldElementData> SavedElementsData;
00020     }
00021 }
```

7.245 FieldLogic.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Core.Messages;
00006 using MergeIt.Game.Messages;
00007 using MergeIt.SimpleDI;
00008 using MergeIt.SimpleDI.ReservedInterfaces;
00009
00010 namespace MergeIt.Game.Field
00011 {
00012     public class FieldLogic : IFieldLogic, IInitializable, IDisposable
00013     {
00014         [Introduce]
00015         private IMessageBus _messageBus;
00016
00017         [Introduce]
00018         private FieldLogicModel _fieldLogicModel;
00019
00020         public void Initialize()
00021         {
00022             _messageBus.AddListener<CreateElementMessage>(OnCreateElementMessageHandler);
00023             _messageBus.AddListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00024             _messageBus.AddListener<SwapElementsMessage>(OnSwapElementsMessageHandler);
00025             _messageBus.AddListener<SplitElementMessage>(OnSplitElementMessageHandler);
00026             _messageBus.AddListener<MergeElementsMessage>(OnMergeElementMessageHandler);
00027         }
00028
00029         public void Dispose()
00030         {
00031             _messageBus.RemoveListener<CreateElementMessage>(OnCreateElementMessageHandler);
00032             _messageBus.RemoveListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00033             _messageBus.RemoveListener<SwapElementsMessage>(OnSwapElementsMessageHandler);
00034             _messageBus.RemoveListener<SplitElementMessage>(OnSplitElementMessageHandler);
00035             _messageBus.RemoveListener<MergeElementsMessage>(OnMergeElementMessageHandler);
00036         }
00037
00038         private void OnCreateElementMessageHandler(CreateElementMessage message)
00039         {
00040             IFIELD_ELEMENT fieldElement = message.NewElement;
00041             fieldElement.InfoParameters.LogicPosition = messageToPoint;
00042
00043             _fieldLogicModel.FieldElements[messageToPoint] = fieldElement;
00044         }
00045
00046         private void OnRemoveElementMessageHandler(RemoveElementMessage message)
00047         {
00048             _fieldLogicModel.FieldElements.Remove(message.RemoveAtPoint);
00049         }
00050
00051         private void OnSwapElementsMessageHandler(SwapElementsMessage message)
00052         {
00053             var fromPoint = message.From;
00054             var toPoint = message.To;
00055
00056             IFIELD_ELEMENT firstElement = _fieldLogicModel.FieldElements[fromPoint];
00057
00058             if (_fieldLogicModel.FieldElements.TryGetValue(toPoint, out IFIELD_ELEMENT secondElement))
00059             {
00060                 secondElement.InfoParameters.LogicPosition = fromPoint;
00061                 firstElement.InfoParameters.LogicPosition = toPoint;
00062
00063                 _fieldLogicModel.FieldElements[fromPoint] = secondElement;
00064                 _fieldLogicModel.FieldElements[toPoint] = firstElement;
00065             }
00066             else
00067             {
00068                 firstElement.InfoParameters.LogicPosition = toPoint;
00069
00070                 _fieldLogicModel.FieldElements[toPoint] = firstElement;
00071                 _fieldLogicModel.FieldElements.Remove(fromPoint);
00072             }
00073         }
00074
00075         private void OnSplitElementMessageHandler(SplitElementMessage message)
00076         {
00077             IFIELD_ELEMENT fieldElement1 = message.SplitElement1;
00078             IFIELD_ELEMENT fieldElement2 = message.SplitElement2;
00079
00080             GridPoint initPoint = message.SplitElement1.InfoParameters.LogicPosition;
00081             GridPoint secondPoint = message.SplitElement2.InfoParameters.LogicPosition;
00082
00083             _fieldLogicModel.FieldElements[initPoint] = fieldElement1;
00084             _fieldLogicModel.FieldElements[secondPoint] = fieldElement2;
00085         }
}
```

```

00086     private void OnMergeElementMessageHandler(MergeElementsMessage message)
00087     {
00088         _fieldLogicModel.FieldElements[message.NewElement.InfoParameters.LogicPosition] =
00089             message.NewElement;
00090     }
00091 }
00092 }
```

7.246 FieldLogicModel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.FieldElements;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Game.Field
00008 {
00009     public class FieldLogicModel
00010     {
00011         public Dictionary<GridPoint, FieldCellComponent> CellComponents { get; } = new();
00012         public Dictionary<GridPoint, IFieldElement> FieldElements { get; } = new();
00013         public IFieldElement OpeningGenerator { get; set; }
00014         public List<IFieldElement> AllGenerators { get; set; } = new();
00015         public int FieldWidth { get; set; }
00016         public int FieldHeight { get; set; }
00017         public float CellSize { get; set; }
00018         public Rect FieldRect { get; set; }
00019         public float PositionOffset { get; set; }
00020         public float CellSpacing { get; set; }
00021
00022         public Vector3 GetPosition(GridPoint fromPoint)
00023         {
00024             if (CellComponents.TryGetValue(fromPoint, out FieldCellComponent cell))
00025             {
00026                 return cell.ObjectContainer.position;
00027             }
00028
00029             return Vector3.zero;
00030         }
00031     }
00032 }
```

7.247 FieldModel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.MVP;
00004
00005 namespace MergeIt.Game.Field
00006 {
00007     public class FieldModel : Model
00008     {
00009
00010     }
00011 }
```

7.248 FieldPresenter.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Core.Messages;
00006 using MergeIt.Core.MVP;
00007 using MergeIt.Core.Services;
00008 using MergeIt.Game.Effects;
00009 using MergeIt.Game.Effects.Controllers;
00010 using MergeIt.Game.Effects.Parameters;
00011 using MergeIt.Game.Factories.Field;
00012 using MergeIt.Game.Factories.FieldElement;
00013 using MergeIt.Game.Messages;
00014 using MergeIt.SimpleDI;
00015 using UnityEngine;
```

```
00016
00017 namespace MergeIt.Game.Field
00018 {
00019     public class FieldPresenter : Presenter<FieldView, FieldModel>
00020     {
00021         private readonly FieldLogicModel _fieldLogicModel;
00022         private readonly IFieldFactory _fieldFactory;
00023         private readonly IFieldElementVisualFactory _fieldElementVisualFactory;
00024         private readonly IMessageBus _messageBus;
00025         private readonly IGameSaveService _saveService;
00026         private readonly IEffectsFactory _effectsFactory;
00027
00028         private Dictionary<GridPoint, FieldCellComponent> _cellComponents;
00029
00030         public FieldPresenter()
00031         {
00032             _fieldLogicModel = DiContainer.Get<FieldLogicModel>();
00033             _fieldFactory = DiContainer.Get<IFieldFactory>();
00034             _fieldElementVisualFactory = DiContainer.Get<IFieldElementVisualFactory>();
00035             _messageBus = DiContainer.Get<IMessageBus>();
00036             _saveService = DiContainer.Get<IGameSaveService>();
00037             _effectsFactory = DiContainer.Get<IEffectsFactory>();
00038         }
00039
00040         protected override void OnDispose()
00041         {
00042             _messageBus.RemoveListener<SelectedElementMessage>(OnSelectedElementMessageHandler);
00043             _messageBus.RemoveListener<ResetSelectionMessage>(OnResetSelectionMessageHandler);
00044             _messageBus.RemoveListener<MergeElementsMessage>(OnMergeElementsMessageHandler);
00045             _messageBus.RemoveListener<UnlockElementMessage>(OnUnlockElementMessageHandler);
00046             _messageBus.RemoveListener<SplitElementMessage>(OnSplitResultMessageHandler);
00047             _messageBus.RemoveListener<ResetPositionMessage>(OnResetPositionsMessageHandler);
00048             _messageBus.RemoveListener<SwapElementsMessage>(OnSwapElementsMessageHandler);
00049             _messageBus.RemoveListener<CreateElementMessage>(OnCreateElementMessageHandler);
00050             _messageBus.RemoveListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00051         }
00052
00053         public void Initialize()
00054         {
00055             _messageBus.AddListener<SelectedElementMessage>(OnSelectedElementMessageHandler);
00056             _messageBus.AddListener<ResetSelectionMessage>(OnResetSelectionMessageHandler);
00057             _messageBus.AddListener<MergeElementsMessage>(OnMergeElementsMessageHandler);
00058             _messageBus.AddListener<SplitElementMessage>(OnSplitResultMessageHandler);
00059             _messageBus.AddListener<UnlockElementMessage>(OnUnlockElementMessageHandler);
00060             _messageBus.AddListener<ResetPositionMessage>(OnResetPositionsMessageHandler);
00061             _messageBus.AddListener<SwapElementsMessage>(OnSwapElementsMessageHandler);
00062             _messageBus.AddListener<CreateElementMessage>(OnCreateElementMessageHandler);
00063             _messageBus.AddListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00064
00065             _cellComponents = _fieldLogicModel.CellComponents;
00066
00067             var fieldRect = View.CellsContainer.rect;
00068             var fieldRectSize = fieldRect.size;
00069
00070             _fieldLogicModel.FieldRect = fieldRect;
00071
00072             int fieldWidth = _fieldLogicModel.FieldWidth;
00073             int fieldHeight = _fieldLogicModel.FieldHeight;
00074
00075             float sizeByWidth = (fieldRectSize.x - (fieldWidth - 1) * View.CellSpacing) / fieldWidth;
00076             float sizeByHeight = (fieldRectSize.y - (fieldHeight - 1) * View.CellSpacing) /
00077                 fieldHeight;
00078
00079             float cellSize = Mathf.Min(sizeByHeight, sizeByWidth);
00080
00081             bool horizontalAlign = Mathf.Approximately(cellSize, sizeByHeight);
00082             float offset = 0f;
00083
00084             if (horizontalAlign)
00085             {
00086                 offset = (fieldRectSize.x - ((fieldWidth - 1) * View.CellSpacing + cellSize *
00087                     fieldWidth)) / 2f;
00088             }
00089             else
00090             {
00091                 offset = (fieldRectSize.y - ((fieldHeight - 1) * View.CellSpacing + cellSize *
00092                     fieldHeight)) / 2f;
00093             }
00094
00095             _fieldLogicModel.CellSize = cellSize;
00096             _fieldLogicModel.PositionOffset = offset;
00097             _fieldLogicModel.CellSpacing = View.CellSpacing;
00098
00099             for (int i = 0; i < fieldHeight; i++)
00100             {
00101                 for (int j = 0; j < fieldWidth; j++)
00102                 {
00103
00104
00105
00106
00107
00108
00109
00110
00111
00112
00113
00114
00115
00116
00117
00118
00119
00120
00121
00122
00123
00124
00125
00126
00127
00128
00129
00130
00131
00132
00133
00134
00135
00136
00137
00138
00139
00140
00141
00142
00143
00144
00145
00146
00147
00148
00149
00150
00151
00152
00153
00154
00155
00156
00157
00158
00159
00160
00161
00162
00163
00164
00165
00166
00167
00168
00169
00170
00171
00172
00173
00174
00175
00176
00177
00178
00179
00180
00181
00182
00183
00184
00185
00186
00187
00188
00189
00190
00191
00192
00193
00194
00195
00196
00197
00198
00199
00200
00201
00202
00203
00204
00205
00206
00207
00208
00209
00210
00211
00212
00213
00214
00215
00216
00217
00218
00219
00220
00221
00222
00223
00224
00225
00226
00227
00228
00229
00230
00231
00232
00233
00234
00235
00236
00237
00238
00239
00240
00241
00242
00243
00244
00245
00246
00247
00248
00249
00250
00251
00252
00253
00254
00255
00256
00257
00258
00259
00260
00261
00262
00263
00264
00265
00266
00267
00268
00269
00270
00271
00272
00273
00274
00275
00276
00277
00278
00279
00280
00281
00282
00283
00284
00285
00286
00287
00288
00289
00290
00291
00292
00293
00294
00295
00296
00297
00298
00299
00300
00301
00302
00303
00304
00305
00306
00307
00308
00309
00310
00311
00312
00313
00314
00315
00316
00317
00318
00319
00320
00321
00322
00323
00324
00325
00326
00327
00328
00329
00330
00331
00332
00333
00334
00335
00336
00337
00338
00339
00340
00341
00342
00343
00344
00345
00346
00347
00348
00349
00350
00351
00352
00353
00354
00355
00356
00357
00358
00359
00360
00361
00362
00363
00364
00365
00366
00367
00368
00369
00370
00371
00372
00373
00374
00375
00376
00377
00378
00379
00380
00381
00382
00383
00384
00385
00386
00387
00388
00389
00390
00391
00392
00393
00394
00395
00396
00397
00398
00399
00400
00401
00402
00403
00404
00405
00406
00407
00408
00409
00410
00411
00412
00413
00414
00415
00416
00417
00418
00419
00420
00421
00422
00423
00424
00425
00426
00427
00428
00429
00430
00431
00432
00433
00434
00435
00436
00437
00438
00439
00440
00441
00442
00443
00444
00445
00446
00447
00448
00449
00450
00451
00452
00453
00454
00455
00456
00457
00458
00459
00460
00461
00462
00463
00464
00465
00466
00467
00468
00469
00470
00471
00472
00473
00474
00475
00476
00477
00478
00479
00480
00481
00482
00483
00484
00485
00486
00487
00488
00489
00490
00491
00492
00493
00494
00495
00496
00497
00498
00499
00500
00501
00502
00503
00504
00505
00506
00507
00508
00509
00510
00511
00512
00513
00514
00515
00516
00517
00518
00519
00520
00521
00522
00523
00524
00525
00526
00527
00528
00529
00530
00531
00532
00533
00534
00535
00536
00537
00538
00539
00540
00541
00542
00543
00544
00545
00546
00547
00548
00549
00550
00551
00552
00553
00554
00555
00556
00557
00558
00559
00560
00561
00562
00563
00564
00565
00566
00567
00568
00569
00570
00571
00572
00573
00574
00575
00576
00577
00578
00579
00580
00581
00582
00583
00584
00585
00586
00587
00588
00589
00590
00591
00592
00593
00594
00595
00596
00597
00598
00599
00600
00601
00602
00603
00604
00605
00606
00607
00608
00609
00610
00611
00612
00613
00614
00615
00616
00617
00618
00619
00620
00621
00622
00623
00624
00625
00626
00627
00628
00629
00630
00631
00632
00633
00634
00635
00636
00637
00638
00639
00640
00641
00642
00643
00644
00645
00646
00647
00648
00649
00650
00651
00652
00653
00654
00655
00656
00657
00658
00659
00660
00661
00662
00663
00664
00665
00666
00667
00668
00669
00670
00671
00672
00673
00674
00675
00676
00677
00678
00679
00680
00681
00682
00683
00684
00685
00686
00687
00688
00689
00690
00691
00692
00693
00694
00695
00696
00697
00698
00699
00700
00701
00702
00703
00704
00705
00706
00707
00708
00709
00710
00711
00712
00713
00714
00715
00716
00717
00718
00719
00720
00721
00722
00723
00724
00725
00726
00727
00728
00729
00730
00731
00732
00733
00734
00735
00736
00737
00738
00739
00740
00741
00742
00743
00744
00745
00746
00747
00748
00749
00750
00751
00752
00753
00754
00755
00756
00757
00758
00759
00760
00761
00762
00763
00764
00765
00766
00767
00768
00769
00770
00771
00772
00773
00774
00775
00776
00777
00778
00779
00780
00781
00782
00783
00784
00785
00786
00787
00788
00789
00790
00791
00792
00793
00794
00795
00796
00797
00798
00799
00800
00801
00802
00803
00804
00805
00806
00807
00808
00809
00810
00811
00812
00813
00814
00815
00816
00817
00818
00819
00820
00821
00822
00823
00824
00825
00826
00827
00828
00829
00830
00831
00832
00833
00834
00835
00836
00837
00838
00839
00840
00841
00842
00843
00844
00845
00846
00847
00848
00849
00850
00851
00852
00853
00854
00855
00856
00857
00858
00859
00860
00861
00862
00863
00864
00865
00866
00867
00868
00869
00870
00871
00872
00873
00874
00875
00876
00877
00878
00879
00880
00881
00882
00883
00884
00885
00886
00887
00888
00889
00890
00891
00892
00893
00894
00895
00896
00897
00898
00899
00900
00901
00902
00903
00904
00905
00906
00907
00908
00909
00910
00911
00912
00913
00914
00915
00916
00917
00918
00919
00920
00921
00922
00923
00924
00925
00926
00927
00928
00929
00930
00931
00932
00933
00934
00935
00936
00937
00938
00939
00940
00941
00942
00943
00944
00945
00946
00947
00948
00949
00950
00951
00952
00953
00954
00955
00956
00957
00958
00959
00960
00961
00962
00963
00964
00965
00966
00967
00968
00969
00970
00971
00972
00973
00974
00975
00976
00977
00978
00979
00980
00981
00982
00983
00984
00985
00986
00987
00988
00989
00990
00991
00992
00993
00994
00995
00996
00997
00998
00999
00999
```

```

00100         var point = GridPoint.Create(i, j);
00101         var cell = _fieldFactory.CreateFieldCell(View.CellsContainer.transform, point);
00102         cell.RectTransform.localScale = Vector3.one;
00103         cell.RectTransform.pivot = Vector2.zero;
00104         cell.RectTransform.sizeDelta = new Vector2(cellSize, cellSize);
00105         cell.RectTransform.anchoredPosition = horizontalAlign
00106             ? new Vector2(offset + j * (cellSize + View.CellSpacing), (fieldHeight - 1 -
00107     i) * (cellSize + View.CellSpacing))
00108             : new Vector2(j * (cellSize + View.CellSpacing), offset + i * (cellSize +
00109     View.CellSpacing));
00110
00111         cell.gameObject.name = $"FieldCell [{point.X}, {point.Y}]";
00112
00113         _cellComponents[point] = cell;
00114
00115         if (_fieldLogicModel.FieldElements.TryGetValue(point, out var fieldElement))
00116         {
00117             var fieldElementPresenter =
00118                 _fieldElementVisualFactory.CreateFieldElement(fieldElement);
00119                 cell.AttachObject(fieldElementPresenter);
00120             }
00121         }
00122
00123     private void OnSelectedElementMessageHandler(SelectedElementMessage message)
00124     {
00125         UpdateSelection(message.Point);
00126     }
00127
00128     private void OnResetSelectionMessageHandler(ResetSelectionMessage message)
00129     {
00130         UpdateSelection(GridPoint.Default);
00131     }
00132
00133     private void OnMergeElementsMessageHandler(MergeElementsMessage message)
00134     {
00135         GridPoint toPoint = message.NewElement.InfoParameters.LogicPosition;
00136         FieldCellComponent toCell = _cellComponents[toPoint];
00137
00138         toCell.DetachObject();
00139
00140         var fieldElementPresenter =
00141             _fieldElementVisualFactory.CreateFieldElement(message.NewElement);
00142             toCell.AttachObject(fieldElementPresenter);
00143             fieldElementPresenter.Select(true);
00144     }
00145
00146     private void OnSplitResultMessageHandler(SplitElementMessage message)
00147     {
00148         _messageBus.Fire<ResetHintsMessage>();
00149
00150         IFieldElement splitElement1 = message.SplitElement1;
00151         IFieldElement splitElement2 = message.SplitElement2;
00152
00153         GridPoint initPoint = splitElement1.InfoParameters.LogicPosition;
00154         GridPoint secondPoint = splitElement2.InfoParameters.LogicPosition;
00155
00156         FieldCellComponent initialCell = _cellComponents[initPoint];
00157         FieldCellComponent secondCell = _cellComponents[secondPoint];
00158
00159         IFieldElementPresenter fromPresenter = initialCell.FieldElementPresenter;
00160         Vector3 fromPosition = initialCell.ObjectContainer.position;
00161
00162         fromPresenter.Remove();
00163         initialCell.DetachObject();
00164
00165         IFieldElementPresenter fieldElementPresenter1 =
00166             _fieldElementVisualFactory.CreateFieldElement(splitElement1);
00167             initialCell.AttachObject(fieldElementPresenter1);
00168
00169         IFieldElementPresenter fieldElementPresenter2 =
00170             _fieldElementVisualFactory.CreateFieldElement(splitElement2);
00171             secondCell.AttachObject(fieldElementPresenter2, false);
00172             fieldElementPresenter2.Transform.position = fromPosition;
00173
00174         var moveParameters = new MoveEffectParameters(fromPosition,
00175             secondCell.ObjectContainer.position);
00176         _effectsFactory.CreateEffect<MoveEffectWithPresenter>(fieldElementPresenter2,
00177             moveParameters);
00178         _effectsFactory.CreateEffect<JumpEffectWithPresenter>(fieldElementPresenter1);
00179     }
00180
00181     private void OnUnlockElementMessageHandler(UnlockElementMessage message)
00182     {
00183         GridPoint point = message.Element.InfoParameters.LogicPosition;

```

```
00179
00180      if (_cellComponents.TryGetValue(point, out FieldCellComponent cellComponent) &&
00181          cellComponent.FieldElementPresenter != null)
00182      {
00183          cellComponent.FieldElementPresenter.SetLock(false);
00184      }
00185  }
00186
00187  private void OnSwapElementsMessageHandler(SwapElementsMessage message)
00188  {
00189      GridPoint fromPoint = message.From;
00190      GridPoint toPoint = message.To;
00191
00192      FieldCellComponent fromCell = _cellComponents[fromPoint];
00193      FieldCellComponent toCell = _cellComponents[toPoint];
00194
00195      IFieldElementPresenter fromPresenter = fromCell.FieldElementPresenter;
00196      IFieldElementPresenter toPresenter = toCell.FieldElementPresenter;
00197
00198      fromCell.DetachObject();
00199      toCell.DetachObject();
00200
00201      fromCell.AttachObject(toPresenter);
00202      toCell.AttachObject(fromPresenter);
00203
00204      toPresenter?.Select(false);
00205      fromPresenter?.Select(true);
00206  }
00207
00208  private void OnResetPositionsMessageHandler(ResetPositionMessage message)
00209  {
00210      GridPoint fromPoint = message.From;
00211      FieldCellComponent fromCell = _cellComponents[fromPoint];
00212      IFieldElementPresenter fromPresenter = fromCell.FieldElementPresenter;
00213
00214      fromPresenter.ResetPosition();
00215      fromPresenter.Select(true);
00216  }
00217
00218  private void OnCreateElementMessageHandler(CreateElementMessage message)
00219  {
00220      FieldCellComponent toCell = _cellComponents[messageToPoint];
00221
00222      IFieldElementPresenter fieldElementPresenter =
00223      _fieldElementVisualFactory.CreateFieldElement(message.NewElement);
00224      toCell.AttachObject(fieldElementPresenter, false);
00225
00226      Transform transform = fieldElementPresenter.Transform;
00227
00228      if (message.FromPosition != null)
00229      {
00230          transform.position = message.FromPosition.Value;
00231
00232          var parameters = new MoveEffectParameters(transform.position,
00233              toCell.ObjectContainer.position);
00234          _effectsFactory.CreateEffect<MoveEffectWithPresenter>(fieldElementPresenter,
00235          parameters);
00236      }
00237
00238  private void OnRemoveElementMessageHandler(RemoveElementMessage message)
00239  {
00240      RemoveElement(message.RemoveAtPoint);
00241
00242  private void RemoveElement(GridPoint point)
00243  {
00244      FieldCellComponent cell = _cellComponents[point];
00245      IFieldElementPresenter cellPresenter = cell.FieldElementPresenter;
00246      cell.ToggleSelector(false);
00247      cellPresenter.Remove();
00248      cell.DetachObject();
00249  }
00250
00251  private void UpdateSelection(GridPoint point)
00252  {
00253      var cells = _cellComponents;
00254
00255      if (cells != null)
00256      {
00257          foreach (var fieldCellComponent in cells)
00258          {
00259              bool activateCell = fieldCellComponent.Key.Equals(point);
00260              fieldCellComponent.Value.ToggleSelector(activateCell);
00261          }
00262      }
00263  }
```

```
00263     }
00264 }
```

7.249 FieldView.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.MVP;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Field
00007 {
00008     public class FieldView : View
00009     {
00010         [SerializeField]
00011         private RectTransform _cellsContainer;
00012
00013         [SerializeField]
00014         private float _cellSpacing;
00015
00016         public RectTransform CellsContainer => _cellsContainer;
00017         public float CellSpacing => _cellSpacing;
00018     }
00019 }
```

7.250 IFieldLogic.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Game.Field
00004 {
00005     public interface IFieldLogic
00006     {
00007     }
00008 }
```

7.251 GameRoot.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004 using MergeIt.Core.WindowSystem;
00005 using MergeIt.Core.WindowSystem.Factory;
00006 using MergeIt.Game.Configs.Services;
00007 using MergeIt.Game.Converters;
00008 using MergeIt.Game.Effects;
00009 using MergeIt.Game.Factories.ElementInfo;
00010 using MergeIt.Game.Factories.Field;
00011 using MergeIt.Game.Factories.FieldElement;
00012 using MergeIt.Game.Factories.Icons;
00013 using MergeIt.Game.Factories.Inventory;
00014 using MergeIt.Game.Field;
00015 using MergeIt.Game.Managers;
00016 using MergeIt.Game.Messages;
00017 using MergeIt.Game.Services;
00018 using MergeIt.Game.Services.Saves;
00019 using MergeIt.Game.Services.Saves.Strategies;
00020 using MergeIt.Game.Windows.ElementInfo;
00021 using MergeIt.Game.Windows.Inventory;
00022 using MergeIt.Game.Windows.UserProgress;
00023 using MergeIt.SimpleDI;
00024
00025 namespace MergeIt.Game
00026 {
00027     public class GameRoot : DiRoot
00028     {
00029         protected override void OnInstall()
00030         {
00031             DiContainer.RegisterInterfacesFor<MessageBus>().AsSingleton();
00032
00033             DiContainer.RegisterInterfacesFor<GameService>().AsSingleton();
00034             DiContainer.RegisterInterfacesFor<GameFieldService>().AsSingleton();
00035             DiContainer.RegisterInterfacesFor<GameFieldActionsService>().AsSingleton();
00036             DiContainer.RegisterInterfacesFor<GameSaveEveryIntervalService>().AsSingleton();
00037             DiContainer.RegisterInterfacesFor<UserService>().AsSingleton();

```

```

00038     DiContainer.RegisterInterfacesFor<UserProgressService>().AsSingleton();
00039     DiContainer.RegisterInterfacesFor<InventoryService>().AsSingleton();
00040     DiContainer.RegisterInterfacesFor<EnergyService>().AsSingleton();
00041     DiContainer.RegisterInterfacesFor<GeneratorService>().AsSingleton();
00042     DiContainer.RegisterInterfacesFor<InfoPanelService>().AsSingleton();
00043     DiContainer.RegisterInterfacesFor<CurrencyService>().AsSingleton();
00044     DiContainer.RegisterInterfacesFor<ElementService>().AsSingleton();
00045     DiContainer.RegisterInterfacesFor<ElementsStockService>().AsSingleton();
00046     DiContainer.RegisterInterfacesFor<EvolutionsService>().AsSingleton();
00047
00048     DiContainer.RegisterInterfacesFor<FieldLogic>().AsSingleton();
00049
00050     DiContainer.RegisterInterfacesFor<EffectsFactory>().AsSingleton();
00051     DiContainer.RegisterInterfacesFor<EffectsManager>().AsSingleton();
00052
00053     DiContainer.RegisterInterfacesFor<ConfigsService>().AsSingleton();
00054     DiContainer.RegisterInterfacesFor<ConfigProcessor>().AsSingleton();
00055     DiContainer.RegisterInterfacesFor<GameLoadService>().AsSingleton();
00056     DiContainer.RegisterInterfacesFor<ResourcesLoaderService>().AsSingleton();
00057
00058     DiContainer.RegisterInterfacesFor<FieldElementVisualFactory>().AsSingleton();
00059     DiContainer.RegisterInterfacesFor<FieldElementFactory>().AsSingleton();
00060     DiContainer.RegisterInterfacesFor<FieldFactory>().AsSingleton();
00061     DiContainer.RegisterInterfacesFor<InventoryFactory>().AsSingleton();
00062     DiContainer.RegisterInterfacesFor<ElementInfoFactory>().AsSingleton();
00063     DiContainer.RegisterInterfacesFor<IconFactory>().AsSingleton();
00064
00065     DiContainer.RegisterInterfacesFor<HintsManager>().AsSingleton();
00066
00067     DiContainer.Register<FieldLogicModel>().AsSingleton();
00068     DiContainer.Register<GameServiceModel>().AsSingleton();
00069     DiContainer.Register<UserServiceModel>().AsSingleton();
00070     DiContainer.Register<InventoryServiceModel>().AsSingleton();
00071     DiContainer.Register<ElementsStockServiceModel>().AsSingleton();
00072
00073     DiContainer.RegisterInterfacesFor<WindowsSystem>().AsSingleton();
00074     DiContainer.RegisterInterfacesFor<WindowFactory>().AsSingleton();
00075
00076     DiContainer.RegisterInterfacesFor<JsonSerializeStrategy>().AsSingleton();
00077 }
00078
00079     protected override void Run()
00080     {
00081         RegisterWindows();
00082
00083         var messageBus = DiContainer.Get<IMessageBus>();
00084         messageBus.Fire<StartGameMessage>();
00085     }
00086
00087     private void RegisterWindows()
00088     {
00089         WindowsStorage.Register<InventoryPresenter, InventoryWindow>(WindowName.Inventory,
00090             SortingLayers.Main);
00091         WindowsStorage.Register<UserInfoPresenter, UserInfoWindow>(WindowName.UserInfo,
00092             SortingLayers.Main);
00093         WindowsStorage.Register<ElementInfoPresenter, ElementInfoWindow>(WindowName.ElementInfo,
00094             SortingLayers.Popup);
00095         WindowsStorage.Register<ElementInfoPresenter, ElementInfoWindow>(WindowName.ElementInfo,
00096             SortingLayers.Popup);
00097     }
00098
00099 }
00100 }
```

7.252 FieldElementExtensions.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Configs.Elements;
00005 using MergeIt.Core.FieldElements;
00006
00007 namespace MergeIt.Game.Helpers
00008 {
00009     public static class FieldElementExtensions
0010     {
0011         public static ElementConfig GetNextInEvolution(this IFieldElement fieldElement)
0012         {
0013             ElementConfig elementConfig = fieldElement.ConfigParameters.ElementConfig;
```

```

00014     List<ElementConfig> evolutionChain = fieldElement.ConfigParameters.EvolutionData.Chain;
00015
00016     int index = evolutionChain.IndexOf(elementConfig);
00017     if (index == evolutionChain.Count - 1)
00018     {
00019         return null;
00020     }
00021
00022     index += 1;
00023
00024     return evolutionChain[index];
00025 }
00026
00027     public static ElementConfig GetPreviousInEvolution(this IFieldElement fieldElement)
00028     {
00029         ElementConfig elementConfig = fieldElement.ConfigParameters.ElementConfig;
00030         List<ElementConfig> evolutionChain = fieldElement.ConfigParameters.EvolutionData.Chain;
00031
00032         int index = evolutionChain.IndexOf(elementConfig);
00033         if (index == 0)
00034         {
00035             return null;
00036         }
00037
00038         index -= 1;
00039
00040         return evolutionChain[index];
00041     }
00042
00043     public static string GetDescription(this ElementConfig config)
00044     {
00045         string description = config.CommonSettings.Description;
00046
00047         if (config.CommonSettings.IsConsumable)
00048         {
00049             description = string.Format(description,
00050             config.CommonSettings.ConsumableSettings.Value);
00051         }
00052
00053         return description;
00054     }
00055 }

```

7.253 FieldGeneratorElementExtensions.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using System.Linq;
00006 using MergeIt.Core.Configs.Elements;
00007 using MergeIt.Core.Configs.Types;
00008 using MergeIt.Core.FieldElements;
00009
00010 namespace MergeIt.Game.Helpers
00011 {
00012     public static class FieldGeneratorElementExtensions
00013     {
00014         public static void UpdateGenerator(this IFieldElement generator)
00015         {
00016             if (generator.InfoParameters.Type != ElementType.Generator)
00017             {
00018                 return;
00019             }
00020
00021             ElementGeneratorSettings settings =
00022             generator.ConfigParameters.ElementConfig.GeneratorSettings;
00023             IGeneratorParameters parameters = generator.GeneratorParameters;
00024             IGeneratorOpenParameters openParameters = generator.GeneratorOpenParameters;
00025             long currentTime = DateTimeOffset.UtcNow.ToUnixTimeSeconds();
00026
00027             if (openParameters is { IsOpening: true })
00028             {
00029                 long openTimePassed = currentTime - openParameters.StartOpeningTime;
00030                 if (openTimePassed < settings.OpenTime)
00031                 {
00032                     openParameters.RemainingTime.Value = settings.OpenTime - openTimePassed;
00033
00034                 }
00035             }
00036             generator.GeneratorOpenParameters = null;

```

```
00037         }
00038
00039         if (parameters.StartChargingTime == 0)
00040     {
00041         return;
00042     }
00043
00044     int availableToDrop = parameters.AvailableToDrop;
00045     int maxPossibleDrops = !settings.IsUnlimited ? settings.MaxDrop * settings.Stack :
00046     settings.MaxDrop;
00047     int maxAvailableToDrop = maxPossibleDrops - parameters.DroppedElements;
00048     maxAvailableToDrop = Math.Min(maxAvailableToDrop, settings.MaxDrop);
00049     int minDrop = Math.Min(settings.MinDrop, maxAvailableToDrop);
00050
00051     long passedTime = currentTime - parameters.StartChargingTime;
00052     long restPassedTime = passedTime % settings.ChargeTime;
00053     int charged = (int)(passedTime / settings.ChargeTime) + parameters.ChargedCount;
00054     charged = Math.Min(charged, maxAvailableToDrop);
00055
00056     parameters.MinDropFullChargeTime = minDrop * settings.ChargeTime;
00057
00058     if (availableToDrop == 0)
00059     {
00060         if (charged >= minDrop)
00061         {
00062             availableToDrop = charged;
00063             parameters.ChargedCount = 0;
00064             parameters.StartChargingTime = currentTime - restPassedTime;
00065         }
00066         else
00067         {
00068             parameters.ChargedCount = charged;
00069         }
00070     }
00071     else
00072     {
00073         availableToDrop += charged;
00074         parameters.StartChargingTime = currentTime - restPassedTime;
00075         parameters.ChargedCount = 0;
00076     }
00077
00078     availableToDrop = Math.Min(availableToDrop, maxAvailableToDrop);
00079
00080     float restTime = settings.ChargeTime - restPassedTime;
00081     parameters.RemainChargeTime = restTime;
00082
00083     if (availableToDrop == 0)
00084     {
00085         parameters.MinDropChargeTime.Value = (minDrop - 1 - charged) * settings.ChargeTime +
00086         restTime;
00087     }
00088     else if (availableToDrop < maxAvailableToDrop)
00089     {
00090         parameters.MinDropChargeTime.Value = 0;
00091     }
00092     else
00093     {
00094         parameters.MinDropChargeTime.Value = 0;
00095         parameters.RemainChargeTime = 0;
00096         parameters.StartChargingTime = 0;
00097     }
00098
00099     parameters.AvailableToDrop = availableToDrop;
00100 }
00101
00102     public static bool? TryGenerate(this IFieldElement generator)
00103     {
00104         if (generator.InfoParameters.Type != ElementType.Generator)
00105         {
00106             return null;
00107         }
00108
00109         var settings = generator.ConfigParameters.ElementConfig.GeneratorSettings;
00110         var parameters = generator.GeneratorParameters;
00111         int maxPossibleDrops = !settings.IsUnlimited ? settings.MaxDrop * settings.Stack :
00112         settings.MaxDrop;
00113         parameters.AvailableToDrop--;
00114
00115         if (!settings.IsUnlimited)
00116         {
00117             parameters.DroppedElements++;
00118
00119             if (parameters.AvailableToDrop == 0 &&
00120                 parameters.DroppedElements >= maxPossibleDrops)
00121             {
00122                 return false;
00123             }
00124         }
00125     }
00126 }
```

```

00121         }
00122     }
00123
00124     int availableToDrop = parameters.AvailableToDrop;
00125     int maxAvailableToDrop = maxPossibleDrops - parameters.DroppedElements;
00126     maxAvailableToDrop = Math.Min(maxAvailableToDrop, settings.MaxDrop);
00127     int minDrop = Math.Min(settings.MinDrop, maxAvailableToDrop);
00128
00129     parameters.MinDropFullChargeTime = minDrop * settings.ChargeTime;
00130
00131     if (!parameters.Charging)
00132     {
00133         parameters.StartChargingTime = DateTimeOffset.UtcNow.UnixTimeSeconds();
00134         parameters.RemainChargeTime = settings.ChargeTime;
00135     }
00136
00137     if (availableToDrop == 0)
00138     {
00139         parameters.MinDropChargeTime.Value = (minDrop - 1 - parameters.ChargedCount) *
00140             settings.ChargeTime + parameters.RemainChargeTime;
00141     }
00142
00143     return true;
00144 }
00145
00146     public static void ChargeGenerator(this IFIELDElement generator, int count = 1)
00147     {
00148         if (generator.InfoParameters.Type != ElementType.Generator)
00149         {
00150             return;
00151         }
00152
00153         var settings = generator.ConfigParameters.ElementConfig.GeneratorSettings;
00154         var parameters = generator.GeneratorParameters;
00155
00156         ChargeGenerator(parameters, settings, charged: count);
00157     }
00158
00159     public static void SkipCharging(this IFIELDElement generator)
00160     {
00161         if (generator.InfoParameters.Type != ElementType.Generator)
00162         {
00163             return;
00164         }
00165
00166         var settings = generator.ConfigParameters.ElementConfig.GeneratorSettings;
00167         var parameters = generator.GeneratorParameters;
00168
00169         int maxPossibleDrops = !settings.IsUnlimited ? settings.MaxDrop * settings.Stack :
00170             settings.MaxDrop;
00171         int maxAvailableToDrop = maxPossibleDrops - parameters.DroppedElements;
00172         int chargedCount = Math.Min(settings.MinDrop, maxAvailableToDrop);
00173
00174         int timeDiff = settings.ChargeTime - (int)parameters.RemainChargeTime;
00175
00176         parameters.MinDropChargeTime.Value = 0;
00177         parameters.ChargedCount = 0;
00178         parameters.StartChargingTime = DateTimeOffset.UtcNow.UnixTimeSeconds() - timeDiff;
00179         parameters.AvailableToDrop = chargedCount;
00180     }
00181
00182     public static void TrySkipOpeningTime(this IFIELDElement generator, int seconds)
00183     {
00184         if (generator.InfoParameters.Type != ElementType.Generator)
00185         {
00186             return;
00187         }
00188
00189         var parameters = generator.GeneratorOpenParameters;
00190
00191         if (parameters is { IsOpening: true })
00192         {
00193             generator.GeneratorOpenParameters.RemainingTime.Value -= seconds;
00194         }
00195
00196     public static void TrySkipChargingTime(this IFIELDElement generator, int seconds)
00197     {
00198         if (generator.InfoParameters.Type != ElementType.Generator)
00199         {
00200             return;
00201         }
00202
00203         var settings = generator.ConfigParameters.ElementConfig.GeneratorSettings;
00204         var parameters = generator.GeneratorParameters;
00205
00206         if (!parameters.Charging)
00207         {
00208             parameters.StartChargingTime = DateTimeOffset.UtcNow.UnixTimeSeconds();
00209             parameters.RemainChargeTime = settings.ChargeTime;
00210         }
00211
00212         int availableToDrop = parameters.AvailableToDrop;
00213         int maxAvailableToDrop = maxPossibleDrops - parameters.DroppedElements;
00214         maxAvailableToDrop = Math.Min(maxAvailableToDrop, settings.MaxDrop);
00215         int minDrop = Math.Min(settings.MinDrop, maxAvailableToDrop);
00216
00217         parameters.MinDropFullChargeTime = minDrop * settings.ChargeTime;
00218
00219         if (availableToDrop == 0)
00220         {
00221             parameters.MinDropChargeTime.Value = (minDrop - 1 - parameters.ChargedCount) *
00222                 settings.ChargeTime + parameters.RemainChargeTime;
00223         }
00224
00225         return true;
00226     }
00227 }
```

```
00206         {
00207             return;
00208         }
00209
00210         int diff = (int)parameters.RemainChargeTime - seconds;
00211         parameters.MinDropChargeTime.Value -= seconds;
00212
00213         if (diff < 0)
00214         {
00215             int chargedCount = 1;
00216
00217             diff = Math.Abs(diff);
00218             int restCharged = diff / settings.ChargeTime;
00219             int restTime = diff % settings.ChargeTime;
00220
00221             chargedCount += restCharged;
00222
00223             ChargeGenerator(parameters, settings, chargedCount, restTime);
00224         }
00225         else
00226         {
00227             parameters.RemainChargeTime = diff;
00228             parameters.StartChargingTime = DateTimeOffset.UtcNow.ToUnixTimeSeconds() - seconds;
00229         }
00230     }
00231
00232     public static GeneratableElement GetCandidateToGenerate(this IFIELDElement generator)
00233     {
00234         if (generator.InfoParameters.Type != ElementType.Generator)
00235         {
00236             return null;
00237         }
00238
00239         ElementGeneratorSettings generatorSettings =
00240             generator.ConfigParameters.ElementConfig.GeneratorSettings;
00241         List<GeneratableElement> generatedItems = generatorSettings.GenerateItems;
00242         int sumPossibilities = generatedItems.Sum(x => x.Possibility);
00243         int random = UnityEngine.Random.Range(0, sumPossibilities);
00244
00245         int min = 0;
00246         for (int i = 0; i < generatedItems.Count; i++)
00247         {
00248             var item = generatedItems[i];
00249             int max = min + item.Possibility;
00250
00251             if (random >= min && random < max)
00252             {
00253                 return item;
00254             }
00255
00256             min = max;
00257         }
00258
00259         return null;
00260     }
00261
00262     private static void ChargeGenerator(IGeneratorParameters parameters, ElementGeneratorSettings
00263     settings, int charged = 1, int skipTime = 0)
00264     {
00265         parameters.ChargedCount += charged;
00266
00267         long currentTime = DateTimeOffset.UtcNow.ToUnixTimeSeconds();
00268
00269         int availableToDrop = parameters.AvailableToDrop;
00270         int maxPossibleDrops = !settings.IsUnlimited ? settings.MaxDrop * settings.Stack :
00271             settings.MaxDrop;
00272         int maxAvailableToDrop = maxPossibleDrops - parameters.DroppedElements;
00273         maxAvailableToDrop = Math.Min(maxAvailableToDrop, settings.MaxDrop);
00274         int minDrop = Math.Min(settings.MinDrop, maxAvailableToDrop);
00275
00276         parameters.RemainChargeTime = settings.ChargeTime - skipTime;
00277         parameters.MinDropFullChargeTime = minDrop * settings.ChargeTime;
00278         parameters.StartChargingTime = currentTime - skipTime;
00279
00280         if (availableToDrop == 0 && parameters.ChargedCount >= minDrop ||
00281             availableToDrop > 0)
00282         {
00283             availableToDrop += parameters.ChargedCount;
00284
00285             parameters.ChargedCount = 0;
00286         }
00287
00288         availableToDrop = Math.Min(availableToDrop, maxAvailableToDrop);
00289
00290         if (availableToDrop == maxAvailableToDrop)
00291         {
00292             parameters.StartChargingTime = 0;
```

```

00290         parameters.RemainChargeTime = 0;
00291         parameters.ChargedCount = 0;
00292     }
00293
00294     parameters.AvailableToDrop = availableToDrop;
00295 }
00296 }
00297 }
```

7.254 HudTargetComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Game.Enums;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Helpers
00007 {
00008     public class HudTargetComponent : MonoBehaviour
00009     {
00010         [SerializeField]
00011         private HudPanelType _hudPanelType;
00012
00013         private void Awake()
00014         {
00015             HudTargets.AddTarget(_hudPanelType, transform);
00016         }
00017     }
00018 }
```

7.255 HudTargets.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Game.Enums;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Game.Helpers
00008 {
00009     public static class HudTargets
00010     {
00011         private static Dictionary<HudPanelType, Transform> _hudTargets = new();
00012
00013         public static void AddTarget(HudPanelType hudPanelType, Transform transform)
00014         {
00015             _hudTargets[hudPanelType] = transform;
00016         }
00017
00018         public static Transform GetTarget(HudPanelType hudPanelType)
00019         {
00020             _hudTargets.TryGetValue(hudPanelType, out Transform target);
00021
00022             return target;
00023         }
00024     }
00025 }
```

7.256 EnergyComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections;
00004 using MergeIt.Core.Helpers;
00005 using MergeIt.Core.Messages;
00006 using MergeIt.Game.Enums;
00007 using MergeIt.Game.Messages;
00008 using MergeIt.Game.Services;
00009 using MergeIt.SimpleDI;
00010 using TMPro;
00011 using UnityEngine;
00012
00013 namespace MergeIt.Game.HUD
00014 {
00015     public class EnergyComponent : HudPanelBase
```

```
00016      {
00017          [SerializeField]
00018          private TMP_Text _value;
00019
00020          [SerializeField]
00021          private TMP_Text _timerLabel;
00022
00023          public override HudPanelType Type
00024          {
00025              get => HudPanelType.Energy;
00026          }
00027
00028          private IMESSAGEBUS _messageBus;
00029          private long _remainingTime;
00030          private Coroutine _timerUpdate;
00031
00032          protected override void Start()
00033          {
00034              base.Start();
00035
00036              _messageBus = DiContainer.Get<IMESSAGEBUS>();
00037              _messageBus.AddListener<StartChargingMessage>(OnStartChargingMessageHandler);
00038          }
00039
00040          private void OnDestroy()
00041          {
00042              UserServiceModel.Energy.Unsubscribe(OnEnergyChanged);
00043              _messageBus.RemoveListener<StartChargingMessage>(OnStartChargingMessageHandler);
00044          }
00045
00046          private void OnStartChargingMessageHandler(StartChargingMessage message)
00047          {
00048              if (message.RemainingTime != -1)
00049              {
00050                  _timerLabel.gameObject.SetActive(true);
00051
00052                  _remainingTime = message.RemainingTime;
00053
00054                  if (_timerUpdate != null)
00055                  {
00056                      StopCoroutine(_timerUpdate);
00057                  }
00058
00059                  _timerLabel.text = _remainingTime.FormatTime();
00060                  _timerUpdate = StartCoroutine(UpdateTimer());
00061              }
00062              else
00063              {
00064                  if (_timerUpdate != null)
00065                  {
00066                      StopCoroutine(_timerUpdate);
00067                  }
00068
00069                  _timerLabel.gameObject.SetActive(false);
00070              }
00071          }
00072
00073          protected override void OnApplyModel(UserServiceModel userServiceModel)
00074          {
00075              UserServiceModel.Energy.Subscribe(OnEnergyChanged, true);
00076          }
00077
00078          private void OnEnergyChanged(int current)
00079          {
00080              _value.text = $"{current}";
00081          }
00082
00083          private IEnumerator UpdateTimer()
00084          {
00085              while (_remainingTime > 0)
00086              {
00087                  yield return new WaitForSecondsRealtime(1f);
00088
00089                  _remainingTime -= 1;
00090                  _timerLabel.text = _remainingTime.FormatTime();
00091              }
00092
00093              _messageBus.Fire<EnergyRestoredMessage>();
00094          }
00095      }
00096 }
```

7.257 HardCurrencyComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Game.Enums;
00004 using MergeIt.Game.Services;
00005 using TMPro;
00006 using UnityEngine;
00007
00008 namespace MergeIt.Game.HUD
00009 {
00010     public class HardCurrencyComponent : HudPanelBase
00011     {
00012         [SerializeField]
00013         private TMP_Text _value;
00014
00015         public override HudPanelType Type
00016         {
00017             get => HudPanelType.HardCurrency;
00018         }
00019
00020         protected override void OnApplyModel(UserServiceModel userServiceModel)
00021         {
00022             userServiceModel.HardCurrency.Subscribe(OnHardCurrencyChanged, true);
00023         }
00024
00025         private void OnDestroy()
00026         {
00027             userServiceModel.HardCurrency.Unsubscribe(OnHardCurrencyChanged);
00028         }
00029
00030         private void OnHardCurrencyChanged(int current)
00031         {
00032             _value.text = $"{current}";
00033         }
00034     }
00035 }
```

7.258 HudComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004 using MergeIt.Core.Services;
00005 using MergeIt.Core.WindowSystem;
00006 using MergeIt.Game.Messages;
00007 using MergeIt.Game.Services;
00008 using MergeIt.SimpleDI;
00009 using UnityEngine;
00010
00011 namespace MergeIt.Game.HUD
00012 {
00013     public class HudComponent : MonoBehaviour
00014     {
00015         [SerializeField]
00016         private ProgressComponent _progressComponent;
00017
00018         [SerializeField]
00019         private SoftCurrencyComponent _softCurrencyComponent;
00020
00021         [SerializeField]
00022         private HardCurrencyComponent _hardCurrencyComponent;
00023
00024         [SerializeField]
00025         private EnergyComponent _energyComponent;
00026
00027         private UserServiceModel _userServiceModel;
00028         private IUserProgressService _userProgressService;
00029         private IMessageBus _messageBus;
00030
00031         private void Start()
00032         {
00033             _messageBus = DiContainer.Get<IMessageBus>();
00034             _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00035             _messageBus.AddListener<LevelUpdatedMessage>(OnLevelUpdatedMessageHandler);
00036         }
00037
00038         private void OnDestroy()
00039         {
00040             _messageBus.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00041             _messageBus.RemoveListener<LevelUpdatedMessage>(OnLevelUpdatedMessageHandler);
00042         }
00043     }
00044 }
```

```

00044     private void OnLoadedGameMessageHandler(LoadedGameMessage message)
00045     {
00046         _userServiceModel = DiContainer.Get<UserServiceModel>();
00047         _userProgressService = DiContainer.Get<IUserProgressService>();
00048         var windowSystem = DiContainer.Get<IWindowSystem>();
00049
00050         int maxExp = _userProgressService.GetCurrentLevelMaxExp();
00051         _progressComponent.Initialize(windowSystem);
00052         _progressComponent.SetMaxProgress(maxExp);
00053         _progressComponent.ApplyModel(_userServiceModel);
00054         _progressComponent.UpdateProgress();
00055
00056         _energyComponent.ApplyModel(_userServiceModel);
00057         _softCurrencyComponent.ApplyModel(_userServiceModel);
00058         _hardCurrencyComponent.ApplyModel(_userServiceModel);
00059     }
00060
00061     private void OnLevelUpdatedMessageHandler(LevelUpdatedMessage message)
00062     {
00063         _progressComponent.SetMaxProgress(message.NextLevelExp);
00064         _progressComponent.UpdateProgress();
00065     }
00066 }
00067 }
```

7.259 HudPanelBase.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Game.Enums;
00004 using MergeIt.Game.Helpers;
00005
00006 namespace MergeIt.Game.HUD
00007 {
00008     public abstract class HudPanelBase : UserListenerComponent, IHudPanel
00009     {
00010         public virtual HudPanelType Type { get; }
00011
00012         protected virtual void Start()
00013         {
00014             HudTargets.AddTarget(Type, transform);
00015         }
00016     }
00017 }
```

7.260 IHudPanel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Game.Enums;
00004
00005 namespace MergeIt.Game.HUD
00006 {
00007     public interface IHudPanel
00008     {
00009         HudPanelType Type { get; }
00010     }
00011
00012 }
```

7.261 IUserListener.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Game.Services;
00004
00005 namespace MergeIt.Game.HUD
00006 {
00007     public interface IUserListener
00008     {
00009         void ApplyModel(UserServiceModel userServiceModel);
00010     }
00011 }
```

7.262 ProgressComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Helpers;
00004 using MergeIt.Core.WindowSystem;
00005 using MergeIt.Game.Enums;
00006 using MergeIt.Game.Services;
00007 using MergeIt.Game.Windows.UserProgress;
00008 using TMPro;
00009 using UnityEngine;
00010 using UnityEngine.UI;
00011
00012 namespace MergeIt.Game.HUD
00013 {
00014     public class ProgressComponent : HudPanelBase
00015     {
00016         [SerializeField]
00017         private TMP_Text _levelValue;
00018
00019         [SerializeField]
00020         private SlicedFilledImage _progressImage;
00021
00022         [SerializeField]
00023         private GameObject _giftImageContainer;
00024
00025         [SerializeField]
00026         private Button _progressButton;
00027
00028         public override HudPanelType Type
00029         {
00030             get => HudPanelType.UserProgress;
00031         }
00032
00033         private int _nextExperience;
00034         private float _prevProgress;
00035         private float _nextProgress;
00036         private bool _animate;
00037         private float _tempProgress;
00038
00039         private IWindowSystem _windowSystem;
00040
00041         protected override void Start()
00042         {
00043             base.Start();
00044
00045             _progressButton.onClick.AddListener(OnProgressButtonClick);
00046         }
00047
00048         public void OnDestroy()
00049         {
00050             _progressButton.onClick.RemoveListener(OnProgressButtonClick);
00051
00052             UserServiceModel.Level.Unsubscribe(OnLevelChanged);
00053             UserServiceModel.Experience.Unsubscribe(OnExperienceChanged);
00054         }
00055
00056         public void Initialize(IWindowSystem windowSystem)
00057         {
00058             _windowSystem = windowSystem;
00059         }
00060
00061         public void SetMaxProgress(int nextExperience)
00062         {
00063             _nextExperience = nextExperience;
00064         }
00065
00066         public void UpdateProgress()
00067         {
00068             SetProgress(_nextExperience, UserServiceModel.Experience.Value, false);
00069         }
00070
00071         protected override void OnApplyModel(UserServiceModel userServiceModel)
00072         {
00073             UserServiceModel.Level.Subscribe(OnLevelChanged, true);
00074             UserServiceModel.Experience.Subscribe(OnExperienceChanged, true);
00075         }
00076
00077
00078         private void OnProgressButtonClick()
00079         {
00080             _windowSystem.OpenWindow<UserInfoPresenter>(enableBlackout: true);
00081         }
00082
00083         private void OnLevelChanged(int value)
00084         {
00085             _levelValue.text = $"{value}";

```

```

00086         }
00087
00088     private void OnExperienceChanged(int prev, int current)
00089     {
00090         SetProgress(prev, current, true);
00091     }
00092
00093     private void SetProgress(int prev, int current, bool animated)
00094     {
00095         if (animated)
00096         {
00097             _prevProgress = prev / (float)_nextExperience;
00098             _nextProgress = current / (float)_nextExperience;
00099             _tempProgress = _prevProgress;
00100             _animate = true;
00101         }
00102         else
00103         {
00104             _progressImage.fillAmount = current / (float)_nextExperience;
00105         }
00106     }
00107
00108     private void Update()
00109     {
00110         if (_animate)
00111         {
00112             _tempProgress += Time.deltaTime;
00113
00114             if (_tempProgress >= _nextProgress)
00115             {
00116                 _tempProgress = 0f;
00117                 _progressImage.fillAmount = _nextProgress;
00118                 _animate = false;
00119             }
00120             else
00121             {
00122                 _progressImage.fillAmount = _tempProgress;
00123             }
00124         }
00125     }
00126 }
00127 }
```

7.263 SoftCurrencyComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Game.Enums;
00004 using MergeIt.Game.Services;
00005 using TMPro;
00006 using UnityEngine;
00007
00008 namespace MergeIt.Game.HUD
00009 {
0010     public class SoftCurrencyComponent : HudPanelBase
0011     {
0012         [SerializeField]
0013         private TMP_Text _value;
0014
0015         public override HudPanelType Type
0016         {
0017             get => HudPanelType.SoftCurrency;
0018         }
0019
0020         protected override void OnApplyModel(UserServiceModel userServiceModel)
0021         {
0022             userServiceModel.SoftCurrency.Subscribe(OnSoftCurrencyChanged, true);
0023         }
0024
0025         private void OnDestroy()
0026         {
0027             userServiceModel.SoftCurrency.Unsubscribe(OnSoftCurrencyChanged);
0028         }
0029
0030         private void OnSoftCurrencyChanged(int current)
0031         {
0032             _value.text = $"{current}";
0033         }
0034     }
0035 }
```

7.264 UserListenerComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Game.Services;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.HUD
00007 {
00008     public abstract class UserListenerComponent : MonoBehaviour, IUserListener
00009     {
00010         protected UserServiceModel UserServiceModel { get; private set; }
00011
00012         public void ApplyModel(UserServiceModel userServiceModel)
00013         {
00014             UserServiceModel = userServiceModel;
00015
00016             OnApplyModel(userServiceModel);
00017         }
00018
00019         protected abstract void OnApplyModel(UserServiceModel userServiceModel);
00020     }
00021 }
```

7.265 HintsManager.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using MergeIt.Core.Configs.Elements;
00006 using MergeIt.Core.FieldElements;
00007 using MergeIt.Core.Helpers;
00008 using MergeIt.Core.Messages;
00009 using MergeIt.Core.Services;
00010 using MergeIt.Game.Field;
00011 using MergeIt.Game.Helpers;
00012 using MergeIt.Game.Messages;
00013 using MergeIt.SimpleDI;
00014 using MergeIt.SimpleDI.ReservedInterfaces;
00015 using UnityEngine;
00016
00017 namespace MergeIt.Game.Managers
00018 {
00019     public class HintsManager : IHintsManager, IInitializable, IDisposable, IUpdatable
00020     {
00021         [Introduce] private IMessageBus _messageBus;
00022
00023         [Introduce] private FieldLogicModel _fieldLogicModel;
00024
00025         [Introduce] private IConfigsService _configsService;
00026
00027         private float _time;
00028         private bool _animationInProcess;
00029         private bool _active;
00030
00031         private IFIELDElementPresenter _candidate1;
00032         private IFIELDElementPresenter _candidate2;
00033         private IList<int> _randomHeight;
00034         private IList<int> _randomWidth;
00035         private Dictionary<ElementConfig, List<IFIELDElement>> _sameElements = new();
00036
00037         private bool Available => _active && !_animationInProcess;
00038
00039         public void Initialize()
00040         {
00041             _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00042             _messageBus.AddListener<ActivateHintsMessage>(OnActivateHintsMessageHandler);
00043             _messageBus.AddListener<ResetHintsMessage>(OnResetHintsMessageHandler);
00044         }
00045
00046         public void Dispose()
00047         {
00048             _messageBus.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00049             _messageBus.RemoveListener<ActivateHintsMessage>(OnActivateHintsMessageHandler);
00050             _messageBus.RemoveListener<ResetHintsMessage>(OnResetHintsMessageHandler);
00051         }
00052
00053         public void Update()
00054         {
00055             if (Available && _candidate1 == null && _candidate2 == null)
00056             {
00057                 _time += Time.deltaTime;
00058             }
00059         }
00060     }
00061 }
```

```
00058
00059     if (_time >= _configsService.HintsConfig.Interval)
00060     {
00061         _time = 0f;
00062         FindCandidates();
00063     }
00064 }
00065 else if (_candidate1 != null && _candidate2 != null)
00066 {
00067     if (_candidate1.State == FieldElementState.Idle &&
00068         _candidate2.State == FieldElementState.Idle)
00069     {
00070         _candidate1 = null;
00071         _candidate2 = null;
00072
00073         _animationInProcess = false;
00074     }
00075 }
00076
00077
00078 private void OnLoadedGameMessageHandler(LoadedGameMessage _)
00079 {
00080     _randomHeight = ListExtensions.GenerateShuffledArray(_fieldLogicModel.FieldHeight);
00081     _randomWidth = ListExtensions.GenerateShuffledArray(_fieldLogicModel.FieldWidth);
00082
00083     _active = true;
00084     _animationInProcess = false;
00085 }
00086
00087 private void OnActivateHintsMessageHandler(ActivateHintsMessage message)
00088 {
00089     _active = message.Active;
00090
00091     if (!_active)
00092     {
00093         ResetHint();
00094     }
00095 }
00096
00097 private void OnResetHintsMessageHandler(ResetHintsMessage message)
00098 {
00099     ResetHint();
00100 }
00101
00102 private void ResetHint()
00103 {
00104     _candidate1?.SetState(FieldElementState.Idle);
00105     _candidate2?.SetState(FieldElementState.Idle);
00106
00107     _candidate1 = null;
00108     _candidate2 = null;
00109
00110     _animationInProcess = false;
00111 }
00112
00113 private void FindCandidates()
00114 {
00115     _sameElements.Clear();
00116
00117     _randomHeight.Shuffle();
00118     _randomWidth.Shuffle();
00119
00120     List<IFieldElement> candidates = null;
00121
00122     for (int i = 0; i < _randomHeight.Count; i++)
00123     {
00124         for (int j = 0; j < _randomWidth.Count; j++)
00125         {
00126             int row = _randomHeight[i];
00127             int column = _randomWidth[j];
00128
00129             GridPoint point = GridPoint.Create(row, column);
00130             if (_fieldLogicModel.FieldElements.TryGetValue(point, out IFIELDElement
fieldElement))
00131             {
00132                 ElementConfig config = fieldElement.ConfigParameters.ElementConfig;
00133                 if (fieldElement.GetNextInEvolution() != null)
00134                 {
00135                     if (!_sameElements.ContainsKey(config))
00136                     {
00137                         _sameElements[config] = new List<IFieldElement>();
00138                     }
00139
00140                     var list = _sameElements[config];
00141
00142                     if (list.Exists(x => x.InfoParameters.IsBlocked) &&
fieldElement.InfoParameters.IsBlocked)
00143
```

```

00144         {
00145             continue;
00146         }
00147         list.Add(fieldElement);
00148     }
00149     if (list.Count > 1)
00150     {
00151         candidates = list;
00152         break;
00153     }
00154 }
00155 }
00156 }
00157 }
00158 }
00159 if (candidates != null)
00160 {
00161     for (var i = 0; i < 2; i++)
00162     {
00163         IFieldElement fieldElement = candidates[i];
00164         FieldCellComponent cell =
00165             _fieldLogicModel.CellComponents[fieldElement.InfoParameters.LogicPosition];
00166         IFieldElementPresenter candidate = cell.FieldElementPresenter;
00167
00168         if (_candidate1 != null)
00169         {
00170             _candidate2 = candidate;
00171         }
00172         else
00173         {
00174             _candidate1 = candidate;
00175         }
00176
00177         _animationInProgress = true;
00178         candidate.SetState(FieldElementState.Hint);
00179     }
00180 }
00181 }
00182 }
00183 }
00184 }
```

7.266 IHintsManager.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Game.Managers
00004 {
00005     public interface IHintsManager
00006     {
00007     }
00008 }
```

7.267 ActivateHintsMessage.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class ActivateHintsMessage : IMessage
00008     {
00009         public bool Active { get; set; }
00010     }
00011 }
```

7.268 CheckGeneratorMessage.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
```

```
00007 {  
00008     public class CheckGeneratorMessage : IMessage  
00009     {  
0010         public GridPoint GeneratorPoint { get; set; }  
0011     }  
0012 }
```

7.269 ClickElementMessage.cs

```
00001 // Copyright (c) 2024, Awessets  
00002  
00003 using MergeIt.Core.Messages;  
00004 using MergeIt.Game.Field;  
00005  
00006 namespace MergeIt.Game.Messages  
00007 {  
00008     public class ClickElementMessage : IMessage  
00009     {  
0010         public FieldCellComponent Cell { get; set; }  
0011     }  
0012 }
```

7.270 CreateElementMessage.cs

```
00001 // Copyright (c) 2024, Awessets  
00002  
00003 using MergeIt.Core.FieldElements;  
00004 using MergeIt.Core.Messages;  
00005 using UnityEngine;  
00006  
00007 namespace MergeIt.Game.Messages  
00008 {  
00009     public class CreateElementMessage : IMessage  
00010     {  
0011         public IFIELDElement NewElement { get; set; }  
0012         public Vector3? FromPosition { get; set; }  
0013         public GridPoint ToPoint { get; set; }  
0014     }  
0015  
0016 }
```

7.271 ElementActionMessage.cs

```
00001 // Copyright (c) 2024, Awessets  
00002  
00003 using MergeIt.Core.FieldElements;  
00004 using MergeIt.Core.Messages;  
00005 using MergeIt.Game.UI.InfoPanel;  
00006  
00007 namespace MergeIt.Game.Messages  
00008 {  
00009     public class ElementActionMessage : IMessage  
00010     {  
0011         public IFIELDElement Element { get; set; }  
0012         public ElementActionType ActionType { get; set; }  
0013     }  
0014 }
```

7.272 EndDragElementMessage.cs

```
00001 // Copyright (c) 2024, Awessets  
00002  
00003 using MergeIt.Core.FieldElements;  
00004 using MergeIt.Core.Messages;  
00005 using UnityEngine;  
00006  
00007 namespace MergeIt.Game.Messages  
00008 {  
00009     public class EndDragElementMessage : IMessage  
00010     {  
0011         public GridPoint FromPoint { get; set; }  
0012         public GameObject ToGameObject { get; set; }  
0013         public Vector2 Position { get; set; }  
0014     }  
0015 }
```

7.273 EnergyRestoredMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class EnergyRestoredMessage : IMessage
00008     {
00009     }
00010 }
```

7.274 FieldFullMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class FieldFullMessage : IMessage
00008     {
00009     }
00010 }
```

7.275 GeneratorBusyMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class GeneratorBusyMessage : IMessage
00009     {
00010         public GridPoint Point { get; set; }
00011     }
00012 }
```

7.276 GeneratorOpenedMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class GeneratorOpenedMessage : IMessage
00009     {
00010         public GridPoint GeneratorPoint { get; set; }
00011     }
00012 }
```

7.277 GeneratorOpenStartMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class GeneratorOpenStartMessage : IMessage
00009     {
00010         public GridPoint GeneratorPoint { get; set; }
00011     }
00012 }
```

7.278 GeneratorRestoredMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class GeneratorRestoredMessage : IMessage
00009     {
00010         public GridPoint GeneratorPoint { get; set; }
00011     }
00012 }
```

7.279 LevelUpdatedMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class LevelUpdatedMessage : IMessage
00008     {
00009         public int NextLevelExp { get; set; }
00010     }
00011 }
```

7.280 LevelUpMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class LevelUpMessage : IMessage
00008     {
00009     }
00010 }
```

7.281 LoadedGameMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class LoadedGameMessage : IMessage
00008     {
00009     }
00010
00011 }
```

7.282 MergeElementsMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class MergeElementsMessage : IMessage
00009     {
00010         public GridPoint From { get; set; }
00011         public IFIELDElement NewElement { get; set; }
00012     }
00013
00014 }
```

7.283 MoveElementMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class MoveElementMessage : IMessage
00009     {
0010         public IFIELDVIEW FieldElementView { get; set; }
0011         public GridPoint FromPoint { get; set; }
0012         public GridPoint ToPoint { get; set; }
0013     }
0014 }
```

7.284 PrepareGameMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class PrepareGameMessage : IMessage
00008     {
00009     }
0010 }
```

7.285 RemoveElementMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class RemoveElementMessage : IMessage
00009     {
0010         public GridPoint RemoveAtPoint { get; set; }
0011     }
0012 }
```

7.286 RemoveFromInventoryMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class RemoveFromInventoryMessage : IMessage
00009     {
0010         public IFIELDELEMENT FieldElement { get; set; }
0011     }
0012 }
```

7.287 ResetHintsMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class ResetHintsMessage : IMessage
00008     {
00009     }
0010 }
```

7.288 ResetPositionMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class ResetPositionMessage : IMessage
00009     {
00010         public GridPoint From { get; set; }
00011     }
00012 }
```

7.289 ResetSelectionMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class ResetSelectionMessage : IMessage
00008     {
00009     }
00010 }
```

7.290 SelectedElementMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class SelectedElementMessage : IMessage
00009     {
00010         public GridPoint Point { get; set; }
00011     }
00012 }
```

7.291 SkipTimeMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class SkipTimeMessage : IMessage
00008     {
00009         public int Seconds { get; set; }
00010     }
00011 }
```

7.292 SplitElementMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class SplitElementMessage : IMessage
00009     {
00010         public IFIELDElement SplitElement1 { get; set; }
00011         public IFIELDElement SplitElement2 { get; set; }
00012     }
00013 }
```

7.293 StartChargingMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class StartChargingMessage : IMessage
00008     {
00009         public long RemainingTime { get; set; }
00010     }
00011 }
00012 }
```

7.294 StartGameMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class StartGameMessage : IMessage
00008     {
00009     }
00010 }
00011 }
```

7.295 SwapElementsMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class SwapElementsMessage : IMessage
00009     {
00010         public GridPoint From { get; set; }
00011         public GridPoint To { get; set; }
00012     }
00013 }
```

7.296 UnlockElementMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005
00006 namespace MergeIt.Game.Messages
00007 {
00008     public class UnlockElementMessage : IMessage
00009     {
00010         public IFIELDELEMENT Element { get; set; }
00011     }
00012 }
```

7.297 UpdateStockMessage.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004
00005 namespace MergeIt.Game.Messages
00006 {
00007     public class UpdateStockMessage : IMessage
00008     {
00009     }
00010 }
```

7.298 CurrencyService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.Configs.Types;
00005 using MergeIt.Core.Services;
00006 using MergeIt.SimpleDI;
00007
00008 namespace MergeIt.Game.Services
00009 {
00010     public class CurrencyService : ICurrencyService
00011     {
00012         [Introduce]
00013         private UserModel _userServiceModel;
00014
00015         public bool TryPay(CurrencySettings currencySettings)
00016         {
00017             bool result = false;
00018             switch (currencySettings.Currency)
00019             {
00020                 case CurrencyType.Soft:
00021                     if (_userServiceModel.SoftCurrency.Value > currencySettings.Amount)
00022                     {
00023                         _userServiceModel.SoftCurrency.Value -= currencySettings.Amount;
00024                         result = true;
00025                     }
00026
00027                     break;
00028
00029                 case CurrencyType.Hard:
00030                     if (_userServiceModel.HardCurrency.Value > currencySettings.Amount)
00031                     {
00032                         _userServiceModel.HardCurrency.Value -= currencySettings.Amount;
00033                         result = true;
00034                     }
00035
00036                     break;
00037
00038                 case CurrencyType.Splitter:
00039                     if (_userServiceModel.Splitters.Value > currencySettings.Amount)
00040                     {
00041                         _userServiceModel.Splitters.Value -= currencySettings.Amount;
00042                         result = true;
00043                     }
00044
00045                     break;
00046
00047                 case CurrencyType.Energy:
00048                     if (_userServiceModel.Energy.Value > currencySettings.Amount)
00049                     {
00050                         _userServiceModel.Energy.Value -= currencySettings.Amount;
00051                         result = true;
00052                     }
00053
00054                     break;
00055             }
00056
00057             return result;
00058         }
00059
00060         public void Sell(CurrencySettings currencySettings)
00061         {
00062             switch (currencySettings.Currency)
00063             {
00064                 case CurrencyType.Soft:
00065                     _userServiceModel.SoftCurrency.Value += currencySettings.Amount;
00066                     break;
00067
00068                 case CurrencyType.Hard:
00069                     _userServiceModel.HardCurrency.Value += currencySettings.Amount;
00070                     break;
00071
00072                 case CurrencyType.Splitter:
00073                     _userServiceModel.Splitters.Value += currencySettings.Amount;
00074                     break;
00075
00076                 case CurrencyType.Energy:
00077                     _userServiceModel.Energy.Value += currencySettings.Amount;
00078                     break;
00079             }
00080         }
00081     }
00082 }
```

7.299 ElementService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Core.Messages;
00006 using MergeIt.Core.Saves;
00007 using MergeIt.Core.Services;
00008 using MergeIt.Game.Factories.FieldElement;
00009 using MergeIt.Game.Field;
00010 using MergeIt.Game.Helpers;
00011 using MergeIt.Game.Messages;
00012 using MergeIt.SimpleDI;
00013
00014 namespace MergeIt.Game.Services
00015 {
00016     public class ElementService : IELEMENTService
00017     {
00018         [Introduce]
00019         private ICurrencyService _currencyService;
00020
00021         [Introduce]
00022         private IFieldElementFactory _fieldElementFactory;
00023
00024         [Introduce]
00025         private FieldLogicModel _fieldLogicModel;
00026
00027         [Introduce]
00028         private IGameFieldService _gameFieldService;
00029
00030         [Introduce]
00031         private IMessageBus _messageBus;
00032
00033         [Introduce]
00034         private IGameSaveService _saveService;
00035
00036         [Introduce]
00037         private UserServiceModel _userServiceModel;
00038
00039         public void TrySell(IFIELDElement fieldElement)
00040         {
00041             CurrencySettings costSettings =
00042                 fieldElement.ConfigParameters.ElementConfig.CommonSettings.SellCostSettings;
00043
00044             _currencyService.Sell(costSettings);
00045
00046             GridPoint point = fieldElement.InfoParameters.LogicPosition;
00047
00048             var removeMessage = new RemoveElementMessage {RemoveAtPoint = point};
00049             _messageBus.Fire(removeMessage);
00050             _messageBus.Fire<ResetSelectionMessage>();
00051
00052             _saveService.Save(GameSaveType.Field);
00053         }
00054
00055         public void TryUnlock(IFIELDElement fieldElement)
00056         {
00057             CurrencySettings costSettings =
00058                 fieldElement.ConfigParameters.ElementConfig.CommonSettings.UnlockCostSettings;
00059
00060             if (_currencyService.TryPay(costSettings))
00061             {
00062                 fieldElement.InfoParameters.IsBlocked = false;
00063
00064                 _saveService.Save(GameSaveType.Field);
00065
00066                 var unlockMessage = new UnlockElementMessage {Element = fieldElement};
00067                 _messageBus.Fire(unlockMessage);
00068             }
00069         }
00070
00071         public void TrySplit(IFIELDElement element)
00072         {
00073             var freeCellPoint = _gameFieldService.GetFreeCell();
00074             if (freeCellPoint == null)
00075             {
00076                 return;
00077             }
00078
00079             GridPoint secondPoint = freeCellPoint.Value;
00080
00081             var splitResult = TrySplit(element, secondPoint);
00082             if (splitResult != null)
00083             {
00084                 CurrencySettings costSettings =
00085                     element.ConfigParameters.ElementConfig.CommonSettings.SplitCostSettings;

```

```

00083             if (_currencyService.TryPay(costSettings))
00084             {
00085                 SendSplitMessage(splitResult.Value);
00086 
00087                 _saveService.Save(GameSaveType.Field);
00088                 _messageBus.Fire<ResetSelectionMessage>();
00089             }
00090         }
00091     }
00092 }
00093 
00094     private (IFieldElement element1, IFIELDelement element2)? TrySplit(IFIELDelement fieldElement,
00095     GridPoint point2)
00096     {
00097         ElementConfig previousInEvolution = fieldElement.GetPreviousInEvolution();
00098         if (previousInEvolution)
00099         {
00100             IFIELDelement splitElement1 =
00101                 _fieldElementFactory.CreateFieldElement(previousInEvolution,
00102                 fieldElement.InfoParameters.LogicPosition);
00103             IFIELDelement splitElement2 =
00104                 _fieldElementFactory.CreateFieldElement(previousInEvolution, point2);
00105 
00106             return (splitElement1, splitElement2);
00107         }
00108     }
00109 
00110     private void SendSplitMessage(IFIELDelement element1, IFIELDelement element2) splitResult)
00111     {
00112         var splitResultMessage = new SplitElementMessage
00113         {
00114             SplitElement1 = splitResult.element1,
00115             SplitElement2 = splitResult.element2
00116         };
00117 
00118         _messageBus.Fire(splitResultMessage);
00119     }
00120 }

```

7.300 ElementsStockService.cs

```

00001 // Copyright (c) 2024, Awessets
00002 
00003 using System.Linq;
00004 using MergeIt.Core.Configs.Elements;
00005 using MergeIt.Core.ElementsStock;
00006 using MergeIt.Core.Messages;
00007 using MergeIt.Core.Saves;
00008 using MergeIt.Core.Services;
00009 using MergeIt.Game.ElementsStock;
00010 using MergeIt.SimpleDI;
00011 
00012 namespace MergeIt.Game.Services
00013 {
00014     public class ElementsStockService : IElementsStockService
00015     {
00016         [Introduce]
00017         private IConfigsService _configsService;
00018 
00019         private ElementConfig _currentElement;
00020 
00021         [Introduce]
00022         private IMessageBus _messageBus;
00023 
00024         [Introduce]
00025         private IGameSaveService _saveService;
00026 
00027         [Introduce]
00028         private ElementsStockServiceModel _serviceModel;
00029 
00030         public void CreateStock()
00031         {
00032             var stockData = new ElementsStockData();
00033             SetupStock(stockData);
00034 
00035             _saveService.Save(GameSaveType.Stock);
00036         }
00037 
00038         public void SetupStock(IElementsStockData stockData)
00039         {

```

```

00040         SetModel(stockData);
00041     }
00042
00043     public void Add(ElementConfig elementConfig)
00044     {
00045         bool stockIsEmpty = !_serviceModel.Elements.Any();
00046         _serviceModel.Elements.Enqueue(elementConfig);
00047
00048         if (stockIsEmpty)
00049         {
00050             _currentElement = elementConfig;
00051         }
00052
00053         _saveService.Save(GameSaveType.Stock);
00054     }
00055
00056     public bool Remove()
00057     {
00058         return _serviceModel.Elements.TryDequeue(out_);
00059     }
00060
00061     public ElementConfig GetNext()
00062     {
00063         _serviceModel.Elements.TryPeek(out ElementConfig config);
00064         _currentElement = config;
00065         return _currentElement;
00066     }
00067
00068     public ElementConfig GetCurrent()
00069     {
00070         return _currentElement;
00071     }
00072
00073     public IElementsStockData GetData()
00074     {
00075         var data = new ElementsStockData();
00076
00077         if (_serviceModel.Elements.Count > 0)
00078         {
00079             data.Elements = _serviceModel.Elements.Select(x => x.Id).ToArray();
00080         }
00081
00082         return data;
00083     }
00084
00085     private void SetModel(IElementsStockData data)
00086     {
00087         if (data?.Elements != null)
00088         {
00089             foreach (string configId in data.Elements)
00090             {
00091                 ElementConfig config = _configsService.GetConfig(configId);
00092                 _serviceModel.Elements.Enqueue(config);
00093             }
00094         }
00095     }
00096 }
00097 }
```

7.301 ElementsStockServiceModel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Configs.Elements;
00005
00006 namespace MergeIt.Game.Services
00007 {
00008     public class ElementsStockServiceModel
00009     {
00010         public Queue<ElementConfig> Elements { get; } = new();
00011     }
00012 }
```

7.302 EnergyService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
```

```
00004 using MergeIt.Core.Configs;
00005 using MergeIt.Core.Messages;
00006 using MergeIt.Core.Saves;
00007 using MergeIt.Core.Services;
00008 using MergeIt.Game.Messages;
00009 using MergeIt.SimpleDI;
00010 using MergeIt.SimpleDI.ReservedInterfaces;
00011 using UnityEngine;
00012
00013 namespace MergeIt.Game.Services
00014 {
00015     public class EnergyService : IEnergyService, IInitializable, IDisposable
00016     {
00017         private GameConfig _config;
00018
00019         [Introduce]
00020         private IConfigsService _configsService;
00021
00022         [Introduce]
00023         private IGameSaveService _gameSaveService;
00024
00025         [Introduce]
00026         private IMessageBus _messageBus;
00027
00028         private int _skipSeconds;
00029
00030         [Introduce]
00031         private UserServiceModel _userServiceModel;
00032
00033         public void Dispose()
00034         {
00035             _messageBus.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00036             _messageBus.RemoveListener<EnergyRestoredMessage>(OnEnergyRestoredMessageHandler);
00037             _messageBus.RemoveListener<SkipTimeMessage>(OnSkipTimeMessageHandler);
00038
00039             _userServiceModel.Energy.Unsubscribe(OnEnergyChanged);
00040         }
00041
00042         public void Initialize()
00043         {
00044             _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00045             _messageBus.AddListener<EnergyRestoredMessage>(OnEnergyRestoredMessageHandler);
00046             _messageBus.AddListener<SkipTimeMessage>(OnSkipTimeMessageHandler);
00047         }
00048
00049         private void OnLoadedGameMessageHandler(LoadedGameMessage message)
00050         {
00051             _config = _configsService.GameConfig;
00052             _userServiceModel.Energy.Subscribe(OnEnergyChanged, true);
00053         }
00054
00055         private void OnEnergyRestoredMessageHandler(EnergyRestoredMessage restoredMessage)
00056         {
00057             int newEnergy = _userServiceModel.Energy.Value + 1;
00058             newEnergy = Mathf.Clamp(newEnergy, 0, _config.EnergyCap);
00059
00060             _userServiceModel.EnergyRestoringStartTime = -1;
00061             _userServiceModel.Energy.Value = newEnergy;
00062         }
00063
00064         private void OnSkipTimeMessageHandler(SkipTimeMessage message)
00065         {
00066             if (_userServiceModel.EnergyRestoringStartTime != -1)
00067             {
00068                 _skipSeconds = message.Seconds;
00069                 int energyRestoredPoints = _skipSeconds / _config.EnergyRestoreTime;
00070                 _skipSeconds %= _config.EnergyRestoreTime;
00071
00072                 int energyRestored = energyRestoredPoints + _userServiceModel.Energy.Value;
00073                 energyRestored = Mathf.Clamp(energyRestored, 0, _config.EnergyCap);
00074                 _userServiceModel.Energy.Value = energyRestored;
00075
00076                 if (energyRestoredPoints == 0)
00077                 {
00078                     OnEnergyChanged(energyRestored);
00079                 }
00080             }
00081         }
00082
00083         private void OnEnergyChanged(int value)
00084         {
00085             if (value < _config.EnergyCap)
00086             {
00087                 if (_userServiceModel.EnergyRestoringStartTime == -1)
00088                 {
00089                     _userServiceModel.EnergyRestoringStartTime =
00090                         DateTimeOffset.UtcNow.ToUnixTimeSeconds();
```

```

00090         }
00091         else
00092         {
00093             _userServiceModel.EnergyRestoringStartTime -= _skipSeconds;
00094         }
00095     }
00096     else
00097     {
00098         _userServiceModel.EnergyRestoringStartTime = -1;
00099         _skipSeconds = 0;
00100     }
00101
00102     CheckTime();
00103
00104     _gameSaveService.Save(GameSaveType.User);
00105 }
00106
00107     private void CheckTime()
00108     {
00109         if (_userServiceModel.EnergyRestoringStartTime > -1)
00110         {
00111             long currentTime = DateTimeOffset.UtcNow.ToUnixTimeSeconds();
00112             long diffTime = currentTime - _userServiceModel.EnergyRestoringStartTime;
00113             long remainingTime = _config.EnergyRestoreTime - diffTime;
00114
00115             SendStartTimer(remainingTime);
00116         }
00117         else
00118         {
00119             SendStartTimer(-1);
00120         }
00121     }
00122
00123     private void SendStartTimer(long remainingTime)
00124     {
00125         var message = new StartChargingMessage
00126         {
00127             RemainingTime = remainingTime
00128         };
00129
00130         _messageBus.Fire(message);
00131     }
00132 }
00133 }

```

7.303 EvolutionsService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using System.Linq;
00005 using MergeIt.Core.Configs.Data;
00006 using MergeIt.Core.Configs.Elements;
00007 using MergeIt.Core.Evolutions;
00008 using MergeIt.Core.Messages;
00009 using MergeIt.Core.Saves;
00010 using MergeIt.Core.Schemes;
00011 using MergeIt.Core.Services;
00012 using MergeIt.Game.Evolutions;
00013 using MergeIt.SimpleDI;
00014
00015 namespace MergeIt.Game.Services
00016 {
00017     public class EvolutionsService : IEvolutionsService
00018     {
00019         private readonly EvolutionsServiceModel _serviceModel = new();
00020
00021         [Introduce]
00022         private IConfigsService _configsService;
00023
00024         [Introduce]
00025         private IMessageBus _messageBus;
00026
00027         [Introduce]
00028         private IGameSaveService _saveService;
00029
00030         public List<ElementConfig> GetEvolutionChain(EvolutionData evolutionData)
00031         {
00032             var chain = evolutionData?.Chain;
00033
00034             return chain;
00035         }
00036

```

```
00037     public List<ElementConfig> GetGeneratedBy(ElementConfig config)
00038     {
00039         var generators = _configsService.GetConfigs(x =>
00040         {
00041             ElementGeneratorSettings generatorSettings = x.GeneratorSettings;
00042             return generatorSettings != null && generatorSettings.GenerateItems.Any(el =>
00043                 el.Element == config);
00044         }).ToList();
00045 
00046         return generators;
00047     }
00048 
00049     public List<ElementConfig> Generates(ElementConfig config)
00050     {
00051         List<ElementConfig> generatesElements = null;
00052         if (config.GeneratorSettings != null)
00053         {
00054             generatesElements = new List<ElementConfig>();
00055             var generatedItems = config.GeneratorSettings.GenerateItems;
00056 
00057             for (int i = 0; i < generatedItems.Count; i++)
00058             {
00059                 GeneratableElement generatedItem = generatedItems[i];
00060 
00061                 ElementConfig elementConfig = generatedItem.Element;
00062 
00063                 if (elementConfig != null)
00064                 {
00065                     generatesElements.Add(elementConfig);
00066                 }
00067             }
00068         }
00069 
00070         return generatesElements;
00071     }
00072 
00073     public void UpdateProgress(ElementConfig elementId)
00074     {
00075         EvolutionData evolution = _configsService.GetEvolutionByElement(elementId);
00076 
00077         if (evolution == null || evolution.Discovered)
00078         {
00079             return;
00080         }
00081 
00082         string evolutionId = evolution.Id;
00083 
00084         if (_serviceModel.EvolutionsProgress.TryGetValue(evolutionId, out int progress))
00085         {
00086             int index = evolution.Chain.IndexOf(elementId);
00087             int order = index + 1;
00088 
00089             if (index != -1 && order > progress)
00090             {
00091                 _serviceModel.EvolutionsProgress[evolutionId] = order;
00092             }
00093         }
00094     }
00095 
00096     public int GetEvolutionProgress(string id)
00097     {
00098         _serviceModel.EvolutionsProgress.TryGetValue(id, out int progress);
00099 
00100         return progress;
00101     }
00102 
00103     public void SetupEvolutionsProgress(IEvolutionsProgressData data)
00104     {
00105         var evolutions = data.EvolutionsProgress;
00106 
00107         if (evolutions != null)
00108         {
00109             for (int i = 0; i < evolutions.Count; i++)
00110             {
00111                 EvolutionProgressData evolutionData = evolutions[i];
00112                 _serviceModel.EvolutionsProgress[evolutionData.EvolutionId] =
00113                     evolutionData.Progress;
00114             }
00115         }
00116 
00117         public IEvolutionsProgressData GetData()
00118     {
00119         var evolutions = new EvolutionsProgressData();
00120 
00121         foreach (var evoProgressItem in _serviceModel.EvolutionsProgress)
```

```

00122         {
00123             var evoProgressData = new EvolutionProgressData
00124             {
00125                 Progress = evoProgressItem.Value,
00126                 EvolutionId = evoProgressItem.Key
00127             };
00128
00129             evolutions.EvolutionsProgress.Add(evoProgressData);
00130         }
00131
00132         return evolutions;
00133     }
00134
00135     public void CreateEvolutionsProgress()
00136     {
00137         SchemeObject evolutionsConfig = _configsService.LevelConfig.EvolutionsScheme;
00138         var evolutions = new EvolutionsProgressData();
00139
00140         for (int i = 0; i < evolutionsConfig.Evolution.Count; i++)
00141         {
00142             EvolutionData evolution = evolutionsConfig.Evolution[i];
00143
00144             _serviceModel.EvolutionsProgress[evolution.Id] = evolution.Discovered ?
00145             evolution.Chain.Count : 1;
00146         }
00147
00148         SetupEvolutionsProgress(evolutions);
00149
00150         _saveService.Save(GameSaveType.EvolutionsProgress);
00151     }
00152 }

```

7.304 EvolutionsServiceModel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004
00005 namespace MergeIt.Game.Services
00006 {
00007     public class EvolutionsServiceModel
00008     {
00009         public Dictionary<string, int> EvolutionsProgress { get; } = new();
00010     }
00011 }

```

7.305 GameFieldActionsService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using MergeIt.Core.Messages;
00006 using MergeIt.Core.Services;
00007 using MergeIt.Game.Field.Actions;
00008 using MergeIt.Game.Messages;
00009 using MergeIt.SimpleDI;
00010 using MergeIt.SimpleDI.ReservedInterfaces;
00011
00012 namespace MergeIt.Game.Services
00013 {
00014     public class GameFieldActionsService : IGameFieldActionsService, IInitializable, IDisposable
00015     {
00016         private readonly List<IFieldActionProcessor> _fieldActionProcessors = new();
00017
00018         [Introduce]
00019         private IMessageBus _messageBus;
00020
00021         public void Dispose()
00022         {
00023             _messageBus.RemoveListener<ClickElementMessage>(OnClickElementMessageHandler);
00024             _messageBus.RemoveListener<EndDragElementMessage>(OnEndDragElementMessageHandler);
00025         }
00026
00027         public void Initialize()
00028         {
00029             _messageBus.AddListener<ClickElementMessage>(OnClickElementMessageHandler);
00030             _messageBus.AddListener<EndDragElementMessage>(OnEndDragElementMessageHandler);

```

```
00031     _fieldActionProcessors.Add(new FieldMergeProcessor());
00032     _fieldActionProcessors.Add(new FieldGenerationProcessor());
00033     _fieldActionProcessors.Add(new FieldConsumableProcessor());
00034 }
00035 }
00036
00037 private void OnClickElementMessageHandler(ClickElementMessage message)
00038 {
00039     for (int i = 0; i < _fieldActionProcessors.Count; i++)
00040     {
00041         _fieldActionProcessors[i].ProcessClick(message.Cell);
00042     }
00043 }
00044
00045 private void OnEndDragElementMessageHandler(EndDragElementMessage message)
00046 {
00047     for (int i = 0; i < _fieldActionProcessors.Count; i++)
00048     {
00049         _fieldActionProcessors[i].ProcessEndDrag(message.FromPoint, message.ToGameObject);
00050     }
00051 }
00052 }
00053 }
```

7.306 GameFieldService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Configs.Elements;
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Core.Helpers;
00007 using MergeIt.Core.Messages;
00008 using MergeIt.Core.Services;
00009 using MergeIt.Game.Factories.Field;
00010 using MergeIt.Game.Factories.FieldElement;
00011 using MergeIt.Game.Field;
00012 using MergeIt.Game.Messages;
00013 using MergeIt.SimpleDI;
00014 using MergeIt.SimpleDI.ReservedInterfaces;
00015
00016 namespace MergeIt.Game.Services
00017 {
00018     public class GameFieldService : IGameFieldService, IInitializable, IDisposable
00019     {
00020         [Introduce]
00021         private IConfigsService _configsService;
00022
00023         [Introduce]
00024         private IFieldElementFactory _fieldElementFactory;
00025
00026         [Introduce]
00027         private IFieldFactory _fieldFactory;
00028
00029         [Introduce]
00030         private FieldLogicModel _fieldLogicModel;
00031
00032         [Introduce]
00033         private GameServiceModel _gameServiceModel;
00034
00035         [Introduce]
00036         private IMessageBus _messageBus;
00037
00038         public void Dispose()
00039         {
00040             _messageBus.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00041         }
00042
00043         public GridPoint? GetFreeCell()
00044         {
00045             int fieldHeight = _fieldLogicModel.FieldHeight;
00046             int fieldWidth = _fieldLogicModel.FieldWidth;
00047
00048             var randomHeight = ListExtensions.GenerateShuffledArray(fieldHeight);
00049             var randomWidth = ListExtensions.GenerateShuffledArray(fieldWidth);
00050
00051             for (int i = 0; i < randomHeight.Count; i++)
00052             {
00053                 for (int j = 0; j < randomWidth.Count; j++)
00054                 {
00055                     int row = randomHeight[i];
00056                     int column = randomWidth[j];
00057
00058                     if (_fieldElementFactory.CreateFieldElement(row, column))
00059                     {
00060                         return new GridPoint(row, column);
00061                     }
00062                 }
00063             }
00064
00065             return null;
00066         }
00067
00068         void OnLoadedGameMessageHandler(LoadedGameMessage message)
00069         {
00070             _gameServiceModel.OnGameLoaded();
00071         }
00072     }
00073 }
```

```

00058         var point = GridPoint.Create(row, column);
00059         if (!_fieldLogicModel.FieldElements.ContainsKey(point))
00060         {
00061             return point;
00062         }
00063     }
00064 }
00065
00066     return null;
00067 }
00068
00069     public IFIELDElement CreateNewElement(ElementConfig config, GridPoint point, bool blocked =
00070     false)
00071     {
00072         IFIELDElement newElement =
00073             _fieldElementFactory.CreateFieldElement(config, point, blocked);
00074
00075         return newElement;
00076     }
00077
00078     public void Initialize()
00079     {
00080         _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00081     }
00082
00083     private void OnLoadedGameMessageHandler(LoadedGameMessage message)
00084     {
00085         FieldPresenter field = _fieldFactory.CreateField(_gameServiceModel.MainCanvas.transform);
00086         field.Initialize();
00087     }
00088 }

```

7.307 GameService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Commands;
00005 using MergeIt.Core.Messages;
00006 using MergeIt.Core.Services;
00007 using MergeIt.Game.Commands;
00008 using MergeIt.Game.Converters;
00009 using MergeIt.Game.Factories.Field;
00010 using MergeIt.Game.Messages;
00011 using MergeIt.SimpleDI;
00012 using MergeIt.SimpleDI.ReservedInterfaces;
00013
00014 namespace MergeIt.Game.Services
00015 {
00016     public class GameService : IGameService, IInitializable, IDisposable
00017     {
00018         [Introduce]
00019         private IConfigProcessor _configProcessor;
00020
00021         [Introduce]
00022         private IConfigsService _configsService;
00023
00024         [Introduce]
00025         private IFIELDFactory _fieldFactory;
00026
00027         [Introduce]
00028         private IGameLoadService _gameLoadService;
00029
00030         [Introduce]
00031         private GameServiceModel _gameServiceModel;
00032
00033         [Introduce]
00034         public IMessageBus _messageBus;
00035
00036         [Introduce]
00037         private IGameSaveService _saveService;
00038
00039         public void Dispose()
00040         {
00041             _messageBus.RemoveListener<StartGameMessage>(StartGameMessageHandler);
00042         }
00043
00044         public void Initialize()
00045         {
00046             _messageBus.AddListener<StartGameMessage>(StartGameMessageHandler);
00047         }
00048

```

```

00049     private async void StartGameMessageHandler(StartGameMessage message)
00050     {
00051         var manager = new CommandManager();
00052
00053         manager.Add(new LoadConfigsCommand());
00054         manager.Add(new PrepareUserCommand());
00055         manager.Add(new PrepareEnergyCommand());
00056         manager.Add(new PrepareStockCommand());
00057         manager.Add(new PrepareInventoryCommand());
00058         manager.Add(new PrepareFieldCommand());
00059         manager.Add(new CheckEvolutionsProgressCommand());
00060
00061         await manager.RunAsync();
00062
00063         _messageBus.Fire<LoadedGameMessage>();
00064     }
00065 }
00066 }
```

7.308 GameServiceModel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Game.Field;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Services
00007 {
00008     public class GameServiceModel
00009     {
00010         private Camera _mainCamera;
00011         private Canvas _mainCanvas;
00012
00013         public FieldData LoadedLevel { get; set; }
00014
00015         public Camera MainCamera
00016         {
00017             get
00018             {
00019                 if (!_mainCamera)
00020                 {
00021                     _mainCamera = Camera.main;
00022                 }
00023
00024                 return _mainCamera;
00025             }
00026         }
00027
00028         public Canvas MainCanvas
00029         {
00030             get
00031             {
00032                 if (!_mainCanvas)
00033                 {
00034                     _mainCanvas =
00035                         GameObject.FindGameObjectWithTag("MainCanvas").GetComponent<Canvas>();
00036                 }
00037
00038                 return _mainCanvas;
00039             }
00040         }
00041 }
```

7.309 GeneratorService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Linq;
00005 using MergeIt.Core.Configs.Elements;
00006 using MergeIt.Core.Configs.Types;
00007 using MergeIt.Core.FieldElements;
00008 using MergeIt.Core.Messages;
00009 using MergeIt.Core.Services;
00010 using MergeIt.Game.Field;
00011 using MergeIt.Game.Helpers;
00012 using MergeIt.Game.Messages;
00013 using MergeIt.SimpleDI;
```

```

00014 using MergeIt.SimpleDI.ReservedInterfaces;
00015 using UnityEngine;
00016
00017 namespace MergeIt.Game.Services
00018 {
00019     public class GeneratorService : IGeneratorsService, IInitializable, IDisposable, IUpdatable
00020     {
00021         [Introduce] private IConfigsService _configsService;
00022
00023         [Introduce] private ICurrencyService _currencyService;
00024
00025         [Introduce] private FieldLogicModel _fieldLogicModel;
00026
00027         [Introduce] private IMessageBus _messageBus;
00028
00029         [Introduce] private IGameSaveService _saveService;
00030
00031         [Introduce] private UserServiceModel _userServiceModel;
00032
00033         public void Initialize()
00034         {
00035             _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00036             _messageBus.AddListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00037             _messageBus.AddListener<MergeElementsMessage>(MergeElementsMessageHandler);
00038             _messageBus.AddListener<SplitElementMessage>(SplitElementsMessageHandler);
00039             _messageBus.AddListener<CreateElementMessage>(CreateElementMessageHandler);
00040             _messageBus.AddListener<SkipTimeMessage>(OnSkipTimeMessageHandler);
00041         }
00042
00043         public void Dispose()
00044         {
00045             _messageBus.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00046             _messageBus.RemoveListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00047             _messageBus.RemoveListener<MergeElementsMessage>(MergeElementsMessageHandler);
00048             _messageBus.RemoveListener<SplitElementMessage>(SplitElementsMessageHandler);
00049             _messageBus.RemoveListener<CreateElementMessage>(CreateElementMessageHandler);
00050             _messageBus.RemoveListener<SkipTimeMessage>(OnSkipTimeMessageHandler);
00051         }
00052
00053         public void TryOpen(IFieldElement generator)
00054         {
00055             if (_fieldLogicModel.OpeningGenerator != null)
00056             {
00057                 Debug.Log("Another generator is already opening.");
00058
00059                 return;
00060             }
00061
00062             ElementConfig generatorConfig = generator.ConfigParameters.ElementConfig;
00063             generator.GeneratorOpenParameters.StartOpeningTime =
00064                 DateTimeOffset.UtcNow.ToUnixTimeSeconds();
00065             generator.GeneratorOpenParameters.RemainingTime.Value =
00066                 generatorConfig.GeneratorSettings.OpenTime;
00067             _fieldLogicModel.OpeningGenerator = generator;
00068
00069             _messageBus.Fire(new GeneratorOpenStartMessage
00070             {
00071                 GeneratorPoint = generator.InfoParameters.LogicPosition
00072             });
00073
00074         public void TrySkipOpening(IFieldElement generator)
00075         {
00076             ElementGeneratorSettings generatorSettings =
00077                 generator.ConfigParameters.ElementConfig.GeneratorSettings;
00078             if (_currencyService.TryPay(generatorSettings.SkipOpenCostSettings))
00079             {
00080                 ClearOpenableGenerator(generator);
00081
00082                 _messageBus.Fire(new GeneratorOpenedMessage
00083                 {
00084                     GeneratorPoint = generator.InfoParameters.LogicPosition
00085                 });
00086
00087         public void TrySkipCharging(IFieldElement generator)
00088         {
00089             ElementGeneratorSettings generatorSettings =
00090                 generator.ConfigParameters.ElementConfig.GeneratorSettings;
00091             if (_currencyService.TryPay(generatorSettings.SkipChargeCostSettings))
00092             {
00093                 generator.SkipCharging();
00094
00095                 _messageBus.Fire(new GeneratorRestoredMessage
00096                 {
00097                     GeneratorPoint = generator.InfoParameters.LogicPosition

```

```
00097         });
00098
00099     _messageBus.Fire(new CheckGeneratorMessage
00100 {
00101     GeneratorPoint = generator.InfoParameters.LogicPosition
00102 });
00103 }
00104 }
00105
00106 public void Update()
00107 {
00108     if (_fieldLogicModel != null)
00109     {
00110         float deltaTime = Time.unscaledDeltaTime;
00111         for (var index = _fieldLogicModel.AllGenerators.Count - 1; index >= 0; index--)
00112         {
00113             var generator = _fieldLogicModel.AllGenerators[index];
00114             IGeneratorOpenParameters generatorOpenParameters =
00115             generator.GeneratorOpenParameters;
00116             IGeneratorParameters parameters = generator.GeneratorParameters;
00117             if (parameters.Charging)
00118             {
00119                 if (parameters.MinDropChargeTime.Value > 0f)
00120                 {
00121                     parameters.MinDropChargeTime.Value -= deltaTime;
00122                 }
00123
00124                 parameters.RemainChargeTime -= deltaTime;
00125
00126                 if (parameters.RemainChargeTime <= 0f)
00127                 {
00128                     generator.ChargeGenerator();
00129                 }
00130
00131                 if (parameters.MinDropChargeTime.Value <= 0f)
00132                 {
00133                     _messageBus.Fire(new GeneratorRestoredMessage
00134                     {
00135                         GeneratorPoint = generator.InfoParameters.LogicPosition
00136                     });
00137                 }
00138             else if (generatorOpenParameters is { IsOpening: true })
00139             {
00140                 generatorOpenParameters.RemainingTime.Value -= deltaTime;
00141
00142                 if (generatorOpenParameters.RemainingTime.Value <= 0f)
00143                 {
00144                     ClearOpenableGenerator(generator);
00145
00146                     _messageBus.Fire(new GeneratorOpenedMessage
00147                     {
00148                         GeneratorPoint = generator.InfoParameters.LogicPosition
00149                     });
00150                 }
00151             }
00152         }
00153     }
00154 }
00155
00156 private void OnLoadedGameMessageHandler(LoadedGameMessage message)
00157 {
00158     var elements = _fieldLogicModel.FieldElements;
00159     IFieldElement openingGenerator = elements.FirstOrDefault(x =>
00160         x.Value.GeneratorOpenParameters is { IsOpening: true }).Value;
00161
00162     _fieldLogicModel.OpeningGenerator = openingGenerator;
00163     _fieldLogicModel.AllGenerators = elements
00164         .Where(x => x.Value.GeneratorParameters != null)
00165         .Select(x => x.Value).ToList();
00166 }
00167
00168 private void OnSkipTimeMessageHandler(SkipTimeMessage message)
00169 {
00170     for (var index = _fieldLogicModel.AllGenerators.Count - 1; index >= 0; index--)
00171     {
00172         var fieldElement = _fieldLogicModel.AllGenerators[index];
00173         if (fieldElement.GeneratorParameters == null)
00174         {
00175             continue;
00176         }
00177
00178         if (fieldElement.GeneratorOpenParameters is { IsOpening: true })
00179         {
00180             fieldElement.TrySkipOpeningTime(message.Seconds);
00181         }
00182     }
```

```

00183         if (fieldElement.GeneratorParameters.Charging)
00184             fieldElement.TrySkipChargingTime(message.Seconds);
00185
00186         _messageBus.Fire(new CheckGeneratorMessage
00187         {
00188             GeneratorPoint = fieldElement.InfoParameters.LogicPosition
00189         });
00190     }
00191 }
00192 }
00193 }
00194
00195 private void OnRemoveElementMessageHandler(RemoveElementMessage message)
00196 {
00197     IFieldElement removedGenerator =
00198     _fieldLogicModel.AllGenerators
00199     .FirstOrDefault(x => x.InfoParameters.LogicPosition == message.RemoveAtPoint);
00200
00201     if (removedGenerator != null)
00202     {
00203         if (_fieldLogicModel.OpeningGenerator == removedGenerator)
00204         {
00205             _fieldLogicModel.OpeningGenerator = null;
00206         }
00207
00208         _fieldLogicModel.AllGenerators.Remove(removedGenerator);
00209     }
00210 }
00211
00212 private void MergeElementsMessageHandler(MergeElementsMessage message)
00213 {
00214     CheckCreatedElement(message.NewElement);
00215 }
00216
00217 private void SplitElementsMessageHandler(SplitElementMessage message)
00218 {
00219     var position = message.SplitElement1.InfoParameters.LogicPosition;
00220
00221     var generator =
00222     _fieldLogicModel.AllGenerators.FirstOrDefault(x => x.InfoParameters.LogicPosition ==
00223     position);
00224
00225     if (generator != null)
00226     {
00227         _fieldLogicModel.AllGenerators.Remove(generator);
00228     }
00229
00230     CheckCreatedElement(message.SplitElement1);
00231     CheckCreatedElement(message.SplitElement2);
00232 }
00233
00234 private void CreateElementMessageHandler(CreateElementMessage message)
00235 {
00236     CheckCreatedElement(message.NewElement);
00237 }
00238
00239 private void ClearOpenableGenerator(IFieldElement generator)
00240 {
00241     generator.GeneratorOpenParameters.RemainingTime.Value = 0f;
00242     generator.GeneratorOpenParameters = null;
00243     _fieldLogicModel.OpeningGenerator = null;
00244 }
00245
00246 private void CheckCreatedElement(IFieldElement element)
00247 {
00248     if (element.InfoParameters.Type == ElementType.Generator)
00249     {
00250         element.UpdateGenerator();
00251         _fieldLogicModel.AllGenerators.Add(element);
00252     }
00253 }
00254 }

```

7.310 InfoPanelService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.FieldElements;
00005 using MergeIt.Core.Messages;
00006 using MergeIt.Core.Services;
00007 using MergeIt.Game.Messages;

```

```

00008 using MergeIt.Game.UI.InfoPanel;
00009 using MergeIt.SimpleDI;
00010 using MergeIt.SimpleDI.ReservedInterfaces;
00011
00012 namespace MergeIt.Game.Services
00013 {
00014     public class InfoPanelService : IInfoPanelService, IInitializable, IDisposable
00015     {
00016         [Introduce]
00017         private IElementService _elementService;
00018
00019         [Introduce]
00020         private IGeneratorsService _generatorsService;
00021
00022         [Introduce]
00023         private IMessageBus _messageBus;
00024
00025         public void Dispose()
00026         {
00027             _messageBus.RemoveListener<ElementActionMessage>(OnElementActionMessageHandler);
00028         }
00029
00030         public void Initialize()
00031         {
00032             _messageBus.AddListener<ElementActionMessage>(OnElementActionMessageHandler);
00033         }
00034
00035         private void OnElementActionMessageHandler(ElementActionMessage message)
00036         {
00037             IFieldElement fieldElement = message.Element;
00038             switch (message.ActionType)
00039             {
00040                 case ElementActionType.SkipCharging:
00041                     _generatorsService.TrySkipCharging(fieldElement);
00042                     break;
00043
00044                 case ElementActionType.SkipOpening:
00045                     _generatorsService.TrySkipOpening(fieldElement);
00046                     break;
00047
00048                 case ElementActionType.Sell:
00049                     _elementService.TrySell(fieldElement);
00050                     break;
00051
00052                 case ElementActionType.Split:
00053                     _elementService.TrySplit(fieldElement);
00054                     break;
00055
00056                 case ElementActionType.Open:
00057                     _generatorsService.TryOpen(fieldElement);
00058                     break;
00059
00060                 case ElementActionType.Unlock:
00061                     _elementService.TryUnlock(fieldElement);
00062                     break;
00063             }
00064         }
00065     }
00066 }

```

7.311 InventoryService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Configs.Inventory;
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Core.Inventory;
00007 using MergeIt.Core.Saves;
00008 using MergeIt.Core.Services;
00009 using MergeIt.Core.WindowSystem;
00010 using MergeIt.Game.Converters;
00011 using MergeIt.Game.Factories.Inventory;
00012 using MergeIt.Game.Windows.Inventory;
00013 using MergeIt.SimpleDI;
00014
00015 namespace MergeIt.Game.Services
00016 {
00017     public class InventoryService : IInventoryService
00018     {
00019         [Introduce]
00020         private IConfigProcessor _configProcessor;
00021

```

```

00022     [Introduce]
00023     private IConfigsService _configsService;
00024
00025     [Introduce]
00026     private IGameSaveService _saveService;
00027
00028     [Introduce]
00029     private InventoryServiceModel _serviceModel;
00030
00031     [Introduce]
00032     private IWindowSystem _windowSystem;
00033
00034     public void CreateInventory()
00035     {
00036         InventoryConfig inventoryConfig = _configsService.InventoryConfig;
00037         var inventoryData = new InventoryData
00038         {
00039             InventorySize = inventoryConfig.InitialCapacity
00040         };
00041
00042         SetupInventory(inventoryData);
00043
00044         _saveService.Save(GameSaveType.Inventory);
00045
00046     }
00047
00048     public void SetupInventory(IInventoryData inventoryData)
00049     {
00050         _serviceModel.InventorySize = inventoryData.InventorySize;
00051         var elements = new List<IFieldElement>();
00052
00053         if (inventoryData.InventoryElements != null)
00054         {
00055             for (int i = 0; i < inventoryData.InventoryElements.Length; i++)
00056             {
00057                 FieldElementData elementData = inventoryData.InventoryElements[i];
00058                 IFIELD_ELEMENT element = _configProcessor.ConvertToFieldElement(elementData);
00059                 elements.Add(element);
00060             }
00061
00062             _serviceModel.InventoryElements.AddRange(elements);
00063         }
00064     }
00065
00066     public IInventoryData GetData()
00067     {
00068         var data = new InventoryData();
00069         data.InventorySize = _serviceModel.InventorySize;
00070
00071         if (_serviceModel.InventoryElements != null)
00072         {
00073             data.InventoryElements = new FieldElementData[_serviceModel.InventoryElements.Count];
00074
00075             for (int i = 0; i < _serviceModel.InventoryElements.Count; i++)
00076             {
00077                 IFIELD_ELEMENT element = _serviceModel.InventoryElements[i];
00078                 FieldElementData elementData =
00079                 _configProcessor.ConvertToFieldElementData(element);
00080
00081                 data.InventoryElements[i] = elementData;
00082             }
00083         }
00084
00085         return data;
00086     }
00087
00088     public void Add(IFIELD_ELEMENT fieldElement)
00089     {
00090         _serviceModel.InventoryElements.Add(fieldElement);
00091         _saveService.Save(GameSaveType.Inventory);
00092     }
00093
00094     public bool Remove(IFIELD_ELEMENT fieldElement)
00095     {
00096         bool result = _serviceModel.InventoryElements.Remove(fieldElement);
00097
00098         if (result)
00099         {
00100             _saveService.Save(GameSaveType.Inventory);
00101
00102         }
00103
00104         return result;
00105     }
00106
00107     public void OpenWindow()
00108     {
00109         _windowSystem.OpenWindow<InventoryPresenter>(enableBlackout: true);
00110     }

```

```
00108 }
```

7.312 InventoryServiceModel.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.FieldElements;
00005
00006 namespace MergeIt.Game.Services
00007 {
00008     public class InventoryServiceModel
00009     {
00010         public int InventorySize { get; set; }
00011         public List<IFieldElement> InventoryElements { get; } = new();
00012
00013         public bool IsFull()
00014         {
00015             return InventoryElements.Count == InventorySize;
00016         }
00017     }
00018 }
```

7.313 ResourcesLoaderService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Services;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Services
00007 {
00008     public class ResourcesLoaderService : IResourcesLoaderService
00009     {
00010         public TResource GetObject<TResource>(string path) where TResource : Object
00011         {
00012             var resource = Resources.Load<TResource>(path);
00013
00014             return resource;
00015         }
00016     }
00017 }
```

7.314 GameLoadService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004 using MergeIt.Core.Saves;
00005 using MergeIt.Core.Services;
00006 using MergeIt.Game.Services.Saves.Strategies;
00007 using MergeIt.SimpleDI;
00008
00009 namespace MergeIt.Game.Services.Saves
00010 {
00011     public class GameLoadService : IGameLoadService
00012     {
00013         [Introduce]
00014         private IMessageBus _messageBus;
00015
00016         [Introduce]
00017         private ISerializeStrategy _serializeStrategy;
00018
00019         public T Load<T>() where T : class, ISavable
00020         {
00021             var data = _serializeStrategy.Load<T>();
00022
00023             return data;
00024         }
00025     }
00026 }
```

7.315 GameSaveEveryIntervalService.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using System.Linq;
00006 using Cysharp.Threading.Tasks;
00007 using MergeIt.Core.FieldElements;
00008 using MergeIt.Core.Helpers;
00009 using MergeIt.Core.Messages;
00010 using MergeIt.Core.Saves;
00011 using MergeIt.Core.Services;
00012 using MergeIt.Game.Converters;
00013 using MergeIt.Game.ElementsStock;
00014 using MergeIt.Game.Evolutions;
00015 using MergeIt.Game.Factories.Inventory;
00016 using MergeIt.Game.Field;
00017 using MergeIt.Game.Messages;
00018 using MergeIt.Game.Services.Saves.Strategies;
00019 using MergeIt.Game.User;
00020 using MergeIt.SimpleDI;
00021 using MergeIt.SimpleDI.ReservedInterfaces;
00022 using UnityEditor;
00023 using UnityEngine;
00024
00025 namespace MergeIt.Game.Services.Saves
00026 {
00027     public class GameSaveEveryIntervalService : IGameSaveService, IInitializable, IDisposable,
00028         IUpdatable, IMonoApplicationQuitHandler
00029     {
00030         private readonly HashSet<GameSaveType> _saveQueue = new();
00031         private bool _canSave;
00032         private float _time;
00033         private float _interval;
00034
00035         [Introduce]
00036         private IConfigProcessor _configProcessor;
00037
00038         [Introduce]
00039         private IConfigsService _configsService;
00040
00041         [Introduce]
00042         private IEvolutionsService _evolutionsService;
00043
00044         [Introduce]
00045         private FieldLogicModel _fieldLogicModel;
00046
00047         [Introduce]
00048         private GameServiceModel _gameServiceModel;
00049
00050         [Introduce]
00051         private IInventoryService _inventoryService;
00052
00053         [Introduce]
00054         private IMessageBus _messageBus;
00055
00056         [Introduce]
00057         private IElementsStockService _stockService;
00058
00059         [Introduce]
00060         private UserServiceModel _userServiceModel;
00061
00062         [Introduce]
00063         private ISerializeStrategy _serializeStrategy;
00064
00065         public async UniTask Save(GameSaveType gameSaveType)
00066         {
00067             _saveQueue.Add(gameSaveType);
00068
00069             await UniTask.Yield();
00070         }
00071
00072         public void Initialize()
00073         {
00074             _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00075             MonoEventsListener.Instance.SubscribeOnApplicationQuit(this);
00076         }
00077
00078         public void Dispose()
00079         {
00080             _messageBus.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00081             MonoEventsListener.Instance.UnsubscribeFromApplicationQuit(this);
00082         }
00083
00084         public async void OnApplicationQuit()
00085         {
```

```
00085         await TrySave(GameSaveType.All);
00086     }
00087
00088     public async void Update()
00089     {
00090         if (_canSave)
00091         {
00092             _time += Time.unscaledDeltaTime;
00093
00094             if (_time >= _interval)
00095             {
00096                 _canSave = false;
00097                 await TrySave();
00098
00099                 _time = 0f;
00100                 _canSave = true;
00101             }
00102         }
00103     }
00104
00105     private async void OnLoadedGameMessageHandler(LoadedGameMessage _)
00106     {
00107         _interval = _configsService.GameConfig.SaveInterval;
00108         _canSave = true;
00109
00110         await TrySave();
00111     }
00112
00113     private async UniTask SaveLevel()
00114     {
00115         FieldData loadedLevel = _gameServiceModel.LoadedLevel;
00116         loadedLevel.SavedElementsData.Clear();
00117
00118         foreach (var fieldElement in _fieldLogicModel.FieldElements)
00119         {
00120             FieldElementData savedElement =
00121             _configProcessor.ConvertToFieldElementData(fieldElement.Value);
00122             loadedLevel.SavedElementsData.Add(savedElement);
00123         }
00124
00125         await SaveAsync(loadedLevel);
00126     }
00127
00128     private async UniTask SaveUser()
00129     {
00130         UserData userData = _userServiceModel.GetData();
00131
00132         await SaveAsync(userData);
00133     }
00134
00135     private async UniTask SaveStock()
00136     {
00137         var stockData = _stockService.GetData() as ElementsStockData;
00138
00139         await SaveAsync(stockData);
00140     }
00141
00142     private async UniTask SaveInventory()
00143     {
00144         var inventoryData = _inventoryService.GetData() as InventoryData;
00145
00146         await SaveAsync(inventoryData);
00147     }
00148
00149     private async UniTask SaveEvolutionsProgress()
00150     {
00151         var evolutionsProgressData = _evolutionsService.GetData() as EvolutionsProgressData;
00152
00153         await SaveAsync(evolutionsProgressData);
00154     }
00155
00156     private async UniTask SaveAsync<T>(T data) where T : class, ISavable
00157     {
00158         await _serializeStrategy.Save(data);
00159 #if UNITY_EDITOR
00160         AssetDatabase.Refresh();
00161 #endif
00162     }
00163
00164     private async UniTask TrySave()
00165     {
00166         if (_saveQueue.Count > 0)
00167         {
00168             var saves = _saveQueue.Distinct().ToArray();
00169             foreach (GameSaveType saveType in saves)
00170             {
00171                 await SaveAsync(saveType);
00172             }
00173         }
00174     }
00175 }
```

```

00171             _saveQueue.Remove(saveType);
00172             await TrySave(saveType);
00173         }
00174     }
00175 }
00176
00177     private async UniTask TrySave(GameSaveType gameSaveType)
00178     {
00179         if ((gameSaveType & GameSaveType.Field) == GameSaveType.Field)
00180         {
00181             await SaveLevel();
00182         }
00183
00184         if ((gameSaveType & GameSaveType.User) == GameSaveType.User)
00185         {
00186             await SaveUser();
00187         }
00188
00189         if ((gameSaveType & GameSaveType.Inventory) == GameSaveType.Inventory)
00190         {
00191             await SaveInventory();
00192         }
00193
00194         if ((gameSaveType & GameSaveType.Stock) == GameSaveType.Stock)
00195         {
00196             await SaveStock();
00197         }
00198
00199         if ((gameSaveType & GameSaveType.EvolutionsProgress) == GameSaveType.EvolutionsProgress)
00200         {
00201             await SaveEvolutionsProgress();
00202         }
00203     }
00204 }
00205 }
```

7.316 GameSaveOnEveryActionService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using Cysharp.Threading.Tasks;
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Core.Helpers;
00007 using MergeIt.Core.Saves;
00008 using MergeIt.Core.Services;
00009 using MergeIt.Game.Converters;
00010 using MergeIt.Game.ElementsStock;
00011 using MergeIt.Game.Evolutions;
00012 using MergeIt.Game.Factories.Inventory;
00013 using MergeIt.Game.Field;
00014 using MergeIt.Game.Services.Saves.Strategies;
00015 using MergeIt.Game.User;
00016 using MergeIt.SimpleDI;
00017 using MergeIt.SimpleDI.ReservedInterfaces;
00018 using UnityEditor;
00019
00020 namespace MergeIt.Game.Services.Saves
00021 {
00022     public class GameSaveOnEveryActionService : IGameSaveService, IInitializable, IDisposable,
00023     IMonoApplicationQuitHandler
00024     {
00025         private const string SavePath = "Saves";
00026
00027         [Introduce]
00028         private IConfigProcessor _configProcessor;
00029
00030         [Introduce]
00031         private IEvolutionsService _evolutionsService;
00032
00033         [Introduce]
00034         private FieldLogicModel _fieldLogicModel;
00035
00036         [Introduce]
00037         private GameServiceModel _gameServiceModel;
00038
00039         [Introduce]
00040         private IInventoryService _inventoryService;
00041
00042         [Introduce]
00043         private IElementsStockService _stockService;
00044
00045         [Introduce]
```

```
00045     private UserServiceModel _userServiceModel;
00046
00047     [Introduce]
00048     private ISerializeStrategy _serializeStrategy;
00049
00050     public async UniTask Save(GameSaveType gameSaveType)
00051     {
00052         if ((gameSaveType & GameSaveType.Field) == GameSaveType.Field)
00053         {
00054             await SaveLevel();
00055         }
00056
00057         if ((gameSaveType & GameSaveType.User) == GameSaveType.User)
00058         {
00059             await SaveUser();
00060         }
00061
00062         if ((gameSaveType & GameSaveType.Inventory) == GameSaveType.Inventory)
00063         {
00064             await SaveInventory();
00065         }
00066
00067         if ((gameSaveType & GameSaveType.Stock) == GameSaveType.Stock)
00068         {
00069             await SaveStock();
00070         }
00071
00072         if ((gameSaveType & GameSaveType.EvolutionsProgress) == GameSaveType.EvolutionsProgress)
00073         {
00074             await SaveEvolutionsProgress();
00075         }
00076     }
00077
00078     public void Initialize()
00079     {
00080         MonoEventsListener.Instance.SubscribeOnApplicationQuit(this);
00081     }
00082
00083     public void Dispose()
00084     {
00085         MonoEventsListener.Instance.UnsubscribeFromApplicationQuit(this);
00086     }
00087
00088     public async void OnApplicationQuit()
00089     {
00090         await Save(GameSaveType.All);
00091     }
00092
00093     private async UniTask SaveLevel()
00094     {
00095         FieldData loadedLevel = _gameServiceModel.LoadedLevel;
00096         loadedLevel.SavedElementsData.Clear();
00097
00098         foreach (var fieldElement in _fieldLogicModel.FieldElements)
00099         {
00100             FieldElementData savedElement =
00101             _configProcessor.ConvertToFieldElementData(fieldElement.Value);
00102             loadedLevel.SavedElementsData.Add(savedElement);
00103
00104             await SaveAsync(loadedLevel);
00105         }
00106
00107         private async UniTask SaveUser()
00108     {
00109         UserData userData = _userServiceModel.GetData();
00110
00111         await SaveAsync(userData);
00112     }
00113
00114     private async UniTask SaveStock()
00115     {
00116         var stockData = _stockService.GetData() as ElementsStockData;
00117
00118         await SaveAsync(stockData);
00119     }
00120
00121     private async UniTask SaveInventory()
00122     {
00123         var inventoryData = _inventoryService.GetData() as InventoryData;
00124
00125         await SaveAsync(inventoryData);
00126     }
00127
00128     private async UniTask SaveEvolutionsProgress()
00129     {
00130         var evolutionsProgressData = _evolutionsService.GetData() as EvolutionsProgressData;
```

```

00131         await SaveAsync(evolutionProgressData);
00132     }
00133 }
00134
00135     private async UniTask SaveAsync<T>(T data) where T : class, ISavable
00136     {
00137         await _serializeStrategy.Save(data);
00138 #if UNITY_EDITOR
00139         AssetDatabase.Refresh();
00140 #endif
00141     }
00142 }
00143 }
```

7.317 GameSaveOnQuitService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using Cysharp.Threading.Tasks;
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Core.Helpers;
00007 using MergeIt.Core.Saves;
00008 using MergeIt.Core.Services;
00009 using MergeIt.Game.Converters;
00010 using MergeIt.Game.ElementsStock;
00011 using MergeIt.Game.Evolutions;
00012 using MergeIt.Game.Factories.Inventory;
00013 using MergeIt.Game.Field;
00014 using MergeIt.Game.Services.Saves.Strategies;
00015 using MergeIt.Game.User;
00016 using MergeIt.SimpleDI;
00017 using MergeIt.SimpleDI.ReservedInterfaces;
00018 using UnityEditor;
00019 using UnityEngine;
00020
00021 namespace MergeIt.Game.Services.Saves
00022 {
00023     public class GameSaveOnQuitService : IGameSaveService, IInitializable, IDisposable,
00024     IMonoApplicationQuitHandler
00025     {
00026         [Introduce]
00027         private IConfigProcessor _configProcessor;
00028
00029         [Introduce]
00030         private IEvolutionsService _evolutionsService;
00031
00032         [Introduce]
00033         private FieldLogicModel _fieldLogicModel;
00034
00035         [Introduce]
00036         private GameServiceModel _gameServiceModel;
00037
00038         [Introduce]
00039         private IInventoryService _inventoryService;
00040
00041         [Introduce]
00042         private IElementsStockService _stockService;
00043
00044         [Introduce]
00045         private UserServiceModel _userServiceModel;
00046
00047         [Introduce]
00048         private ISerializeStrategy _serializeStrategy;
00049
00050         public void Dispose()
00051         {
00052             MonoEventsListener.Instance.UnsubscribeFromApplicationQuit(this);
00053         }
00054
00055         public async UniTask Save(GameSaveType gameSaveType)
00056         {
00057             if (gameSaveType != GameSaveType.All)
00058             {
00059                 Debug.Log("This save strategy works only on application quit.");
00060
00061                 return;
00062             }
00063
00064             await SaveLevel();
00065
00066             await SaveUser();
00067     }
```

```

00067         await SaveInventory();
00068
00069         await SaveStock();
00070
00071         await SaveEvolutionsProgress();
00072     }
00073
00074     public void Initialize()
00075     {
00076         MonoEventsListener.Instance.SubscribeOnApplicationQuit(this);
00077     }
00078
00079     public async void OnApplicationQuit()
00080     {
00081         await Save(GameSaveType.All);
00082     }
00083
00084     private async UniTask SaveLevel()
00085     {
00086         FieldData loadedLevel = _gameServiceModel.LoadedLevel;
00087         loadedLevel.SavedElementsData.Clear();
00088
00089         foreach (var fieldElement in _fieldLogicModel.FieldElements)
00090         {
00091             FieldElementData savedElement =
00092                 _configProcessor.ConvertToFieldElementData(fieldElement.Value);
00093                 loadedLevel.SavedElementsData.Add(savedElement);
00094             }
00095
00096             await SaveAsync(loadedLevel);
00097         }
00098
00099     private async UniTask SaveUser()
00100     {
00101         UserData userData = _userServiceModel.GetData();
00102
00103         await SaveAsync(userData);
00104     }
00105
00106     private async UniTask SaveStock()
00107     {
00108         var stockData = _stockService.GetData() as ElementsStockData;
00109
00110         await SaveAsync(stockData);
00111     }
00112
00113     private async UniTask SaveInventory()
00114     {
00115         var inventoryData = _inventoryService.GetData() as InventoryData;
00116
00117         await SaveAsync(inventoryData);
00118     }
00119
00120     private async UniTask SaveEvolutionsProgress()
00121     {
00122         var evolutionsProgressData = _evolutionsService.GetData() as EvolutionsProgressData;
00123
00124         await SaveAsync(evolutionsProgressData);
00125     }
00126
00127     private async UniTask SaveAsync<T>(T data) where T : class, ISavable
00128     {
00129         await _serializeStrategy.Save(data);
00130 #if UNITY_EDITOR
00131         AssetDatabase.Refresh();
00132 #endif
00133     }
00134 }
00135 }
```

7.318 BaseSerializeStrategy.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using Cysharp.Threading.Tasks;
00004 using MergeIt.Core.Saves;
00005
00006 namespace MergeIt.Game.Services.Saves.Strategies
00007 {
00008     public abstract class BaseSerializeStrategy : ISerializeStrategy
00009     {
00010         public virtual string SaveDir
```

```

00011      {
00012          get => "Saves";
00013      }
00014
00015      public abstract UniTask Save<T>(T data) where T : class, ISavable;
00016
00017      public abstract T Load<T>() where T : class, ISavable;
00018  }
00019 }
```

7.319 BinaryFormatterStrategy.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.IO;
00004 using System.Runtime.Serialization.Formatters.Binary;
00005 using Cysharp.Threading.Tasks;
00006 using MergeIt.Core.Helpers;
00007 using UnityEngine;
00008 using Debug = UnityEngine.Debug;
00009
00010 namespace MergeIt.Game.Services.Saves.Strategies
00011 {
00012     public class BinaryFormatterStrategy : BaseSerializeStrategy
00013     {
00014         private readonly BinaryFormatter _formatter = new();
00015
00016         public async override UniTask Save<T>(T data)
00017         {
00018             using var memoryStream = new MemoryStream();
00019
00020             _formatter.Serialize(memoryStream, data);
00021
00022             byte[] serializedData = memoryStream.ToArray();
00023             string fileName = SavesHelper.GetFileNameWithExtension<T>();
00024
00025             await WriteSerializedDataAsync(serializedData, fileName);
00026         }
00027
00028         public override T Load<T>()
00029         {
00030             string fileName = SavesHelper.GetFileNameWithExtension<T>();
00031             string loadPath = Path.Combine(Application.streamingAssetsPath, SaveDir, fileName);
00032             if (File.Exists(loadPath))
00033             {
00034                 FileStream fileStream = File.Open(loadPath, FileMode.Open);
00035
00036                 var loadedData = _formatter.Deserialize(fileStream) as T;
00037
00038                 fileStream.Close();
00039
00040                 return loadedData;
00041             }
00042
00043             Debug.Log($"There is no saved {fileName} found.");
00044
00045             return null;
00046         }
00047
00048         private async UniTask WriteSerializedDataAsync(byte[] bytes, string fileName)
00049         {
00050             string path = Path.Combine(Application.streamingAssetsPath, SaveDir);
00051             if (!Directory.Exists(path))
00052             {
00053                 Directory.CreateDirectory(path);
00054             }
00055
00056             path = Path.Combine(path, fileName);
00057
00058             await using var fileStream =
00059                 new FileStream(path, FileMode.Create, FileAccess.Write, FileShare.None, 4096, true);
00060
00061             await fileStream.WriteAsync(bytes, 0, bytes.Length);
00062         }
00063     }
00064 }
```

7.320 ISerializeStrategy.cs

```
00001 // Copyright (c) 2024, Awessets
```

```

00002
00003 using Cysharp.Threading.Tasks;
00004 using MergeIt.Core.Saves;
00005
00006 namespace MergeIt.Game.Services.Saves.Strategies
00007 {
00008     public interface ISerializeStrategy
00009     {
00010         string SaveDir { get; }
00011
00012         UniTask Save<T>(T data) where T : class, ISavable;
00013         T Load<T>() where T : class, ISavable;
00014     }
00015 }
```

7.321 JsonSerializerStrategy.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.IO;
00004 using Cysharp.Threading.Tasks;
00005 using MergeIt.Core.Helpers;
00006 using Newtonsoft.Json;
00007 using UnityEngine;
00008
00009 namespace MergeIt.Game.Services.Saves.Strategies
00010 {
00011     public class JsonSerializerStrategy : BaseSerializeStrategy
00012     {
00013         public async override UniTask Save<T>(T data)
00014         {
00015             string fileName = SavesHelper.GetFileNameWithExtension<T>();
00016             string serialized = JsonConvert.SerializeObject(data);
00017
00018             string path = Path.Combine(Application.streamingAssetsPath, SaveDir);
00019             if (!Directory.Exists(path))
00020             {
00021                 Directory.CreateDirectory(path);
00022             }
00023
00024             path = Path.Combine(path, fileName);
00025
00026             await File.WriteAllTextAsync(path, serialized);
00027         }
00028
00029         public override T Load<T>()
00030         {
00031             string fileName = SavesHelper.GetFileNameWithExtension<T>();
00032             string loadPath = Path.Combine(Application.streamingAssetsPath, SaveDir, fileName);
00033             if (File.Exists(loadPath))
00034             {
00035                 string fileStream = File.ReadAllText(loadPath);
00036
00037                 var loadedData = JsonConvert.DeserializeObject<T>(fileStream);
00038
00039                 return loadedData;
00040             }
00041
00042             Debug.Log($"There is no saved {fileName} found.");
00043
00044             return null;
00045         }
00046     }
00047 }
```

7.322 UserProgressService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Configs.Elements;
00005 using MergeIt.Core.Configs.LevelUp;
00006 using MergeIt.Core.Messages;
00007 using MergeIt.Core.Services;
00008 using MergeIt.Game.Messages;
00009 using MergeIt.SimpleDI;
00010 using MergeIt.SimpleDI.ReservedInterfaces;
00011
```

```

00012 namespace MergeIt.Game.Services
00013 {
00014     public class UserProgressService : IUserProgressService, IInitializable, IDisposable
00015     {
00016         [Introduce]
00017         private IConfigsService _configService;
00018
00019         [Introduce]
00020         private IMessageBus _messageBus;
00021
00022         [Introduce]
00023         private UserServiceModel _userServiceModel;
00024
00025         public void Dispose()
00026         {
00027             _messageBus.RemoveListener<LevelUpMessage>(OnLevelUpMessageHandler);
00028         }
00029
00030         public void Initialize()
00031         {
00032             _messageBus.AddListener<LevelUpMessage>(OnLevelUpMessageHandler);
00033         }
00034
00035         public int GetCurrentLevelMaxExp()
00036         {
00037             LevelUpParameters levelUpParameters =
00038             _configService.GetLevelUpData(_userServiceModel.Level.Value);
00039
00040             return levelUpParameters.Experience;
00041         }
00042
00043         public bool CanLevelUp()
00044         {
00045             return _userServiceModel.Experience.Value >= GetCurrentLevelMaxExp();
00046         }
00047
00048         public ElementConfig[] GetLevelUpPrizes()
00049         {
00050             LevelUpParameters levelUpParameters =
00051             _configService.GetLevelUpData(_userServiceModel.Level.Value);
00052
00053             if (levelUpParameters != null)
00054             {
00055                 return levelUpParameters.Bonuses;
00056             }
00057
00058             return null;
00059         }
00060
00061         private void OnLevelUpMessageHandler(LevelUpMessage message)
00062         {
00063             if (CanLevelUp())
00064             {
00065                 int userLevel = _userServiceModel.Level.Value;
00066                 LevelUpParameters currentLevelParameters = _configService.GetLevelUpData(userLevel);
00067
00068                 int experienceDiff = _userServiceModel.Experience.Value -
00069                 currentLevelParameters.Experience;
00070                 if (experienceDiff >= 0)
00071                 {
00072                     userLevel++;
00073
00074                     LevelUpParameters nextLevelParameters = _configService.GetLevelUpData(userLevel);
00075
00076                     if (nextLevelParameters != null)
00077                     {
00078                         _userServiceModel.Level.Value = userLevel;
00079                         _userServiceModel.Experience.SetValueSilently(experienceDiff);
00080
00081                         SendLevelUpdated(nextLevelParameters.Experience);
00082                     }
00083
00084                 private void SendLevelUpdated(int maxExp)
00085                 {
00086                     var message = new LevelUpdatedMessage {NextLevelExp = maxExp};
00087
00088                     _messageBus.Fire(message);
00089                 }
00090             }
00091         }

```

7.323 UserService.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs;
00004 using MergeIt.Core.Messages;
00005 using MergeIt.Core.Saves;
00006 using MergeIt.Core.Services;
00007 using MergeIt.Core.User;
00008 using MergeIt.Game.User;
00009 using MergeIt.SimpleDI;
00010 using Random = System.Random;
00011
00012 namespace MergeIt.Game.Services
00013 {
00014     public class UserService : IUserService
00015     {
00016         [Introduce]
00017         private IConfigsService _configsService;
00018
00019         [Introduce]
00020         private IMessageBus _messageBus;
00021
00022         [Introduce]
00023         private IGameSaveService _saveService;
00024
00025         [Introduce]
00026         private UserServiceModel _userServiceModel;
00027
00028         public void CreateUser()
00029     {
00030         GameConfig config = _configsService.GameConfig;
00031         var userData = new UserData
00032     {
00033         Name = $"User{new Random().Next(ushort.MinValue, ushort.MaxValue)}",
00034         Energy = config.EnergyCap,
00035         Experience = 0,
00036         SoftCurrency = config.InitialSoftCurrency,
00037         HardCurrency = config.InitialHardCurrency,
00038         Splitters = config.InitialSplittersCount,
00039         Level = 1
00040     };
00041         SetupUser(userData);
00042
00043         _saveService.Save(GameSaveType.User);
00044     }
00045
00046
00047     public void SetupUser(IUserData userData)
00048     {
00049         _userServiceModel.Set(userData);
00050     }
00051 }
00052
00053 }
```

7.324 UserServiceModel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.User;
00004 using MergeIt.Core.Utils;
00005 using MergeIt.Game.User;
00006
00007 namespace MergeIt.Game.Services
00008 {
00009     public class UserServiceModel
00010     {
00011         public Bindable<string> Name { get; } = new();
00012         public Bindable<int> Energy { get; } = new();
00013         public Bindable<int> SoftCurrency { get; } = new();
00014         public Bindable<int> HardCurrency { get; } = new();
00015         public Bindable<int> Splitters { get; } = new();
00016         public Bindable<int> Level { get; } = new();
00017         public Bindable<int> Experience { get; } = new();
00018
00019         public long EnergyRestoringStartTime { get; set; }
00020
00021         public void Set(IUserData data)
00022         {
00023             Name.Value = data.Name;
00024             Energy.Value = data.Energy;
00025             SoftCurrency.Value = data.SoftCurrency;
```

```

00026     HardCurrency.Value = data.HardCurrency;
00027     Splitters.Value = data.Splitters;
00028     Level.Value = data.Level;
00029     Experience.Value = data.Experience;
00030     EnergyRestoringStartTime = data.EnergyRestoringStartTime;
00031 }
00032
00033     public UserData GetData()
00034 {
00035         var userData = new UserData
00036     {
00037         Name = Name.Value,
00038         Energy = Energy.Value,
00039         Experience = Experience.Value,
00040         SoftCurrency = SoftCurrency.Value,
00041         HardCurrency = HardCurrency.Value,
00042         Splitters = Splitters.Value,
00043         Level = Level.Value,
00044         EnergyRestoringStartTime = EnergyRestoringStartTime
00045     };
00046
00047     return userData;
00048 }
00049 }
00050 }
```

7.325 TopPanelComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Messages;
00004 using MergeIt.Game.ElementsStock;
00005 using MergeIt.Game.Messages;
00006 using MergeIt.SimpleDI;
00007 using UnityEngine;
00008
00009 namespace MergeIt.Game.TopPanel
00010 {
00011     public class TopPanelComponent : MonoBehaviour
00012     {
00013         [SerializeField]
00014         private ElementsStockComponent _stockComponent;
00015
00016         private IMessageBus _messageBus;
00017
00018         private void Start()
00019     {
00020             _messageBus = DiContainer.Get<IMessageBus>();
00021             _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00022     }
00023
00024         private void OnDestroy()
00025     {
00026             _messageBus.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00027     }
00028
00029         private void OnLoadedGameMessageHandler(LoadedGameMessage _)
00030     {
00031             _stockComponent.Initialize();
00032     }
00033 }
00034 }
```

7.326 ElementActionType.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004
00005 namespace MergeIt.Game.UI.InfoPanel
00006 {
00007     [Flags]
00008     public enum Element ActionType
00009     {
00010         None = 0,
00011         Common = 1 << 0,
00012         SkipCharging = 1 << 1,
00013         SkipOpening = 1 << 2,
00014         Sell = 1 << 3,
```

```
00015     Split = 1 << 4,
00016     Open = 1 << 5,
00017     Unlock = 1 << 6
00018 }
00019 }
```

7.327 ElementCommonInfo.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using TMPro;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Game.UI.InfoPanel
00008 {
00009     public class ElementCommonInfo : ElementInfo
00010     {
00011         [SerializeField]
00012         private TMP_Text _nameLabel;
00013
00014         [SerializeField]
00015         private TMP_Text _descriptionLabel;
00016
00017         public override Element ActionType ActionType
00018         {
00019             get => Element ActionType.Common;
00020         }
00021
00022         protected override bool OnTrySetup()
00023         {
00024             gameObject.SetActive(true);
00025
00026             IInfoParameters infoParameters = SelectedElement.InfoParameters;
00027
00028             _nameLabel.text = infoParameters.Name;
00029             _descriptionLabel.text = infoParameters.Description;
00030
00031             return true;
00032         }
00033     }
00034 }
```

7.328 ElementInfo.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004 using MergeIt.Core.Messages;
00005 using MergeIt.Core.Services;
00006 using MergeIt.Game.Messages;
00007 using MergeIt.SimpleDI;
00008 using UnityEngine;
00009
00010 namespace MergeIt.Game.UI.InfoPanel
00011 {
00012     public abstract class ElementInfo : MonoBehaviour
00013     {
00014         protected IConfigsService ConfigsService;
00015         protected IMessageBus MessageBus;
00016
00017         protected IFieldElement SelectedElement;
00018         public abstract Element ActionType { get; }
00019
00020         private void Start()
00021         {
00022             MessageBus = DiContainer.Get<IMessageBus>();
00023
00024             OnStart();
00025         }
00026
00027         private void OnDisable()
00028         {
00029             Clear();
00030         }
00031
00032         private void OnDestroy()
00033         {
00034             Destroy();
00035         }
00036     }
00037 }
```

```

00035         }
00036
00037     public void TrySetup(IFieldElement fieldElement)
00038     {
00039         ConfigsService = DiContainer.Get<IConfigsService>();
00040         SelectedElement = fieldElement;
00041
00042         if (!OnTrySetup())
00043         {
00044             SelectedElement = null;
00045             gameObject.SetActive(false);
00046         }
00047     }
00048
00049     protected abstract bool OnTrySetup();
00050
00051     protected void Clear()
00052     {
00053         SelectedElement = null;
00054
00055         OnClear();
00056     }
00057
00058     protected virtual void OnStart()
00059     {
00060
00061     }
00062
00063     protected virtual void OnClear()
00064     {
00065
00066     }
00067     protected virtual void Destroy()
00068     {
00069
00070     }
00071
00072     protected void ActionButtonClick()
00073     {
00074         OnActionButtonClick();
00075     }
00076
00077     protected virtual void OnActionButtonClick()
00078     {
00079         SendActionMessage();
00080     }
00081
00082     protected void SendActionMessage()
00083     {
00084         var message = new ElementActionMessage
00085         {
00086             Element = SelectedElement,
00087             ActionType = ActionType
00088         };
00089
00090         MessageBus.Fire(message);
00091     }
00092 }
00093 }
```

7.329 ElementInfoPanelComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using System.Linq;
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Core.Messages;
00007 using MergeIt.Core.WindowSystem;
00008 using MergeIt.Game.Field;
00009 using MergeIt.Game.Messages;
00010 using MergeIt.Game.Windows.ElementInfo;
00011 using MergeIt.SimpleDI;
00012 using TMPro;
00013 using UnityEngine;
00014 using UnityEngine.UI;
00015
00016 namespace MergeIt.Game.UI.InfoPanel
00017 {
00018     public class ElementInfoPanelComponent : MonoBehaviour
00019     {
00020         [SerializeField]
00021         private TMP_Text _commonLabel;
```

```
00022
00023     [SerializeField]
00024     private ElementInfo[] _elementInfos;
00025
00026     [SerializeField]
00027     private Button _infoButton;
00028
00029     private Dictionary<ElementActionType, ElementInfo> _elementInfoDict = new();
00030     private FieldLogicModel _fieldLogicModel;
00031     private IMessageBus _messageBus;
00032     private IFieldElement _selectedElement;
00033     private IWindowSystem _windowSystem;
00034
00035     private void Start()
00036     {
00037         _infoButton.onClick.AddListener(OnItemInfoClicked);
00038
00039         _elementInfoDict = _elementInfos.ToDictionary(el => el.ActionType, el => el);
00040         _commonLabel.gameObject.SetActive(true);
00041
00042         _windowSystem = DiContainer.Get<IWindowSystem>();
00043         _messageBus = DiContainer.Get<IMessageBus>();
00044         _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00045         _messageBus.AddListener<SelectedElementMessage>(OnSelectedElementMessageHandler);
00046         _messageBus.AddListener<ResetSelectionMessage>(OnResetSelectionMessageHandler);
00047         _messageBus.AddListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00048         _messageBus.AddListener<GeneratorOpenStartMessage>(OnGeneratorOpenStartMessageHandler);
00049         _messageBus.AddListener<GeneratorOpenedMessage>(OnGeneratorOpenedMessageHandler);
00050         _messageBus.AddListener<GeneratorRestoredMessage>(OnGeneratorRestoredMessageHandler);
00051         _messageBus.AddListener<CheckGeneratorMessage>(OnCheckGeneratorMessageHandler);
00052         _messageBus.AddListener<UnlockElementMessage>(OnUnlockElementMessageHandler);
00053     }
00054
00055     private void OnDestroy()
00056     {
00057         _infoButton.onClick.RemoveListener(OnItemInfoClicked);
00058
00059         _messageBus?.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00060         _messageBus?.RemoveListener<SelectedElementMessage>(OnSelectedElementMessageHandler);
00061         _messageBus?.RemoveListener<ResetSelectionMessage>(OnResetSelectionMessageHandler);
00062         _messageBus?.RemoveListener<RemoveElementMessage>(OnRemoveElementMessageHandler);
00063
00064         _messageBus?.RemoveListener<GeneratorOpenStartMessage>(OnGeneratorOpenStartMessageHandler);
00065         _messageBus?.RemoveListener<GeneratorOpenedMessage>(OnGeneratorOpenedMessageHandler);
00066         _messageBus?.RemoveListener<GeneratorRestoredMessage>(OnGeneratorRestoredMessageHandler);
00067         _messageBus?.RemoveListener<CheckGeneratorMessage>(OnCheckGeneratorMessageHandler);
00068         _messageBus?.RemoveListener<UnlockElementMessage>(OnUnlockElementMessageHandler);
00069     }
00070
00071     private void OnLoadedGameMessageHandler(LoadedGameMessage message)
00072     {
00073         _fieldLogicModel = DiContainer.Get<FieldLogicModel>();
00074
00075         private void DefineInfoTypes(IFieldElement fieldElement)
00076     {
00077         foreach (var elementInfo in _elementInfoDict)
00078         {
00079             elementInfo.Value.TrySetup(fieldElement);
00080         }
00081     }
00082
00083         private void OnSelectedElementMessageHandler(SelectedElementMessage message)
00084     {
00085             _selectedElement = _fieldLogicModel.FieldElements[message.Point];
00086             UpdatePanel(message.Point);
00087     }
00088
00089         private void OnResetSelectionMessageHandler(ResetSelectionMessage message)
00090     {
00091             ResetSelection();
00092     }
00093
00094         private void OnRemoveElementMessageHandler(RemoveElementMessage message)
00095     {
00096             if (_selectedElement != null &&
00097                 _selectedElement.InfoParameters.LogicPosition == message.RemoveAtPoint)
00098             {
00099                 ResetSelection();
00100             }
00101     }
00102
00103         private void OnGeneratorOpenStartMessageHandler(GeneratorOpenStartMessage message)
00104     {
00105             UpdatePanel(message.GeneratorPoint);
00106     }
00107
```

```

00108     private void OnGeneratorOpenedMessageHandler(GeneratorOpenedMessage message)
00109     {
00110         UpdatePanel(message.GeneratorPoint);
00111     }
00112
00113     private void OnGeneratorRestoredMessageHandler(GeneratorRestoredMessage message)
00114     {
00115         UpdatePanel(message.GeneratorPoint);
00116     }
00117
00118     private void OnCheckGeneratorMessageHandler(CheckGeneratorMessage message)
00119     {
00120         UpdatePanel(message.GeneratorPoint);
00121     }
00122
00123     private void OnUnlockElementMessageHandler(UnlockElementMessage message)
00124     {
00125         UpdatePanel(message.Element.InfoParameters.LogicPosition);
00126     }
00127
00128     private void ResetSelection()
00129     {
00130         _infoButton.gameObject.SetActive(false);
00131         _selectedElement = null;
00132         _commonLabel.gameObject.SetActive(true);
00133
00134         foreach (var elementInfo in _elementInfoDict)
00135         {
00136             elementInfo.Value.gameObject.SetActive(false);
00137         }
00138     }
00139
00140     private void UpdatePanel(GridPoint point)
00141     {
00142         if (_selectedElement == null)
00143         {
00144             return;
00145         }
00146
00147         _infoButton.gameObject.SetActive(true);
00148
00149         IFieldElement element = _fieldLogicModel.FieldElements[point];
00150
00151         if (element != _selectedElement)
00152         {
00153             return;
00154         }
00155
00156         _commonLabel.gameObject.SetActive(false);
00157
00158         DefineInfoTypes(_selectedElement);
00159     }
00160
00161     private void OnItemInfoClicked()
00162     {
00163         if (_selectedElement != null)
00164         {
00165             var infoArgs = new ElementInfoArgs {ElementConfig =
00166             _selectedElement?.ConfigParameters.ElementConfig};
00167
00168             _windowSystem.OpenWindow<ElementInfoPresenter>(enableBlackout: true, args: infoArgs);
00169         }
00170     }
00171 }

```

7.330 ElementOpenInfo.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Types;
00004 using MergeIt.Core.Helpers;
00005 using TMPro;
00006 using UnityEngine;
00007 using UnityEngine.UI;
00008
00009 namespace MergeIt.Game.UI.InfoPanel
00010 {
00011     public class ElementOpenInfo : ElementInfo
00012     {
00013         [SerializeField]
00014         private TMP_Text _openTime;
00015

```

```

00016     [SerializeField]
00017     private Button _openButton;
00018
00019     public override ElementActionType ActionType => ElementActionType.Open;
00020
00021     protected override void OnStart()
00022     {
00023         base.OnStart();
00024
00025         _openButton.onClick.AddListener(ActionButtonClick);
00026     }
00027
00028     protected override bool OnTrySetup()
00029     {
00030         if (SelectedElement.InfoParameters.Type == ElementType.Generator &&
00031             !SelectedElement.InfoParameters.IsBlocked)
00032         {
00033             if (SelectedElement.GeneratorOpenParameters is { IsOpening: false})
00034             {
00035                 int openTime =
00036                     SelectedElement.ConfigParameters.ElementConfig.GeneratorSettings.OpenTime;
00037
00038                 gameObject.SetActive(true);
00039
00040                 _openTime.text = openTime.FormatTime();
00041
00042                 return true;
00043             }
00044         }
00045
00046         return false;
00047     }
00048
00049     protected override void Destroy()
00050     {
00051         base.Destroy();
00052
00053         _openButton.onClick.RemoveListener(ActionButtonClick);
00054     }
00055 }
00056 }
```

7.331 ElementSellInfo.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.Configs.Types;
00005 using TMPro;
00006 using UnityEngine;
00007 using UnityEngine.UI;
00008
00009 namespace MergeIt.Game.UI.InfoPanel
00010 {
00011     public class ElementSellInfo : ElementInfo
00012     {
00013         [SerializeField]
00014         private TMP_Text _costLabel;
00015
00016         [SerializeField]
00017         private Image _costTypeIcon;
00018
00019         [SerializeField]
00020         private Button _sellButton;
00021
00022         public override ElementActionType ActionType
00023         {
00024             get => ElementActionType.Sell;
00025         }
00026
00027         protected override void OnStart()
00028         {
00029             base.OnStart();
00030
00031             _sellButton.onClick.AddListener(ActionButtonClick);
00032         }
00033
00034         protected override bool OnTrySetup()
00035         {
00036             CurrencySettings sellCostSettings =
00037                 SelectedElement.ConfigParameters.ElementConfig.CommonSettings.SellCostSettings;
00038
00039             if (sellCostSettings.Amount > 0 && !SelectedElement.InfoParameters.IsBlocked)
```

```

00040         {
00041             if (SelectedElement.InfoParameters.Type == ElementType.Generator)
00042             {
00043                 if (SelectedElement.GeneratorParameters.AvailableToDrop == 0)
00044                 {
00045                     return false;
00046                 }
00047             }
00048             gameObject.SetActive(true);
00049
00050             _costTypeIcon.sprite = ConfigsService.GetCurrencyIcon(skipChargeCostSettings.Currency);
00051             _costLabel.text = skipChargeCostSettings.Amount.ToString();
00052
00053             return true;
00054         }
00055     }
00056
00057     return false;
00058 }
00059
00060 protected override void Destroy()
00061 {
00062     base.Destroy();
00063
00064     _sellButton.onClick.AddListener(ActionButtonClick);
00065 }
00066 }
00067 }
```

7.332 ElementSpeedUpChargingInfo.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.Configs.Types;
00005 using TMPro;
00006 using UnityEngine;
00007 using UnityEngine.UI;
00008
00009 namespace MergeIt.Game.UI.InfoPanel
00010 {
00011     public class ElementSpeedUpChargingInfo : ElementInfo
00012     {
00013         [SerializeField]
00014         private TMP_Text _costLabel;
00015
00016         [SerializeField]
00017         private Image _costTypeIcon;
00018
00019         [SerializeField]
00020         private Button _skipChargeButton;
00021
00022         public override ElementActionType ActionType
00023         {
00024             get => ElementActionType.SkipCharging;
00025         }
00026
00027         protected override void OnStart()
00028         {
00029             base.OnStart();
00030
00031             _skipChargeButton.onClick.AddListener(OnActionButtonClick);
00032         }
00033
00034         protected override bool OnTrySetup()
00035         {
00036             if (SelectedElement.InfoParameters.Type == ElementType.Generator &&
00037                 !SelectedElement.InfoParameters.IsBlocked)
00038             {
00039                 CurrencySettings skipChargeCostSettings =
00040
00041                 SelectedElement.ConfigParameters.ElementConfig.GeneratorSettings.SkipChargeCostSettings;
00042
00043                 bool generatorCharging = SelectedElement.GeneratorParameters.AvailableToDrop == 0;
00044
00045                 if (generatorCharging && skipChargeCostSettings.Amount > 0)
00046                 {
00047                     gameObject.SetActive(true);
00048
00049                     _costTypeIcon.sprite =
00050                         ConfigsService.GetCurrencyIcon(skipChargeCostSettings.Currency);
00051                     _costLabel.text = skipChargeCostSettings.Amount.ToString();
00052
00053                 }
00054             }
00055         }
00056
00057     }
00058 }
```

```

00051             return true;
00052         }
00053     }
00054
00055     return false;
00056 }
00057
00058 protected override void Destroy()
00059 {
00060     base.Destroy();
00061
00062     _skipChargeButton.onClick.AddListener(OnActionButtonClick);
00063 }
00064 }
00065 }
```

7.333 ElementSpeedUpOpeningInfo.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.Configs.Types;
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Core.Helpers;
00007 using MergeIt.Core.Utils;
00008 using TMPro;
00009 using UnityEngine;
00010 using UnityEngine.UI;
00011
00012 namespace MergeIt.Game.UI.InfoPanel
00013 {
00014     public class ElementSpeedUpOpeningInfo : ElementInfo
00015     {
00016         [SerializeField]
00017         private TMP_Text _costLabel;
00018
00019         [SerializeField]
00020         private TMP_Text _openingTimerLabel;
00021
00022         [SerializeField]
00023         private Image _costTypeIcon;
00024
00025         [SerializeField]
00026         private Button _skipOpeningButton;
00027
00028         private Bindable<float> _remainingTime;
00029
00030         public override ElementActionType ActionType
00031         {
00032             get => ElementActionType.SkipOpening;
00033         }
00034
00035         protected override void OnStart()
00036         {
00037             base.OnStart();
00038             _skipOpeningButton.onClick.AddListener(OnActionButtonClick);
00039         }
00040
00041         protected override bool OnTrySetup()
00042         {
00043             if (SelectedElement.InfoParameters.Type == ElementType.Generator &&
00044                 !SelectedElement.InfoParameters.IsBlocked)
00045             {
00046                 CurrencySettings skipOpenCostSettings =
00047
SelectedElement.ConfigParameters.ElementConfig.GeneratorSettings.SkipOpenCostSettings;
00048
IGeneratorOpenParameters openParameters = SelectedElement.GeneratorOpenParameters;
00049
00050
if (openParameters is { IsOpening: true } &&
00051             openParameters.RemainingTime.Value > 0f)
00052             {
00053                 gameObject.SetActive(true);
00054
00055                 _remainingTime?.Unsubscribe(OnRemainingTimeChanged);
00056                 _remainingTime = openParameters.RemainingTime;
00057
00058                 _remainingTime.Subscribe(OnRemainingTimeChanged);
00059                 _openingTimerLabel.text = _remainingTime.Value.FormatTime();
00060
ConfiggsService.GetCurrencyIcon(skipOpenCostSettings.Currency);
00061
00062             if (skipOpenCostSettings.Amount == 0)
00063             {
```

```

00064             _skipOpeningButton.gameObject.SetActive(false);
00065         }
00066     else
00067     {
00068         _costLabel.text = skipOpenCostSettings.Amount.ToString();
00069     }
00070
00071     return true;
00072 }
00073 }
00074
00075     return false;
00076 }
00077
00078     protected override void OnClear()
00079     {
00080         base.OnClear();
00081
00082         _remainingTime?.Unsubscribe(OnRemainingTimeChanged);
00083         _remainingTime = null;
00084     }
00085
00086     private void OnRemainingTimeChanged(float newValue)
00087     {
00088         _openingTimerLabel.text = newValue.FormatTime();
00089     }
00090 }
00091 }
```

7.334 ElementSplitInfo.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.Helpers;
00005 using MergeIt.Game.Helpers;
00006 using MergeIt.Game.Services;
00007 using MergeIt.SimpleDI;
00008 using TMPro;
00009 using UnityEngine;
00010 using UnityEngine.UI;
00011
00012 namespace MergeIt.Game.UI.InfoPanel
00013 {
00014     public class ElementSplitInfo : ElementInfo
00015     {
00016         [SerializeField]
00017         private TMP_Text _costLabel;
00018
00019         [SerializeField]
00020         private TMP_Text _splittersAmountLabel;
00021
00022         [SerializeField]
00023         private Button _splitButton;
00024
00025         private UserServiceModel _userServiceModel;
00026
00027         public override ElementActionType ActionType
00028         {
00029             get => ElementActionType.Split;
00030         }
00031
00032         protected override void OnStart()
00033         {
00034             base.OnStart();
00035
00036             TrySetupUser();
00037
00038             _splitButton.onClick.AddListener(ActionButtonClick);
00039         }
00040
00041         protected override bool OnTrySetup()
00042         {
00043             TrySetupUser();
00044
00045             ElementCommonSettings commonSettings =
00046             SelectedElement.ConfigParameters.ElementConfig.CommonSettings;
00047
00048             if (!SelectedElement.InfoParameters.IsBlocked &&
00049                 SelectedElement.GetPreviousInEvolution() &&
00050                 commonSettings.SplitCostSettings.Amount > 0)
00051             {
00052                 gameObject.SetActive(true);
00053             }
00054         }
00055     }
00056 }
```

```

00052         _costLabel.text = commonSettings.SplitCostSettings.Amount.ToString();
00053
00054         return true;
00055     }
00056
00057     return false;
00058 }
00059
00060
00061     protected override void Destroy()
00062     {
00063         base.Destroy();
00064
00065         _splitButton.onClick.AddListener(ActionButtonClick);
00066         _userServiceModel?.Splitters.Unsubscribe(OnSplittersValueChanged);
00067     }
00068
00069     private void TrySetupUser()
00070     {
00071         if (_userServiceModel == null)
00072         {
00073             _userServiceModel = DiContainer.Get<UserServiceModel>();
00074             _userServiceModel.Splitters.Subscribe(OnSplittersValueChanged, true);
00075         }
00076     }
00077
00078     private void OnSplittersValueChanged(int newValue)
00079     {
00080         MainThreadDispatcher.Instance.RunOnMainThread(() => { _splittersAmountLabel.text =
00081             _userServiceModel.Splitters.Value.ToString(); });
00082     }
00083 }

```

7.335 ElementUnlockInfo.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using TMPro;
00005 using UnityEngine;
00006 using UnityEngine.UI;
00007
00008 namespace MergeIt.Game.UI.InfoPanel
00009 {
00010     public class ElementUnlockInfo : ElementInfo
00011     {
00012         [SerializeField]
00013         private TMP_Text _costLabel;
00014
00015         [SerializeField]
00016         private Image _costTypeIcon;
00017
00018         [SerializeField]
00019         private Button _unlockButton;
00020
00021         public override Element ActionType ActionType
00022         {
00023             get => Element ActionType.Unlock;
00024         }
00025
00026         protected override void OnStart()
00027         {
00028             base.OnStart();
00029
00030             _unlockButton.onClick.AddListener(ActionButtonClick);
00031         }
00032
00033         protected override bool OnTrySetup()
00034         {
00035             CurrencySettings unlockCostSettings =
00036                 SelectedElement.ConfigParameters.ElementConfig.CommonSettings.UnlockCostSettings;
00037
00038             if (unlockCostSettings.Amount > 0 && SelectedElement.InfoParameters.IsBlocked)
00039             {
00040                 gameObject.SetActive(true);
00041
00042                 _costTypeIcon.sprite = ConfigsService.GetCurrencyIcon(unlockCostSettings.Currency);
00043                 _costLabel.text = unlockCostSettings.Amount.ToString();
00044
00045             return true;
00046         }
00047

```

```

00048         return false;
00049     }
00050
00051     protected override void Destroy()
00052     {
00053         base.Destroy();
00054
00055         _unlockButton.onClick.AddListener(ActionButtonClick);
00056     }
00057 }
00058 }
```

7.336 InventoryPanelComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using System.Linq;
00005 using MergeIt.Core.FieldElements;
00006 using MergeIt.Core.Messages;
00007 using MergeIt.Core.Services;
00008 using MergeIt.Game.Factories.Inventory;
00009 using MergeIt.Game.Field;
00010 using MergeIt.Game.Messages;
00011 using MergeIt.Game.Services;
00012 using MergeIt.SimpleDI;
00013 using UnityEngine;
00014 using UnityEngine.UI;
00015
00016 namespace MergeIt.Game.UI.InventoryPanel
00017 {
00018     public class InventoryPanelComponent : MonoBehaviour
00019     {
00020         [SerializeField]
00021         private Button _openButton;
00022
00023         [SerializeField]
00024         private RectTransform _itemsContent;
00025
00026         private float _contentHeight;
00027         private FieldLogicModel _fieldLogicModel;
00028         private IGameFieldService _gameFieldService;
00029         private IInventoryFactory _inventoryFactory;
00030         private IInventoryService _inventoryService;
00031         private InventoryServiceModel _inventoryServiceModel;
00032         private readonly HashSet<InventoryPanelItemPair> _items = new();
00033
00034         private IMessageBus _messageBus;
00035         private RectTransform _rectTransform;
00036
00037         private void Start()
00038         {
00039             _rectTransform = GetComponent<RectTransform>();
00040             _openButton.onClick.AddListener(OnOpenInventoryClick);
00041
00042             _inventoryFactory = DiContainer.Get<IInventoryFactory>();
00043             _fieldLogicModel = DiContainer.Get<FieldLogicModel>();
00044             _inventoryServiceModel = DiContainer.Get<InventoryServiceModel>();
00045             _inventoryService = DiContainer.Get<IInventoryService>();
00046             _gameFieldService = DiContainer.Get<IGameFieldService>();
00047
00048             _messageBus = DiContainer.Get<IMessageBus>();
00049             _messageBus.AddListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00050             _messageBus.AddListener<EndDragElementMessage>(OnEndDragElementMessageHandler);
00051
00052             _contentHeight = _itemsContent.rect.height;
00053         }
00054
00055         private void OnDestroy()
00056         {
00057             _openButton.onClick.RemoveListener(OnOpenInventoryClick);
00058
00059             if (_messageBus != null)
00060             {
00061                 _messageBus.RemoveListener<LoadedGameMessage>(OnLoadedGameMessageHandler);
00062                 _messageBus.RemoveListener<EndDragElementMessage>(OnEndDragElementMessageHandler);
00063
00064                 _messageBus.RemoveListener<RemoveFromInventoryMessage>(OnRemoveFromInventoryMessageHandler);
00065             }
00066
00067             private void OnLoadedGameMessageHandler(LoadedGameMessage __)
00068             {
```

```
00069         _messageBus.AddListener<RemoveFromInventoryMessage>(OnRemoveFromInventoryMessageHandler);
00070
00071         var elements = _inventoryServiceModel.InventoryElements;
00072
00073         if (elements != null)
00074         {
00075             for (int i = 0; i < elements.Count; i++)
00076             {
00077                 CreateItem(elements[i]);
00078             }
00079         }
00080     }
00081
00082     private void OnEndDragElementMessageHandler(EndDragElementMessage message)
00083     {
00084         if (!message.ToGameObject)
00085         {
00086             return;
00087         }
00088
00089         if (_inventoryServiceModel.IsFull())
00090         {
00091             return;
00092         }
00093
00094         if (!_fieldLogicModel.FieldElements.TryGetValue(message.FromPoint, out IFieldElement
00095             element))
00096         {
00097             return;
00098         }
00099
00100        bool isInRect = RectTransformUtility.RectangleContainsScreenPoint(_rectTransform,
00101            message.Position, Camera.main);
00102
00103        if (isInRect)
00104        {
00105            CreateItem(element);
00106
00107            _inventoryService.Add(element);
00108
00109            var remove = new RemoveElementMessage
00110            {
00111                RemoveAtPoint = message.FromPoint
00112            };
00113            _messageBus.Fire(remove);
00114        }
00115
00116     private void OnRemoveFromInventoryMessageHandler(RemoveFromInventoryMessage message)
00117     {
00118         RemoveItemFromInventory(message.FieldElement);
00119     }
00120
00121     private void CreateItem(IFieldElement fieldElement)
00122     {
00123         InventoryPanelItemComponent component =
00124             _inventoryFactory.CreateInventoryPanelItem(fieldElement);
00125         component.transform.SetParent(_itemsContent);
00126         component.transform.localScale = Vector3.one;
00127         component.RectTransform.sizeDelta = new Vector2(_contentHeight, _contentHeight);
00128
00129         component.ClickEvent += OnItemClick;
00130
00131         _items.Add(new InventoryPanelItemPair(component, fieldElement));
00132     }
00133
00134     private void OnItemClick(InventoryPanelItemComponent item)
00135     {
00136         Vector3 fromPosition = item.gameObject.transform.position;
00137         var freeCell = _gameFieldService.GetFreeCell();
00138         if (freeCell != null)
00139         {
00140             item.ClickEvent -= OnItemClick;
00141             InventoryPanelItemPair pair = _items.FirstOrDefault(x => x.Component == item);
00142             if (pair != null)
00143             {
00144                 pair.Element.InfoParameters.LogicPosition = freeCell.Value;
00145                 _messageBus.Fire(new CreateElementMessage
00146                 {
00147                     NewElement = pair.Element,
00148                     FromPosition = fromPosition,
00149                     ToPoint = freeCell.Value
00150                 });
00151
00152             _inventoryService.Remove(pair.Element);
00153             Destroy(pair.Component.gameObject);
00154         }
00155     }
00156 }
```

```

00153             _items.Remove(pair);
00154         }
00155     }
00156 }
00157
00158     private void RemoveItemFromInventory(IFieldElement fieldElement)
00159 {
00160     InventoryPanelItemPair pair = _items.FirstOrDefault(x => x.Element == fieldElement);
00161
00162     if (pair is {Component: not null})
00163     {
00164         Destroy(pair.Component.gameObject);
00165         _items.Remove(pair);
00166     }
00167 }
00168
00169     private void OnOpenInventoryClick()
00170 {
00171     _inventoryService.OpenWindow();
00172 }
00173 }
00174 }
```

7.337 InventoryPanelItemComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using UnityEngine;
00005 using UnityEngine.UI;
00006
00007 namespace MergeIt.Game.UI.InventoryPanel
00008 {
00009     public class InventoryPanelItemComponent : MonoBehaviour
00010     {
00011         [SerializeField]
00012         private Button _button;
00013
00014         [SerializeField]
00015         private GameObject _maxLevelIcon;
00016
00017         private RectTransform _rectTransform;
00018
00019         public RectTransform RectTransform
00020         {
00021             get
00022             {
00023                 if (!_rectTransform)
00024                 {
00025                     _rectTransform = GetComponent<RectTransform>();
00026                 }
00027
00028                 return _rectTransform;
00029             }
00030         }
00031
00032         private void Awake()
00033         {
00034             _button.onClick.AddListener(OnClick);
00035         }
00036
00037         private void OnDestroy()
00038         {
00039             _button.onClick.RemoveListener(OnClick);
00040         }
00041         public event Action<InventoryPanelItemComponent> ClickEvent;
00042
00043         private void OnClick()
00044         {
00045             ClickEvent?.Invoke(this);
00046         }
00047     }
00048 }
```

7.338 InventoryPanelItemPair.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
```

```

00004
00005 namespace MergeIt.Game.UI.InventoryPanel
00006 {
00007     public class InventoryPanelItemPair
00008     {
00009         public InventoryPanelItemPair(InventoryPanelItemComponent component, IFieldElement
00010             fieldElement)
00011         {
00012             Component = component;
00013             Element = fieldElement;
00014         }
00015         public InventoryPanelItemComponent Component { get; }
00016
00017         public IFieldElement Element { get; }
00018     }
00019 }

```

7.339 UserData.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using MergeIt.Core.Saves;
00005 using MergeIt.Core.User;
00006 using Newtonsoft.Json;
00007
00008 namespace MergeIt.Game.User
00009 {
00010     [Serializable, Savable("user", "dat")]
00011     public class UserData : IUserData
00012     {
00013         [JsonProperty("n")]
00014         public string Name { get; set; }
00015         [JsonProperty("e")]
00016         public int Energy { get; set; }
00017         [JsonProperty("s")]
00018         public int SoftCurrency { get; set; }
00019         [JsonProperty("h")]
00020         public int HardCurrency { get; set; }
00021         [JsonProperty("sp")]
00022         public int Splitters { get; set; }
00023         [JsonProperty("l")]
00024         public int Level { get; set; }
00025         [JsonProperty("ex")]
00026         public int Experience { get; set; }
00027         [JsonProperty("er")]
00028         public long EnergyRestoringStartTime { get; set; }
00029     }
00030 }

```

7.340 ElementInfoArgs.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Configs.Elements;
00004 using MergeIt.Core.WindowSystem.Data;
00005
00006 namespace MergeIt.Game.Windows.ElementInfo
00007 {
00008     public class ElementInfoArgs : WindowArgs
00009     {
00010         public ElementConfig ElementConfig { get; set; }
00011     }
00012 }

```

7.341 ElementInfoItemComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using TMPro;
00004 using UnityEngine;
00005 using UnityEngine.Events;
00006 using UnityEngine.UI;
00007

```

```
00008 namespace MergeIt.Game.Windows.ElementInfo
00009 {
00010     public class ElementInfoItemComponent : MonoBehaviour
00011     {
00012         [SerializeField]
00013         private Button _button;
00014
00015         [SerializeField]
00016         private Color _selectedColor;
00017
00018         [SerializeField]
00019         private Image _backgroundImage;
00020
00021         [SerializeField]
00022         private GameObject _generatorIcon;
00023
00024         [SerializeField]
00025         private Transform _iconContainer;
00026
00027         [SerializeField]
00028         private TMP_Text _numberText;
00029
00030         [SerializeField]
00031         private GameObject _numberContainer;
00032
00033         [SerializeField]
00034         private GameObject _unknownContainer;
00035
00036         [SerializeField]
00037         private GameObject _lockerObject;
00038
00039         [SerializeField]
00040         private GameObject _maxLevelIcon;
00041
00042
00043     public UnityAction<ElementInfoItemComponent> ClickEvent = delegate { };
00044
00045     public RectTransform RectTransform { get; private set; }
00046
00047     public void Awake()
00048     {
00049         RectTransform = GetComponent<RectTransform>();
00050         _button.onClick.AddListener(OnClick);
00051     }
00052
00053     public void OnDestroy()
00054     {
00055         ClickEvent = null;
00056         _button.onClick.RemoveListener(OnClick);
00057     }
00058
00059     public void Setup(bool isGenerator, bool isSelected, int number = -1)
00060     {
00061         _backgroundImage.color = isSelected ? _selectedColor : Color.white;
00062         _generatorIcon.SetActive(isGenerator);
00063
00064         if (number > -1 && _numberText)
00065         {
00066             _numberContainer.SetActive(true);
00067             _numberText.text = number.ToString();
00068         }
00069         else
00070         {
00071             _numberContainer.SetActive(false);
00072         }
00073     }
00074
00075     public void SetIcon(Transform iconTransform, bool isLocked = false)
00076     {
00077         iconTransform.SetParent(_iconContainer);
00078         iconTransform.localScale = Vector3.one;
00079
00080         if (_lockerObject)
00081         {
00082             _lockerObject.SetActive(isLocked);
00083         }
00084     }
00085
00086     private void OnClick()
00087     {
00088         ClickEvent?.Invoke(this);
00089     }
00090
00091     public void SetUnknown()
00092     {
00093         _unknownContainer.SetActive(true);
00094     }
```

```
00095     }
00096
00097 }
```

7.342 ElementInfoItemsPanelComponent.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using TMPro;
00004 using UnityEngine;
00005 using UnityEngine.UI;
00006
00007 namespace MergeIt.Game.Windows.ElementInfo
00008 {
00009     public class ElementInfoItemsPanelComponent : MonoBehaviour
00010     {
00011         [SerializeField]
00012         private TMP_Text _panelTitle;
00013
00014         [SerializeField]
00015         private GridLayoutGroup _itemsGrid;
00016
00017         private VerticalLayoutGroup _layoutGroup;
00018         private RectTransform _rectTransform;
00019
00020         public GridLayoutGroup ItemsGrid
00021         {
00022             get => _itemsGrid;
00023         }
00024
00025         public RectTransform RectTransform
00026         {
00027             get
00028             {
00029                 if (!_rectTransform)
00030                 {
00031                     _rectTransform = GetComponent<RectTransform>();
00032                 }
00033
00034                 return _rectTransform;
00035             }
00036         }
00037
00038         public TMP_Text PanelTitle
00039         {
00040             get => _panelTitle;
00041         }
00042
00043         private void Awake()
00044         {
00045             _layoutGroup = GetComponent<VerticalLayoutGroup>();
00046         }
00047
00048         public void SetSpacing(float spacing)
00049         {
00050             _layoutGroup.spacing = spacing;
00051         }
00052
00053         public void SetTitle(string titleText)
00054         {
00055             _panelTitle.text = titleText;
00056         }
00057
00058         public void SetItem(Transform item)
00059         {
00060             item.SetParent(_itemsGrid.transform);
00061             item.localScale = Vector3.one;
00062         }
00063     }
00064 }
```

7.343 ElementInfoModel.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.WindowSystem.Windows;
00004
00005 namespace MergeIt.Game.Windows.ElementInfo
00006 {
```

```

00007     public class ElementInfoModel : WindowModel
00008     {
00009     }
00010 }
```

7.344 ElementInfoPresenter.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using System.Linq;
00005 using MergeIt.Core.Configs.Data;
00006 using MergeIt.Core.Configs.Elements;
00007 using MergeIt.Core.Configs.Types;
00008 using MergeIt.Core.Schemes;
00009 using MergeIt.Core.Services;
00010 using MergeIt.Core.WindowSystem.Data;
00011 using MergeIt.Core.WindowSystem.Windows;
00012 using MergeIt.Game.Factories.ElementInfo;
00013 using MergeIt.SimpleDI;
00014 using UnityEngine;
00015 using UnityEngine.UI;
00016
00017 namespace MergeIt.Game.Windows.ElementInfo
00018 {
00019     public class ElementInfoPresenter : WindowPresenter<ElementInfoWindow, ElementInfoModel>
00020     {
00021
00022         private IConfigsService _configsService;
00023
00024         private readonly Dictionary<ElementInfoItemComponent, ElementConfig> _elementConfigs = new();
00025         private ElementInfoArgs _elementInfoArgs;
00026         private IElememtInfoFactory _elementInfoFactory;
00027         private IEvolutionsService _evolutionsService;
00028         private RectTransform _prevSetUpPanel;
00029         private ElementConfig _selectedConfig;
00030
00031         protected override void OnInitialize(IWindowArgs args = null)
00032         {
00033             base.OnInitialize(args);
00034
00035             _elementInfoFactory = DiContainer.Get<IElememtInfoFactory>();
00036             _configsService = DiContainer.Get<IConfigsService>();
00037             _evolutionsService = DiContainer.Get<IEvolutionsService>();
00038
00039             _elementInfoArgs = args as ElementInfoArgs;
00040
00041             if (_elementInfoArgs != null)
00042             {
00043                 _selectedConfig = _elementInfoArgs.ElementConfig;
00044                 SchemeObject evolutionConfig = _configsService.LevelConfig.EvolutionsScheme;
00045                 EvolutionData evolutionData = evolutionConfig.Evolution.FirstOrDefault(x =>
00046                     x.Chain.Contains(_selectedConfig));
00047
00048                 View.WindowTitleText.text = evolutionData?.Name;
00049
00050                 var elementChain = _evolutionsService.GetEvolutionChain(evolutionData);
00051                 var generatorsChain = _evolutionsService.GetGeneratedBy(_selectedConfig);
00052                 var generates = _evolutionsService.GetGenerates(_selectedConfig);
00053
00054                 bool lockedElements = _selectedConfig.Type == ElementType.Generator &&
00055                     _selectedConfig.GeneratorSettings.GenerateBlocked;
00056
00057                 SetupPanel(View.EvolutionPanel, elementChain, _prevSetUpPanel,
00058                     evolutionData.Description, evolutionData, true);
00059                 SetupPanel(View.GeneratesPanel, generates, _prevSetUpPanel, "Produce:", lockedItems:
00060                     lockedElements);
00061                 SetupPanel(View.CreatedFromPanel, generatorsChain, _prevSetUpPanel, "Created from:");
00062
00063                 CalculateScrollSize(View.EvolutionPanel.RectTransform,
00064                     View.GeneratesPanel.RectTransform, View.CreatedFromPanel.RectTransform);
00065             }
00066
00067         protected override void OnDestroyWindow()
00068         {
00069             base.OnDestroyWindow();
00070
00071             foreach (var elementConfig in _elementConfigs)
00072             {
00073                 if (elementConfig.Key)
00074                 {
00075                     elementConfig.Key.ClickEvent -= OnClickElement;
```

```
00072         }
00073     }
00074 
00075     _elementConfigs.Clear();
00076 }
00077 
00078     private void SetupPanel(ElementInfoItemsPanelComponent panelComponent, List<ElementConfig>
00079     elementConfigs,
00080     RectTransform prevPanelTransform, string panelTitle, EvolutionData evolutionData = null,
00081     bool showNumbers = false,
00082     bool lockedItems = false)
00083 {
00084     if (elementConfigs.Count > 0)
00085     {
00086         panelComponent.gameObject.SetActive(true);
00087         panelComponent.SetSpacing(View.SpacingInsidePanels);
00088         panelComponentSetTitle(panelTitle);
00089 
00090         SetupPanelSize(panelComponent, elementConfigs.Count, prevPanelTransform);
00091 
00092         for (int i = 0; i < elementConfigs.Count; i++)
00093         {
00094             ElementConfig elementConfig = elementConfigs[i];
00095             ElementInfoItemComponent item = null;
00096             bool isDiscovered = true;
00097 
00098             if (evolutionData != null)
00099             {
00100                 int index = evolutionData.Chain.IndexOf(elementConfig) + 1;
00101                 int progress = _evolutionsService.GetEvolutionProgress(evolutionData.Id);
00102                 isDiscovered = index <= progress;
00103 
00104                 if (isDiscovered)
00105                 {
00106                     item = CreateComponent(elementConfig, lockedItems);
00107                 }
00108                 else
00109                 {
00110                     item = CreateComponent(elementConfig, lockedItems);
00111                 }
00112 
00113                 if (item)
00114                 {
00115                     bool isGenerator = isDiscovered && elementConfig.Type ==
00116                     ElementType.Generator;
00117                     panelComponent.SetItem(item.transform);
00118 
00119                     item.Setup(isGenerator, elementConfig == _selectedConfig, showNumbers ? i + 1
00120 : -1);
00121                 }
00122             }
00123         }
00124 
00125         _prevSetUpPanel = panelComponent.RectTransform;
00126     }
00127     else
00128     {
00129         panelComponent.gameObject.SetActive(false);
00130     }
00131 }
00132 
00133     private ElementInfoItemComponent CreateComponent(ElementConfig elementConfig, bool isLocked =
00134     false)
00135     {
00136         ElementInfoItemComponent item = _elementInfoFactory.CreateElementWindowItem(elementConfig,
00137         isLocked: isLocked);
00138         item.ClickEvent += OnClickElement;
00139 
00140         _elementConfigs[item] = elementConfig;
00141 
00142         return item;
00143     }
00144 
00145     private void SetupPanelSize(ElementInfoItemsPanelComponent panel, int itemsCount,
00146     RectTransform prevPanelTransform)
00147     {
00148         GridLayoutGroup grid = panel.ItemsGrid;
00149         float cellHeight = grid.cellSize.y;
00150         int rows = Mathf.CeilToInt((float)itemsCount / grid.constraintCount);
00151 
00152         float resultSize = 0f;
00153         resultSize += panel.PanelTitle.textInfo.textComponent.rectTransform.rect.height;
00154         resultSize += View.SpacingInsidePanels;
```

```

00152         float gridSize = 0f;
00153         gridSize += cellHeight * rows;
00154         gridSize += grid.spacing.y * (rows - 1);
00155         gridSize += grid.padding.top + grid.padding.bottom;
00156
00157         var gridRect = panel.ItemsGrid.GetComponent<RectTransform>();
00158         Vector2 panelSize = panel.RectTransform.sizeDelta;
00159         Vector2 gridLayoutSize = gridRect.sizeDelta;
00160
00161         panelSize.y = resultSize + gridSize;
00162         gridLayoutSize.y = gridSize;
00163
00164         gridRect.sizeDelta = gridLayoutSize;
00165         panel.RectTransform.sizeDelta = panelSize;
00166
00167         if (prevPanelTransform != default)
00168         {
00169             float newPosition = prevPanelTransform.anchoredPosition.y - View.SpacingBetweenPanels
00170             - prevPanelTransform.sizeDelta.y;
00171             panel.RectTransform.anchoredPosition = new
00172             Vector2(panel.RectTransform.anchoredPosition.x, newPosition);
00173         }
00174     }
00175
00176     private void CalculateScrollSize(params RectTransform[] panels)
00177     {
00178         float sumHeight =
00179             panels.Sum(x => x.gameObject.activeSelf ? x.sizeDelta.y : 0f) + panels.Length *
00180             View.SpacingBetweenPanels;
00181
00182         View.ScrollContent.sizeDelta = new Vector2(View.ScrollContent.sizeDelta.x, sumHeight);
00183     }
00184
00185     private void OnClickElement(ElementInfoItemComponent component)
00186     {
00187         if (_elementConfigs.TryGetValue(component, out ElementConfig config) &&
00188             config != _selectedConfig)
00189         {
00190             var args = new ElementInfoArgs
00191             {
00192                 ElementConfig = config
00193             };
00194
00195             WindowSystem.OpenWindow<ElementInfoPresenter>(true, true, args);
00196         }
00197     }
00198 }
```

7.345 ElementInfoWindow.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.WindowSystem.Windows;
00004 using TMPro;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Game.Windows.ElementInfo
00008 {
00009     public class ElementInfoWindow : WindowBase
00010     {
00011         [SerializeField]
00012         private TMP_Text _windowTitleText;
00013
00014         [SerializeField]
00015         private float _spacingBetweenPanels;
00016
00017         [SerializeField]
00018         private float _spacingInsidePanels;
00019
00020         [SerializeField]
00021         private ElementInfoItemsPanelComponent _evolutionPanel;
00022
00023         [SerializeField]
00024         private ElementInfoItemsPanelComponent _generatesPanel;
00025
00026         [SerializeField]
00027         private ElementInfoItemsPanelComponent _createdFromPanel;
00028
00029         [SerializeField]
00030         private RectTransform _scrollContent;
00031 }
```

```

00032     public TMP_Text WindowTitleText
00033     {
00034         get => _windowTitleText;
00035     }
00036
00037     public float SpacingBetweenPanels
00038     {
00039         get => _spacingBetweenPanels;
00040     }
00041
00042     public float SpacingInsidePanels
00043     {
00044         get => _spacingInsidePanels;
00045     }
00046
00047     public ElementInfoItemsPanelComponent EvolutionPanel
00048     {
00049         get => _evolutionPanel;
00050     }
00051
00052     public ElementInfoItemsPanelComponent GeneratesPanel
00053     {
00054         get => _generatesPanel;
00055     }
00056
00057     public ElementInfoItemsPanelComponent CreatedFromPanel
00058     {
00059         get => _createdFromPanel;
00060     }
00061
00062     public RectTransform ScrollContent
00063     {
00064         get => _scrollContent;
00065     }
00066 }
00067 }
```

7.346 ElementInfoWindowContent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004
00005 namespace MergeIt.Game.Windows.ElementInfo
00006 {
00007     public class ElementInfoWindowContent : MonoBehaviour
00008     {
00009     }
00010 }
```

7.347 InventoryModel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.WindowSystem.Windows;
00004
00005 namespace MergeIt.Game.Windows.Inventory
00006 {
00007     public class InventoryModel : WindowModel
00008     {
00009     }
00010 }
```

7.348 InventoryPresenter.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using System.Linq;
00005 using MergeIt.Core.Configs.Elements;
00006 using MergeIt.Core.FieldElements;
00007 using MergeIt.Core.Services;
00008 using MergeIt.Core.WindowSystem.Data;
00009 using MergeIt.Core.WindowSystem.Windows;
00010 using MergeIt.Game.Factories.Inventory;
```

```

00011 using MergeIt.Game.Messages;
00012 using MergeIt.Game.Services;
00013 using MergeIt.Game.Windows.ElementInfo;
00014 using MergeIt.SimpleDI;
00015 using UnityEngine;
00016
00017 namespace MergeIt.Game.Windows.Inventory
00018 {
00019     public class InventoryPresenter : WindowPresenter<InventoryWindow, InventoryModel>
00020     {
00021         private IConfigsService _configsService;
00022         private ICurrencyService _currencyService;
00023         private IGameFieldService _fieldService;
00024         private int _initialSize;
00025         private IInventoryFactory _inventoryFactory;
00026         private IInventoryService _inventoryService;
00027         private InventoryServiceModel _inventoryServiceModel;
00028
00029         private List<InventoryWindowItemPair> _itemComponents;
00030         private InventoryWindowPaidCellComponent _paidCell;
00031         private CurrencySettings[] _paidCellsPrices;
00032
00033         protected override void OnInitialize(IWindowArgs args = null)
00034         {
00035             base.OnInitialize(args);
00036
00037             _inventoryService = DiContainer.Get<IInventoryService>();
00038             _inventoryServiceModel = DiContainer.Get<InventoryServiceModel>();
00039             _configsService = DiContainer.Get<IConfigsService>();
00040             _inventoryFactory = DiContainer.Get<IInventoryFactory>();
00041             _currencyService = DiContainer.Get<ICurrencyService>();
00042             _fieldService = DiContainer.Get<IGameFieldService>();
00043
00044             _paidCellsPrices = _configsService.InventoryConfig.PaidCells;
00045             _initialSize = _configsService.InventoryConfig.InitialCapacity;
00046
00047             FillGrid();
00048         }
00049
00050         protected override void OnDestroyWindow()
00051         {
00052             base.OnDestroyWindow();
00053
00054             for (int i = 0; i < _itemComponents.Count; i++)
00055             {
00056                 InventoryWindowItemPair item = _itemComponents[i];
00057                 item.Component.InfoClickEvent -= OnItemInfoClick;
00058                 item.Component.ItemClickEvent -= OnItemClick;
00059             }
00060
00061             if (_paidCell)
00062             {
00063                 _paidCell.BuyCellEvent -= OnBuyCellClick;
00064             }
00065         }
00066
00067         private void FillGrid()
00068         {
00069             _itemComponents = new List<InventoryWindowItemPair>();
00070             var inventoryElements = _inventoryServiceModel.InventoryElements;
00071             int elementsCount = inventoryElements.Count;
00072             if (elementsCount > 0)
00073             {
00074                 for (int i = 0; i < elementsCount; i++)
00075                 {
00076                     IFIELDElement element = inventoryElements[i];
00077                     InventoryWindowItemComponent cell =
00078                         _inventoryFactory.CreateInventoryWindowItem(element);
00079
00080                     if (cell)
00081                     {
00082                         cell.ItemClickEvent += OnItemClick;
00083                         cell.InfoClickEvent += OnItemInfoClick;
00084
00085                         _itemComponents.Add(new InventoryWindowItemPair(cell, element));
00086
00087                         PlaceCellInGrid(cell.transform);
00088                     }
00089                 }
00090
00091             int diff = _inventoryServiceModel.InventorySize - elementsCount;
00092             for (int i = 0; i < diff; i++)
00093             {
00094                 CreateEmptyCell();
00095             }
00096         }

```

```

00097         TrySetupPaidCell(true);
00098     }
00099
00100     private void OnItemClick(InventoryWindowItemComponent item)
00101     {
00102         item.ItemClickEvent -= OnItemClick;
00103         InventoryWindowItemPair itemPair = _itemComponents.FirstOrDefault(x => x.Component == item);
00104         if (itemPair != null)
00105         {
00106             var freeCell = _fieldService.GetFreeCell();
00107
00108             if (freeCell != null)
00109             {
00110                 if (_inventoryService.Remove(itemPair.Element))
00111                 {
00112                     int index = itemPair.Component.transform.GetSiblingIndex();
00113
00114                     Object.Destroy(itemPair.Component.gameObject);
00115
00116                     CreateEmptyCell(index);
00117
00118                     // TODO: animate
00119
00120                     MessageBus.Fire(new RemoveFromInventoryMessage
00121                     {
00122                         FieldElement = itemPair.Element
00123                     });
00124
00125                     MessageBus.Fire(new createElementMessage
00126                     {
00127                         NewElement = itemPair.Element,
00128                         FromPosition = null,
00129                         ToPoint = freeCell.Value
00130                     });
00131
00132                 }
00133             }
00134         }
00135
00136         private void OnItemInfoClick(InventoryWindowItemComponent item)
00137         {
00138             IFIELDElement fieldElement = _itemComponents.FirstOrDefault(x => x.Component == item)?.Element;
00139             var infoArgs = new ElementInfoArgs {ElementConfig =
00140                 fieldElement?.ConfigParameters.ElementConfig};
00141             WindowSystem.OpenWindow<ElementInfoPresenter>(enableBlackout: true, args: infoArgs);
00142         }
00143
00144         private void OnBuyCellClick()
00145         {
00146             int priceIndex = _inventoryServiceModel.InventorySize - _initialSize;
00147             if (_currencyService.TryPay(_paidCellsPrices[priceIndex]))
00148             {
00149                 _inventoryServiceModel.InventorySize++;
00150
00151                 GameObject emptyCell = _inventoryFactory.CreateWindowEmptyCell();
00152                 PlaceCellInGrid(emptyCell.transform);
00153
00154                 if (_paidCell)
00155                 {
00156                     int index = _paidCell.transform.GetSiblingIndex();
00157                     emptyCell.transform.SetSiblingIndex(index);
00158
00159                     TrySetupPaidCell();
00160                 }
00161             }
00162         }
00163
00164         private void TrySetupPaidCell(bool createCell = false)
00165         {
00166             int index = _inventoryServiceModel.InventorySize - _initialSize;
00167             if (index < _paidCellsPrices.Length)
00168             {
00169                 if (createCell)
00170                 {
00171                     _paidCell = _inventoryFactory.CreateWindowPaidCell();
00172                     _paidCell.BuyCellEvent += OnBuyCellClick;
00173
00174                     PlaceCellInGrid(_paidCell.transform);
00175                 }
00176
00177                 CurrencySettings priceSettings = _paidCellsPrices[index];
00178                 string priceText = priceSettings.Amount.ToString();
00179                 Sprite priceSprite = _configsService.GetCurrencyIcon(priceSettings.Currency);
00180

```

```

00181         _paidCell.Setup(priceText, priceSprite);
00182     }
00183     else if (_paidCell)
00184     {
00185         _paidCell.BuyCellEvent -= OnBuyCellClick;
00186
00187         Object.Destroy(_paidCell.gameObject);
00188
00189         _paidCell = null;
00190     }
00191 }
00192
00193 private void CreateEmptyCell(int siblingIndex = -1)
00194 {
00195     GameObject cell = _inventoryFactory.CreateWindowEmptyCell();
00196
00197     PlaceCellInGrid(cell.transform, siblingIndex);
00198 }
00199
00200 private void PlaceCellInGrid(Transform cell, int siblingIndex = -1)
00201 {
00202     cell.SetParent(View.ItemsGrid);
00203     cell.localScale = Vector3.one;
00204
00205     if (siblingIndex != -1)
00206     {
00207         cell.SetSiblingIndex(siblingIndex);
00208     }
00209 }
00210 }
00211 }
```

7.349 InventoryWindow.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.WindowSystem.Windows;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Game.Windows.Inventory
00007 {
00008     public class InventoryWindow : WindowBase
00009     {
00010         [SerializeField]
00011         private RectTransform _itemsGrid;
00012
00013         public RectTransform ItemsGrid
00014         {
00015             get => _itemsGrid;
00016         }
00017     }
00018 }
```

7.350 InventoryWidgetItemComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using UnityEngine;
00005 using UnityEngine.UI;
00006
00007 namespace MergeIt.Game.Windows.Inventory
00008 {
00009     public class InventoryWidgetItemComponent : MonoBehaviour
00010     {
00011
00012         [SerializeField]
00013         private Button _itemButton;
00014
00015         [SerializeField]
00016         private Button _infoButton;
00017
00018         [SerializeField]
00019         private RectTransform _iconContainer;
00020
00021         public void Awake()
00022         {
00023             _itemButton.onClick.AddListener(OnItemClick);
00024             _infoButton.onClick.AddListener(OnInfoClick);
00025         }
00026     }
00027 }
```

```

00025         }
00026
00027     public void OnDestroy()
00028     {
00029         _itemButton.onClick.AddListener(OnItemClick);
00030         _infoButton.onClick.AddListener(OnInfoClick);
00031     }
00032     public event Action<InventoryWidgetItemComponent> ItemClickEvent;
00033     public event Action<InventoryWidgetItemComponent> InfoClickEvent;
00034
00035     public void SetIcon(RectTransform rectTransform)
00036     {
00037         rectTransform.SetParent(_iconContainer);
00038         rectTransform.localScale = Vector3.one;
00039         ;
00040     }
00041
00042     private void OnItemClick()
00043     {
00044         ItemClickEvent?.Invoke(this);
00045     }
00046
00047     private void OnInfoClick()
00048     {
00049         InfoClickEvent?.Invoke(this);
00050     }
00051 }
00052 }
```

7.351 InventoryWindowItemPair.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.FieldElements;
00004
00005 namespace MergeIt.Game.Windows.Inventory
00006 {
00007     public class InventoryWidgetItemPair
00008     {
00009
00010         public InventoryWidgetItemPair(InventoryWidgetItemComponent component, IFieldElement
00011             fieldElement)
00011         {
00012             Component = component;
00013             Element = fieldElement;
00014         }
00015
00016         public InventoryWidgetItemComponent Component { get; }
00017
00018         public IFieldElement Element { get; }
00019     }
00020 }
```

7.352 InventoryWindowPaidCellComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using TMPro;
00005 using UnityEngine;
00006 using UnityEngine.UI;
00007
00008 namespace MergeIt.Game.Windows.Inventory
00009 {
00010     public class InventoryWindowPaidCellComponent : MonoBehaviour
00011     {
00012
00013         [SerializeField]
00014         private Button _buyButton;
00015
00016         [SerializeField]
00017         private TMP_Text _priceLabel;
00018
00019         [SerializeField]
00020         private Image _priceIcon;
00021
00022         public void Awake()
00023         {
00024             _buyButton.onClick.AddListener(OnBuyButtonClick);
```

```

00025         }
00026
00027     public void OnDestroy()
00028     {
00029         _buyButton.onClick.AddListener(OnBuyButtonClick);
00030     }
00031     public event Action BuyCellEvent;
00032
00033     public void Setup(string priceText, Sprite priceImage)
00034     {
00035         _priceLabel.text = priceText;
00036         _priceIcon.sprite = priceImage;
00037     }
00038
00039     private void OnBuyButtonClick()
00040     {
00041         BuyCellEvent?.Invoke();
00042     }
00043 }
00044 }
```

7.353 UserInfoModel.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.WindowSystem.Windows;
00004
00005 namespace MergeIt.Game.Windows.UserProgress
00006 {
00007     public class UserInfoModel : WindowModel
00008     {
00009     }
00010 }
```

7.354 UserInfoPresenter.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using MergeIt.Core.Configs.Elements;
00005 using MergeIt.Core.Messages;
00006 using MergeIt.Core.Services;
00007 using MergeIt.Core.WindowSystem.Windows;
00008 using MergeIt.Game.Effects;
00009 using MergeIt.Game.Effects.Controllers;
00010 using MergeIt.Game.Effects.Parameters;
00011 using MergeIt.Game.Enums;
00012 using MergeIt.Game.Factories.ElementInfo;
00013 using MergeIt.Game.Messages;
00014 using MergeIt.Game.Services;
00015 using MergeIt.Game.Windows.ElementInfo;
00016 using MergeIt.SimpleDI;
00017 using UnityEngine;
00018
00019 namespace MergeIt.Game.Windows.UserProgress
00020 {
00021     public class UserInfoPresenter : WindowPresenter<UserInfoWindow, UserInfoModel>
00022     {
00023         private readonly Dictionary<ElementInfoItemComponent, ElementConfig> _elementConfigs = new();
00024         private IEffectorFactory _effectorFactory;
00025         private IElementInfoFactory _elementInfoFactory;
00026         private IElementsStockService _elementsStockService;
00027
00028         private IMessageBus _messageBus;
00029         private IUserProgressService _userProgressService;
00030         private UserServiceModel _userServiceModel;
00031
00032         protected override void OnInitialize(UserInfoWindow view)
00033         {
00034             base.OnInitialize(view);
00035
00036             _userServiceModel = DiContainer.Get<UserServiceModel>();
00037             _userProgressService = DiContainer.Get<IUserProgressService>();
00038             _elementsStockService = DiContainer.Get<IElementsStockService>();
00039             _elementInfoFactory = DiContainer.Get<IElementInfoFactory>();
00040             _effectorFactory = DiContainer.Get<IEffectorFactory>();
00041
00042             _messageBus = DiContainer.Get<IMessageBus>();
00043             _messageBus.AddListener<LevelUpdatedMessage>(LevelUpdatedMessageHandler);
```

```
00044         View.LevelUpEvent += LevelUpEventHandler;
00045
00046         CheckLevelUpButtonState();
00047         FillElements();
00048         UpdateProgress();
00049     }
00050
00051
00052     protected override void OnDestroyWindow()
00053     {
00054         base.OnDestroyWindow();
00055
00056         _messageBus.RemoveListener<LevelUpdatedMessage>(LevelUpdatedMessageHandler);
00057
00058         View.LevelUpEvent -= LevelUpEventHandler;
00059
00060         base.OnDestroyWindow();
00061
00062         foreach (var elementConfig in _elementConfigs)
00063         {
00064             if (elementConfig.Key)
00065             {
00066                 elementConfig.Key.ClickEvent -= OnClickElement;
00067             }
00068         }
00069
00070         _elementConfigs.Clear();
00071     }
00072
00073     private void LevelUpEventHandler()
00074     {
00075         _messageBus.Fire(new LevelUpMessage());
00076     }
00077
00078     private void CheckLevelUpButtonState()
00079     {
00080         bool canLevelUp = _userProgressService.CanLevelUp();
00081         View.EnableLevelUpButton(canLevelUp);
00082     }
00083
00084     private void FillElements()
00085     {
00086         var prizes = _userProgressService.GetLevelUpPrizes();
00087
00088         if (prizes is {Length: > 0})
00089         {
00090             for (int i = 0; i < prizes.Length; i++)
00091             {
00092                 ElementConfig prize = prizes[i];
00093                 ElementInfoItemComponent elementInfo =
00094                     _elementInfoFactory.CreateElementWindowItem(prize,
ElementInfoType.UserProgressWindow);
00095
00096                 if (elementInfo != null)
00097                 {
00098                     _elementConfigs[elementInfo] = prize;
00099                     elementInfo.ClickEvent += OnClickElement;
00100                     elementInfo.transform.SetParent(View.PrizesContainer);
00101                     elementInfo.transform.localScale = Vector3.one;
00102                 }
00103             }
00104         }
00105     }
00106
00107     private void UpdateProgress()
00108     {
00109         int currentMaxExp = _userProgressService.GetCurrentLevelMaxExp();
00110         int currentExp = _userServiceModel.Experience.Value;
00111         int level = _userServiceModel.Level.Value;
00112
00113         View.UpdateProgress(level, currentExp, currentMaxExp);
00114     }
00115
00116     private void OnClickElement(ElementInfoItemComponent component)
00117     {
00118         if (_elementConfigs.TryGetValue(component, out ElementConfig config))
00119         {
00120             var args = new ElementInfoArgs
00121             {
00122                 ElementConfig = config
00123             };
00124
00125             WindowSystem.OpenWindow<ElementInfoPresenter>(true, true, args);
00126         }
00127     }
00128
00129     private void LevelUpdatedMessageHandler(LevelUpdatedMessage _)
```

```

00130      {
00131          View.EnableLevelUpButton(false);
00132          int effectsCount = _elementConfigs.Count;
00133          foreach (var config in _elementConfigs)
00134          {
00135              _elementsStockService.Add(config.Value);
00136
00137              RectTransform elementTransform = config.Key.RectTransform;
00138
00139              var parameters = new MoveEffectParameters(elementTransform.position, default,
00140                  HudPanelType.ElementStock, EffectName.MoveToStock);
00141              _effectsFactory.CreateEffect<MoveToUiEffect>(config.Key.RectTransform, parameters, ())
00142          =>
00143              {
00144                  effectsCount--;
00145                  Object.Destroy(elementTransform.gameObject);
00146
00147                  CheckEffects();
00148              });
00149
00150      void CheckEffects()
00151      {
00152          if (effectsCount == 0)
00153          {
00154              _elementConfigs.Clear();
00155              CheckLevelUpButtonState();
00156              FillElements();
00157              UpdateProgress();
00158
00159          }
00160      }
00161  }
00162 }
00163 }
```

7.355 UserInfoWindow.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.WindowSystem.Windows;
00004 using UnityEngine;
00005 using UnityEngine.Events;
00006 using UnityEngine.UI;
00007
00008 namespace MergeIt.Game.Windows.UserProgress
00009 {
00010     public class UserInfoWindow : WindowBase
00011     {
00012
00013         [SerializeField]
00014         private Button _levelUpButton;
00015
00016         [SerializeField]
00017         private RectTransform _prizesContainer;
00018
00019         [SerializeField]
00020         private UserProgressComponent _userProgressComponent;
00021
00022         public UnityAction LevelUpEvent;
00023
00024         public RectTransform PrizesContainer
00025         {
00026             get => _prizesContainer;
00027         }
00028
00029         public override void OnOpenFinished()
00030         {
00031             base.OnOpenFinished();
00032
00033             _levelUpButton.onClick.AddListener(LevelUpEvent);
00034         }
00035
00036         public override void Close()
00037         {
00038             base.Close();
00039
00040             _levelUpButton.onClick.RemoveListener(LevelUpEvent);
00041         }
00042
00043         public void EnableLevelUpButton(bool enable)
00044         {
```

```

00045         _levelUpButton.interactable = enable;
00046     }
00047
00048     public void UpdateProgress(int level, int currentExp, int nextExp)
00049     {
00050         _userProgressComponent.UpdateProgress(level, currentExp, nextExp);
00051     }
00052 }
00053 }
```

7.356 UserProgressComponent.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Helpers;
00004 using TMPro;
00005 using UnityEngine;
00006
00007 namespace MergeIt.Game.Windows.UserProgress
00008 {
00009     public class UserProgressComponent : MonoBehaviour
00010     {
00011         [SerializeField]
00012         private TMP_Text _levelValue;
00013
00014         [SerializeField]
00015         private TMP_Text _progressText;
00016
00017         [SerializeField]
00018         private SlicedFilledImage _progressImage;
00019
00020         public void UpdateProgress(int level, int currentExp, int nextExp)
00021     {
00022         _levelValue.text = $"{level}";
00023         _progressText.text = $"{currentExp}/{nextExp}";
00024         _progressImage.fillAmount = (float)currentExp / nextExp;
00025     }
00026 }
00027 }
```

7.357 DiContainer.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using MergeIt.SimpleDI.ReservedInterfaces;
00006 using UnityEngine;
00007
00008 namespace MergeIt.SimpleDI
00009 {
00010     public class DiContainer
00011     {
00012         private static readonly HashSet<Type> ReservedInterfaces = new()
00013         {
00014             typeof(IInitializable),
00015             typeof(IUpdatable),
00016             typeof(IDisposable)
00017         };
00018
00019         internal static readonly HashSet<IUpdatable> UpdatableObjects = new();
00020
00021         private static readonly Dictionary<Type, Dictionary<string, DiHandler>> TypesHandlers = new();
00022
00023         internal static void Update()
00024     {
00025         foreach (IUpdatable updatable in UpdatableObjects)
00026     {
00027             updatable.Update();
00028         }
00029     }
00030
00031         public static DiHandler Register<TInterface, TType>(string key = "")
00032             where TInterface : class
00033             where TType : TInterface
00034     {
00035             Type type = typeof(TInterface);
00036             var diHandler = new DiHandler();
00037             diHandler.AddDependency<TType>();
```

```

00038
00039         return Bind<TType>(type, diHandler, key);
00040     }
00041
00042     public static DiHandler Register<TType>(string key = "")
00043         where TType : class
00044     {
00045         Type type = typeof(TType);
00046         var diHandler = new DiHandler();
00047         diHandler.AddDependency<TType>();
00048
00049         return Bind<TType>(type, diHandler, key);
00050     }
00051
00052     public static DiHandler RegisterInterfacesFor<TType>(string key = "")
00053         where TType : class
00054     {
00055         Type type = typeof(TType);
00056         Type[] interfaces = type.GetInterfaces();
00057
00058         var diHandler = new DiHandler();
00059         diHandler.AddDependency<TType>();
00060
00061         for (int i = 0; i < interfaces.Length; i++)
00062     {
00063             Type interfaceType = interfaces[i];
00064
00065             if (!ReservedInterfaces.Contains(interfaceType))
00066             {
00067                 Bind<TType>(interfaceType, diHandler, key);
00068             }
00069         }
00070
00071         return diHandler;
00072     }
00073
00074     public static TInterface Get<TInterface>(string key = "")
00075         where TInterface : class
00076     {
00077         return Get(typeof(TInterface), key) as TInterface;
00078     }
00079
00080     public static void Drop<TInterface>(string key = "")
00081         where TInterface : class
00082     {
00083         Type type = typeof(TInterface);
00084
00085         if (!TypesHandlers.TryGetValue(type, out Dictionary<string, DiHandler>
00086             dependencyHandlers))
00087         {
00088             Debug.LogWarning($"Binding for {type} was not found");
00089             return;
00090         }
00091         dependencyHandlers.Remove(key);
00092     }
00093
00094     public static void Clear()
00095     {
00096         foreach (var typesHandler in TypesHandlers)
00097     {
00098             typesHandler.Value.Clear();
00099         }
00100
00101         TypesHandlers.Clear();
00102
00103         UpdatableObjects.Clear();
00104     }
00105
00106     private static DiHandler Bind<TType>(Type type, DiHandler diHandler, string key = "")
00107     {
00108         if (!TypesHandlers.TryGetValue(type, out Dictionary<string, DiHandler> handlers))
00109         {
00110             handlers = new Dictionary<string, DiHandler> {{key, diHandler}};
00111
00112             TypesHandlers.Add(type, handlers);
00113         }
00114         else
00115         {
00116             if (handlers.ContainsKey(key))
00117             {
00118                 throw new Exception(
00119                     $"Container already has dependency with key '{key}': {type} ->
00120                     {typeof(TType)}");
00121             }
00122         }
00123         handlers.Add(key, diHandler);
00124     }

```

```

00123         }
00124
00125         return diHandler;
00126     }
00127
00128     internal static object Get(Type type, string key = "")
00129     {
00130         if (!TypesHandlers.TryGetValue(type, out Dictionary<string, DiHandler> dependencyHandlers)
00131             ||
00132             !dependencyHandlers.TryGetValue(key, out DiHandler handler))
00133         {
00134             Debug.LogError($"Bindings for {type} were not found");
00135             return default;
00136         }
00137
00138         var instance = handler.Setup();
00139
00140         return instance;
00141     }
00142
00143     internal static void PostProcess()
00144     {
00145         foreach (var typesHandler in TypesHandlers)
00146         {
00147             foreach (var diHandler in typesHandler.Value)
00148             {
00149                 diHandler.Value.TrySetup();
00150             }
00151         }
00152     }
00153 }

```

7.358 DiHandler.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Linq;
00005 using System.Reflection;
00006 using MergeIt.SimpleDI.ReservedInterfaces;
00007
00008 namespace MergeIt.SimpleDI
00009 {
0010     public class DiHandler
0011     {
0012         private Type _type;
0013         private object _instance;
0014         private bool _isSingleton;
0015         private bool _needSetup = true;
0016
0017         internal bool NeedSetup
0018         {
0019             get => _needSetup;
0020             set => _needSetup = value;
0021         }
0022
0023         internal void TrySetup()
0024         {
0025             Setup();
0026         }
0027
0028         internal object Setup()
0029         {
0030             if (!_needSetup)
0031             {
0032                 return _instance;
0033             }
0034
0035             _needSetup = false;
0036
0037             _instance = GetOrCreateInstance();
0038
0039             var type = _instance.GetType();
0040             var properties = type
0041                 .GetProperties(BindingFlags.Instance | BindingFlags.GetProperty | BindingFlags.Public
0042                 | BindingFlags.NonPublic)
0043                 .Where(x => x.GetCustomAttribute<IntroduceAttribute>() != null);
0044
0045             var fields = type
0046                 .GetFields(BindingFlags.Instance | BindingFlags.NonPublic | BindingFlags.Public)
0047                 .Where(x => x.GetCustomAttribute<IntroduceAttribute>() != null);

```

```

00048         foreach ( PropertyInfo propertyInfo in properties )
00049         {
00050             var attribute = propertyInfo.GetCustomAttribute<IntroduceAttribute>();
00051             var attributeKey = attribute.GetType().GetField("_key", BindingFlags.Instance |
00052                 BindingFlags.NonPublic);
00053             string value = attributeKey.GetValue(attribute) as string;
00054             propertyInfo.SetValue(_instance, DiContainer.Get(propertyInfo.PropertyType, value));
00055         }
00056
00057         foreach ( FieldInfo fieldInfo in fields )
00058         {
00059             var attribute = fieldInfo.GetCustomAttribute<IntroduceAttribute>();
00060             var attributeKey = attribute.GetType().GetField("_key", BindingFlags.Instance |
00061                 BindingFlags.NonPublic);
00062             string value = attributeKey.GetValue(attribute) as string;
00063             fieldInfo.SetValue(_instance, DiContainer.Get(fieldInfo.FieldType, value));
00064         }
00065
00066         if (_instance is IInitializable initializable)
00067         {
00068             initializable.Initialize();
00069         }
00070
00071         if (_instance is IUpdatable updatable)
00072         {
00073             DiContainer.UpdatableObjects.Add(updatable);
00074         }
00075
00076         return _instance;
00077     }
00078
00079     public void AddDependency<TType>()
00080     {
00081         _type = typeof(TType);
00082     }
00083
00084     public DiHandler AssSingleton()
00085     {
00086         _isSingleton = true;
00087
00088         GetOrGetInstance();
00089
00090         return this;
00091     }
00092
00093     public DiHandler AssSingleton<TType>(TType instance) where TType : class
00094     {
00095         _isSingleton = true;
00096         _instance = instance;
00097         _needSetup = false;
00098
00099         return this;
00100    }
00101
00102    private object GetOrGetInstance()
00103    {
00104        if (_isSingleton)
00105        {
00106            return _instance ??= Activator.CreateInstance(_type);
00107        }
00108
00109        var instance = Activator.CreateInstance(_type);
00110
00111        return instance;
00112    }
00113 }
00114 }
```

7.359 DiRoot.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using MergeIt.Core.Helpers;
00004 using UnityEngine;
00005
00006 namespace MergeIt.SimpleDI
00007 {
00008     public abstract class DiRoot : MonoBehaviour
00009     {
00010         protected abstract void OnInstall();
00011 }
```

```

00012     protected virtual void Run()
00013     {
00014     }
00015
00016     private void Awake()
00017     {
00018         _ = MainThreadDispatcher.Instance;
00019         Install();
00020         Run();
00021     }
00022
00023
00024     private void Update()
00025     {
00026         DiContainer.Update();
00027     }
00028
00029     private void Install()
00030     {
00031         OnInstall();
00032         PostInstall();
00033     }
00034
00035     private void PostInstall()
00036     {
00037         DiContainer.PostProcess();
00038     }
00039 }
00040 }
```

7.360 IntroduceAttribute.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004
00005 namespace MergeIt.SimpleDI
00006 {
00007     public class IntroduceAttribute : Attribute
00008     {
00009         private string _key;
00010
00011         public IntroduceAttribute(string key = "")
00012         {
00013             _key = key;
00014         }
00015     }
00016 }
```

7.361 IInitializable.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.SimpleDI.ReservedInterfaces
00004 {
00005     public interface IInitializable
00006     {
00007         void Initialize();
00008     }
00009 }
```

7.362 IUpdatable.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.SimpleDI.ReservedInterfaces
00004 {
00005     public interface IUpdatable
00006     {
00007         void Update();
00008     }
00009 }
```

7.363 Bindable.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005
00006 namespace MergeIt.Core.Utils
00007 {
00008     public class Bindable<T>
00009     {
00010         private event Action<T, T> ChangedFromToEvent = delegate { };
00011         private event Action<T> ChangedToEvent = delegate { };
00012
00013         private T _value;
00014         private T _holdValue;
00015         private bool _isHeld;
00016
00017         public T Value
00018         {
00019             get => _value;
00020             set
00021             {
00022                 if (EqualityComparer<T>.Default.Equals(_value, value))
00023                 {
00024                     return;
00025                 }
00026
00027                 if (_isHeld)
00028                 {
00029                     _holdValue = _value;
00030                     _value = value;
00031
00032                     return;
00033                 }
00034
00035                 T prevValue = _value;
00036
00037                 _value = value;
00038
00039                 ChangedFromToEvent?.Invoke(prevValue, value);
00040                 ChangedToEvent?.Invoke(value);
00041             }
00042         }
00043
00044         public void Subscribe(Action<T, T> onChanged, bool immediateCheck = false)
00045         {
00046             ChangedFromToEvent += onChanged;
00047
00048             if (immediateCheck)
00049             {
00050                 onChanged?.Invoke(default, _value);
00051             }
00052         }
00053
00054         public void Subscribe(Action<T> onChanged, bool immediateCheck = false)
00055         {
00056             ChangedToEvent += onChanged;
00057
00058             if (immediateCheck)
00059             {
00060                 onChanged?.Invoke(_value);
00061             }
00062         }
00063
00064         public void Unsubscribe(Action<T, T> onChanged)
00065         {
00066             ChangedFromToEvent -= onChanged;
00067         }
00068
00069         public void Unsubscribe(Action<T> onChanged)
00070         {
00071             ChangedToEvent -= onChanged;
00072         }
00073
00074         public void SetValueSilently(T value)
00075         {
00076             _value = value;
00077         }
00078
00079         public void Hold()
00080         {
00081             _isHeld = true;
00082         }
00083
00084         public void Release()
00085         {

```

```
00086         if (!isHeld)
00087         {
00088             return;
00089         }
00090         isHeld = false;
00091
00092         T temp = _holdValue;
00093         _holdValue = default;
00094
00095         if (!temp.Equals(_value))
00096         {
00097             ChangedFromToEvent?.Invoke(temp, _value);
00098             ChangedToEvent?.Invoke(_value);
00099         }
00100     }
00101 }
00102 }
00103 }
```

7.364 IMonoApplicationQuitHandler.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Helpers
00004 {
00005     public interface IMonoApplicationQuitHandler
00006     {
00007         void OnApplicationQuit();
00008     }
00009 }
```

7.365 IMonoUpdateHandler.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 namespace MergeIt.Core.Helpers
00004 {
00005     public interface IMonoUpdateHandler
00006     {
00007         void Update();
00008     }
00009 }
```

7.366 MainThreadDispatcher.cs

```
00001 // Copyright (c) 2024, Awessets
00002
00003 using System;
00004 using System.Collections.Generic;
00005 using System.Threading;
00006 using UnityEngine;
00007
00008 namespace MergeIt.Core.Helpers
00009 {
00010     public class MainThreadDispatcher : MonoBehaviour
00011     {
00012         private static MainThreadDispatcher _instance;
00013         public static MainThreadDispatcher Instance
00014         {
00015             get
00016             {
00017                 if (!_instance)
00018                 {
00019                     var updater = new GameObject($"nameof(MainThreadDispatcher)");
00020                     _instance = updater.AddComponent<MainThreadDispatcher>();
00021                     DontDestroyOnLoad(updater);
00022                 }
00023
00024                 return _instance;
00025             }
00026         }
00027
00028         private static int _mainThreadId;
00029         private static readonly Queue<Action> ExecutionQueue = new Queue<Action>();
```

```

00030
00031     private void Awake()
00032     {
00033         _mainThreadId = Thread.CurrentThread.ManagedThreadId;
00034     }
00035
00036     private void Update()
00037     {
00038         lock (ExecutionQueue)
00039         {
00040             while (ExecutionQueue.Count > 0)
00041             {
00042                 ExecutionQueue.Dequeue().Invoke();
00043             }
00044         }
00045     }
00046
00047     public void Enqueue(Action action)
00048     {
00049         if (action == null)
00050         {
00051             Debug.LogError("No action to enqueue.");
00052             return;
00053         }
00054
00055         lock (ExecutionQueue)
00056         {
00057             ExecutionQueue.Enqueue(action);
00058         }
00059     }
00060
00061     public void RunOnMainThread(Action action)
00062     {
00063         if (action == null)
00064         {
00065             Debug.LogError("No action to run on main thread.");
00066             return;
00067         }
00068
00069         if (IsMainThread())
00070         {
00071             action();
00072         }
00073         else
00074         {
00075             Enqueue(action);
00076         }
00077     }
00078
00079     public bool IsMainThread()
00080     {
00081         return Thread.CurrentThread.ManagedThreadId == _mainThreadId;
00082     }
00083 }
00084 }
```

7.367 MonoEventsListener.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using System.Collections.Generic;
00004 using UnityEngine;
00005
00006 namespace MergeIt.Core.Helpers
00007 {
00008     public class MonoEventsListener : MonoBehaviour
00009     {
00010         private static MonoEventsListener _instance;
00011         public static MonoEventsListener Instance
00012         {
00013             get
00014             {
00015                 if (!_instance)
00016                 {
00017                     var updater = new GameObject($"{nameof(MonoEventsListener)}");
00018                     _instance = updater.AddComponent<MonoEventsListener>();
00019                     DontDestroyOnLoad(updater);
00020                 }
00021             }
00022             return _instance;
00023         }
00024     }
00025 }
```

```

00026     private static readonly List<IMonoUpdateHandler> UpdatableHandlers = new
00027         List<IMonoUpdateHandler>();
00028     private static readonly List<IMonoApplicationQuitHandler> ApplicationQuitHandlers = new
00029         List<IMonoApplicationQuitHandler>();
00030     public void SubscribeOnUpdate(IMonoUpdateHandler monoUpdateHandler)
00031     {
00032         UpdatableHandlers.Add(monoUpdateHandler);
00033     }
00034     public void UnsubscribeFromUpdate(IMonoUpdateHandler monoUpdateHandler)
00035     {
00036         UpdatableHandlers.Remove(monoUpdateHandler);
00037     }
00038     public void SubscribeOnApplicationQuit(IMonoApplicationQuitHandler monoApplicationQuitHandler)
00039     {
00040         ApplicationQuitHandlers.Add(monoApplicationQuitHandler);
00041     }
00042     public void UnsubscribeFromApplicationQuit(IMonoApplicationQuitHandler
00043 monoApplicationQuitHandler)
00044     {
00045         ApplicationQuitHandlers.Remove(monoApplicationQuitHandler);
00046     }
00047     private void Update()
00048     {
00049         for (int i = 0; i < UpdatableHandlers.Count; i++)
00050         {
00051             UpdatableHandlers[i].Update();
00052         }
00053     }
00054     private void OnApplicationQuit()
00055     {
00056         for (int i = 0; i < ApplicationQuitHandlers.Count; i++)
00057         {
00058             ApplicationQuitHandlers[i].OnApplicationQuit();
00059         }
00060     }
00061 }
00062 }
00063 }
00064 }
00065 }

```

7.368 ResponsiveGridLayout.cs

```

00001 // Copyright (c) 2024, Awessets
00002
00003 using UnityEngine;
00004 using UnityEngine.UI;
00005
00006 namespace MergeIt.Core.Utils
00007 {
00008     [RequireComponent(typeof(GridLayoutGroup))]
00009     public class ResponsiveGridLayout : MonoBehaviour
00010     {
00011         private GridLayoutGroup _gridLayout;
00012         private RectTransform _parentRectTransform;
00013
00014         private bool _initialized;
00015         private void Awake()
00016         {
00017             _gridLayout = GetComponent<GridLayoutGroup>();
00018             _parentRectTransform = _gridLayout.transform.parent.GetComponent<RectTransform>();
00019
00020             _initialized = true;
00021
00022             AdjustCellSize();
00023         }
00024
00025         private void AdjustCellSize()
00026         {
00027             int columns = _gridLayout.constraintCount;
00028             float containerWidth = _parentRectTransform.rect.width - (_gridLayout.padding.left +
00029             _gridLayout.padding.right) -
00030                             _gridLayout.spacing.x * (columns - 1);
00031
00032             float cellSize = containerWidth / columns;
00033
00034             _gridLayout.cellSize = new Vector2(cellSize, cellSize);
00035         }
00036         private void OnRectTransformDimensionsChange()

```

```

00037      {
00038          if (_initialized)
00039          {
00040              AdjustCellSize();
00041          }
00042      }
00043  }
00044 }
```

7.369 SlicedFilledImage.cs

```

00001 #if UNITY_2017_4 || UNITY_2018_2_OR_NEWER
00002 #endif
00003 using System;
00004 using System.Collections.Generic;
00005 using UnityEngine;
00006 using UnityEngine.U2D;
00007 using UnityEngine.UI;
00008 using Sprites = UnityEngine.Sprites;
00009
00010 // Custom Editor to order the variables in the Inspector similar to Image component
00011
00012 // Credit: https://bitbucket.org/Unity-Technologies/ui/src/2018.4/UnityEngine.UI/UI/Core/Image.cs
00013 namespace MergeIt.Core.Helpers
00014 {
00015     [RequireComponent( typeof( CanvasRenderer ) )]
00016     [AddComponentMenu( "UI/Sliced Filled Image", 11 )]
00017     public class SlicedFilledImage : MaskableGraphic, ISerializationCallbackReceiver, ILayoutElement,
00018         ICanvasRaycastFilter
00019     {
00020         private static class SetPropertyUtility
00021         {
00022             public static bool SetStruct<T>( ref T currentValue, T newValue ) where T : struct
00023             {
00024                 if( EqualityComparer<T>.Default.Equals( currentValue, newValue ) )
00025                     return false;
00026
00027                 currentValue = newValue;
00028                 return true;
00029             }
00030
00031             public static bool SetClass<T>( ref T currentValue, T newValue ) where T : class
00032             {
00033                 if( ( currentValue == null && newValue == null ) || ( currentValue != null &&
00034                     currentValue.Equals( newValue ) ) )
00035                     return false;
00036
00037                 currentValue = newValue;
00038                 return true;
00039             }
00040
00041             public enum FillDirection { Right = 0, Left = 1, Up = 2, Down = 3 }
00042
00043             private static readonly Vector3[] s_Vertices = new Vector3[4];
00044             private static readonly Vector2[] s_UVs = new Vector2[4];
00045             private static readonly Vector2[] s_SlicedVertices = new Vector2[4];
00046             private static readonly Vector2[] s_SlicedUVs = new Vector2[4];
00047
00048 #pragma warning disable 1692
00049 #pragma warning disable IDE1006 // Suppress 'Naming rule violation' warnings
00050 #pragma warning disable 0649
00051     [SerializeField]
00052     private Sprite m_Sprite;
00053     public Sprite sprite
00054     {
00055         get { return m_Sprite; }
00056         set
00057         {
00058             if( SetPropertyUtility.SetClass( ref m_Sprite, value ) )
00059             {
00060                 SetAllDirty();
00061                 TrackImage();
00062             }
00063         }
00064     }
00065     [SerializeField]
00066     private FillDirection m_FillDirection;
00067     public FillDirection fillDirection
00068     {
00069         get { return m_FillDirection; }
00070         set
```

```
00071         {
00072             if( SetPropertyUtility.SetStruct( ref m_FillDirection, value ) )
00073                 SetVerticesDirty();
00074         }
00075     }
00076 
00077     [Range( 0, 1 )]
00078     [SerializeField]
00079     private float m_FillAmount = 1f;
00080     public float fillAmount
00081     {
00082         get { return m_FillAmount; }
00083         set
00084         {
00085             if( SetPropertyUtility.SetStruct( ref m_FillAmount, Mathf.Clamp01( value ) ) )
00086                 SetVerticesDirty();
00087         }
00088     }
00089 
00090     [SerializeField]
00091     private bool m_FillCenter = true;
00092     public bool fillCenter
00093     {
00094         get { return m_FillCenter; }
00095         set
00096         {
00097             if( SetPropertyUtility.SetStruct( ref m_FillCenter, value ) )
00098                 SetVerticesDirty();
00099         }
00100     }
00101 
00102     [SerializeField]
00103     private float m_PixelsPerUnitMultiplier = 1f;
00104     public float pixelsPerUnitMultiplier
00105     {
00106         get { return m_PixelsPerUnitMultiplier; }
00107         set { m_PixelsPerUnitMultiplier = Mathf.Max( 0.01f, value ); }
00108     }
00109 
00110     public float pixelsPerUnit
00111     {
00112         get
00113         {
00114             float spritePixelsPerUnit = 100;
00115             if( activeSprite )
00116                 spritePixelsPerUnit = activeSprite.pixelsPerUnit;
00117 
00118             float referencePixelsPerUnit = 100;
00119             if( canvas )
00120                 referencePixelsPerUnit = canvas.referencePixelsPerUnit;
00121 
00122             return m_PixelsPerUnitMultiplier * spritePixelsPerUnit / referencePixelsPerUnit;
00123         }
00124     }
00125 #pragma warning restore 0649
00126 
00127     [NonSerialized]
00128     private Sprite m_OverrideSprite;
00129     public Sprite overrideSprite
00130     {
00131         get { return activeSprite; }
00132         set
00133         {
00134             if( SetPropertyUtility.SetClass( ref m_OverrideSprite, value ) )
00135             {
00136                 SetAllDirty();
00137                 TrackImage();
00138             }
00139         }
00140     }
00141 
00142     private Sprite activeSprite { get { return m_OverrideSprite != null ? m_OverrideSprite :
00143         m_Sprite; } }
00144 
00145     public override Texture mainTexture
00146     {
00147         get
00148         {
00149             if( activeSprite != null )
00150                 return activeSprite.texture;
00151 
00152             return material != null && material.mainTexture != null ? material.mainTexture :
00153                 s_WhiteTexture;
00154         }
00155     }
00156 
00157     public bool hasBorder
```

```

00156      {
00157          get
00158          {
00159              if( activeSprite != null )
00160              {
00161                  Vector4 v = activeSprite.border;
00162                  return v.sqrMagnitude > 0f;
00163              }
00164          }
00165          return false;
00166      }
00167  }
00168
00169  public override Material material
00170  {
00171      get
00172      {
00173          if( m_Material != null )
00174              return m_Material;
00175
00176          if( activeSprite && activeSprite.associatedAlphaSplitTexture != null )
00177          {
00178 #if UNITY_EDITOR
00179             if( Application.isPlaying )
00180 #endif
00181             return Image.defaultETC1GraphicMaterial;
00182         }
00183
00184         return defaultMaterial;
00185     }
00186     set { base.material = value; }
00187 }
00188
00189  public float alphaHitTestMinimumThreshold { get; set; }
00190 #pragma warning restore IDE1006
00191 #pragma warning restore 1692
00192
00193  protected SlicedFilledImage()
00194  {
00195      useLegacyMeshGeneration = false;
00196  }
00197
00198  protected override void OnEnable()
00199  {
00200      base.OnEnable();
00201      TrackImage();
00202  }
00203
00204  protected override void OnDisable()
00205  {
00206      base.OnDisable();
00207
00208      if( m_Tracked )
00209          UnTrackImage();
00210  }
00211
00212 #if UNITY_EDITOR
00213     protected override void OnValidate()
00214     {
00215         base.OnValidate();
00216         m_PixelsPerUnitMultiplier = Mathf.Max( 0.01f, m_PixelsPerUnitMultiplier );
00217     }
00218 #endif
00219
00220  protected override void OnPopulateMesh( VertexHelper vh )
00221  {
00222      if( activeSprite == null )
00223      {
00224          base.OnPopulateMesh( vh );
00225          return;
00226      }
00227
00228      GenerateSlicedFilledSprite( vh );
00229  }
00230
00231  protected override void UpdateMaterial()
00232  {
00233      base.UpdateMaterial();
00234
00235      // Check if this sprite has an associated alpha texture (generated when splitting RGBA =
00236      // RGB + A as two textures without alpha)
00237      if( activeSprite == null )
00238      {
00239          canvasRenderer.SetAlphaTexture( null );
00240          return;
00241      }
00242  }
00243
00244

```

```

00245         Texture2D alphaTex = activeSprite.associatedAlphaSplitTexture;
00246         if( alphaTex != null )
00247             canvasRenderer.SetAlphaTexture( alphaTex );
00248     }
00249
00250     private void GenerateSlicedFilledSprite( VertexHelper vh )
00251     {
00252         vh.Clear();
00253
00254         if( m_FillAmount < 0.001f )
00255             return;
00256
00257         Rect rect = GetPixelAdjustedRect();
00258         Vector4 outer = Sprites.DataUtility.GetOuterUV( activeSprite );
00259         Vector4 padding = Sprites.DataUtility.GetPadding( activeSprite );
00260
00261         if( !hasBorder )
00262     {
00263         Vector2 size = activeSprite.rect.size;
00264
00265         int spriteW = Mathf.RoundToInt( size.x );
00266         int spriteH = Mathf.RoundToInt( size.y );
00267
00268         // Image's dimensions used for drawing. X = left, Y = bottom, Z = right, W = top.
00269         Vector4 vertices = new Vector4(
00270             rect.x + rect.width * ( padding.x / spriteW ),
00271             rect.y + rect.height * ( padding.y / spriteH ),
00272             rect.x + rect.width * ( ( spriteW - padding.z ) / spriteW ),
00273             rect.y + rect.height * ( ( spriteH - padding.w ) / spriteH ) );
00274
00275         GenerateFilledSprite( vh, vertices, outer, m_FillAmount );
00276         return;
00277     }
00278
00279     Vector4 inner = Sprites.DataUtility.GetInnerUV( activeSprite );
00280     Vector4 border = GetAdjustedBorders( activeSprite.border / pixelsPerUnit, rect );
00281
00282     padding = padding / pixelsPerUnit;
00283
00284     s_SlicedVertices[0] = new Vector2( padding.x, padding.y );
00285     s_SlicedVertices[3] = new Vector2( rect.width - padding.z, rect.height - padding.w );
00286
00287     s_SlicedVertices[1].x = border.x;
00288     s_SlicedVertices[1].y = border.y;
00289
00290     s_SlicedVertices[2].x = rect.width - border.z;
00291     s_SlicedVertices[2].y = rect.height - border.w;
00292
00293     for( int i = 0; i < 4; ++i )
00294     {
00295         s_SlicedVertices[i].x += rect.x;
00296         s_SlicedVertices[i].y += rect.y;
00297     }
00298
00299     s_SlicedUVs[0] = new Vector2( outer.x, outer.y );
00300     s_SlicedUVs[1] = new Vector2( inner.x, inner.y );
00301     s_SlicedUVs[2] = new Vector2( inner.z, inner.w );
00302     s_SlicedUVs[3] = new Vector2( outer.z, outer.w );
00303
00304     float rectStartPos;
00305     float _1OverTotalSize;
00306     if( m_FillDirection == FillDirection.Left || m_FillDirection == FillDirection.Right )
00307     {
00308         rectStartPos = s_SlicedVertices[0].x;
00309
00310         float totalSize = ( s_SlicedVertices[3].x - s_SlicedVertices[0].x );
00311         _1OverTotalSize = totalSize > 0f ? 1f / totalSize : 1f;
00312     }
00313     else
00314     {
00315         rectStartPos = s_SlicedVertices[0].y;
00316
00317         float totalSize = ( s_SlicedVertices[3].y - s_SlicedVertices[0].y );
00318         _1OverTotalSize = totalSize > 0f ? 1f / totalSize : 1f;
00319     }
00320
00321     for( int x = 0; x < 3; x++ )
00322     {
00323         int x2 = x + 1;
00324
00325         for( int y = 0; y < 3; y++ )
00326         {
00327             if( !m_FillCenter && x == 1 && y == 1 )
00328                 continue;
00329
00330             int y2 = y + 1;
00331

```

```

00332         float sliceStart, sliceEnd;
00333         switch( m_FillDirection )
00334     {
00335         case FillDirection.Right:
00336             sliceStart = ( s_SlicedVertices[x].x - rectStartPos ) * _1OverTotalSize;
00337             sliceEnd = ( s_SlicedVertices[x2].x - rectStartPos ) * _1OverTotalSize;
00338             break;
00339         case FillDirection.Up:
00340             sliceStart = ( s_SlicedVertices[y].y - rectStartPos ) * _1OverTotalSize;
00341             sliceEnd = ( s_SlicedVertices[y2].y - rectStartPos ) * _1OverTotalSize;
00342             break;
00343         case FillDirection.Left:
00344             sliceStart = 1f - ( s_SlicedVertices[x2].x - rectStartPos ) *
00345             _1OverTotalSize;
00346             sliceEnd = 1f - ( s_SlicedVertices[x].x - rectStartPos ) *
00347             _1OverTotalSize;
00348             break;
00349         case FillDirection.Down:
00350             sliceStart = 1f - ( s_SlicedVertices[y2].y - rectStartPos ) *
00351             _1OverTotalSize;
00352             sliceEnd = 1f - ( s_SlicedVertices[y].y - rectStartPos ) *
00353             _1OverTotalSize;
00354             break;
00355         default: // Just there to get rid of the "Use of unassigned local variable"
00356             compiler_error
00357             sliceStart = sliceEnd = 0f;
00358             break;
00359     }
00360 
00361     if( sliceStart >= m_FillAmount )
00362         continue;
00363 
00364     Vector4 vertices = new Vector4( s_SlicedVertices[x].x, s_SlicedVertices[y].y,
00365     s_SlicedVertices[x2].x, s_SlicedVertices[y2].y );
00366     Vector4 uvs = new Vector4( s_SlicedUVs[x].x, s_SlicedUVs[y].y, s_SlicedUVs[x2].x,
00367     s_SlicedUVs[y2].y );
00368     float fillAmount = ( m_FillAmount - sliceStart ) / ( sliceEnd - sliceStart );
00369     GenerateFilledSprite( vh, vertices, uvs, fillAmount );
00370 }
00371 }
00372 }
00373 }
00374 }
00375 }
00376 }
00377 }
00378 }
00379 }
00380 }
00381 }
00382 }
00383 }
00384 }
00385 }
00386 }
00387 }
00388 }
00389 }
00390 }
00391 }
00392 }
00393 }
00394 }
00395 }
00396 }
00397 }
00398 }
00399     private void GenerateFilledSprite( VertexHelper vh, Vector4 vertices, Vector4 uvs, float
00400     fillAmount )
00401     {
00402         if( m_FillAmount < 0.001f )
00403             return;
00404 
00405         float uvLeft = uvs.x;
00406         float uvBottom = uvs.y;
00407         float uvRight = uvs.z;

```

```
00407         float uvTop = uvs.w;
00408
00409         if( fillAmount < 1f )
00410     {
00411             if( m_FillDirection == FillDirection.Left || m_FillDirection == FillDirection.Right )
00412             {
00413                 if( m_FillDirection == FillDirection.Left )
00414                 {
00415                     vertices.x = vertices.z - ( vertices.z - vertices.x ) * fillAmount;
00416                     uvLeft = uvRight - ( uvRight - uvLeft ) * fillAmount;
00417                 }
00418                 else
00419                 {
00420                     vertices.z = vertices.x + ( vertices.z - vertices.x ) * fillAmount;
00421                     uvRight = uvLeft + ( uvRight - uvLeft ) * fillAmount;
00422                 }
00423             }
00424             else
00425             {
00426                 if( m_FillDirection == FillDirection.Down )
00427                 {
00428                     vertices.y = vertices.w - ( vertices.w - vertices.y ) * fillAmount;
00429                     uvBottom = uvTop - ( uvTop - uvBottom ) * fillAmount;
00430                 }
00431                 else
00432                 {
00433                     vertices.w = vertices.y + ( vertices.w - vertices.y ) * fillAmount;
00434                     uvTop = uvBottom + ( uvTop - uvBottom ) * fillAmount;
00435                 }
00436             }
00437         }
00438
00439         s_Vertices[0] = new Vector3( vertices.x, vertices.y );
00440         s_Vertices[1] = new Vector3( vertices.x, vertices.w );
00441         s_Vertices[2] = new Vector3( vertices.z, vertices.w );
00442         s_Vertices[3] = new Vector3( vertices.z, vertices.y );
00443
00444         s_UVs[0] = new Vector2( uvLeft, uvBottom );
00445         s_UVs[1] = new Vector2( uvLeft, uvTop );
00446         s_UVs[2] = new Vector2( uvRight, uvTop );
00447         s_UVs[3] = new Vector2( uvRight, uvBottom );
00448
00449         int startIndex = vh.currentVertCount;
00450
00451         for( int i = 0; i < 4; i++ )
00452             vh.AddVert( s_Vertices[i], color, s_UVs[i] );
00453
00454         vh.AddTriangle( startIndex, startIndex + 1, startIndex + 2 );
00455         vh.AddTriangle( startIndex + 2, startIndex + 3, startIndex );
00456     }
00457
00458     int ILayoutElement.layoutPriority { get { return 0; } }
00459     float ILayoutElement.minWidth { get { return 0; } }
00460     float ILayoutElement.minHeight { get { return 0; } }
00461     float ILayoutElement.flexibleWidth { get { return -1; } }
00462     float ILayoutElement.flexibleHeight { get { return -1; } }
00463
00464     float ILayoutElement.preferredWidth
00465     {
00466         get
00467         {
00468             if( activeSprite == null )
00469                 return 0;
00470
00471             return Sprites.DataUtility.GetMinSize( activeSprite ).x / pixelsPerUnit;
00472         }
00473     }
00474
00475     float ILayoutElement.preferredHeight
00476     {
00477         get
00478         {
00479             if( activeSprite == null )
00480                 return 0;
00481
00482             return Sprites.DataUtility.GetMinSize( activeSprite ).y / pixelsPerUnit;
00483         }
00484     }
00485
00486     void ILayoutElement.CalculateLayoutInputHorizontal() { }
00487     void ILayoutElement.CalculateLayoutInputVertical() { }
00488
00489     bool ICanvasRaycastFilter.IsRaycastLocationValid( Vector2 screenPoint, Camera eventCamera )
00490     {
00491         if( alphaHitTestMinimumThreshold <= 0 )
00492             return true;
00493     }
```

```

00494         if( alphaHitTestMinimumThreshold > 1 )
00495             return false;
00496
00497         if( activeSprite == null )
00498             return true;
00499
00500         Vector2 local;
00501         if( !RectTransformUtility.ScreenPointToLocalPointInRectangle( rectTransform, screenPoint,
00502             eventCamera, out local ) )
00503             return false;
00504
00505         Rect rect = GetPixelAdjustedRect();
00506
00507         // Convert to have lower left corner as reference point.
00508         local.x += rectTransform.pivot.x * rect.width;
00509         local.y += rectTransform.pivot.y * rect.height;
00510
00511         Rect spriteRect = activeSprite.rect;
00512         Vector4 border = activeSprite.border;
00513         Vector4 adjustedBorder = GetAdjustedBorders( border / pixelsPerUnit, rect );
00514
00515         for( int i = 0; i < 2; i++ )
00516         {
00517             if( local[i] <= adjustedBorder[i] )
00518                 continue;
00519
00520             if( rect.size[i] - local[i] <= adjustedBorder[i + 2] )
00521             {
00522                 local[i] -= ( rect.size[i] - spriteRect.size[i] );
00523                 continue;
00524             }
00525
00526             float lerp = Mathf.InverseLerp( adjustedBorder[i], rect.size[i] - adjustedBorder[i +
00527             2], local[i] );
00528             local[i] = Mathf.Lerp( border[i], spriteRect.size[i] - border[i + 2], lerp );
00529
00530             // Normalize local coordinates.
00531             Rect textureRect = activeSprite.textureRect;
00532             Vector2 normalized = new Vector2( local.x / textureRect.width, local.y /
00533             textureRect.height );
00534
00535             // Convert to texture space.
00536             float x = Mathf.Lerp( textureRect.x, textureRect.xMax, normalized.x ) /
00537             activeSprite.texture.width;
00538             float y = Mathf.Lerp( textureRect.y, textureRect.yMax, normalized.y ) /
00539             activeSprite.texture.height;
00540
00541             switch( m_FillDirection )
00542             {
00543                 case FillDirection.Right:
00544                     if( x > m_FillAmount )
00545                         return false;
00546                     break;
00547                 case FillDirection.Left:
00548                     if( lf - x > m_FillAmount )
00549                         return false;
00550                     break;
00551                 case FillDirection.Up:
00552                     if( y > m_FillAmount )
00553                         return false;
00554                     break;
00555             }
00556
00557             try
00558             {
00559                 return activeSprite.texture.GetPixelBilinear( x, y ).a >=
00560                 alphaHitTestMinimumThreshold;
00561             }
00562             catch( UnityException e )
00563             {
00564                 Debug.LogError( "Using alphaHitTestMinimumThreshold greater than 0 on Image whose
00565                     sprite texture cannot be read. " + e.Message + " Also make sure to disable sprite packing for this
00566                     sprite.", this );
00567
00568                 void ISerializationCallbackReceiver.OnBeforeSerialize() { }
00569                 void ISerializationCallbackReceiver.OnAfterDeserialize()
00570                 {
00571                     m_FillAmount = Mathf.Clamp01( m_FillAmount );
00572                 }
00573             }
00574         }
00575     }
00576 }
```

```
00573     // Whether this is being tracked for Atlas Binding
00574     private bool m_Tracked = false;
00575
00576 #if UNITY_2017_4 || UNITY_2018_2_OR_NEWER
00577     private static List<SlicedFilledImage> m_TrackedTexturelessImages = new
00578         List<SlicedFilledImage>();
00579     private static bool s_Initialized;
00580 #endif
00581
00582     private void TrackImage()
00583     {
00584         if( activeSprite != null && activeSprite.texture == null )
00585         {
00586 #if UNITY_2017_4 || UNITY_2018_2_OR_NEWER
00587             if( !s_Initialized )
00588             {
00589                 SpriteAtlasManager.atlasRegistered += RebuildImage;
00590                 s_Initialized = true;
00591             }
00592
00593             m_TrackedTexturelessImages.Add( this );
00594 #endif
00595             m_Tracked = true;
00596         }
00597     }
00598
00599     private void UnTrackImage()
00600     {
00601 #if UNITY_2017_4 || UNITY_2018_2_OR_NEWER
00602     m_TrackedTexturelessImages.Remove( this );
00603 #endif
00604     m_Tracked = false;
00605     }
00606
00607 #if UNITY_2017_4 || UNITY_2018_2_OR_NEWER
00608     private static void RebuildImage( SpriteAtlas spriteAtlas )
00609     {
00610         for( int i = m_TrackedTexturelessImages.Count - 1; i >= 0; i-- )
00611         {
00612             SlicedFilledImage image = m_TrackedTexturelessImages[i];
00613             if( spriteAtlas.CanBindTo( image.activeSprite ) )
00614             {
00615                 image.SetAllDirty();
00616                 m_TrackedTexturelessImages.RemoveAt( i );
00617             }
00618         }
00619     }
00620 #endif
00621 }
00622 }
```


Index

_messageBus	Mergelt.Core.Configs.Elements.CurrencySettings, 111
Mergelt.Game.Services.GameService, 269	
ActionButtonClick	AmountBinding
Mergelt.Game.UI.InfoPanel.ElementInfo, 134	Mergelt.Editor.Configs.CurrencyComponent, 108
ActionCommandBase	AnimationController
Mergelt.Editor.LevelEditor.Commands.ActionCommandBase, 499	Mergelt.Core.WindowSystem.Windows.WindowBase,
51	
ActionType	Mergelt.Game.Field.Elements.FieldElementView,
Mergelt.Game.Messages.ElementActionMessage, 122	239
Mergelt.Game.UI.InfoPanel.ElementCommonInfo, 124	Animator
Mergelt.Game.UI.InfoPanel.ElementInfo, 136	Mergelt.Game.Effects.Controllers.BaseEffect, 61
Mergelt.Game.UI.InfoPanel.ElementOpenInfo, 156	Any
Mergelt.Game.UI.InfoPanel.ElementSellInfo, 159	Mergelt.Editor.Core.LevelEditor.Commands.LimitedStack<
Mergelt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo, 163	T >, 410
Mergelt.Game.UI.InfoPanel.ElementSpeedUpOpeningInfo, 165	ApplyCell
Mergelt.Game.UI.InfoPanel.ElementSplitInfo, 168	Mergelt.Editor.LevelEditor.LevelEditorWindow, 403
Mergelt.Game.UI.InfoPanel.ElementUnlockInfo, 175	ApplyCellCommand
Activate	Mergelt.Editor.LevelEditor.Commands.ApplyCellCommand,
Mergelt.Game.Field.Elements.FieldElementPresenter< TView, TModel >, 230	58
Active	ApplyModel
Mergelt.Game.Messages.ActivateHintsMessage, 54	Mergelt.Game.HUD.UserListenerComponent, 486
Add	Arrow
Mergelt.Core.Commands.CommandManager, 86	Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer,
Mergelt.Game.Services.ElementsStockService, 171	69
Mergelt.Game.Services.InventoryService, 364	Assets/Mergelt/Editor/Core/Configs/Components/CurrencyComponent.cs,
AddDependency< TType >	525
Mergelt.SimpleDI.DiHandler, 114	Assets/Mergelt/Editor/Core/Configs/Components/CurrencyComponentsTrans.cs,
AddEffect	525
Mergelt.Game.Effects.EffectsManager, 121	Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/BaseSettingsDrawer.cs,
AddListener< T >	526
Mergelt.Core.Messages.IMessageBus, 349	Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/CommonSettingsDrawer.cs,
Mergelt.Core.Messages.MessageBus, 418	527
AdSkip	Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/ElementConfigInspector.cs,
Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 130	528
AllGenerators	Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/GeneratorSettingsBase.cs,
Mergelt.Game.Field.FieldLogicModel, 248	530
alphaHitTestMinimumThreshold	Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/GeneratorSettingsEvol.cs,
Mergelt.Core.Helpers.SlicedFilledImage, 468	531
Amount	Assets/Mergelt/Editor/Core/Configs/ItemsDrawers/GeneratorSettingsInspec.cs,
	532
	Assets/Mergelt/Editor/Core/Constants.cs, 533
	Assets/Mergelt/Editor/Core/EvolutionsEditor/Components/EvolutionsGraph.cs,
	533
	Assets/Mergelt/Editor/Core/EvolutionsEditor/Components/EvolutionsGraph.cs,
	539
	Assets/Mergelt/Editor/Core/EvolutionsEditor/EvolutionsEditorWindow.cs,
	540

Assets/MergeIt/Editor/Core/EvolutionsEditor/ItemsNodes/BaseNode.cs
Assets/MergeIt/Source/Core/Animations/IWindowAnimationController.cs,
571
Assets/MergeIt/Editor/Core/EvolutionsEditor/ItemsNodes/AssetBase.cs
Assets/MergeIt/Source/Core/Animations/IWindowAnimationListener.cs,
571
Assets/MergeIt/Editor/Core/EvolutionsEditor/ItemsNodes/AssetList.cs
Assets/MergeIt/Source/Core/Animations/MecanimController.cs,
571
Assets/MergeIt/Editor/Core/EvolutionsEditor/ItemsNodes/AssetModel.cs
Assets/MergeIt/Source/Core/Animations/WindowDefaultAnimationController.cs,
571
Assets/MergeIt/Editor/Core/EvolutionsEditor/ItemsNodes/AssetModel.cs
Assets/MergeIt/Source/Core/Animations/WindowMecanimAnimationController.cs,
572
Assets/MergeIt/Editor/Core/EvolutionsEditor/ItemsNodes/AssetModel.cs
Assets/MergeIt/Source/Core/Commands/Command.cs,
573
Assets/MergeIt/Editor/Core/EvolutionsEditor/ItemsNodes/AssetModel.cs
Assets/MergeIt/Source/Core/Commands/CommandManager.cs,
573
Assets/MergeIt/Editor/Core/EvolutionsEditor/SchemeObject.cs
Assets/MergeIt/Source/Core/Commands/ICommand.cs,
575
Assets/MergeIt/Editor/Core/GeneratedItemsCreator.cs, Assets/MergeIt/Source/Core/Commands/ICommandManager.cs,
548 575
Assets/MergeIt/Editor/Core/GeneratedItemUI.cs, 550 Assets/MergeIt/Source/Core/Configs/ConfigsConstants.cs,
Assets/MergeIt/Editor/Core/LevelEditor/Commands/ActionCommandBase.cs,
552 Assets/MergeIt/Source/Core/Configs/Data/EvolutionData.cs,
Assets/MergeIt/Editor/Core/LevelEditor/Commands/ActionCommandManager.cs,
552 Assets/MergeIt/Source/Core/Configs/Data/LevelElementData.cs,
Assets/MergeIt/Editor/Core/LevelEditor/Commands/ApplyCellCommand.cs,
553 Assets/MergeIt/Source/Core/Configs/Effects/EffectConfig.cs,
Assets/MergeIt/Editor/Core/LevelEditor/Commands/ChangeCellCommand.cs,
553 Assets/MergeIt/Source/Core/Configs/Elements/ElementConfig.cs,
Assets/MergeIt/Editor/Core/LevelEditor/Commands/ClearCellCommand.cs,
554 Assets/MergeIt/Source/Core/Configs/Elements/FieldElementIconComponent.cs
Assets/MergeIt/Editor/Core/LevelEditor/Commands/IActionCommand.cs,
554 Assets/MergeIt/Source/Core/Configs/Elements/Settings/ConsumableSettings.cs
Assets/MergeIt/Editor/Core/LevelEditor/Commands/IActionCommandManager.cs,
554 Assets/MergeIt/Source/Core/Configs/Elements/Settings/CurrencySettings.cs
Assets/MergeIt/Editor/Core/LevelEditor/Commands/LimitedStack.cs, 559 Assets/MergeIt/Source/Core/Configs/Elements/Settings/ElementCommonSettings.cs
555 Assets/MergeIt/Source/Core/Configs/Elements/Settings/ElementGeneratorSettings.cs
Assets/MergeIt/Editor/Core/LevelEditor/EvolutionSelection.cs, 559 Assets/MergeIt/Source/Core/Configs/Elements/Settings/ElementGeneratorSettings.cs
555 Assets/MergeIt/Source/Core/Configs/Elements/Settings/ElementGeneratorSettings.cs
Assets/MergeIt/Editor/Core/LevelEditor/LevelConfigEditor.cs, 560 Assets/MergeIt/Source/Core/Configs/Elements/Settings/GeneratableElementSettings.cs
556 Assets/MergeIt/Source/Core/Configs/Elements/Settings/GeneratableElementSettings.cs
Assets/MergeIt/Editor/Core/LevelEditor/LevelEditorWindow.cs, 582 Assets/MergeIt/Source/Core/Configs/GameConfig.cs,
557 Assets/MergeIt/Source/Core/Configs/Hints/HintsConfig.cs,
Assets/MergeIt/Editor/Helpers/ConfigsPostprocessor.cs, 567 582
Assets/MergeIt/Editor/Helpers/SavesHelper.cs, 596 582
Assets/MergeIt/Editor/Helpers/SlicedFilledImageEditor.cs, Assets/MergeIt/Source/Core/Configs/Icons/GamelIconsConfig.cs,
568 582
Assets/MergeIt/Source/Core/Animations/AnimationController.cs, Assets/MergeIt/Source/Core/Configs/Icons/IconParameters.cs,
569 583
Assets/MergeIt/Source/Core/Animations/AnimationListener.cs, Assets/MergeIt/Source/Core/Configs/Inventory/InventoryConfig.cs,
569 583
Assets/MergeIt/Source/Core/Animations/AnimationWindow.cs, Assets/MergeIt/Source/Core/Configs/LevelConfig.cs,
569 583
Assets/MergeIt/Source/Core/Animations/AnimatorStateMachine.cs, Assets/MergeIt/Source/Core/Configs/LevelUp/LevelUpConfig.cs,
569 584
Assets/MergeIt/Source/Core/Animations/IAnimationController.cs, Assets/MergeIt/Source/Core/Configs/LevelUp/LevelUpParameters.cs,
570 584
Assets/MergeIt/Source/Core/Animations/IAnimationListener.cs, Assets/MergeIt/Source/Core/Configs/Types/ConsumableOperationType.cs
570 584

Assets/MergeIt/Source/Core/Configs/Types/ConsumableTags.cs	Assets/MergeIt/Source/Core/FieldElements/Parameters/ProducedByParameter.cs
584	593
Assets/MergeIt/Source/Core/Configs/Types/CurrencyType.cs	Assets/MergeIt/Source/Core/FieldElements/Parameters/ProduceParameter.cs
585	593
Assets/MergeIt/Source/Core/Configs/Types/ElementType.cs	Assets/MergeIt/Source/Core/FieldElements/Parameters/Serializable/Save.cs
585	593
Assets/MergeIt/Source/Core/Configs/Windows/LayerParameter.cs	Assets/MergeIt/Source/Core/FieldElements/Parameters/Serializable/Save.cs
585	593
Assets/MergeIt/Source/Core/Configs/Windows/LayersConfig.cs	Assets/MergeIt/Source/Core/FieldElements/Parameters/Serializable/Save.cs
586	594
Assets/MergeIt/Source/Core/Configs/Windows/SortingLayer.cs	Assets/MergeIt/Source/Core/FieldElements/Parameters/Serializable/Save.cs
615	594
Assets/MergeIt/Source/Core/ElementsStock/IElementsStock.cs	Assets/MergeIt/Source/Core/Helpers/CurvesHelper.cs
586	594
Assets/MergeIt/Source/Core/ElementsStock/IElementsStock.cs	Assets/MergeIt/Source/Core/Helpers/ListExtensions.cs
586	595
Assets/MergeIt/Source/Core/Evolutions/EvolutionProgressData.cs	Assets/MergeIt/Source/Core/Helpers/NumericExtensions.cs
586	595
Assets/MergeIt/Source/Core/Evolutions/IEvolutionsProgressData.cs	Assets/MergeIt/Source/Core/Helpers/RectTransformExtensions.cs
587	596
Assets/MergeIt/Source/Core/FieldElements/Animations/FieldElementState.cs	Assets/MergeIt/Source/Core/Helpers/SavesHelper.cs
587	597
Assets/MergeIt/Source/Core/FieldElements/FieldElement.cs	Assets/MergeIt/Source/Core/Helpers/TimeExtensions.cs
587	597
Assets/MergeIt/Source/Core/FieldElements/FieldElementData.cs	Assets/MergeIt/Source/Core/Inventory/IInventoryData.cs
587	598
Assets/MergeIt/Source/Core/FieldElements/GeneratableFieldElement.cs	Assets/MergeIt/Source/Core/Messages/IMessage.cs
588	599
Assets/MergeIt/Source/Core/FieldElements/GridPoint.cs	Assets/MergeIt/Source/Core/Messages/IMessageBus.cs
588	599
Assets/MergeIt/Source/Core/FieldElements/IFieldElement.cs	Assets/MergeIt/Source/Core/Messages/MessageBus.cs
589	599
Assets/MergeIt/Source/Core/FieldElements/IFieldElementModel.cs	Assets/MergeIt/Source/Core/MVP/IModel.cs
589	600
Assets/MergeIt/Source/Core/FieldElements/IFieldElementPresenter.cs	Assets/MergeIt/Source/Core/MVP/IPresenter.cs
589	601
Assets/MergeIt/Source/Core/FieldElements/IFieldElementView.cs	Assets/MergeIt/Source/Core/MVP/IView.cs
589	601
Assets/MergeIt/Source/Core/FieldElements/IFieldElementPresenter.cs	Assets/MergeIt/Source/Core/MVP/Presenter.cs
590	601
Assets/MergeIt/Source/Core/FieldElements/Parameters/ConfigParameters.cs	Assets/MergeIt/Source/Core/MVP/View.cs
590	602
Assets/MergeIt/Source/Core/FieldElements/Parameters/ConfigParameters.cs	Assets/MergeIt/Source/Core/Saves/GameSaveType.cs
590	602
Assets/MergeIt/Source/Core/FieldElements/Parameters/GeneralParameters.cs	Assets/MergeIt/Source/Core/Saves/ISavable.cs
590	602
Assets/MergeIt/Source/Core/FieldElements/Parameters/GeneratorParameters.cs	Assets/MergeIt/Source/Core/Saves/SavableAttribute.cs
591	603
Assets/MergeIt/Source/Core/FieldElements/Parameters/IConfigParameters.cs	Assets/MergeIt/Source/Core/Schemes/Data/EdgeData.cs
591	603
Assets/MergeIt/Source/Core/FieldElements/Parameters/IGeneratorOpenParameters.cs	Assets/MergeIt/Source/Core/Schemes/Data/ElementNodeData.cs
591	603
Assets/MergeIt/Source/Core/FieldElements/Parameters/IGeneratorParameters.cs	Assets/MergeIt/Source/Core/Schemes/Data/EvolutionNodeData.cs
591	603
Assets/MergeIt/Source/Core/FieldElements/Parameters/IGeneratorParameters.cs	Assets/MergeIt/Source/Core/Schemes/Data/IElementNodeData.cs
591	603
Assets/MergeIt/Source/Core/FieldElements/Parameters/IInfoParameters.cs	Assets/MergeIt/Source/Core/Schemes/Data/IEvolutionNodeData.cs
592	603
Assets/MergeIt/Source/Core/FieldElements/Parameters/InfoParameters.cs	Assets/MergeIt/Source/Core/Schemes/SchemeData.cs
592	603
Assets/MergeIt/Source/Core/FieldElements/Parameters/IProducedByParameters.cs	Assets/MergeIt/Source/Core/Schemes/SchemeObject.cs
592	603
Assets/MergeIt/Source/Core/FieldElements/Parameters/IProduceParameters.cs	Assets/MergeIt/Source/Core/Services/IConfigsService.cs
592	603

606	Assets/Mergelt/Source/Core/WindowSystem/SortingLayers.cs,
Assets/Mergelt/Source/Core/Services/ICurrencyService.cs,	615
606	Assets/Mergelt/Source/Core/WindowSystem/WindowName.cs,
Assets/Mergelt/Source/Core/Services/IElementService.cs,	615
606	Assets/Mergelt/Source/Core/WindowSystem/Windows/IWindow.cs,
Assets/Mergelt/Source/Core/Services/IElementsStockService.cs,	615
607	Assets/Mergelt/Source/Core/WindowSystem/Windows/IWindowPresenter.
Assets/Mergelt/Source/Core/Services/IEnergyService.cs,	616
607	Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowBase.cs,
Assets/Mergelt/Source/Core/Services/IEvolutionsService.cs,	616
607	Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowCreateInfo.
Assets/Mergelt/Source/Core/Services/IGameFieldActionsService.cs,	617
607	Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowModel.cs,
Assets/Mergelt/Source/Core/Services/IGameFieldService.cs,	618
608	Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowPresenter.
Assets/Mergelt/Source/Core/Services/IGameLoadService.cs,	618
608	Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowState.cs,
Assets/Mergelt/Source/Core/Services/IGameSaveService.cs,	620
608	Assets/Mergelt/Source/Core/WindowSystem/Windows/WindowStorage.cs,
Assets/Mergelt/Source/Core/Services/IGameService.cs,	620
608	Assets/Mergelt/Source/Core/WindowSystem/WindowsSystem.cs,
Assets/Mergelt/Source/Core/Services/IGeneratorsService.cs,	621
608	Assets/Mergelt/Source/Game/Commands/CheckEvolutionsProgressCom.
Assets/Mergelt/Source/Core/Services/IInfoPanelService.cs,	624
609	Assets/Mergelt/Source/Game/Commands/CheckOpeningGeneratorsCom.
Assets/Mergelt/Source/Core/Services/IInventoryService.cs,	625
609	Assets/Mergelt/Source/Game/Commands/EffectCommand.cs,
Assets/Mergelt/Source/Core/Services/IResourcesLoaderService.cs,	625
609	Assets/Mergelt/Source/Game/Commands/LoadConfigsCommand.cs,
Assets/Mergelt/Source/Core/Services/IUserProgressService.cs,	625
609	Assets/Mergelt/Source/Game/Commands/PrepareEnergyCommand.cs,
Assets/Mergelt/Source/Core/Services/IUserService.cs,	626
609	Assets/Mergelt/Source/Game/Commands/PrepareFieldCommand.cs,
Assets/Mergelt/Source/Core/User/IUserData.cs,	610
610	Assets/Mergelt/Source/Game/Commands/PrepareInventoryCommand.cs,
Assets/Mergelt/Source/Core/WindowSystem/BlackoutCom	Assets/Mergelt/Source/Game/Commands/PrepareInventoryCommand.cs,
610	627
Assets/Mergelt/Source/Core/WindowSystem/Commands/Assets/Mergelt/Source/Game/Commands/PrepareStockCommand.cs,	628
610	628
Assets/Mergelt/Source/Core/WindowSystem/Commands/Assets/Mergelt/Source/Game/Commands/PrepareUserCommand.cs,	628
611	628
Assets/Mergelt/Source/Core/WindowSystem/Commands/Assets/Mergelt/Source/Game/Configs/ConfigsContainer.cs,	629
612	629
Assets/Mergelt/Source/Core/WindowSystem/Data/IWindow	Assets/Mergelt/Source/Game/Configs/ElementPrefab.cs,
612	630
Assets/Mergelt/Source/Core/WindowSystem/Data/IWindow	Assets/Mergelt/Source/Game/Configs/Services/ConfigsService.cs,
612	630
Assets/Mergelt/Source/Core/WindowSystem/Data/Window	Assets/Mergelt/Source/Game/Configs/Services/ConfigsServiceModel.cs,
613	632
Assets/Mergelt/Source/Core/WindowSystem/Data/Window	Assets/Mergelt/Source/Game/Converters/ConfigProcessor.cs,
613	633
Assets/Mergelt/Source/Core/WindowSystem/Factory/IWind	Assets/Mergelt/Source/Game/Converters/IConfigProcessor.cs,
613	637
Assets/Mergelt/Source/Core/WindowSystem/Factory/Wind	Assets/Mergelt/Source/Game/Effects/Configs/JumpEffectConfig.cs,
613	637
Assets/Mergelt/Source/Core/WindowSystem/IWindowSyst	Assets/Mergelt/Source/Game/Effects/Configs/MoveEffectConfig.cs,
614	637
Assets/Mergelt/Source/Core/WindowSystem/Messages/Clos	Assets/Mergelt/Source/Game/Effects/Configs/MoveToUiEffectConfig.cs,
614	637

Assets/MergeIt/Source/Game/Effects/Controllers/BaseEffect.cs	Assets/MergeIt/Source/Game/Factories/FieldElement/IFieldElementVisual.cs
638	654
Assets/MergeIt/Source/Game/Effects/Controllers/BaseEffectWithIcon.cs	Assets/MergeIt/Source/Game/Factories/Icons/IIconFactory.cs
638	654
Assets/MergeIt/Source/Game/Effects/Controllers/IEffect.cs	Assets/MergeIt/Source/Game/Factories/Icons/IIconFactory.cs
639	655
Assets/MergeIt/Source/Game/Effects/Controllers/IEffectWithInventory.cs	Assets/MergeIt/Source/Game/Factories/Inventory/IInventoryFactory.cs
639	655
Assets/MergeIt/Source/Game/Effects/Controllers/JumpEffect.cs	Assets/MergeIt/Source/Game/Factories/Inventory/InventoryData.cs
639	656
Assets/MergeIt/Source/Game/Effects/Controllers/JumpEffectWithInventory.cs	Assets/MergeIt/Source/Game/Factories/Inventory/InventoryFactory.cs
640	656
Assets/MergeIt/Source/Game/Effects/Controllers/MoveEffect.cs	Assets/MergeIt/Source/Game/Field/Actions/FieldActionProcessorBase.cs
641	657
Assets/MergeIt/Source/Game/Effects/Controllers/MoveEffectWithField.cs	Assets/MergeIt/Source/Game/Field/Actions/FieldConsumableProcessor.cs
642	658
Assets/MergeIt/Source/Game/Effects/Controllers/MoveToUnitEffect.cs	Assets/MergeIt/Source/Game/Field/Actions/FieldGenerationProcessor.cs
643	659
Assets/MergeIt/Source/Game/Effects/Controllers/MoveToUnitEffectWithField.cs	Assets/MergeIt/Source/Game/Field/Actions/FieldMergeProcessor.cs
643	661
Assets/MergeIt/Source/Game/Effects/EffectName.cs	Assets/MergeIt/Source/Game/Field/Actions/IFieldActionProcessor.cs
644	664
Assets/MergeIt/Source/Game/Effects/EffectsFactory.cs	Assets/MergeIt/Source/Game/Field/Elements/Animations/FieldElementAnimation.cs
644	664
Assets/MergeIt/Source/Game/Effects/EffectsManager.cs	Assets/MergeIt/Source/Game/Field/Elements/Animations/FieldElementStateManager.cs
645	665
Assets/MergeIt/Source/Game/Effects/IEffectsFactory.cs	Assets/MergeIt/Source/Game/Field/Elements/FieldElementModel.cs
645	665
Assets/MergeIt/Source/Game/Effects/IEffectsManager.cs	Assets/MergeIt/Source/Game/Field/Elements/FieldElementPresenter.cs
646	666
Assets/MergeIt/Source/Game/Effects/Parameters/IEffectParameter.cs	Assets/MergeIt/Source/Game/Field/Elements/FieldElementPresenterBase.cs
646	666
Assets/MergeIt/Source/Game/Effects/Parameters/MoveEffectParameter.cs	Assets/MergeIt/Source/Game/Field/Elements/FieldElementView.cs
646	668
Assets/MergeIt/Source/Game/ElementsStock/ElementsStock.cs	Assets/MergeIt/Source/Game/Field/Elements/Generator/FieldElementGenerator.cs
647	669
Assets/MergeIt/Source/Game/ElementsStock/ElementsStock.cs	Assets/MergeIt/Source/Game/Field/Elements/Generator/FieldElementGenerator.cs
648	670
Assets/MergeIt/Source/Game/Enums/HudPanelType.cs	Assets/MergeIt/Source/Game/Field/Elements/Generator/FieldElementGenerator.cs
649	671
Assets/MergeIt/Source/Game/Evolutions/EvolutionsProgress.cs	Assets/MergeIt/Source/Game/Field/Elements/Generator/GeneratorTimerController.cs
649	671
Assets/MergeIt/Source/Game/Factories/ElementInfo/ElementInfo.cs	Assets/MergeIt/Source/Game/Field/FieldCellComponent.cs
649	672
Assets/MergeIt/Source/Game/Factories/ElementInfo/ElementInfo.cs	Assets/MergeIt/Source/Game/Field/FieldData.cs
650	674
Assets/MergeIt/Source/Game/Factories/ElementInfo/IElementInfo.cs	Assets/MergeIt/Source/Game/Field/FieldLogic.cs
650	675
Assets/MergeIt/Source/Game/Factories/Field/FieldFactory.cs	Assets/MergeIt/Source/Game/Field/FieldLogicModel.cs
650	676
Assets/MergeIt/Source/Game/Factories/Field/IFieldFactory.cs	Assets/MergeIt/Source/Game/Field/FieldModel.cs
651	676
Assets/MergeIt/Source/Game/Factories/Field/IFieldFactory.cs	Assets/MergeIt/Source/Game/Field/FieldPresenter.cs
651	676
Assets/MergeIt/Source/Game/Factories/FieldElement/FieldElement.cs	Assets/MergeIt/Source/Game/Field/FieldView.cs
651	680
Assets/MergeIt/Source/Game/Factories/FieldElement/FieldElement.cs	Assets/MergeIt/Source/Game/Field/IFieldLogic.cs
651	680
Assets/MergeIt/Source/Game/Factories/FieldElement/FieldElement.cs	Assets/MergeIt/Source/Game/GameRoot.cs
653	680
Assets/MergeIt/Source/Game/Factories/FieldElement/FieldElement.cs	Assets/MergeIt/Source/Game/Helpers/FieldElementExtensions.cs
654	681
Assets/MergeIt/Source/Game/Factories/FieldElement/IFieldElement.cs	Assets/MergeIt/Source/Game/Helpers/FieldGeneratorElementExtensions.cs
654	682

Assets/Mergelt/Source/Game/Helpers/HudTargetComponent.cs,	686	698
Assets/Mergelt/Source/Game/Helpers/HudTargets.cs,	686	698
Assets/Mergelt/Source/Game/HUD/EnergyComponent.cs,	686	698
Assets/Mergelt/Source/Game/HUD/HardCurrencyComponent.cs,	688	698
Assets/Mergelt/Source/Game/HUD/HudComponent.cs,	688	698
Assets/Mergelt/Source/Game/HUD/HudPanelBase.cs,	689	699
Assets/Mergelt/Source/Game/HUD/IHudPanel.cs,	689	699
Assets/Mergelt/Source/Game/HUD/IUserListener.cs,	689	699
Assets/Mergelt/Source/Game/HUD/ProgressComponent.cs,	690	699
Assets/Mergelt/Source/Game/HUD/SoftCurrencyComponent.cs,	691	699
Assets/Mergelt/Source/Game/HUD/UserListenerComponent.cs,	692	700
Assets/Mergelt/Source/Game/Managers/HintsManager.cs,	692	700
Assets/Mergelt/Source/Game/Managers/IHintsManager.cs,	694	700
Assets/Mergelt/Source/Game/Messages/ActivateHintsMessage.cs,	694	700
Assets/Mergelt/Source/Game/Messages/CheckGeneratorMessage.cs,	694	700
Assets/Mergelt/Source/Game/Messages/ClickElementMessage.cs,	695	701
Assets/Mergelt/Source/Game/Messages/CreateElementMessage.cs,	695	702
Assets/Mergelt/Source/Game/Messages/ElementActionMessage.cs,	695	703
Assets/Mergelt/Source/Game/Messages/EndDragElementMessage.cs,	695	704
Assets/Mergelt/Source/Game/Messages/EnergyRestoredMessage.cs,	696	704
Assets/Mergelt/Source/Game/Messages/FieldFullMessage.cs,	696	706
Assets/Mergelt/Source/Game/Messages/GeneratorBusyMessage.cs,	696	708
Assets/Mergelt/Source/Game/Messages/GeneratorOpenedMessage.cs,	696	708
Assets/Mergelt/Source/Game/Messages/GeneratorOpenStateMessage.cs,	696	709
Assets/Mergelt/Source/Game/Messages/GeneratorRestoredMessage.cs,	697	710
Assets/Mergelt/Source/Game/Messages/LevelUpdatedMessage.cs,	697	711
Assets/Mergelt/Source/Game/Messages/LevelUpMessage.cs,	697	711
Assets/Mergelt/Source/Game/Messages/LoadedGameMessage.cs,	697	714
Assets/Mergelt/Source/Game/Messages/MergeElementsMessage.cs,	697	715
Assets/Mergelt/Source/Game/Messages/MoveElementMessage.cs,	697	715
Assets/Mergelt/Source/Game/Messages/PrepareGameMessage.cs,	698	715
Assets/Mergelt/Source/Game/Messages/RemoveElementMessage.cs,	698	715
Assets/Mergelt/Source/Game/Messages/RemoveFromInventoryMessage.cs,	698	715
Assets/Mergelt/Source/Game/Messages/ResetHintsMessage.cs,	698	715
Assets/Mergelt/Source/Game/Messages/ResetPositionMessage.cs,	699	715
Assets/Mergelt/Source/Game/Messages/ResetSelectionMessage.cs,	699	715
Assets/Mergelt/Source/Game/Messages/SelectedElementMessage.cs,	699	715
Assets/Mergelt/Source/Game/Messages/SkipTimeMessage.cs,	699	715
Assets/Mergelt/Source/Game/Messages/SplitElementMessage.cs,	699	715
Assets/Mergelt/Source/Game/Messages/StartChargingMessage.cs,	700	715
Assets/Mergelt/Source/Game/Messages/StartGameMessage.cs,	700	715
Assets/Mergelt/Source/Game/Messages/SwapElementsMessage.cs,	700	715
Assets/Mergelt/Source/Game/Messages/UnlockElementMessage.cs,	700	715
Assets/Mergelt/Source/Game/Messages/UpdateStockMessage.cs,	700	715
Assets/Mergelt/Source/Game/Services/CurrencyService.cs,	701	715
Assets/Mergelt/Source/Game/Services/ElementService.cs,	702	715
Assets/Mergelt/Source/Game/Services/ElementsStockService.cs,	703	715
Assets/Mergelt/Source/Game/Services/ElementsStockServiceModel.cs,	704	715
Assets/Mergelt/Source/Game/Services/EnergyService.cs,	704	715
Assets/Mergelt/Source/Game/Services/EvolutionsService.cs,	706	715
Assets/Mergelt/Source/Game/Services/EvolutionsServiceModel.cs,	708	715
Assets/Mergelt/Source/Game/Services/GameFieldActionsService.cs,	708	715
Assets/Mergelt/Source/Game/Services/GameFieldService.cs,	709	715
Assets/Mergelt/Source/Game/Services/GameService.cs,	710	715
Assets/Mergelt/Source/Game/Services/GameServiceModel.cs,	711	715
Assets/Mergelt/Source/Game/Services/GeneratorService.cs,	711	715
Assets/Mergelt/Source/Game/Services/InfoPanelService.cs,	714	715
Assets/Mergelt/Source/Game/Services/InventoryService.cs,	715	715
Assets/Mergelt/Source/Game/Services/InventoryServiceModel.cs,	715	715

717 Assets/MergeIt/Source/Game/Services/ResourcesLoaderService.cs, 743
Assets/MergeIt/Source/Game/Services/ResourcesLoaderService.cs, 717 Assets/MergeIt/Source/Game/Windows/ElementInfo/ElementInfoModel.cs
Assets/MergeIt/Source/Game/Services/Saves/GameLoadService.cs, 743
717 Assets/MergeIt/Source/Game/Windows/ElementInfo/ElementInfoPresenter.cs
Assets/MergeIt/Source/Game/Services/Saves/GameSaveEveryIntervalService.cs,
718 Assets/MergeIt/Source/Game/Windows/ElementInfo/ElementInfoWindow.cs
Assets/MergeIt/Source/Game/Services/Saves/GameSaveOnEveryActionService.cs,
720 Assets/MergeIt/Source/Game/Windows/ElementInfo/ElementInfoWindow.cs
Assets/MergeIt/Source/Game/Services/Saves/GameSaveOnQuitService.cs,
722 Assets/MergeIt/Source/Game/Windows/Inventory/InventoryModel.cs,
Assets/MergeIt/Source/Game/Services/Saves/Strategies/BaseSerializ
723 Assets/MergeIt/Source/Game/Windows/Inventory/InventoryPresenter.cs,
Assets/MergeIt/Source/Game/Services/Saves/Strategies/BinaryFormatterStrategy.cs,
724 Assets/MergeIt/Source/Game/Windows/Inventory/InventoryWindow.cs,
Assets/MergeIt/Source/Game/Services/Saves/Strategies/ISerializeStrategy.cs,
724 Assets/MergeIt/Source/Game/Windows/Inventory/InventoryWindowItemC
Assets/MergeIt/Source/Game/Services/Saves/Strategies/JsonSerializeStrategy.cs,
725 Assets/MergeIt/Source/Game/Windows/Inventory/InventoryWindowItemP
Assets/MergeIt/Source/Game/Services/UserProgressService.cs, 751 Assets/MergeIt/Source/Game/Windows/Inventory/InventoryWindowPaidCe
725 Assets/MergeIt/Source/Game/Services/UserService.cs,
727 Assets/MergeIt/Source/Game/Windows/UserProgress/UserInfoModel.cs,
Assets/MergeIt/Source/Game/Services/UserServiceModel.cs, 752 Assets/MergeIt/Source/Game/Windows/UserProgress/UserInfoPresenter.cs
727 Assets/MergeIt/Source/Game/TopPanel/TopPanelComponent.cs, 752 Assets/MergeIt/Source/Game/Windows/UserProgress/UserInfoWindow.cs
728 Assets/MergeIt/Source/Game/UI/InfoPanel/ElementActionType.cs, 754 Assets/MergeIt/Source/Game/Windows/UserProgress/UserProgressComp
728 Assets/MergeIt/Source/Game/UI/InfoPanel/ElementCommonInfo.cs, 755 Assets/MergeIt/Source/SimpleDI/DiContainer.cs, 755
Assets/MergeIt/Source/Game/UI/InfoPanel/ElementInfo.cs, Assets/MergeIt/Source/SimpleDI/DiHandler.cs, 757
729 Assets/MergeIt/Source/SimpleDI/DiRoot.cs, 758
Assets/MergeIt/Source/Game/UI/InfoPanel/ElementInfoPart.cs, Assets/MergeIt/Source/SimpleDI/IntroduceAttribute.cs,
730 Assets/MergeIt/Source/SimpleDI/ReservedInterfaces/IInitializable.cs,
Assets/MergeIt/Source/Game/UI/InfoPanel/ElementOpenInfo.cs, 759 732 Assets/MergeIt/Source/SimpleDI/ReservedInterfaces/IUpdatable.cs,
Assets/MergeIt/Source/Game/UI/InfoPanel/ElementSellInfo.cs, 759 Assets/MergeIt/Source/Utils/Bindable.cs, 760
Assets/MergeIt/Source/Game/UI/InfoPanel/ElementSpeedUpOpening.cs, 761 Assets/MergeIt/Source/Utils/IMonoApplicationQuitHandler.cs,
735 Assets/MergeIt/Source/Utils/IMonoUpdateHandler.cs,
Assets/MergeIt/Source/Game/UI/InfoPanel/ElementSplitInfo.cs, 761 Assets/MergeIt/Source/Utils/MainThreadDispatcher.cs,
736 Assets/MergeIt/Source/Game/UI/InfoPanel/ElementUnlockInfo.cs, 761 Assets/MergeIt/Source/Utils/MonoEventsListener.cs,
737 Assets/MergeIt/Source/Game/UI/InventoryPanel/InventoryPanelComponent.cs,
738 Assets/MergeIt/Source/Utils/ResponsiveGridLayout.cs,
Assets/MergeIt/Source/Game/UI/InventoryPanel/InventoryPanelItemComponent.cs,
740 Assets/MergeIt/Source/Utils/SlicedFilledImage.cs, 764
Assets/MergeIt/Source/Game/UI/InventoryPanel/InventoryPanelItem.cs, Assets/MergeIt/Source/Utils/Singleton< TType >
740 Mergelt.SimpleDI.DiHandler, 114
Assets/MergeIt/Source/Game/User/UserData.cs, 741 Mergelt.SimpleDI.DiHandler, 115
Assets/MergeIt/Source/Game/Windows/ElementInfo/ElementInfoFieldCellComponent, 207
741 AttachObject
Assets/MergeIt/Source/Game/Windows/ElementInfo/ElementInfoFieldCellComponent, 207
AvailableToDrop

Mergelt.Core.FieldElements.GeneratorParameters, 280
Mergelt.Core.FieldElements.IGeneratorParameters, 339
Mergelt.Core.FieldElements.SavedGeneratorParameters, 459
Awake
Mergelt.Game.Windows.ElementInfo.ElementInfoItemComponent, 139
Mergelt.Game.Inventory.InventoryWindowItemComponent, 370
Mergelt.Game.Inventory.InventoryWindowPaidCellComponent, 372
BakeEvolutionsData
Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView, 195
BaseSettingsDrawer
Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer, 68
Blackout
Mergelt.Core.WindowSystem.WindowsSystem, 522
Bonuses
Mergelt.Core.Configs.LevelUp.LevelUpParameters, 409
BuildContextMenu
Mergelt.Editor.EvolutionsEditor.ElementNode, 150
Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView, 195
BuildLevel
Mergelt.Game.Converters.ConfigProcessor, 92
BuyCellEvent
Mergelt.Game.Windows.Inventory.InventoryWindowPaidCellComponent, 373
canCopySelection
Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView, 199
canDuplicateSelection
Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView, 199
CanLevelUp
Mergelt.Game.Services.UserProgressService, 488
Canvas
Mergelt.Core.FieldElements.IFieldElementPresenter, 331
Mergelt.Core.FieldElements.IFieldElementView, 333
Mergelt.Core.WindowSystem.Windows.WindowBase, 499
Mergelt.Game.Field.Elements.FieldElementPresenterBase<
TView, TModel >, 234
Mergelt.Game.Field.Elements.FieldElementView, 239
Cell
Mergelt.Game.Messages.ClickElementMessage, 82
CellComponents
Mergelt.Game.Field.FieldLogicModel, 248
CellsContainer
Mergelt.Game.Field.FieldView, 255
CellSize
Mergelt.Game.Field.FieldLogicModel, 248
CellSpacing
Mergelt.Game.Field.FieldLogicModel, 248
Chain
Mergelt.Core.Configs.Data.EvolutionData, 181
ChangeCellCommand
Mergelt.Editor.LevelEditor.Commands.ChangeCellCommand, 76
Charged
Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 130
ChargedCount
Mergelt.Core.FieldElements.GeneratorParameters, 280
Mergelt.Core.FieldElements.IGeneratorParameters, 339
Mergelt.Core.FieldElements.SavedGeneratorParameters, 459
ChargeTime
Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 130
Charging
Mergelt.Core.FieldElements.GeneratorParameters, 280
Mergelt.Core.FieldElements.IGeneratorParameters, 339
Clear
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer, 280
Mergelt.Editor.Core.LevelEditor.Commands.LimitedStack<
T >, 410
Mergelt.Editor.GeneratedItemsCreator, 272
Mergelt.Game.UI.InfoPanel.ElementInfo, 134
Mergelt.SimpleDI.DiContainer, 112
ClearCell
Mergelt.Editor.LevelEditor.LevelEditorWindow, 403
ClearCellCommand
Mergelt.Editor.LevelEditor.Commands.ClearCellCommand, 81
ClearNodes
Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView, 195
ClearSaves
Mergelt.Editor.Helpers.SavesHelper, 461
ClearSelection
Mergelt.Editor.LevelEditor.EvolutionSelection, 191
CLICKEvent
Mergelt.Game.UI.InventoryPanel.InventoryPanelItemComponent, 360
Mergelt.Game.Windows.ElementInfo.ElementInfoItemComponent, 140
ClickOutsideArea
Mergelt.Core.WindowSystem.Windows.WindowBase,

	499	Mergelt.Editor.EvolutionsEditor.ElementNode, 152
ClicksCount		Mergelt.Editor.EvolutionsEditor.IElementNode, 317
Mergelt.Core.FieldElements.IFieldElementModel,	329	ConfigParameters
Mergelt.Game.Field.Elements.FieldElementModel,	226	Mergelt.Core.FieldElements.FieldElement, 214
Close		Mergelt.Core.FieldElements.FieldElementData,
Mergelt.Core.WindowSystem.Windows.WindowBase, ConfigsService	497	217
Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >, 516		Mergelt.Core.FieldElements.IFieldElement, 327
Mergelt.Game.UISystem.ElementInfo	135	Mergelt.Game.Effects.Controllers.BaseEffect, 61
Mergelt.Game.Windows.UserInfoWindow, Mergelt.Core.Configs.Types	31	ConsumableOperationType
484		ConsumableSettings
CloseButton		Mergelt.Core.Configs.Elements.ElementCommonSettings,
Mergelt.Core.WindowSystem.Windows.WindowBase,	124	149
384		ConsumableTargetType
CloseEndEvent		Mergelt.Core.Configs.Types, 31
Mergelt.Core.WindowSystem.Windows.IWindow,	Convert	Mergelt.Game.Converters.ConfigProcessor, 92
384	Mergelt.Game.Converters.ConfigProcessor, 92	ConvertToFieldElement
Mergelt.Core.WindowSystem.Windows.WindowBase, ConvertToFieldElement	500	Mergelt.Game.Converters.ConfigProcessor, 92
500		ConvertToFieldElementData
ClosePrevious		Mergelt.Core.WindowSystem.Data.IWindowOpenParameter
388	Mergelt.Game.Converters.ConfigProcessor, 93, 94	Copy
Mergelt.Core.WindowSystem.Data.WindowOpenParameter	514	Mergelt.Core.FieldElements.GridPoint, 294
514		Copy< T >
CloseStartEvent		Mergelt.Core.Schemes.Data.ElementNodeData,
Mergelt.Core.WindowSystem.Windows.IWindow,	153	384
384	Mergelt.Core.Schemes.Data.EvolutionNodeData,	500
Mergelt.Core.WindowSystem.Windows.WindowBase,	186	318
500	Mergelt.Core.Schemes.Data.IElementNodeData,	
CloseWindow		Mergelt.Core.Schemes.Data.IEvolutionNodeData,
Mergelt.Core.Animations.WindowDefaultAnimationController	324	504
Mergelt.Core.Animations.WindowMecanimAnimationController	510	CopyItem
Mergelt.Core.WindowSystem.WindowsSystem,	521	Mergelt.Core.Configs.Data.LevelElementData, 406
521	Mergelt.Core.FieldElements.GeneratorOpenParameters,	
CommonSettings		277
Mergelt.Core.Configs.Elements.ElementConfig,	127	Mergelt.Core.FieldElements.GeneratorParameters,
127	280	280
CommonSettingsDrawer		Mergelt.Core.FieldElements.SavedGeneratorOpenParameters,
Mergelt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer	89	458
Mergelt.Core.FieldElements.GridPoint, 294		Mergelt.Core.FieldElements.SavedGeneratorParameters,
CompareTo		459
Mergelt.Core.Configs.Windows.LayerParameters,	398	Count
398	Mergelt.Editor.Core.LevelEditor.Commands.LimitedStack< T >, 411	
Mergelt.Core.FieldElements.GridPoint, 294		Create
Component		Mergelt.Core.FieldElements.GridPoint, 295
Mergelt.Game.UI.InventoryPanel.InventoryPanel	361	CreateFromPanel
Mergelt.Game.Windows.Inventory.InventoryWindow	371	Mergelt.Game.Windows.ElementInfo.ElementInfoWindow,
InventoryWindow	147	147
InventoryWindow	371	CreateEffect< T >
Config		Mergelt.Game.Effects.EffectsFactory, 120
Mergelt.Core.FieldElements.GeneratableFieldElement,	271	Mergelt.Game.Effects.IEffectsFactory, 314
271		CreateElementWindowItem
Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer, Mergelt.Game.Factories.ElementInfoFactory,	69	137

CreateEvolutionsProgress
Mergelt.Game.Services.EvolutionsService, 202

CreateField
Mergelt.Game.Factories.Field.FieldFactory, 242

CreateFieldCell
Mergelt.Game.Factories.Field.FieldFactory, 242

CreateFieldElement
Mergelt.Game.Factories.FieldElement.FieldElementFactory, 219
Mergelt.Game.Factories.FieldElement.FieldElementVisualFactory, 241

CreateGUI
Mergelt.Editor.LevelEditor.LevelEditorWindow, 403

CreateIcon
Mergelt.Game.Factories.Icons.IconFactory, 308

CreateInspectorGUI
Mergelt.Editor.Configs.ItemsDrawers.ElementConfigInspector, 129

CreateInventory
Mergelt.Game.Services.InventoryService, 364

CreateInventoryPanelItem
Mergelt.Game.Factories.Inventory.InventoryFactory, 357

CreateInventoryWindowItem
Mergelt.Game.Factories.Inventory.InventoryFactory, 357

CreateNewElement
Mergelt.Game.Services.GameFieldService, 259

CreateStock
Mergelt.Game.Services.ElementsStockService, 171

CreateUnknownElementWindowItem
Mergelt.Game.Factories.ElementInfo.ElementInfoFactory, 137

CreateUser
Mergelt.Game.Services.UserService, 490

CreateWindow< TPresenter >
Mergelt.Core.WindowSystem.Factory.IWindowFactory, 387
Mergelt.Core.WindowSystem.Factory.WindowFactory, 506

CreateWindowEmptyCell
Mergelt.Game.Factories.Inventory.InventoryFactory, 357

CreateWindowPaidCell
Mergelt.Game.Factories.Inventory.InventoryFactory, 358

Currency
Mergelt.Core.Configs.Elements.CurrencySettings, 111

CurrencyComponent
Mergelt.Editor.Configs.CurrencyComponent, 108

CurrencyIcon
Mergelt.Core.Configs.Icons.IconParameters, 312

CurrencyType
Mergelt.Core.Configs.Icons.IconParameters, 312
Mergelt.Core.Configs.Types, 31

CurveY

Mergelt.Game.Effects.Configs.JumpEffectConfig, 395

Data
Mergelt.Editor.EvolutionsEditor.ElementNode, 152
Mergelt.Editor.EvolutionsEditor.EvolutionNode, 184
Mergelt.Editor.EvolutionsEditor.IElementNode, 317
Mergelt.Editor.EvolutionsEditor.IEvolutionNode, 323

Default
Mergelt.Core.FieldElements.GridPoint, 296

DefineMessage< T >
Mergelt.Core.Messages.IMessageBus, 349
Mergelt.Core.Messages.MessageBus, 418

Description
Mergelt.Core.Configs.Data.EvolutionData, 181
Mergelt.Core.Configs.Elements.ElementCommonSettings, 124

Mergelt.Core.FieldElements.IInfoParameters, 344
Mergelt.Core.FieldElements.InfoParameters, 353

Mergelt.Core.Schemes.Data.EvolutionNodeData, 186

Mergelt.Core.Schemes.Data.IEvolutionNodeData, 324

Mergelt.Editor.EvolutionsEditor.EvolutionNode, 184
Mergelt.Editor.EvolutionsEditor.IEvolutionNode, 323

Destroy
Mergelt.Game.UI.InfoPanel.ElementInfo, 134
Mergelt.Game.UI.InfoPanel.ElementOpenInfo, 155
Mergelt.Game.UI.InfoPanel.ElementSellInfo, 158
Mergelt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo, 162
Mergelt.Game.UI.InfoPanel.ElementSplitInfo, 167
Mergelt.Game.UI.InfoPanel.ElementUnlockInfo, 174

DestroyEvent
Mergelt.Core.MVP.IView, 383
Mergelt.Core.MVP.View, 495

DestroyWindow
Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >, 516

DetachObject
Mergelt.Game.Field.FieldCellComponent, 207

Discovered
Mergelt.Core.Configs.Data.EvolutionData, 181
Mergelt.Core.Schemes.Data.EvolutionNodeData, 186
Mergelt.Core.Schemes.Data.IEvolutionNodeData, 324

Mergelt.Editor.EvolutionsEditor.EvolutionNode, 184
Mergelt.Editor.EvolutionsEditor.IEvolutionNode, 323

Dispose
Mergelt.Core.Commands.Command, 84

Mergelt.Core.MVP.Presenter< TView, TModel >, 446
Mergelt.Core.WindowSystem.WindowsSystem, 521
Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer, 68
Mergelt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer, 89
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer, 287
Mergelt.Editor.EvolutionsEditor.BaseNode, 65
Mergelt.Editor.EvolutionsEditor.ElementNode, 151
Mergelt.Editor.EvolutionsEditor.EvolutionNode, 184
Mergelt.Editor.GeneratedItemsCreator, 272
Mergelt.Editor.GeneratedItemUI, 275
Mergelt.Game.Field.FieldLogic, 246
Mergelt.Game.Managers.HintsManager, 300
Mergelt.Game.Services.EnergyService, 179
Mergelt.Game.Services.GameFieldActionsService, 258
Mergelt.Game.Services.GameFieldService, 259
Mergelt.Game.Services.GameService, 268
Mergelt.Game.Services.GeneratorService, 283
Mergelt.Game.Services.InfoPanelService, 352
Mergelt.Game.Services.Saves.GameSaveEveryIntervalService, 263
Mergelt.Game.Services.Saves.GameSaveOnEveryActionService, 265
Mergelt.Game.Services.Saves.GameSaveOnQuitService, 267
Mergelt.Game.Services.UserProgressService, 488
Draw
Mergelt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer, 89
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer, 287
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingElementConfigs, 290
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingElementConfigs, 292
Drop< TInterface >
Mergelt.SimpleDI.DiContainer, 112
DropCount
Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 130
DroppedElements
Mergelt.Core.FieldElements.GeneratorParameters, 280
Mergelt.Core.FieldElements.IGeneratorParameters, 340
Mergelt.Core.FieldElements.SavedGeneratorParameters, 459
Duration
Mergelt.Core.Configs.EffectConfig, 119
Edges
Mergelt.Core.Schemes.SchemeData, 462
EffectCommand
Mergelt.Core.Game.Commands.EffectCommand, 118
EffectConfigs
Mergelt.Game.Configs.ConfigsContainer, 95
Mergelt.Game.Configs.Services.ConfigsServiceModel, 104
Element
Mergelt.Core.Configs.Data.LevelElementData, 407
Mergelt.Core.Configs.Elements.GeneratableElement, 270
Mergelt.Game.Messages.ElementActionMessage, 122
Mergelt.Game.Messages.UnlockElementMessage, 477
Mergelt.Game.UI.InventoryPanel.InventoryPanelItemPair, 361
Mergelt.Game.Windows.Inventory.InventoryWindowItemPair, 371
ElementActionType
Mergelt.Game.UI.InfoPanel, 47
ElementConfig
Mergelt.Core.FieldElements.ConfigParameters, 91
Mergelt.Core.FieldElements.IConfigParameters, 309
Mergelt.Core.Schemes.Data.ElementNodeData, 153
ElementData
Mergelt.Core.Schemes.Data.IElementNodeData, 319
ElementInfoArgs
Mergelt.Game.Windows.ElementInfo.ElementInfoArgs, 136
ElementConfigs
Mergelt.Game.Configs.Services.ConfigsServiceModel, 104
ElementData
Mergelt.Editor.LevelEditor.EvolutionSelection, 193
ElementId
Mergelt.Core.FieldElements.SavedConfigParameters, 457
ElementConfigs
Mergelt.Game.Factories.ElementInfo, 43
ElementDrawer
Mergelt.Editor.EvolutionsEditor.ElementNode, 150
Elements
Mergelt.Core.ElementsStock.IElementsStockData, 321
Mergelt.Core.FieldElements.IProducedByParameters, 374
Mergelt.Core.FieldElements.IProduceParameters, 375
Mergelt.Core.FieldElements.ProducedByParameters, 448
Mergelt.Core.FieldElements.ProduceParameters, 449
Mergelt.Game.ElementsStock.ElementsStockData, 170
Mergelt.Game.Services.ElementsStockServiceModel, 173
ElementsNodesData
Mergelt.Core.Schemes.SchemeData, 462

ElementType
 Mergelt.Core.Configs.Types, 32

EnableLevelUpButton
 Mergelt.Game.Windows.UserProgress.UserInfoWindow, 194
 484

EndDrag
 Mergelt.Game.Field.Elements.FieldElementPresenterTView, TModel >, 230

Energy
 Mergelt.Core.User.IUserData, 379
 Mergelt.Game.Services.UserServiceModel, 491
 Mergelt.Game.User.UserData, 478

EnergyCap
 Mergelt.Core.Configs.GameConfig, 256

EnergyCost
 Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 131

EnergyRestoreTime
 Mergelt.Core.Configs.GameConfig, 256

EnergyRestoringStartTime
 Mergelt.Core.User.IUserData, 379
 Mergelt.Game.Services.UserServiceModel, 491
 Mergelt.Game.User.UserData, 478

Enqueue
 Mergelt.Core.Helpers.MainThreadDispatcher, 414

Equals
 Mergelt.Core.Configs.Data.EvolutionData, 181
 Mergelt.Core.Configs.Windows.LayerParameters, 398
 Mergelt.Core.FieldElements.GridPoint, 295
 Mergelt.Core.WindowSystem.Windows.WindowCreateInfo, 503

EvolItem
 Mergelt.Editor.LevelEditor.EvolutionSelection, 193

Evolution
 Mergelt.Core.Schemes.SchemeObject, 463
 Mergelt.Editor.LevelEditor.EvolutionSelection, 193

EvolutionData
 Mergelt.Core.Configs.Data.EvolutionData, 180
 Mergelt.Core.FieldElements.ConfigParameters, 91
 Mergelt.Core.FieldElements.IConfigParameters, 309

EvolutionId
 Mergelt.Core.Configs.Data.LevelElementData, 407
 Mergelt.Core.Evolutions.EvolutionProgressData, 188
 Mergelt.Core.FieldElements.SavedConfigParameters, 457

EvolutionNode
 Mergelt.Editor.EvolutionsEditor.EvolutionNode, 183

EvolutionNodesData
 Mergelt.Editor.EvolutionsEditor.SerializedElementNodes, 465

EvolutionPanel
 Mergelt.Game.Windows.ElementInfo.ElementInfoWindow, 125
 147

EvolutionsData

Mergelt.Core.Schemes.SchemeData, 462

EvolutionsGraphView
 Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView, 194

EvolutionsNodesData
 Mergelt.Core.Schemes.SchemeData, 462

BasktionsProgress
 Mergelt.Core.Evolutions.IEvolutionsProgressData, 325

Mergelt.Game.Evolutions.EvolutionsProgressData, 201

Mergelt.Game.Services.EvolutionsServiceModel, 205

EvolutionsScheme
 Mergelt.Core.Configs.LevelConfig, 401

Execute
 Mergelt.Core.Commands.Command, 84
 Mergelt.Editor.LevelEditor.Commands.ActionCommandBase, 52
 Mergelt.Editor.LevelEditor.Commands.ApplyCellCommand, 59
 Mergelt.Editor.LevelEditor.Commands.ChangeCellCommand, 76
 Mergelt.Editor.LevelEditor.Commands.ClearCellCommand, 81
 Mergelt.Game.Commands.CheckEvolutionsProgressCommand, 78
 Mergelt.Game.Commands.CheckOpeningGeneratorsCommand, 80
 Mergelt.Game.Commands.EffectCommand, 118
 Mergelt.Game.Commands.PrepareEnergyCommand, 439
 Mergelt.Game.Commands.PrepareFieldCommand, 440
 Mergelt.Game.Commands.PrepareInventoryCommand, 442
 Mergelt.Game.Commands.PrepareStockCommand, 443
 Mergelt.Game.Commands.PrepareUserCommand, 445

ExecuteAsync
 Mergelt.Core.Commands.Command, 84
 Mergelt.Game.Commands.LoadConfigsCommand, 413

ExecuteCommand
 Mergelt.Editor.LevelEditor.Commands.ActionCommandManager, 53

Executing
 Mergelt.Core.Commands.CommandManager, 87
 Mergelt.Core.Commands.ICommandManager, 307

Expand
 Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer, 68

Expanded
 Mergelt.Core.Configs.Elements.ElementCommonSettings, 125

Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 131

Experience
Mergelt.Core.Configs.LevelUp.LevelUpParameters, 409
Mergelt.Core.User.IUserData, 379
Mergelt.Game.Services.UserServiceModel, 491
Mergelt.Game.User(userData, 478)

Extension
Mergelt.Core.Saves.SavableAttribute, 457

FieldCellObjectEvenPath
Mergelt.Game.Factories.Field.FieldFactory, 243

FieldCellObjectOddPath
Mergelt.Game.Factories.Field.FieldFactory, 243

FieldElement
Mergelt.Game.Field.Elements.FieldElementPresenterBase<
TView, TModel >, 234
Mergelt.Game.Messages.RemoveFromInventoryMessage, 453

FieldElementGeneratorPresenter
Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorPresenter, 221

FieldElementPresenter
Mergelt.Game.Field.FieldCellComponent, 210

FieldElements
Mergelt.Game.Field.FieldLogicModel, 248

FieldElementsData
Mergelt.Core.Configs.LevelConfig, 401

FieldElementState
Mergelt.Core.FieldElements, 33

FieldElementView
Mergelt.Game.Messages.MoveElementMessage, 431

FieldGameObjectPath
Mergelt.Game.Factories.Field.FieldFactory, 243

FieldHeight
Mergelt.Core.Configs.LevelConfig, 401
Mergelt.Game.Field.FieldData, 213
Mergelt.Game.Field.FieldLogicModel, 248

FieldLogicModel
Mergelt.Game.Field.Actions.FieldActionProcessorBase, 206

FieldPresenter
Mergelt.Game.Field.FieldPresenter, 253

FieldRect
Mergelt.Game.Field.FieldLogicModel, 248

FieldWidth
Mergelt.Core.Configs.LevelConfig, 401
Mergelt.Game.Field.FieldData, 213
Mergelt.Game.Field.FieldLogicModel, 249

fillAmount
Mergelt.Core.Helpers.SlicedFilledImage, 468

fillCenter
Mergelt.Core.Helpers.SlicedFilledImage, 469

FillDirection
Mergelt.Core.Helpers.SlicedFilledImage, 467

fillDirection
Mergelt.Core.Helpers.SlicedFilledImage, 469

Finish
Mergelt.Core.Commands.Command, 85

Mergelt.Game.Effects.Controllers.BaseEffect, 60

FinishCallbackHandler
Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter, 63

Finished
Mergelt.Core.Commands.Command, 85
Mergelt.Core.Commands.CommandManager, 87
Mergelt.Core.Commands.ICommand, 306
Mergelt.Core.Commands.ICommandManager, 307

FinishedCallback
Mergelt.Game.Effects.Controllers.BaseEffect, 61

Fire< T >
Mergelt.Core.Messages.IMessageBus, 349
Mergelt.Core.Messages.MessageBus, 418, 419

From<
From
Mergelt.Game.Messages.MergeElementsMessage, 417
Mergelt.Game.Messages.ResetPositionMessage, 454
Mergelt.Game.Messages.SwapElementsMessage, 476

FromPoint
Mergelt.Game.Messages.EndDragElementMessage, 176
Mergelt.Game.Messages.MoveElementMessage, 431

FromPosition
Mergelt.Game.Effects.Parameters.MoveEffectParameters, 427
Mergelt.Game.Messages.CreateElementMessage, 106

GameConfig
Mergelt.Core.Services.IConfigsService, 311
Mergelt.Game.Configs.ConfigsContainer, 95
Mergelt.Game.Configs.Services.ConfigsService, 100
Mergelt.Game.Configs.Services.ConfigsServiceModel, 104

GameObject
Mergelt.Core.MVP.IView, 382
Mergelt.Core.MVP.View, 494

GameSaveType
Mergelt.Core.Saves, 35

Generate
Mergelt.Editor.GeneratedItemsCreator, 273

GenerateBlocked
Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 131

GeneratedItemsCreator
Mergelt.Editor.GeneratedItemsCreator, 272

GeneratedItemUI
Mergelt.Editor.GeneratedItemUI, 274

GenerateGuid
Mergelt.Core.Configs.Elements.ElementConfig, 127

GenerateItems
Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 131

GeneratesPanel	GetData
Mergelt.Game.Windows.ElementInfo.ElementInfoWindow, Mergelt.Game.Services.ElementsStockService,	
147	171
GeneratorOpenParameters	Mergelt.Game.Services.EvolutionsService, 202
Mergelt.Core.FieldElements.FieldElement, 214	Mergelt.Game.Services.InventoryService, 364
Mergelt.Core.FieldElements.FieldElementData,	Mergelt.Game.Services.UserServiceModel, 491
217	
Mergelt.Core.FieldElements.IFieldElement, 327	
GeneratorParameters	GetEffectConfig< T >
Mergelt.Core.FieldElements.FieldElement, 214	Mergelt.Core.Services.IConfigsService, 310
Mergelt.Core.FieldElements.FieldElementData,	Mergelt.Game.Configs.Services.ConfigsService,
218	98
Mergelt.Core.FieldElements.IFieldElement, 327	
GeneratorPoint	GetElementPrefab
Mergelt.Game.Messages.CheckGeneratorMessage,	Mergelt.Game.Configs.Services.ConfigsService,
79	99
Mergelt.Game.Messages.GeneratorOpenedMessage	GetEvolutionByElement
276	Mergelt.Game.Configs.Services.ConfigsService,
Mergelt.Game.Messages.GeneratorOpenStartMessage	99
279	
Mergelt.Game.Messages.GeneratorRestoredMessage,	GetEvolutionIdByElement
282	Mergelt.Game.Configs.Services.ConfigsService,
GeneratorSettings	99
Mergelt.Core.Configs.Elements.ElementConfig,	
127	
GeneratorSettingsBaseDrawer	GetEvolutionProgress
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSetting	Mergelt.Game.Services.EvolutionsService, 202
286	
GeneratorSettingsEvolutionDrawer	GetGeneratedElementState
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSetting	Mergelt.Game.Field.Elements.Animations.FieldElementAnimationCo
290	216
GeneratorSettingsInspectorDrawer	GetFieldIconDrawer,
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsInspec	Mergelt.Game.Services.GameFieldService, 259
292	
Get< TInterface >	GetGeneratedBy
Mergelt.SimpleDI.DiContainer, 112	Mergelt.Game.Services.EvolutionsService, 203
GetBlackout	GetGenerates
Mergelt.Core.WindowSystem.Factory.WindowFactory,	Mergelt.Game.Services.EvolutionsService, 203
506	
GetClone	GetHashCode
Mergelt.Core.Configs.Data.LevelElementData, 406	Mergelt.Core.Configs.Data.EvolutionData, 181
GetCompatiblePorts	Mergelt.Core.Configs.Windows.LayerParameters,
Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView,GetIconComponent	399
196	
GetConfig	GetCoreFieldPoint
Mergelt.Game.Configs.Services.ConfigsService,	Mergelt.Core.FieldElements.GridPoint, 295
98	
GetConfigs	Mergelt.Core.WindowSystem.Windows.WindowCreateInfo,
Mergelt.Game.Configs.Services.ConfigsService,	503
98	
GetCurrentLevelMaxExp	GetClone
Mergelt.Game.Services.ElementsStockService,	Mergelt.Core.Configs.Elements.ElementConfig,
171	127
GetCurrent	GetConfig
Mergelt.Game.Services.ElementsStockService,	Mergelt.Core.Configs.Elements.FieldElementIconComponent,
171	225
GetNext	GetConfig
Mergelt.Game.Services.ElementsStockService,	Mergelt.Core.Configs.Elements.FieldElementIconComponent,
171	225
GetNext	GetImage
Mergelt.Game.Configs.Services.ConfigsService,	Mergelt.Core.Configs.Elements.FieldElementIconComponent,
171	225
GetNext	GetLevelUpData
Mergelt.Game.Configs.Services.ConfigsService,	Mergelt.Game.Configs.Services.ConfigsService,
171	100
GetNext	GetLevelUpPrizes
Mergelt.Game.Services.ElementsStockService,	Mergelt.Game.Services.UserProgressService, 488
171	
GetNext	GetModel
Mergelt.Game.Services.ElementsStockService,	Mergelt.Game.Field.Elements.FieldElementPresenterBase<
171	TView, TModel >, 230
GetNext	GetModel
Mergelt.Game.Services.ElementsStockService,	Mergelt.Game.Field.Elements.FieldElementPresenterBase<
171	TView, TModel >, 230

172
GetObject< T >
 Mergelt.Core.Services.IResourcesLoaderService,
 375
GetObject< TResource >
 Mergelt.Game.Services.ResourcesLoaderService,
 455
GetPort
 Mergelt.Editor.EvolutionsEditor.BaseNode, 65
GetPosition
 Mergelt.Game.Field.FieldLogicModel, 247
GetPresenter
 Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter,
 63
GetRoot
 Mergelt.Core.WindowSystem.Factory.WindowFactory,
 507
GetUxml
 Mergelt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer,
 90
 Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsBasedDrawer,
 287
GetView
 Mergelt.Game.Field.Elements.FieldElementPresenterBase<
 TView, TModel >, 231
GraphicRaycaster
 Mergelt.Core.FieldElements.IFieldElementView,
 333
 Mergelt.Game.Field.Elements.FieldElementView,
 239
GridPoint
 Mergelt.Core.FieldElements.GridPoint, 294
HandleEvent
 Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView,
 196
HardCurrency
 Mergelt.Core.User.IUserData, 379
 Mergelt.Game.Services.UserServiceModel, 492
 Mergelt.Game.User.UserData, 479
hasBorder
 Mergelt.Core.Helpers.SlicedFilledImage, 469
Hide
 Mergelt.Core.WindowSystem.Windows.WindowPresenter<
 TWindow, TModel >, 516
 Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer,Mergelt.Editor.LevelEditor.EvolutionSelection, 191
 69
HideTimer
 Mergelt.Game.Field.Elements.Generator.FieldElementGenerator,
 224
HintsConfig
 Mergelt.Core.Services.IConfigsService, 311
 Mergelt.Game.Configs.ConfigsContainer, 95
 Mergelt.Game.Configs.Services.ConfigsService,
 100
 Mergelt.Game.Configs.Services.ConfigsServiceModel,
 104
Hold
 Mergelt.Core.Utils.Bindable< T >, 72
HudPanelType
 Mergelt.Game.Enums, 42
Icon
 Mergelt.Core.Configs.Elements.ElementCommonSettings,
 125
IconsConfig
 Mergelt.Game.Configs.ConfigsContainer, 95
 Mergelt.Game.Configs.Services.ConfigsServiceModel,
 104
Id
 Mergelt.Core.Configs.Data.EvolutionData, 182
 Mergelt.Core.Configs.Elements.ElementConfig,
 127
 Mergelt.Core.Schemes.Data.ElementNodeData,
 153
 Mergelt.Core.Schemes.Data.EvolutionNodeData,
 186
 Mergelt.Core.Schemes.Data.IElementNodeData,
 319
 Mergelt.Core.Schemes.Data.IEvolutionNodeData,
 324
 Mergelt.Editor.EvolutionsEditor.BaseNode, 66
 Mergelt.Editor.EvolutionsEditor.ElementNode, 152
 Mergelt.Editor.EvolutionsEditor.EvolutionNode,
 184
 Mergelt.Editor.EvolutionsEditor.ISaveNode, 376
Index
 Mergelt.Editor.GeneratedItemUI, 275
InfoClickEvent
 Mergelt.Game.Windows.Inventory.InventoryWindowItemComponent,
 370
InfoPanelCurrencyIcons
 Mergelt.Core.Configs.Icons.GameIconsConfig,
 260
InfoParameters
 Mergelt.Core.FieldElements.FieldElement, 215
 Mergelt.Core.FieldElements.FieldElementData,
 218
 Mergelt.Core.FieldElements.IFieldElement, 328
Init
 Mergelt.Editor.Configs.CurrencyComponentsTraits,
 109
 Mergelt.Editor.EvolutionsEditor.EvolutionsGraphViewTraits,
 200
InitialCapacity
 Mergelt.Core.Configs.Inventory.InventoryConfig,
 375
InitialHardCurrency
 Mergelt.Core.Configs.GameConfig, 256
Initialize
 Mergelt.Core.Animations.AnimationControllerBase,
 55
 Mergelt.Core.Animations.MecanimController, 416
 Mergelt.Core.Animations.WindowDefaultAnimationController,
 504
 Mergelt.Core.Animations.WindowMecanimAnimationController,
 510

Mergelt.Core.Messages.MessageBus, 419
Mergelt.Core.MVP.Presenter< TView, TModel >, 446
Mergelt.Core.MVP.View, 494
Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >, 516
Mergelt.Core.WindowSystem.WindowsSystem, 521
Mergelt.Editor.EvolutionsEditor.EvolutionsEditorWindow< TInventoryElements >, 189
Mergelt.Game.ElementsStock.ElementsStockComponent, Mergelt.Game.Factories.Inventory.InventoryData, 168
Mergelt.Game.Field.Elements.Animations.FieldElementAnimator< TInventoryElements >, 216
Mergelt.Game.Field.FieldCellComponent, 208
Mergelt.Game.Field.FieldLogic, 246
Mergelt.Game.Field.FieldPresenter, 253
Mergelt.Game.HUD.ProgressComponent, 450
Mergelt.Game.Managers.HintsManager, 300
Mergelt.Game.Services.EnergyService, 179
Mergelt.Game.Services.GameFieldActionsService, 258
Mergelt.Game.Services.GameFieldService, 259
Mergelt.Game.Services.GameService, 268
Mergelt.Game.Services.GeneratorService, 283
Mergelt.Game.Services.InfoPanelService, 352
Mergelt.Game.Services.Saves.GameSaveEveryIntervalService, 263
Mergelt.Game.Services.Saves.GameSaveOnEveryActionService, 265
Mergelt.Game.Services.Saves.GameSaveOnQuitService, 267
Mergelt.Game.Services.UserProgressService, 489
Initialized
Mergelt.Editor.EvolutionsEditor.EvolutionsEditorWindow, 191
InitializeEvent
Mergelt.Core.MVP.IView, 383
Mergelt.Core.MVP.View, 495
InitialSoftCurrency
Mergelt.Core.Configs.GameConfig, 256
InitialSplittersCount
Mergelt.Core.Configs.GameConfig, 257
InitiateCloseEvent
Mergelt.Core.WindowSystem.Windows.IWindow, 384
Mergelt.Core.WindowSystem.Windows.WindowBase< TIsConsumable >, 500
InputNodeID
Mergelt.Core.Schemes.Data.EdgeData, 116
InputPortName
Mergelt.Core.Schemes.Data.EdgeData, 116
Instance
Mergelt.Core.Helpers.MainThreadDispatcher, 415
Mergelt.Core.Helpers.MonoEventsListener, 422
Interval
Mergelt.Core.Configs.Hints.HintsConfig, 299
IntroduceAttribute
Mergelt.SimpleDI.IntroduceAttribute, 354
Mergelt.Core.Services.IConfigsService, 311
Mergelt.Game.Configs.ConfigsContainer, 96
Mergelt.Game.Configs.Services.ConfigsService, 101
Mergelt.Game.Configs.Services.ConfigsServiceModel, 104
Mergelt.Game.Configs.Services.InventoryServiceModel, 367
InventoryPanelItemPair
Mergelt.Game.UI.InventoryPanel.InventoryPanelItemPair, 361
InventorySize
Mergelt.Core.Inventory.IInventoryData, 346
Mergelt.Game.Factories.Inventory.InventoryData, 356
Mergelt.Game.Services.InventoryServiceModel, 367
InventoryWindowItemPair
Mergelt.Game.Windows.Inventory.InventoryWindowItemPair, 371
IsAvailable
Mergelt.Core.FieldElements.IFieldElementPresenter, 381
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >, 234
IsBlocked
Mergelt.Core.Configs.Data.LevelElementData, 407
Mergelt.Core.FieldElements.IInfoParameters, 344
Mergelt.Core.FieldElements.InfoParameters, 353
Mergelt.Core.FieldElements.SavedInfoParameters, 460
IsBusy
Mergelt.Core.FieldElements.IFieldElementModel, 329
Mergelt.Core.FieldElements.IFieldElementPresenter, 331
Mergelt.Game.Field.Elements.FieldElementModel, 226
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >, 235
IsConsumable
Mergelt.Core.Configs.Elements.ElementCommonSettings, 125
IsFull
Mergelt.Game.Services.InventoryServiceModel, 366
IsLocked
Mergelt.Core.FieldElements.IFieldElementModel, 329
Mergelt.Core.FieldElements.IFieldElementPresenter, 331
Mergelt.Game.Field.Elements.FieldElementModel,

226
Mergelt.Game.Field.Elements.FieldElementPresenterBase< T >, 410
TView, TModel >, 235
Load
Mergelt.Editor.Core.LevelEditor.Commands.LimitedStack<
IsMainThread
Mergelt.Core.Helpers.MainThreadDispatcher, 414
Load
Mergelt.Game.Configs.Services.ConfigsService,
100
IsOpening
Load< T >
Mergelt.Core.FieldElements.GeneratorOpenParameters, 277
Mergelt.Core.FieldElements.IGeneratorOpenParameters, 261
Mergelt.Game.Services.Saves.GameLoadService,
336
338
Mergelt.Game.Services.Saves.Strategies.BaseSerializeStrategy,
67
IsUnlimited
Mergelt.Core.Configs.Elements.ElementGeneratorSettings
Mergelt.Game.Services.Saves.Strategies.BinaryFormatterStrategy,
131
ItemClickEvent
Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy,
370
Mergelt.Game.Windows.Inventory.InventoryWindowItemCompo
877
391
ItemsGrid
Mergelt.Game.Windows.ElementInfo.ElementInfoItem
142
Mergelt.Game.Windows.Inventory.InventoryWindow, Lock
369
Mergelt.Game.Field.Elements.FieldElementView,
238
JumpEffect
LockCell
Mergelt.Game.Effects.Controllers.JumpEffect, 393
Mergelt.Editor.LevelEditor.EvolutionSelection, 192
Layer
LogicPosition
Mergelt.Core.WindowSystem.Windows.IWindowPresenter
389
Mergelt.Core.WindowSystem.Windows.WindowPresenter<
TWWindow, TModel >, 520
460
Mergelt.Core.FieldElements.IInfoParameters,
344
Mergelt.Core.FieldElements.InfoParameters, 353
Mergelt.Core.FieldElements.SavedInfoParameters,
MainCamera
MainCanvas
mainTexture
material
MaxDrop
Mergelt.Core.Configs.Elements.ElementGeneratorSettings,
131
Mergelt, 29
Mergelt.Core, 29
Mergelt.Core.Animations, 29
Mergelt.Core.Animations.AnimationControllerBase, 54
Initialize, 55
SetState, 55
SetState< T >, 55
Mergelt.Core.Animations.AnimationListener, 56
ResetAnimationState, 56
Mergelt.Core.Animations.AnimatorStateMachineBehaviour,
56
OnStateEnter, 57
OnStateUpdate, 57
Mergelt.Core.Animations.IAnimationController, 304
SetState< T >, 304
Mergelt.Core.Animations.IAnimationListener, 305
Mergelt.Core.Animations.IWindowAnimationController,
385

Mergelt.Core.Animations.IWindowAnimationListener, 385
Mergelt.Core.Animations.MecanimController, 415
 Initialize, 416
 SetState, 416
 SetState< T >, 416
Mergelt.Core.Animations.WindowDefaultAnimationController, 504
 CloseWindow, 504
 Initialize, 504
 OnCloseEnd, 505
 OnOpenEnd, 505
 OpenWindow, 505
Mergelt.Core.Animations.WindowMecanimAnimationController, 509
 CloseWindow, 510
 Initialize, 510
 OnCloseEnd, 510
 OnOpenEnd, 510
 OpenWindow, 510
Mergelt.Core.Commands, 29
Mergelt.Core.Commands.Command, 83
 Dispose, 84
 Execute, 84
 ExecuteAsync, 84
 Finish, 85
 Finished, 85
 OnDispose, 85
Mergelt.Core.Commands.CommandManager, 85
 Add, 86
 Executing, 87
 Finished, 87
 Run, 86
 RunAsync, 86
 RunSimultaneously, 87
Mergelt.Core.Commands.ICommand, 305
 Finished, 306
Mergelt.Core.Commands.ICommandManager, 306
 Executing, 307
 Finished, 307
Mergelt.Core.Configs, 30
Mergelt.Core.Configs.Data, 30
Mergelt.Core.Configs.Data.EvolutionData, 180
 Chain, 181
 Description, 181
 Discovered, 181
 Equals, 181
 EvolutionData, 180
 GetHashCode, 181
 Id, 182
 Name, 182
Mergelt.Core.Configs.Data.LevelElementData, 406
 CopyFrom, 406
 Element, 407
 EvolutionId, 407
 GetClone, 406
 IsBlocked, 407
 Position, 407
Mergelt.Core.Configs.Effects, 30
Mergelt.Core.Configs.Effects.EffectConfig, 118
 Duration, 119
 Name, 119
Mergelt.Core.Configs.Elements, 30
Mergelt.Core.Configs.Elements.ConsumableSettings, 105
 Operation, 105
 Target, 105
 Value, 106
Mergelt.Core.Configs.Elements.CurrencySettings, 111
 Amount, 111
 Currency, 111
Mergelt.Core.Configs.Elements.ElementCommonSettings, 124
 ConsumableSettings, 124
 Description, 124
 Expanded, 125
 Icon, 125
 IsConsumable, 125
 Name, 125
 SellCostSettings, 125
 SplitCostSettings, 125
 UnlockCostSettings, 126
Mergelt.Core.Configs.Elements.ElementConfig, 126
 CommonSettings, 127
 GenerateGuid, 127
 GeneratorSettings, 127
 GetIconComponent, 127
 Id, 127
 Type, 128
Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 130
 AdSkip, 130
 Charged, 130
 ChargeTime, 130
 DropCount, 130
 EnergyCost, 131
 Expanded, 131
 GenerateBlocked, 131
 GenerateItems, 131
 IsUnlimited, 131
 MaxDrop, 131
 MinDrop, 132
 NeedOpen, 132
 OpenTime, 132
 SkipChargeCostSettings, 132
 SkipOpenCostSettings, 132
 Stack, 132
Mergelt.Core.Configs.Elements.FieldElementIconComponent, 224
 GetImage, 225
 RectTransform, 225
 SetBlocked, 225
Mergelt.Core.Configs.Elements.GeneratableElement, 270
 Element, 270
 Possibility, 270

Mergelt.Core.Configs.GameConfig, 256
EnergyCap, 256
EnergyRestoreTime, 256
InitialHardCurrency, 256
InitialSoftCurrency, 256
InitialSplittersCount, 257
LevelString, 257
SaveInterval, 257
Mergelt.Core.Configs.Hints, 30
Mergelt.Core.Configs.Hints.HintsConfig, 298
 Interval, 299
Mergelt.Core.Configs.Icons, 30
Mergelt.Core.Configs.Icons.GameliconsConfig, 260
 InfoPanelCurrencyIcons, 260
Mergelt.Core.Configs.Icons.IconParameters, 311
 CurrencyIcon, 312
 CurrencyType, 312
Mergelt.Core.Configs.Inventory, 31
Mergelt.Core.Configs.Inventory.InventoryConfig, 354
 InitialCapacity, 355
 PaidCells, 355
Mergelt.Core.Configs.LevelConfig, 400
 EvolutionsScheme, 401
 FieldElementsData, 401
 FieldHeight, 401
 FieldWidth, 401
Mergelt.Core.Configs.LevelUp, 31
Mergelt.Core.Configs.LevelUp.LevelUpConfig, 407
 LevelUp, 408
Mergelt.Core.Configs.LevelUp.Parameters, 409
 Bonuses, 409
 Experience, 409
Mergelt.Core.Configs.Types, 31
 ConsumableOperationType, 31
 ConsumableTargetType, 31
 CurrencyType, 31
 ElementType, 32
Mergelt.Core.Configs.Windows, 32
Mergelt.Core.Configs.Windows.LayerParameters, 398
 CompareTo, 398
 Equals, 398
 GetHashCode, 399
 Name, 399
 Order, 399
Mergelt.Core.Configs.Windows.LayersConfig, 399
 Layers, 400
Mergelt.Core.ElementsStock, 32
Mergelt.Core.ElementsStock.IElementsStockComponent, 320
Mergelt.Core.ElementsStock.IElementsStockData, 320
 Elements, 321
Mergelt.Core.Evolutions, 32
Mergelt.Core.Evolutions.EvolutionProgressData, 187
 EvolutionId, 188
 Progress, 188
Mergelt.Core.Evolutions.IEvolutionsProgressData, 325
 EvolutionsProgress, 325
Mergelt.Core.FieldElements, 33
FieldElementState, 33
Mergelt.Core.FieldElements.ConfigParameters, 90
 ElementConfig, 91
 EvolutionData, 91
Mergelt.Core.FieldElements.FieldElement, 213
 ConfigParameters, 214
 GeneratorOpenParameters, 214
 GeneratorParameters, 214
 InfoParameters, 215
 ProducedByParameters, 215
 ProduceParameters, 215
 ToString, 214
Mergelt.Core.FieldElements.FieldElementData, 217
 ConfigParameters, 217
 GeneratorOpenParameters, 217
 GeneratorParameters, 218
 InfoParameters, 218
Mergelt.Core.FieldElements.GeneratableFieldElement, 271
 Config, 271
 Possibility, 271
Mergelt.Core.FieldElements.GeneratorOpenParameters, 277
 CopyFrom, 277
 IsOpening, 277
 RemainingTime, 277
 StartOpeningTime, 278
Mergelt.Core.FieldElements.GeneratorParameters, 279
 AvailableToDelete, 280
 ChargedCount, 280
 Charging, 280
 CopyFrom, 280
 DroppedElements, 280
 MinDropChargeTime, 280
 MinDropFullChargeTime, 281
 RemainChargeTime, 281
 StartChargingTime, 281
Mergelt.Core.FieldElements.GridPoint, 293
 CompareTo, 294
 Copy, 294
 Create, 295
 Default, 296
 Equals, 295
 GetHashCode, 295
 GridPoint, 294
 operator!=, 295
 operator==, 296
 ToString, 296
 X, 296
 Y, 296
Mergelt.Core.FieldElements.IConfigParameters, 308
 ElementConfig, 309
 EvolutionData, 309
Mergelt.Core.FieldElements.IFieldElement, 327
 ConfigParameters, 327
 GeneratorOpenParameters, 327
 GeneratorParameters, 327
 InfoParameters, 328

ProducedByParameters, 328
 ProduceParameters, 328
Mergelt.Core.FieldElements.IFieldElementModel, 329
 ClicksCount, 329
 IsBusy, 329
 IsLocked, 329
 Point, 330
 Selected, 330
Mergelt.Core.FieldElements.IFieldElementPresenter, 330
 Canvas, 331
 IsAvailable, 331
 IsBusy, 331
 IsLocked, 331
 RectTransform, 331
 State, 332
 Transform, 332
Mergelt.Core.FieldElements.IFieldElementView, 332
 Canvas, 333
 GraphicRaycaster, 333
 RectTransform, 333
Mergelt.Core.FieldElements.IGeneratorOpenParameters, 337
 IsOpening, 338
 RemainingTime, 338
 StartOpeningTime, 338
Mergelt.Core.FieldElements.IGeneratorParameters, 339
 AvailableToDrop, 339
 ChargedCount, 339
 Charging, 339
 DroppedElements, 340
 MinDropChargeTime, 340
 MinDropFullChargeTime, 340
 RemainChargeTime, 340
 StartChargingTime, 340
Mergelt.Core.FieldElements.IInfoParameters, 343
 Description, 344
 IsBlocked, 344
 LogicPosition, 344
 Name, 344
 Type, 344
Mergelt.Core.FieldElements.InfoParameters, 352
 Description, 353
 IsBlocked, 353
 LogicPosition, 353
 Name, 353
 Type, 353
Mergelt.Core.FieldElements.IProducedByParameters, 374
 Elements, 374
Mergelt.Core.FieldElements.IProduceParameters, 374
 Elements, 375
Mergelt.Core.FieldElements.ProducedByParameters, 447
 Elements, 448
Mergelt.Core.FieldElements.ProduceParameters, 448
 Elements, 449
Mergelt.Core.FieldElements.SavedConfigParameters, 457
 ElementId, 457
 EvolutionId, 457
Mergelt.Core.FieldElements.SavedGeneratorOpenParameters, 458
 CopyFrom, 458
 StartOpeningTime, 458
Mergelt.Core.FieldElements.SavedGeneratorParameters, 458
 AvailableToDrop, 459
 ChargedCount, 459
 CopyFrom, 459
 DroppedElements, 459
 StartChargingTime, 459
Mergelt.Core.FieldElements.SavedInfoParameters, 460
 IsBlocked, 460
 LogicPosition, 460
Mergelt.Core.Helpers, 34
Mergelt.Core.Helpers.IMonoApplicationQuitHandler, 350
Mergelt.Core.Helpers.IMonoUpdateHandler, 351
Mergelt.Core.Helpers.MainThreadDispatcher, 413
 Enqueue, 414
 Instance, 415
 IsMainThread, 414
 RunOnMainThread, 414
Mergelt.Core.Helpers.MonoEventsListener, 421
 Instance, 422
 SubscribeOnApplicationQuit, 421
 SubscribeOnUpdate, 421
 UnsubscribeFromApplicationQuit, 422
 UnsubscribeFromUpdate, 422
Mergelt.Core.Helpers.SlicedFilledImage, 466
 alphaHitTestMinimumThreshold, 468
 fillAmount, 468
 fillCenter, 469
 FillDirection, 467
 fillDirection, 469
 hasBorder, 469
 mainTexture, 469
 material, 470
 OnDisable, 467
 OnEnable, 467
 OnPopulateMesh, 468
 overrideSprite, 470
 pixelsPerUnit, 470
 pixelsPerUnitMultiplier, 470
 SlicedFilledImage, 467
 sprite, 471
 UpdateMaterial, 468
Mergelt.Core.Inventory, 34
Mergelt.Core.Inventory.IInventoryData, 345
 InventoryElements, 346
 InventorySize, 346
Mergelt.Core.Messages, 34
Mergelt.Core.Messages.IMessage, 348
Mergelt.Core.Messages.IMessageBus, 348

AddListener< T >, 349
DefineMessage< T >, 349
Fire< T >, 349
RemoveListener< T >, 349
Mergelt.Core.Messages.MessageBus, 417
AddListener< T >, 418
DefineMessage< T >, 418
Fire< T >, 418, 419
Initialize, 419
RemoveListener< T >, 420
Mergelt.Core.MVP, 34
Mergelt.Core.MVP.IModel, 350
Mergelt.Core.MVP.IPresenter, 373
Mergelt.Core.MVP.IView, 382
DestroyEvent, 383
GameObject, 382
InitializeEvent, 383
Mergelt.Core.MVP.Model, 420
Mergelt.Core.MVP.Presenter< TView, TModel >, 445
Dispose, 446
Initialize, 446
Model, 447
OnDispose, 446
OnInitialize, 446
View, 447
Mergelt.Core.MVP.View, 493
DestroyEvent, 495
GameObject, 494
Initialize, 494
InitializeEvent, 495
OnDestroy, 494
Mergelt.Core.Saves, 34
GameSaveType, 35
Mergelt.Core.Saves.ISavable, 376
Mergelt.Core.Saves.SavableAttribute, 456
Extension, 457
Name, 457
SavableAttribute, 456
Mergelt.Core.Schemes, 35
Mergelt.Core.Schemes.Data, 35
Mergelt.Core.Schemes.Data.EdgeData, 116
InputNodeID, 116
InputPortName, 116
OutputNodeID, 116
OutputPortName, 117
Mergelt.Core.Schemes.Data.ElementNodeData, 152
Copy< T >, 153
ElementConfig, 153
Id, 153
Position, 154
Mergelt.Core.Schemes.Data.EvolutionNodeData, 185
Copy< T >, 186
Description, 186
Discovered, 186
Id, 186
Name, 187
Position, 187
Mergelt.Core.Schemes.Data.IElementNodeData, 318
Copy< T >, 318
ElementConfig, 319
Id, 319
Position, 319
Mergelt.Core.Schemes.Data.IEvolutionNodeData, 323
Copy< T >, 324
Description, 324
Discovered, 324
Id, 324
Name, 324
Position, 325
Mergelt.Core.Schemes.SchemeData, 461
Edges, 462
ElementsNodesData, 462
EvolutionsData, 462
EvolutionsNodesData, 462
Mergelt.Core.Schemes.SchemeObject, 462
Evolution, 463
SchemeData, 463
Mergelt.Core.Services, 35
Mergelt.Core.Services.IConfigsService, 310
GameConfig, 311
GetEffectConfig< T >, 310
HintsConfig, 311
InventoryConfig, 311
LevelConfig, 311
Mergelt.Core.Services.ICurrencyService, 312
Mergelt.Core.Services.IElementService, 319
Mergelt.Core.Services.IElementsStockService, 321
Mergelt.Core.Services.IEnergyService, 321
Mergelt.Core.Services.IEvolutionsService, 326
Mergelt.Core.Services.IGameFieldActionsService, 335
Mergelt.Core.Services.IGameFieldService, 335
Mergelt.Core.Services.IGameLoadService, 336
Load< T >, 336
Mergelt.Core.Services.IGameSaveService, 337
Mergelt.Core.Services.IGameService, 337
Mergelt.Core.Services.IGeneratorsService, 341
Mergelt.Core.Services.IInfoPanelService, 343
Mergelt.Core.Services.IInventoryService, 347
Mergelt.Core.Services.IResourcesLoaderService, 375
GetObject< T >, 375
Mergelt.Core.Services.IUserProgressService, 381
Mergelt.Core.Services.IUserService, 381
Mergelt.Core.User, 36
Mergelt.Core.User.IUserData, 378
Energy, 379
EnergyRestoringStartTime, 379
Experience, 379
HardCurrency, 379
Level, 379
Name, 380
SoftCurrency, 380
Splitters, 380
Mergelt.Core.Utils, 36
Mergelt.Core.Utils.Bindable< T >, 72
Hold, 72
Release, 72

Mergelt.Core.Utils.ResponsiveGridLayout, 455
Mergelt.Core.WindowSystem, 36
Mergelt.Core.WindowSystem.BlackoutComponent, 74
 SetActive, 75
 SetLayer, 75
Mergelt.Core.WindowSystem.Commands, 36
Mergelt.Core.WindowSystem.Commands.WindowCloseCommand, 500
 OnDispose, 502
 Update, 502
 WindowCloseCommand, 501
Mergelt.Core.WindowSystem.Commands.WindowHideCommand, 507
 OnDispose, 509
 Update, 509
 WindowHideCommand, 508
Mergelt.Core.WindowSystem.Commands.WindowOpenCommand, 511
 OnDispose, 513
 Update, 513
 WindowOpenCommand, 512
Mergelt.Core.WindowSystem.Data, 37
Mergelt.Core.WindowSystem.Data.IWindowArgs, 386
Mergelt.Core.WindowSystem.Data.IWindowOpenParameters, 387
 ClosePrevious, 388
 NeedBlackout, 388
 Presenter, 388
Mergelt.Core.WindowSystem.Data.WindowArgs, 495
Mergelt.Core.WindowSystem.Data.WindowOpenParameters, 513
 ClosePrevious, 514
 NeedBlackout, 514
 Presenter, 514
Mergelt.Core.WindowSystem.Factory, 37
Mergelt.Core.WindowSystem.Factory.IWindowFactory, 386
 CreateWindow< TPresenter >, 387
Mergelt.Core.WindowSystem.Factory.WindowFactory, 506
 CreateWindow< TPresenter >, 506
 GetBlackout, 506
 GetRoot, 507
Mergelt.Core.WindowSystem.IWindowSystem, 390
 OpenWindow< TPresenter >, 390
Mergelt.Core.WindowSystem.Messages, 37
Mergelt.Core.WindowSystem.Messages.CloseWindowMessage, 83
 Presenter, 83
Mergelt.Core.WindowSystem.Windows, 37
 WindowState, 37
Mergelt.Core.WindowSystem.Windows.IWindow, 383
 CloseEndEvent, 384
 CloseStartEvent, 384
 InitiateCloseEvent, 384
 ShowEndEvent, 384
 ShowStartEvent, 384
Mergelt.Core.WindowSystem.Windows.IWindowPresenter, 388
 Layer, 389
 State, 389
Mergelt.Core.WindowSystem.Windows.WindowBase, 495
 AnimationController, 499
Mergelt.Core.WindowSystem.Windows.WindowCanvas, 499
 ClickOutsideArea, 499
 Close, 497
 CloseButton, 499
 CloseEndEvent, 500
 CloseStartEvent, 500
 InitiateCloseEvent, 500
 OnCloseFinished, 497
 OnCloseStarted, 497
 OnDestroy, 497
 OnOpenFinished, 498
 OnOpenStarted, 498
 SetLayer, 498
 Show, 498
 ShowEndEvent, 500
 ShowStartEvent, 500
Mergelt.Core.WindowSystem.Windows.WindowCreateInfo, 502
 Equals, 503
 GetHashCode, 503
 LayerName, 503
 PrefabPath, 503
 Type, 504
Mergelt.Core.WindowSystem.Windows.WindowModel, 511
 Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >, 514
 Close, 516
 DestroyWindow, 516
 Hide, 516
 Initialize, 516
 Layer, 520
 MessageBus, 519
 OnDestroyWindow, 517
 OnDispose, 517
 OnInitialize, 517
 OnWindowClosed, 517
 OnWindowShown, 518
 OnWindowStartClosing, 518
 OnWindowStartShowing, 518
 SetWindowActive, 518
 SetWindowLayer, 519
 Show, 519
 State, 520
 WindowSystem, 519
Mergelt.Core.WindowSystem.WindowsSystem, 520
 Blackout, 522
 CloseWindow, 521

Dispose, 521
Initialize, 521
OpenWindow< TPresenter >, 521
Root, 522
Mergelt.Editor, 38
Mergelt.Editor.Configs, 38
Mergelt.Editor.Configs.CurrencyComponent, 107
 AmountBinding, 108
 CurrencyComponent, 108
 Title, 108
 TypeBinding, 108
Mergelt.Editor.Configs.CurrencyComponent.UxmlFactory, 493
Mergelt.Editor.Configs.CurrencyComponentsTraits, 108
 Init, 109
Mergelt.Editor.Configs.ItemsDrawers, 38
Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer, 67
 Arrow, 69
 BaseSettingsDrawer, 68
 Config, 69
 Dispose, 68
 Expand, 68
 Hide, 69
 Parent, 69
 Root, 70
 RotateArrow, 69
 Show, 69
Mergelt.Editor.Configs.ItemsDrawers.CommonSettingsDrawer, Data, 184
 88
 CommonSettingsDrawer, 89
 Dispose, 89
 Draw, 89
 GetUxml, 90
 OnExpand, 90
Mergelt.Editor.Configs.ItemsDrawers.ElementConfigInspector, SaveData, 184
 128
 CreateInspectorGUI, 129
 Root, 129
 Target, 129
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsBaseDrawer, 285
 Clear, 287
 Dispose, 287
 Draw, 287
 GeneratorSettingsBaseDrawer, 286
 GetUxml, 287
 OnClear, 288
 OnExpand, 288
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsEvolutionDrawer, 288
 288
 Draw, 290
 GeneratorSettingsEvolutionDrawer, 290
 OnClear, 290
Mergelt.Editor.Configs.ItemsDrawers.GeneratorSettingsInspectorDrawer, 291
 291
 Draw, 292
 GeneratorSettingsInspectorDrawer, 292
Mergelt.Editor.Core, 38
Mergelt.Editor.Core.LevelEditor, 38
Mergelt.Editor.Core.LevelEditor.Commands, 38
Mergelt.Editor.Core.LevelEditor.Commands.LimitedStack< T >, 410
 Any, 410
 Clear, 410
 Count, 411
 LimitedStack, 410
 Peek, 410
 Pop, 411
 Push, 411
Mergelt.Editor.EvolutionsEditor, 39
Mergelt.Editor.EvolutionsEditor.BaseNode, 64
 Dispose, 65
 GetPort, 65
 Id, 66
 SaveData, 65
Mergelt.Editor.EvolutionsEditor.ElementNode, 149
 BuildContextualMenu, 150
 Config, 152
 Data, 152
 Dispose, 151
 ElementNode, 150
 Id, 152
 SaveData, 151
 Validate, 151
Mergelt.Editor.EvolutionsEditor.EvolutionNode, 182
Mergelt.Editor.EvolutionsEditor.EvolutionsEditorWindow, 188
 Initialize, 189
 Initialized, 191
 OnDestroy, 189
 OnDisable, 189
 OnEnable, 190
 OnFocus, 190
 Reload, 190
 ShowWindow, 190
Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView, 194
 BakeEvolutionsData, 195
 BuildContextualMenu, 195
 canCopySelection, 199
 canDuplicateSelection, 199
 ClearNodes, 195
 EvolutionsGraphView, 194
 OnEvent, 196
 OnIncompatiblePorts, 196
 HandleEvent, 196
 Restore, 197
 Save, 198

Validate, 199
Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView.UxmlFactory, 493
Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView.Traits, 200
Init, 200
Mergelt.Editor.EvolutionsEditor.IElementNode, 317
Config, 317
Data, 317
Mergelt.Editor.EvolutionsEditor.IEvolutionNode, 322
Data, 323
Description, 323
Discovered, 323
Name, 323
Mergelt.Editor.EvolutionsEditor.ISaveNode, 376
Id, 376
Mergelt.Editor.EvolutionsEditor.SchemeObjectEditor, 463
OnInspectorGUI, 464
Mergelt.Editor.EvolutionsEditor.SerializedElementNodes, 465
EvolutionNodesData, 465
Mergelt.Editor.GeneratedItemsCreator, 271
Clear, 272
Dispose, 272
Generate, 273
GeneratedItemsCreator, 272
Mergelt.Editor.GeneratedItemUI, 273
Dispose, 275
GeneratedItemUI, 274
Index, 275
RemoveGeneratedItemEvent, 275
Mergelt.Editor.Helpers, 39
Mergelt.Editor.Helpers.ConfigsPostprocessor, 97
Mergelt.Editor.Helpers.SavesHelper, 461
ClearSaves, 461
Mergelt.Editor.Helpers.SlicedFilledImageEditor, 471
OnInspectorGUI, 472
Mergelt.Editor.LevelEditor, 39
Mergelt.Editor.LevelEditor.Commands, 39
Mergelt.Editor.LevelEditor.Commands.ActionCommandBase, 51
ActionCommandBase, 51
Execute, 52
Undo, 52
Window, 52
Mergelt.Editor.LevelEditor.Commands.ActionCommandManager, 52
ExecuteCommand, 53
Redo, 53
Undo, 53
Mergelt.Editor.LevelEditor.Commands.ApplyCellCommand, 58
ApplyCellCommand, 58
Execute, 59
Undo, 59
Mergelt.Editor.LevelEditor.Commands.ChangeCellCommand, 75
ChangeCellCommand, 76
Execute, 76
Undo, 76
Mergelt.Editor.LevelEditor.Commands.ClearCellCommand, 80
ClearCellCommand, 81
Execute, 81
Undo, 81
Mergelt.Editor.LevelEditor.Commands.IActionCommand, 303
Mergelt.Editor.LevelEditor.Commands.IActionCommandManager, 303
Mergelt.Editor.LevelEditor.EvolutionSelection, 191
ClearSelection, 191
ElementData, 193
EvoltItem, 193
Evolution, 193
Init, 191
LockCell, 192
SelectCell, 192
SelectElement, 192
SelectEvolution, 192
UpdateCell, 193
Mergelt.Editor.LevelConfigEditor, 401
OnInspectorGUI, 402
Mergelt.Editor.LevelEditor.LevelEditorWindow, 402
ApplyCell, 403
ClearCell, 403
CreateGUI, 403
Show, 405
ShowWindow, 405
UndoApplyCell, 405
UndoClearCell, 405
Mergelt.Game, 40
Mergelt.Game.Commands, 40
Mergelt.Game.Commands.CheckEvolutionsProgressCommand, 77
Execute, 78
Mergelt.Game.Commands.CheckOpeningGeneratorsCommand, 79
Execute, 80
Mergelt.Game.Commands.EffectCommand, 117
EffectCommand, 118
Execute, 118
Mergelt.Game.Commands.LoadConfigsCommand, 412
ExecuteAsync, 413
Mergelt.Game.Commands.PrepareEnergyCommand, 438
Execute, 439
Mergelt.Game.Commands.PrepareFieldCommand, 439
Execute, 440
Mergelt.Game.Commands.PrepareInventoryCommand, 441
Execute, 442
Mergelt.Game.Commands.PrepareStockCommand, 442
Mergelt.Game.Commands.PrepareUserCommand, 443
Execute, 443
Mergelt.Game.Commands.PrepareUserCommand, 444

Execute, 445
Mergelt.Game.Configs, 40
Mergelt.Game.Configs.ConfigsContainer, 95
EffectConfigs, 95
GameConfig, 95
HintsConfig, 95
IconsConfig, 95
InventoryConfig, 96
LevelConfig, 96
LevelUpConfig, 96
Prefabs, 96
Mergelt.Game.Configs.ElementPrefab, 156
Prefab, 156
Type, 156
Mergelt.Game.Configs.Services, 40
Mergelt.Game.Configs.Services.ConfigsService, 97
GameConfig, 100
GetConfig, 98
GetConfigs, 98
GetCurrencyIcon, 98
GetEffectConfig< T >, 98
GetElementPrefab, 99
GetEvolutionByElement, 99
GetEvolutionData, 99
GetEvolutionIdByElement, 99
GetLevelUpData, 100
HintsConfig, 100
InventoryConfig, 101
LevelConfig, 101
Load, 100
Mergelt.Game.Configs.Services.ConfigsServiceModel, 101
EffectConfigs, 104
ElementConfigs, 104
GameConfig, 104
HintsConfig, 104
IconsConfig, 104
InventoryConfig, 104
LevelConfig, 104
LevelUpConfig, 105
Prefabs, 105
StoreEffectsConfigs, 102
StoreElementsPrefabs, 102
StoreGameConfig, 102
StoreHintsConfig, 102
StoreIconsConfig, 103
StoreInventoryConfig, 103
StoreLevelConfig, 103
StoreLevelUpConfig, 103
Mergelt.Game.Converters, 40
Mergelt.Game.Converters.ConfigProcessor, 91
BuildLevel, 92
Convert, 92
ConvertToFieldElement, 92
ConvertToFieldElementData, 93, 94
Mergelt.Game.Converters.IConfigProcessor, 309
Mergelt.Game.Effects, 41
Mergelt.Game.Effects.Configs, 41
Mergelt.Game.Effects.Configs.JumpEffectConfig, 394
CurveY, 395
ScaleCurve, 395
Mergelt.Game.Effects.Configs.MoveEffectConfig, 425
PositionXCurve, 426
PositionYCurve, 426
ScaleCurve, 426
Mergelt.Game.Effects.Configs.MoveToUiEffectConfig, 434
PositionXCurve, 435
PositionYCurve, 435
ScaleCurve, 435
Mergelt.Game.Effects.Controllers, 41
Mergelt.Game.Effects.Controllers.BaseEffect, 59
Animator, 61
ConfigsService, 61
Finish, 60
FinishedCallback, 61
OnFinished, 60
OnStarted, 60
Setup, 60
Start, 61
Started, 62
Target, 62
Update, 61
Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter, 62
FinishCallbackHandler, 63
GetPresenter, 63
Presenter, 64
Setup, 64
Mergelt.Game.Effects.Controllers.IEffect, 313
Mergelt.Game.Effects.Controllers.IEffectWithPresenter, 315
Mergelt.Game.Effects.Controllers.JumpEffect, 392
JumpEffect, 393
Setup, 393
Update, 393
Mergelt.Game.Effects.Controllers.JumpEffectWithPresenter, 395
OnStarted, 397
Setup, 397
Start, 397
Update, 397
Mergelt.Game.Effects.Controllers.MoveEffect, 423
MoveEffect, 424
Setup, 424
Update, 424
Mergelt.Game.Effects.Controllers.MoveEffectWithPresenter, 428
OnFinished, 430
OnStarted, 430
Setup, 430
Start, 430
Update, 430
Mergelt.Game.Effects.Controllers.MoveToUiEffect, 432
Setup, 433
Update, 433

Mergelt.Game.Effects.Controllers.MoveToUiEffectWithPreset
 435
 OnStarted, 437
 Setup, 437
 Start, 437
 Update, 437

Mergelt.Game.Effects.EffectsFactory, 119
 CreateEffect< T >, 120

Mergelt.Game.Effects.EffectsManager, 121
 AddEffect, 121
 Update, 121

Mergelt.Game.Effects.IEffectsFactory, 314
 CreateEffect< T >, 314

Mergelt.Game.Effects.IEffectsManager, 315

Mergelt.Game.Effects.Parameters, 41

Mergelt.Game.Effects.Parameters.IEffectParameters,
 313
 OverriddenName, 314

Mergelt.Game.Effects.Parameters.MoveEffectParameters,
 426
 FromPosition, 427
 MoveEffectParameters, 427
 OverriddenName, 427
 TargetInHud, 427
 ToPosition, 428

Mergelt.Game.ElementsStock, 41

Mergelt.Game.ElementsStock.ElementsStockComponent,
 168
 Initialize, 168
 OnDestroy, 168
 PopElement, 169
 SetupElement, 169

Mergelt.Game.ElementsStock.ElementsStockData, 170
 Elements, 170

Mergelt.Game.Enums, 42
 HudPanelType, 42

Mergelt.Game.Evolutions, 42

Mergelt.Game.Evolutions.EvolutionsProgressData, 200
 EvolutionsProgress, 201

Mergelt.Game.Factories, 42

Mergelt.Game.Factories.ElementInfo, 42
 ElementInfoType, 43

Mergelt.Game.Factories.ElementInfo.ElementInfoFactory,
 137
 CreateElementWindowItem, 137
 CreateUnknownElementWindowItem, 137

Mergelt.Game.Factories.ElementInfo.IElementInfoFactory,
 316

Mergelt.Game.Factories.Field, 43

Mergelt.Game.Factories.Field.FieldFactory, 242
 CreateField, 242
 CreateFieldCell, 242
 FieldCellObjectEvenPath, 243
 FieldCellObjectOddPath, 243
 FieldGameObjectPath, 243

Mergelt.Game.Factories.Field.IFieldFactory, 334

Mergelt.Game.Factories.FieldElement, 43
 Mergelt.Game.Factories.FieldElement.Factory, 218
 CreateFieldElement, 219

Mergelt.Game.Factories.FieldElement.FieldElementVisualFactory,
 240
 CreateFieldElement, 241

Mergelt.Game.Factories.FieldElement.IFieldElementFactory,
 328

Mergelt.Game.Factories.FieldElement.IFieldElementVisualFactory,
 334

Mergelt.Game.Factories.Icons, 43
 Mergelt.Game.Factories.Icons.IconFactory, 307
 CreateIcon, 308

Mergelt.Game.Factories.Icons.IconFactory, 342

Mergelt.Game.Factories.Inventory, 43
 Mergelt.Game.Factories.Inventory.InventoryFactory,
 346
 Mergelt.Game.Factories.Inventory.InventoryData, 355
 InventoryElements, 356
 InventorySize, 356

Mergelt.Game.Factories.Inventory.InventoryFactory,
 356
 CreateInventoryPanelItem, 357
 CreateInventoryWidgetItem, 357
 CreateWindowEmptyCell, 357
 CreateWindowPaidCell, 358

Mergelt.Game.Field, 44

Mergelt.Game.Field.Actions, 44

Mergelt.Game.Field.Actions.FieldActionProcessorBase,
 205
 FieldLogicModel, 206
 MessageBus, 206
 ProcessClick, 206
 ProcessEndDrag, 206
 SaveService, 206

Mergelt.Game.Field.Actions.FieldConsumableProcessor,
 210
 ProcessClick, 211

Mergelt.Game.Field.Actions.FieldGenerationProcessor,
 244
 ProcessClick, 245

Mergelt.Game.Field.Actions.FieldMergeProcessor, 249
 ProcessClick, 250
 ProcessEndDrag, 250

Mergelt.Game.Field.Actions.IFieldActionProcessor, 326

Mergelt.Game.Field.Elements, 44
 Mergelt.Game.Field.Elements.Animations, 44
 Mergelt.Game.Field.Elements.Animations.FieldElementAnimationController
 215
 GetFieldElementState, 216
 Initialize, 216
 SetState< T >, 216

Mergelt.Game.Field.Elements.Animations.FieldElementStateMachineBehavior
 236
 OnStateExit, 237

Mergelt.Game.Field.Elements.FieldElementModel, 226
 ClicksCount, 226
 IsBusy, 226

IsLocked, 226
 Point, 227
 Selected, 227
Mergelt.Game.Field.Elements.FieldElementPresenter,
 227
Mergelt.Game.Field.Elements.FieldElementPresenterBase<
 TView, TModel >, 229
 Activate, 230
 Canvas, 234
 EndDrag, 230
 FieldElement, 234
 GetModel, 230
 GetView, 231
 IsAvailable, 234
 IsBusy, 235
 IsLocked, 235
 OnInitialize, 231
 RectTransform, 235
 Release, 231
 Remove, 231
 ResetAnimationState, 232
 ResetPosition, 232
 Select, 232
 SetBusy, 232
 SetLock, 232
 SetParent, 233
 SetPoint, 233
 SetState, 233
 StartDrag, 233
 State, 235
 Transform, 236
 Update, 234
Mergelt.Game.Field.Elements.FieldElementView, 237
 AnimationController, 239
 Canvas, 239
 GraphicRaycaster, 239
 Lock, 238
 RectTransform, 239
 ResetState, 238
 ResetStateEvent, 239
Mergelt.Game.Field.Elements.Generator, 44
Mergelt.Game.Field.Elements.Generator.FieldElementGenerator
 219
Mergelt.Game.Field.Elements.Generator.FieldElementGenerator
 220
 FieldElementGeneratorPresenter, 221
 OnDispose, 222
 Release, 222
 Update, 222
Mergelt.Game.Field.Elements.Generator.FieldElementGenerator
 222
 HideTimer, 224
 SetTimer, 224
Mergelt.Game.Field.Elements.Generator.GeneratorTimer
 293
 StartTimer, 293
Mergelt.Game.Field.FieldCellComponent, 207
 AttachObject, 207
 DetachObject, 207
 FieldElementPresenter, 210
 Initialize, 208
 ObjectContainer, 210
 OnBeginDrag, 208
 OnDrag, 208
 OnEndDrag, 208
 OnPointerDown, 209
 OnPointerUp, 209
 Point, 210
 RectTransform, 210
 ToggleSelector, 209
Mergelt.Game.Field.FieldData, 212
 FieldHeight, 213
 FieldWidth, 213
 SavedElementsData, 213
Mergelt.Game.Field.FieldLogic, 246
 Dispose, 246
 Initialize, 246
Mergelt.Game.Field.FieldLogicModel, 247
 AllGenerators, 248
 CellComponents, 248
 CellSize, 248
 CellSpacing, 248
 FieldElements, 248
 FieldHeight, 248
 FieldRect, 248
 FieldWidth, 249
 GetPosition, 247
 OpeningGenerator, 249
 PositionOffset, 249
Mergelt.Game.Field.FieldModel, 251
Mergelt.Game.Field.FieldPresenter, 252
 FieldPresenter, 253
 Initialize, 253
 OnDispose, 254
Mergelt.Game.Field.FieldView, 254
 CellsContainer, 255
 CellSpacing, 255
Mergelt.Game.Field.IFieldLogic, 335
Mergelt.Game.GameRoot, 261
Mergelt.Game.GameRoot
 262
 Run, 262
Mergelt.Game.Helpers, 45
Mergelt.Game.Helpers.HudTargetComponent, 302
Mergelt.Game.HUD, 45
Mergelt.Game.HUD.EnergyComponent, 176
 OnApplyModel, 177
 Start, 177
Mergelt.Game.HUD.HudComponent, 301
Mergelt.Game.HUD.HudPanelBase, 301
 Start, 302
 Type, 302
Mergelt.Game.HUD.IHudPanel, 342

Type, 342
Mergelt.Game.HUD.IUserListener, 380
Mergelt.Game.HUD.ProgressComponent, 449
Initialize, 450
OnApplyModel, 450
OnDestroy, 450
SetMaxProgress, 450
Start, 451
Type, 451
UpdateProgress, 451
Mergelt.Game.HUD.SoftCurrencyComponent, 472
OnApplyModel, 473
Type, 473
Mergelt.Game.HUD.UserListenerComponent, 486
ApplyModel, 486
UserServiceModel, 486
Mergelt.Game.Managers, 45
Mergelt.Game.Managers.HintsManager, 299
Dispose, 300
Initialize, 300
Update, 300
Mergelt.Game.Managers.IHintsManager, 341
Mergelt.Game.Messages, 45
Mergelt.Game.Messages.ActivateHintsMessage, 53
Active, 54
Mergelt.Game.Messages.CheckGeneratorMessage, 78
GeneratorPoint, 79
Mergelt.Game.Messages.ClickElementMessage, 82
Cell, 82
Mergelt.Game.Messages.CreateElementMessage, 106
FromPosition, 106
NewElement, 106
ToPoint, 107
Mergelt.Game.Messages.ElementActionMessage, 122
ActionType, 122
Element, 122
Mergelt.Game.Messages.EndDragElementMessage, 175
FromPoint, 176
Position, 176
ToGameObject, 176
Mergelt.Game.Messages.EnergyRestoredMessage, 178
Mergelt.Game.Messages.FieldFullMessage, 244
Mergelt.Game.Messages.GeneratorBusyMessage, 275
Point, 276
Mergelt.Game.Messages.GeneratorOpenedMessage, 276
GeneratorPoint, 276
Mergelt.Game.Messages.GeneratorOpenStartMessage, 278
GeneratorPoint, 279
Mergelt.Game.Messages.GeneratorRestoredMessage, 281
GeneratorPoint, 282
Mergelt.Game.Messages.LevelUpdatedMessage, 408
NextLevelExp, 408
Mergelt.Game.Messages.LevelUpMessage, 409
Mergelt.Game.Messages.LoadedGameMessage, 413
Mergelt.Game.Messages.MergeElementsMessage, 417
From, 417
NewElement, 417
Mergelt.Game.Messages.MoveElementMessage, 431
FieldElementView, 431
FromPoint, 431
ToPoint, 432
Mergelt.Game.Messages.PrepareGameMessage, 441
Mergelt.Game.Messages.RemoveElementMessage, 452
RemoveAtPoint, 452
Mergelt.Game.Messages.RemoveFromInventoryMessage, 452
FieldElement, 453
Mergelt.Game.Messages.ResetHintsMessage, 453
Mergelt.Game.Messages.ResetPositionMessage, 453
From, 454
Mergelt.Game.Messages.ResetSelectionMessage, 454
Mergelt.Game.Messages.SelectedElementMessage, 464
Point, 465
Mergelt.Game.Messages.SkipTimeMessage, 465
Seconds, 466
Mergelt.Game.Messages.SplitElementMessage, 474
SplitElement1, 474
SplitElement2, 474
Mergelt.Game.Messages.StartChargingMessage, 474
RemainingTime, 475
Mergelt.Game.Messages.StartGameMessage, 475
Mergelt.Game.Messages.SwapElementsMessage, 475
From, 476
To, 476
Mergelt.Game.Messages.UnlockElementMessage, 477
Element, 477
Mergelt.Game.Messages.UpdateStockMessage, 477
Mergelt.Game.Services, 46
Mergelt.Game.Services.CurrencyService, 109
Sell, 110
TryPay, 110
Mergelt.Game.Services.ElementService, 159
TrySell, 160
TrySplit, 160
TryUnlock, 160
Mergelt.Game.Services.ElementsStockService, 170
Add, 171
CreateStock, 171
GetCurrent, 171
GetData, 171
GetNext, 172
Remove, 172
SetupStock, 172
Mergelt.Game.Services.ElementsStockServiceModel, 173
Elements, 173
Mergelt.Game.Services.EnergyService, 179
Dispose, 179
Initialize, 179

Mergelt.Game.Services.EvolutionsService, 201
 CreateEvolutionsProgress, 202
 GetData, 202
 GetEvolutionChain, 202
 GetEvolutionProgress, 202
 GetGeneratedBy, 203
 GetGenerates, 203
 SetupEvolutionsProgress, 203
 UpdateProgress, 204
Mergelt.Game.Services.EvolutionsServiceModel, 204
 EvolutionsProgress, 205
Mergelt.Game.Services.GameFieldActionsService, 257
 Dispose, 258
 Initialize, 258
Mergelt.Game.Services.GameFieldService, 258
 CreateNewElement, 259
 Dispose, 259
 GetFreeCell, 259
 Initialize, 259
Mergelt.Game.Services.GameService, 268
 _messageBus, 269
 Dispose, 268
 Initialize, 268
Mergelt.Game.Services.GameServiceModel, 269
 LoadedLevel, 269
 MainCamera, 269
 MainCanvas, 269
Mergelt.Game.Services.GeneratorService, 282
 Dispose, 283
 Initialize, 283
 TryOpen, 283
 TrySkipCharging, 283
 TrySkipOpening, 284
 Update, 284
Mergelt.Game.Services.InfoPanelService, 351
 Dispose, 352
 Initialize, 352
Mergelt.Game.Services.InventoryService, 363
 Add, 364
 CreateInventory, 364
 GetData, 364
 OpenWindow, 365
 Remove, 365
 SetupInventory, 365
Mergelt.Game.Services.InventoryServiceModel, 366
 InventoryElements, 367
 InventorySize, 367
 IsFull, 366
Mergelt.Game.Services.ResourcesLoaderService, 454
 GetObject< TResource >, 455
Mergelt.Game.Services.Saves, 46
Mergelt.Game.Services.Saves.GameLoadService, 261
 Load< T >, 261
Mergelt.Game.Services.Saves.GameSaveEveryIntervalService, 263
 Dispose, 263
 Initialize, 263
 OnApplicationQuit, 264
 Save, 264
 Update, 264
Mergelt.Game.Services.Saves.GameSaveOnEveryActionService, 265
 Dispose, 265
 Initialize, 265
 OnApplicationQuit, 265
 Save, 265
Mergelt.Game.Services.Saves.GameSaveOnQuitService, 266
 Dispose, 267
 Initialize, 267
 OnApplicationQuit, 267
 Save, 267
Mergelt.Game.Services.Saves.Strategies, 47
Mergelt.Game.Services.Saves.Strategies.BaseSerializeStrategy, 66
 Load< T >, 67
 Save< T >, 67
 SaveDir, 67
Mergelt.Game.Services.Saves.Strategies.BinaryFormatterStrategy, 70
 Load< T >, 71
 Save< T >, 71
Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy, 377
 Load< T >, 377
 Save< T >, 377
 SaveDir, 378
Mergelt.Game.Services.Saves.Strategies.JsonSerializeStrategy, 390
 Load< T >, 391
 Save< T >, 391
Mergelt.Game.Services.UserProgressService, 487
 CanLevelUp, 488
 Dispose, 488
 GetCurrentLevelMaxExp, 488
 GetLevelUpPrizes, 488
 Initialize, 489
Mergelt.Game.Services.UserService, 489
 CreateUser, 490
 SetupUser, 490
Mergelt.Game.Services.UserServiceModel, 490
 Energy, 491
 EnergyRestoringStartTime, 491
 Experience, 491
 GetData, 491
 HardCurrency, 492
 Level, 492
 Name, 492
 Set, 491
 SoftCurrency, 492
 Splitters, 492
Mergelt.Game.TopPanel, 47
Mergelt.Game.TopPanel.TopPanelComponent, 476
Mergelt.Game.UI, 47
Mergelt.Game.UI.InfoPanel, 47
 ElementActionType, 47

Mergelt.Game.UI.InfoPanel.ElementCommonInfo, 122
 ActionType, 124
 OnTrySetup, 123

Mergelt.Game.UI.InfoPanel.ElementInfo, 133
 ActionButtonClick, 134
 ActionType, 136
 Clear, 134
 ConfigsService, 135
 Destroy, 134
 MessageBus, 135
 OnActionButtonClick, 134
 OnClear, 134
 OnStart, 134
 SelectedElement, 135
 SendActionMessage, 134
 TrySetup, 135

Mergelt.Game.UI.InfoPanel.ElementInfoPanelComponent, 143

Mergelt.Game.UI.InfoPanel.ElementOpenInfo, 154
 ActionType, 156
 Destroy, 155
 OnStart, 155
 OnTrySetup, 155

Mergelt.Game.UI.InfoPanel.ElementSellInfo, 157
 ActionType, 159
 Destroy, 158
 OnStart, 158
 OnTrySetup, 158

Mergelt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo, 161
 ActionType, 163
 Destroy, 162
 OnStart, 162
 OnTrySetup, 162

Mergelt.Game.UI.InfoPanel.ElementSpeedUpOpeningInfo, 163
 ActionType, 165
 OnClear, 164
 OnStart, 164
 OnTrySetup, 165

Mergelt.Game.UI.InfoPanel.ElementSplitInfo, 166
 ActionType, 168
 Destroy, 167
 OnStart, 167
 OnTrySetup, 167

Mergelt.Game.UI.InfoPanel.ElementUnlockInfo, 173
 ActionType, 175
 Destroy, 174
 OnStart, 174
 OnTrySetup, 175

Mergelt.Game.UI.InventoryPanel, 48

Mergelt.Game.UI.InventoryPanel.InventoryPanelComponent, 359

Mergelt.Game.UI.InventoryPanel.InventoryPanelItemComponent, 359
 ClickEvent, 360
 RectTransform, 360

Mergelt.Game.UI.InventoryPanel.InventoryPanelItemPair, 360
 Component, 361
 Element, 361
 InventoryPanelItemPair, 361

Mergelt.Game.User, 48

Mergelt.Game.User(userData, 478
 Energy, 478
 EnergyRestoringStartTime, 478
 Experience, 478
 HardCurrency, 479
 Level, 479
 Name, 479
 SoftCurrency, 479
 Splitters, 479

Mergelt.Game.Windows, 48

Mergelt.Game.Windows.ElementInfo, 48

Mergelt.Game.Windows.ElementInfo.ElementInfoArgs, 136
 ElementConfig, 136

Mergelt.Game.Windows.ElementInfo.ElementInfoItemComponent, 138
 Awake, 139
 ClickEvent, 140
 OnDestroy, 139
 RectTransform, 140
 SetIcon, 139
 SetUnknown, 139
 Setup, 139

Mergelt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent, 140
 ItemsGrid, 142
 PanelTitle, 142
 RectTransform, 142
 SetItem, 141
 SetSpacing, 141
 SetTitle, 141

Mergelt.Game.Windows.ElementInfo.ElementInfoModel, 142

Mergelt.Game.Windows.ElementInfo.ElementInfoPresenter, 143
 OnDestroyWindow, 144
 OnInitialize, 144

Mergelt.Game.Windows.ElementInfo.ElementInfoWindow, 145
 CreatedFromPanel, 147
 EvolutionPanel, 147
 GeneratesPanel, 147
 ScrollContent, 147
 SpacingBetweenPanels, 148
 SpacingInsidePanels, 148
 WindowTitleText, 148

Mergelt.Game.Windows.ElementInfo.ElementInfoWindowContent, 148
 Mergelt.Game.Windows.Inventory, 48

Mergelt.Game.Windows.Inventory.InventoryModel, 358

Mergelt.Game.Windows.Inventory.InventoryPresenter, 361

OnDestroyWindow, 363
OnInitialize, 363
Mergelt.Game.Windows.Inventory.InventoryWindow,
 367
 ItemsGrid, 369
Mergelt.Game.Windows.Inventory.InventoryWindowItemComponent,
 369
 Awake, 370
 InfoClickEvent, 370
 ItemClickEvent, 370
 OnDestroy, 370
 SetIcon, 370
Mergelt.Game.Windows.Inventory.InventoryWindowItemPair,
 371
 Component, 371
 Element, 371
 InventoryWindowItemPair, 371
Mergelt.Game.Windows.Inventory.InventoryWindowPaidCellComponent,
 372
 Awake, 372
 BuyCellEvent, 373
 OnDestroy, 372
 Setup, 372
Mergelt.Game.Windows.UserProgress, 48
Mergelt.Game.Windows.UserProgress.UserInfoModel,
 480
Mergelt.Game.Windows.UserProgress.UserInfoPresenter,
 480
 OnDestroyWindow, 482
 OnInitialize, 482
Mergelt.Game.Windows.UserProgress.UserInfoWindow,
 482
 Close, 484
 EnableLevelUpButton, 484
 LevelUpEvent, 485
 OnOpenFinished, 485
 PrizesContainer, 485
 UpdateProgress, 485
Mergelt.Game.Windows.UserProgress.UserProgressComponent,
 487
 UpdateProgress, 487
Mergelt.SimpleDI, 49
Mergelt.SimpleDI.DiContainer, 112
 Clear, 112
 Drop< TInterface >, 112
 Get< TInterface >, 112
 Register< TInterface, TType >, 113
 Register< TType >, 113
 RegisterInterfacesFor< TType >, 113
Mergelt.SimpleDI.DiHandler, 114
 AddDependency< TType >, 114
 AsSingleton, 114
 AsSingleton< TType >, 115
Mergelt.SimpleDI.DiRoot, 115
 Run, 116
Mergelt.SimpleDI.IntroduceAttribute, 354
 IntroduceAttribute, 354
Mergelt.SimpleDI.ReservedInterfaces, 49
Mergelt.SimpleDI.ReservedInterfaces.IInitializable, 345
Mergelt.SimpleDI.ReservedInterfaces.IUpdatable, 378
MessageBus
 Mergelt.Core.WindowSystem.Windows.WindowPresenter<
 TWindow, TModel >, 519
Mergelt.Game.Actions.FieldActionProcessorBase,
 206
 Mergelt.Game.UI.InfoPanel.ElementInfo, 135
MinDrop
 Mergelt.Core.Configs.Elements.ElementGeneratorSettings,
 132
 MinDropChargeTime
 Mergelt.Core.FieldElements.GeneratorParameters,
 280
 Mergelt.Core.FieldElements.IGeneratorParameters,
 340
 MinDropFullChargeTime
 Mergelt.Core.FieldElements.GeneratorParameters,
 281
 Mergelt.Core.FieldElements.IGeneratorParameters,
 340
 Model
 Mergelt.Core.MVP.Presenter< TView, TModel >,
 447
 MoveEffect
 Mergelt.Game.Effects.Controllers.MoveEffect, 424
 Mergelt.Game.Effects.Parameters.MoveEffectParameters,
 427
 Name
 Mergelt.Core.Configs.Data.EvolutionData, 182
 Mergelt.Core.Configs.Effects.EffectConfig, 119
 Mergelt.Core.Configs.Elements.ElementCommonSettings,
 125
 Mergelt.Core.Configs.Windows.LayerParameters,
 399
 Mergelt.Core.FieldElements.IInfoParameters, 344
 Mergelt.Core.FieldElements.InfoParameters, 353
 Mergelt.Core.Saves.SavableAttribute, 457
 Mergelt.Core.Schemes.Data.EvolutionNodeData,
 187
 Mergelt.Core.Schemes.Data.IEvolutionNodeData,
 324
 Mergelt.Core.User.IUserData, 380
 Mergelt.Editor.EvolutionsEditor.EvolutionNode,
 185
 Mergelt.Editor.EvolutionsEditor.IEvolutionNode,
 323
 Mergelt.Game.Services.UserServiceModel, 492
 Mergelt.Game.User.UserData, 479
 NeedBlackout
 Mergelt.Core.WindowSystem.Data.IWindowOpenParameters,
 388
 Mergelt.Core.WindowSystem.Data.WindowOpenParameters,
 514
 NeedOpen
 Mergelt.Core.Configs.Elements.ElementGeneratorSettings,
 132

TView, TModel >, 231
Mergelt.Game.Windows.ElementInfo.ElementInfoPresenter
 OnStateUpdate
 144
 Mergelt.Core.Animations.AnimatorStateMachineBehaviour,
 57
 OnTrySetup
Mergelt.Game.Windows.UserProgress.UserInfoPresenter, Mergelt.Game.UI.InfoPanel.ElementCommonInfo,
 123
 Mergelt.Game.UI.InfoPanel.ElementOpenInfo, 155
 Mergelt.Game.UI.InfoPanel.ElementSellInfo, 158
 Mergelt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo,
 162
 Mergelt.Game.UI.InfoPanel.ElementSpeedUpOpeningInfo,
 165
 Mergelt.Game.UI.InfoPanel.ElementSplitInfo, 167
 Mergelt.Game.UI.InfoPanel.ElementUnlockInfo,
 175
OnInspectorGUI
 Mergelt.Editor.EvolutionsEditor.SchemeObjectEditor,
 464
 Mergelt.Editor.Helpers.SlicedFilledImageEditor,
 472
 Mergelt.Editor.LevelEditor.LevelConfigEditor, 402
OnInstall
 Mergelt.Game.GameRoot, 262
OnOpenEnd
 Mergelt.Core.Animations.WindowDefaultAnimationController
 505
 Mergelt.Core.Animations.WindowMecanimAnimationController, TWindow, TModel >, 517
 510
 OnWindowClosed
 Mergelt.Core.WindowSystem.Windows.WindowPresenter<
 TWindow, TModel >, 518
 OnWindowShown
 Mergelt.Core.WindowSystem.Windows.WindowPresenter<
 TWindow, TModel >, 518
 OnWindowStartClosing
 Mergelt.Core.WindowSystem.Windows.WindowPresenter<
 TWindow, TModel >, 518
 OnWindowStartShowing
 Mergelt.Core.WindowSystem.Windows.WindowPresenter<
 TWindow, TModel >, 518
OnOpenStarted
 Mergelt.Core.WindowSystem.Windows.WindowBase,
 498
 Mergelt.Core.WindowSystem.Windows.WindowBase,
 498
OnPointerDown
 Mergelt.Game.Field.FieldCellComponent, 209
OnPointerUp
 Mergelt.Game.Field.FieldCellComponent, 209
OnPopulateMesh
 Mergelt.Core.Helpers.SlicedFilledImage, 468
OnStart
 Mergelt.Game.UI.InfoPanel.ElementInfo, 134
 Mergelt.Game.UI.InfoPanel.ElementOpenInfo, 155
 Mergelt.Game.UI.InfoPanel.ElementSellInfo, 158
 Mergelt.Game.UI.InfoPanel.ElementSpeedUpChargingInfo
 162
 Mergelt.Game.Services.InventoryService, 365
 OpenWindow< TPresenter >
 390
 Mergelt.Core.Animations.WindowDefaultAnimationController,
 505
 Mergelt.Core.Animations.WindowMecanimAnimationController,
 510
 Mergelt.Core.WindowSystem.IWindowSystem,
 390
 Mergelt.Core.WindowSystem.WindowsSystem,
 521
Operation
 Mergelt.Core.Configs.Elements.ConsumableSettings,
 105
 operator!=
 Mergelt.Core.FieldElements.GridPoint, 295
 operator==
 Mergelt.Core.FieldElements.GridPoint, 296
 On
 Mergelt.Core.Configs.Windows.LayerParameters,
 399
OnStarted
 Mergelt.Game.Effects.Controllers.BaseEffect, 60
 Mergelt.Game.Effects.Controllers.JumpEffectWithPresenter
 397
 Mergelt.Game.Effects.Controllers.MoveEffectWithPresenter
 430
 Mergelt.Game.Effects.Controllers.MoveToUiEffectWithPresenter
 437
OnStateEnter
 Mergelt.Core.Animations.AnimatorStateMachineBehaviour
 57
OnStateExit
 Mergelt.Game.Field.Elements.Animations.FieldElementState
 57
 Mergelt.Core.Behaviours.Data.EdgeData, 117
 57
 Mergelt.Core.Schemes.Data.EdgeData, 116
 57
 OutputPortName
 57
 Mergelt.Core.Behaviours.Data.EdgeData, 117
 57

OverriddenName
 Mergelt.Game.Effects.Parameters.IEffectParameters, 314
 Mergelt.Game.Effects.Parameters.MoveEffectParameters, Mergelt.Game.Effects.Configs.MoveToUiEffectConfig, 427

overrideSprite
 Mergelt.Core.Helpers.SlicedFilledImage, 470

PaidCells
 Mergelt.Core.Configs.Inventory.InventoryConfig, 355

PanelTitle
 Mergelt.Game.Windows.ElementInfo.ElementInfoItem, 142

Parent
 Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer, 69

Peek
 Mergelt.Editor.Core.LevelEditor.Commands.LimitedStack< T >, 410

pixelsPerUnit
 Mergelt.Core.Helpers.SlicedFilledImage, 470

pixelsPerUnitMultiplier
 Mergelt.Core.Helpers.SlicedFilledImage, 470

Point
 Mergelt.Core.FieldElements.IFieldElementModel, 330
 Mergelt.Game.Field.Elements.FieldElementModel, 227
 Mergelt.Game.Field.FieldCellComponent, 210
 Mergelt.Game.Messages.GeneratorBusyMessage, 276
 Mergelt.Game.Messages.SelectedElementMessage, 465

Pop
 Mergelt.Editor.Core.LevelEditor.Commands.LimitedStack< T >, 411

PopElement
 Mergelt.Game.ElementsStock.ElementsStockComponent, 169

Position
 Mergelt.Core.Configs.Data.LevelElementData, 407
 Mergelt.Core.Schemes.Data.ElementNodeData, 154
 Mergelt.Core.Schemes.Data.EvolutionNodeData, 187
 Mergelt.Core.Schemes.Data.IElementNodeData, 319
 Mergelt.Core.Schemes.Data.IEvolutionNodeData, 325
 Mergelt.Game.Messages.EndDragElementMessage, 176

PositionOffset
 Mergelt.Game.Field.FieldLogicModel, 249

PositionXCurve
 Mergelt.Game.Effects.Configs.MoveEffectConfig, 426
 Mergelt.Game.Effects.Configs.MoveToUiEffectConfig, 435

PositionYCurve
 Mergelt.Game.Effects.Configs.MoveEffectConfig, 426
 Mergelt.Game.Effects.Configs.MoveToUiEffectConfig, 435

Possibility
 Mergelt.Core.Configs.Elements.GeneratableElement, 270
 Mergelt.Core.FieldElements.GeneratableFieldElement, 271

Prefab
 Mergelt.Game.Configs.ElementPrefab, 156
 PrefabPanel
 Mergelt.Core.WindowSystem.Windows.WindowCreateInfo, 503

Prefabs
 Mergelt.Game.Configs.ConfigsContainer, 96
 Mergelt.Game.Configs.Services.ConfigsServiceModel, 105

Presenter
 Mergelt.Core.WindowSystem.Data.IWindowOpenParameters, 388
 Mergelt.Core.WindowSystem.Data.WindowOpenParameters, 514
 Mergelt.Core.WindowSystem.Messages.CloseWindowMessage, 83
 Mergelt.Game.Effects.Controllers.BaseEffectWithPresenter, 64

PrizesContainer
 Mergelt.Game.Windows.UserProgress.UserInfoWindow, 485

ProcessClick
 Mergelt.Game.Field.Actions.FieldActionProcessorBase, 206
 Mergelt.Game.Field.Actions.FieldConsumableProcessor, 211
 Mergelt.Game.Field.Actions.FieldGenerationProcessor, 245

Mergelt.Game.Field.Actions.FieldMergeProcessor, 250

ProcessEndDrag
 Mergelt.Game.Field.Actions.FieldActionProcessorBase, 206
 Mergelt.Game.Field.Actions.FieldMergeProcessor, 250

ProducedByParameters
 Mergelt.Core.FieldElements.FieldElement, 215
 Mergelt.Core.FieldElements.IFieldElement, 328

ProduceParameters
 Mergelt.Core.FieldElements.FieldElement, 215
 Mergelt.Core.FieldElements.IFieldElement, 328

Progress
 Mergelt.Core.Evolutions.EvolutionProgressData, 188

Push
 Mergelt.Editor.Core.LevelEditor.Commands.LimitedStack< T >, 411

RectTransform

Mergelt.Core.Configs.Elements.FieldElementIconComponent
 225
Mergelt.Core.FieldElements.IFieldElementPresenter,
 331
Mergelt.Core.FieldElements.IFieldElementView,
 333
Mergelt.Game.Field.Elements.FieldElementPresenterBase<
 TView, TModel >, 235
Mergelt.Game.Field.Elements.FieldElementView,
 239
Mergelt.Game.Field.FieldCellComponent, 210
Mergelt.Game.UI.InventoryPanel.InventoryPanelItemCompo
 238
Mergelt.Game.Windows.ElementInfo.ElementInfoItem
 140
Mergelt.Game.Windows.ElementInfo.ElementInfoItemsPanelCo
 142
Redo
 Mergelt.Editor.LevelEditor.Commands.ActionCommandManager
 53
Register< TInterface, TType >
 Mergelt.SimpleDI.DiContainer, 113
Register< TType >
 Mergelt.SimpleDI.DiContainer, 113
RegisterInterfacesFor< TType >
 Mergelt.SimpleDI.DiContainer, 113
Release
 Mergelt.Core.Utils.Bindable< T >, 72
 Mergelt.Game.Field.Elements.FieldElementPresenterBase<
 TView, TModel >, 231
 Mergelt.Game.Field.Elements.Generator.FieldElementGen
 222
Reload
 Mergelt.Editor.EvolutionsEditor.EvolutionsEditorWindow
 190
RemainChargeTime
 Mergelt.Core.FieldElements.GeneratorParameters,
 281
 Mergelt.Core.FieldElements.IGeneratorParameters,
 340
RemainingTime
 Mergelt.Core.FieldElements.GeneratorOpenParameters,
 277
 Mergelt.Core.FieldElements.IGeneratorOpenParameters,
 338
 Mergelt.Game.Messages.StartChargingMessage,
 475
Remove
 Mergelt.Game.Field.Elements.FieldElementPresenterBase<
 TView, TModel >, 231
 Mergelt.Game.Services.ElementsStockService,
 172
 Mergelt.Game.Services.InventoryService, 365
RemoveAtPoint
 Mergelt.Game.Messages.RemoveElementMessage,
 452
RemoveGeneratedItemEvent
 Mergelt.Editor.GeneratedItemUI, 275
ElementListener< T >
 Mergelt.Core.Messages.IMessageBus, 349
 Mergelt.Core.Messages.MessageBus, 420
ResetAnimationState
 Mergelt.Core.Animations.AnimationListener, 56
 Mergelt.Game.Field.Elements.FieldElementPresenterBase<
 TView, TModel >, 232
ResetPosition
 Mergelt.Game.Field.Elements.FieldElementPresenterBase<
 TView, TModel >, 232
ResetState
 Mergelt.Game.Field.Elements.FieldElementView,
 238
 Mergelt.Game.Field.Elements.FieldElementView,
 239
 Mergelt.Game.Field.Elements.FieldElementView,
 238
 Mergelt.Game.Field.Elements.FieldElementView,
 239
 Mergelt.Game.Field.Elements.FieldElementView,
 238
 Mergelt.Game.Field.Elements.FieldElementView,
 239
Root
 Mergelt.Core.WindowSystem.WindowsSystem,
 522
 Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer,
 70
 Mergelt.Editor.Configs.ItemsDrawers.ElementConfigInspector,
 129
RotateArrow
 Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer,
 69
Run
 Mergelt.Core.Commands.CommandManager, 86
 Mergelt.Game.GameRoot, 262
 Mergelt.SimpleDI.DiRoot, 116
RunAsync
 Mergelt.Core.Commands.CommandManager, 86
RunOnMainThread
 Mergelt.Core.Helpers.MainThreadDispatcher, 414
RunSimultaneously
 Mergelt.Core.Commands.CommandManager, 87
SaveableAttribute
 Mergelt.Core.Saves.SavableAttribute, 456
Save
 Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView,
 198
 Mergelt.Game.Services.Saves.GameSaveEveryIntervalService,
 264
 Mergelt.Game.Services.Saves.GameSaveOnEveryActionService,
 265
 Mergelt.Game.Services.Saves.GameSaveOnQuitService,
 267
Save< T >
 Mergelt.Game.Services.Saves.Strategies.BaseSerializeStrategy,
 67
 Mergelt.Game.Services.Saves.Strategies.BinaryFormatterStrategy,
 71
 Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy,
 377

Mergelt.Game.Services.Saves.Strategies.JsonSerializerStrategy, 391

SaveData
Mergelt.Editor.EvolutionsEditor.BaseNode, 65
Mergelt.Editor.EvolutionsEditor.ElementNode, 151
Mergelt.Editor.EvolutionsEditor.EvolutionNode, 184

SavedElementsData
Mergelt.Game.Field.FieldData, 213

SaveDir
Mergelt.Game.Services.Saves.Strategies.BaseSerializeStrategy, 67

Mergelt.Game.Services.Saves.Strategies.ISerializeStrategy, 378

SaveInterval
Mergelt.Core.Configs.GameConfig, 257

SaveService
Mergelt.Game.Field.Actions.FieldActionProcessorBase, 206

ScaleCurve
Mergelt.Game.Effects.Configs.JumpEffectConfig, 395
Mergelt.Game.Effects.Configs.MoveEffectConfig, 426
Mergelt.Game.Effects.Configs.MoveToUiEffectConfig, 435

SchemeData
Mergelt.Core.Schemes.SchemeObject, 463

ScrollContent
Mergelt.Game.Windows.ElementInfo.ElementInfoWindow, 147

Seconds
Mergelt.Game.Messages.SkipTimeMessage, 466

Select
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >, 232

SelectCell
Mergelt.Editor.LevelEditor.EvolutionSelection, 192

Selected
Mergelt.Core.FieldElements.IFieldElementModel, 330
Mergelt.Game.Field.Elements.FieldElementModel, 227

SelectedElement
Mergelt.Game.UI.InfoPanel.ElementInfo, 135

SelectElement
Mergelt.Editor.LevelEditor.EvolutionSelection, 192

SelectEvolution
Mergelt.Editor.LevelEditor.EvolutionSelection, 192

Sell
Mergelt.Game.Services.CurrencyService, 110

SellCostSettings
Mergelt.Core.Configs.Elements.ElementCommonSettings, 125

SendActionMessage
Mergelt.Game.UI.InfoPanel.ElementInfo, 134

Set
Mergelt.Game.Services.UserServiceModel, 491

Mergelt.Core.WindowSystem.BlackoutComponent, 75

SetBlocked
Mergelt.Core.Configs.Elements.FieldElementIconComponent, 225

SetBusy
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >, 232

SetIcon
Mergelt.Game.Windows.ElementInfo.ElementInfoItemComponent, 139

Mergelt.Game.Windows.Inventory.InventoryWindowItemComponent, 370

SetItem
Mergelt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent, 141

SetLayer
Mergelt.Core.WindowSystem.BlackoutComponent, 75

Mergelt.Core.WindowSystem.Windows.WindowBase, 498

SetLock
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >, 232

SetMaxProgress
Mergelt.Game.HUD.ProgressBarComponent, 450

SetParent
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >, 233

SetPoint
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >, 233

SetSpacing
Mergelt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent, 141

SetState
Mergelt.Core.Animations.AnimationControllerBase, 55
Mergelt.Core.Animations.MecanimController, 416
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >, 233

SetState< T >
Mergelt.Core.Animations.AnimationControllerBase, 55
Mergelt.Core.Animations.IAnimationController, 304
Mergelt.Core.Animations.MecanimController, 416
Mergelt.Game.Field.Elements.Animations.FieldElementAnimationController, 216

SetTimer
Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorView

SetTitle
Mergelt.Game.Windows.ElementInfo.ElementInfoItemsPanelComponent, 141

SetUnknown
Mergelt.Game.Windows.ElementInfo.ElementInfoItemComponent,

Setup	Mergelt.Editor.EvolutionsEditor.EvolutionsEditorWindow, 139
Mergelt.Game.Effects.Controllers.BaseEffect, 60	Mergelt.Editor.LevelEditor.LevelEditorWindow, 405
Mergelt.Game.Effects.Controllers.BaseEffectWithPressSkipChargeCostSettings, 64	Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 132
Mergelt.Game.Effects.Controllers.JumpEffect, 393	Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 397
Mergelt.Game.Effects.Controllers.JumpEffectWithPressSkipOpenCostSettings, 397	Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 424
Mergelt.Game.Effects.Controllers.MoveEffect, 430	Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 430
Mergelt.Game.Effects.Controllers.MoveEffectWithPresSideFilledImage, 430	Mergelt.Core.Helpers.SlicedFilledImage, 467
Mergelt.Game.Effects.Controllers.MoveToUiEffect, 433	SoftCurrency
Mergelt.Game.Effects.Controllers.MoveToUiEffectWithPresSideFilledImage, 437	Mergelt.Core.User.IUserData, 380
Mergelt.Game.Windows.ElementInfo.ElementInfoItem, 139	Mergelt.Game.Services.UserServiceModel, 492
Mergelt.Game.Windows.Inventory.InventoryWindowPaidCellComponent, 372	Mergelt.Game.User.UserData, 479
SetupElement	SpacingBetweenPanels
Mergelt.Game.ElementsStock.ElementsStockComponent, 169	Mergelt.Game.Windows.ElementInfo.ElementInfoWindow, 148
SetupEvolutionsProgress	SplitCostSettings
Mergelt.Game.Services.EvolutionsService, 203	Mergelt.Core.Configs.Elements.ElementCommonSettings, 125
SetupInventory	SplitElement1
Mergelt.Game.Services.InventoryService, 365	Mergelt.Game.Messages.SplitElementMessage, 474
SetupStock	SplitElement2
Mergelt.Game.Services.ElementsStockService, 172	Mergelt.Game.Messages.SplitElementMessage, 474
SetupUser	Splitters
Mergelt.Game.Services.UserService, 490	Mergelt.Core.User.IUserData, 380
SetValueSilently	Mergelt.Game.Services.UserServiceModel, 492
Mergelt.Core.Utils.Bindable< T >, 72	Mergelt.Game.User.UserData, 479
SetWindowActive	Sprite
Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >, 518	Mergelt.Core.Helpers.SlicedFilledImage, 471
SetWindowLayer	Stack
Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >, 519	Mergelt.Core.Configs.Elements.ElementGeneratorSettings, 132
Show	Start
Mergelt.Core.WindowSystem.Windows.WindowBase, 498	Mergelt.Game.Effects.Controllers.BaseEffect, 61
Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >, 519	Mergelt.Game.Effects.Controllers.JumpEffectWithPresenter, 397
Mergelt.Editor.Configs.ItemsDrawers.BaseSettingsDrawer, 69	Mergelt.Game.Effects.Controllers.MoveEffectWithPresenter, 430
Mergelt.Editor.LevelEditor.LevelEditorWindow, 405	Mergelt.Game.Effects.Controllers.MoveToUiEffectWithPresenter, 437
ShowEndEvent	Mergelt.Game.HUD.EnergyComponent, 177
Mergelt.Core.WindowSystem.Windows.IWindow, 384	Mergelt.Game.HUD.HudPanelBase, 302
Mergelt.Core.WindowSystem.Windows.WindowBase, 500	Mergelt.Game.HUD.ProgressComponent, 451
ShowStartEvent	StartChargingTime
Mergelt.Core.WindowSystem.Windows.IWindow, 384	Mergelt.Core.FieldElements.GeneratorParameters, 281
Mergelt.Core.WindowSystem.Windows.WindowBase, 500	Mergelt.Core.FieldElements.IGeneratorParameters, 340
ShowWindow	StartDrag

Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >, 233

Started
Mergelt.Game.Effects.Controllers.BaseEffect, 62

StartOpeningTime
Mergelt.Core.FieldElements.GeneratorOpenParameters, 278

Mergelt.Core.FieldElements.IGeneratorOpenParameters, 338

Mergelt.Core.FieldElements.SavedGeneratorOpenParameters, 458

StartTimer
Mergelt.Game.Field.Elements.GeneratorTimer, 293

State
Mergelt.Core.FieldElements.IFieldElementPresenter, 332

Mergelt.Core.WindowSystem.Windows.IWindowPresenter, 389

Mergelt.Core.WindowSystem.Windows.WindowPresenter, 520

StoreEffectsConfigs
Mergelt.Game.Configs.Services.ConfigsServiceModel, 102

StoreElementsPrefabs
Mergelt.Game.Configs.Services.ConfigsServiceModel, 102

StoreGameConfig
Mergelt.Game.Configs.Services.ConfigsServiceModel, 102

StoreHintsConfig
Mergelt.Game.Configs.Services.ConfigsServiceModel, 102

StoreIconsConfig
Mergelt.Game.Configs.Services.ConfigsServiceModel, 103

StoreInventoryConfig
Mergelt.Game.Configs.Services.ConfigsServiceModel, 103

StoreLevelConfig
Mergelt.Game.Configs.Services.ConfigsServiceModel, 103

StoreLevelUpConfig
Mergelt.Game.Configs.Services.ConfigsServiceModel, 103

Subscribe
Mergelt.Core.Utils.Bindable< T >, 73

SubscribeOnApplicationQuit
Mergelt.Core.Helpers.MonoEventsListener, 421

SubscribeOnUpdate
Mergelt.Core.Helpers.MonoEventsListener, 421

Target
Mergelt.Core.Configs.Elements.ConsumableSettings, 105

Mergelt.Editor.Configs.ItemsDrawers.ElementConfigInspector, 129

Mergelt.Game.Effects.Controllers.BaseEffect, 62

TargetInHud
Mergelt.Game.Parameters.MoveEffectParameters, 427

Title
To

Mergelt.Game.Messages.SwapElementsMessage, 476

Mergelt.Game.Messages.EndDragElementMessage, 176

ToPoint
Mergelt.Game.Field.FieldCellComponent, 209

ToString
Mergelt.Core.FieldElements.FieldElement, 214

Mergelt.Core.FieldElements.GridPoint, 296

Transform
Mergelt.Game.Field.Elements.FieldElementPresenterBase< TView, TModel >, 236

TryOpen
Mergelt.Game.Services.GeneratorService, 283

TryPay
Mergelt.Game.Services.CurrencyService, 110

TrySell
Mergelt.Game.Services.ElementService, 160

TrySetup
Mergelt.Game.UI.InfoPanel.ElementInfo, 135

TrySkipCharging
Mergelt.Game.Services.GeneratorService, 283

TrySkipOpening
Mergelt.Game.Services.ElementService, 160

TrySplit
Mergelt.Game.Services.ElementService, 160

TryUnlock
Mergelt.Game.Services.ElementService, 160

Type
Mergelt.Core.Configs.Elements.ElementConfig, 128

Mergelt.Core.FieldElements.IInfoParameters, 344

Mergelt.Core.FieldElements.InfoParameters, 353

Mergelt.Core.WindowSystem.Windows.WindowCreateInfo, 504

Mergelt.Game.Configs.ElementPrefab, 156

Mergelt.Game.HUD.EnergyComponent, 178

Mergelt.Game.HUD.HardCurrencyComponent, 298

Mergelt.Game.HUD.HudPanelBase, 302

Mergelt.Game.HUD.IHudPanel	342	UpdateMaterial	Mergelt.Core.Helpers.SlicedFilledImage , 468
Mergelt.Game.HUD.ProgressComponent	451	UpdateProgress	Mergelt.Game.HUD.ProgressComponent , 451
Mergelt.Game.HUD.SoftCurrencyComponent	473	Mergelt.Game.Services.EvolutionsService	204
TypeBinding		Mergelt.Game.Windows.UserProgress.UserInfoWindow	
Mergelt.Editor.Configs.CurrencyComponent	108	485	
Undo		Mergelt.Game.Windows.UserProgress.UserProgressComponent	
Mergelt.Editor.LevelEditor.Commands.ActionCommandBase	52	487	
Mergelt.Editor.LevelEditor.Commands.ActionCommandManager	53	UserServiceModel	
Mergelt.Editor.LevelEditor.Commands.ApplyCellCommand	59	Mergelt.Game.HUD.UserListenerComponent	486
Mergelt.Editor.LevelEditor.Commands.ChangeCellCommand	76	Validate	
Mergelt.Editor.LevelEditor.Commands.ClearCellCommand	81	Mergelt.Editor.EvolutionsEditor.ElementNode	151
UndoApplyCell		Mergelt.Editor.EvolutionsEditor.EvolutionsGraphView	
Mergelt.Editor.LevelEditor.LevelEditorWindow	405	199	
UndoClearCell		Value	
Mergelt.Editor.LevelEditor.LevelEditorWindow	405	Mergelt.Core.Configs.Elements.ConsumableSettings	
UnlockCostSettings		106	
Mergelt.Core.Configs.Elements.ElementCommonSettings	126	Mergelt.Core.Utils.Bindable< T >	74
Unsubscribe		View	
Mergelt.Core.Utils.Bindable< T >	73	Mergelt.Core.MVP.Presenter< TView, TModel >	
UnsubscribeFromApplicationQuit		447	
Mergelt.Core.Helpers.MonoEventsListener	422	Window	
UnsubscribeFromUpdate		Mergelt.Editor.LevelEditor.Commands.ActionCommandBase	
Mergelt.Core.Helpers.MonoEventsListener	422	52	
Update		WindowCloseCommand	
Mergelt.Core.WindowSystem.Commands.WindowCloseCommand	502	Mergelt.Core.WindowSystem.Commands.WindowCloseCommand	
Mergelt.Core.WindowSystem.Commands.WindowHideCommand	509	501	
Mergelt.Core.WindowSystem.Commands.WindowOpenCommand	513	WindowHideCommand	
Mergelt.Core.WindowSystem.Commands.WindowOpenCommand	513	Mergelt.Core.WindowSystem.Commands.WindowHideCommand	
Mergelt.Game.Effects.Controllers.BaseEffect	61	508	
Mergelt.Game.Effects.Controllers.JumpEffect	393	WindowOpenCommand	
Mergelt.Game.Effects.Controllers.JumpEffectWithPresenter	397	Mergelt.Core.WindowSystem.Commands.WindowOpenCommand	
Mergelt.Game.Effects.Controllers.MoveEffect	424	512	
Mergelt.Game.Effects.Controllers.MoveEffectWithPresenter	430	WindowState	
Mergelt.Game.Effects.Controllers.MoveToUiEffect	433	Mergelt.Core.WindowSystem.Windows	37
Mergelt.Game.Effects.Controllers.MoveToUiEffectWithPresenter	437	WindowSystem	
Mergelt.Game.Effects.EffectsManager	121	Mergelt.Core.WindowSystem.Windows.WindowPresenter< TWindow, TModel >	519
Mergelt.Game.Field.Elements.FieldElementPresenter		X	
Mergelt.Game.Field.Elements.FieldElementPresenter		Y	
Mergelt.Game.Field.Elements.FieldElementPresenterBase		Mergelt.Core.FieldElements.GridPoint	296
TView, TModel >	234	Mergelt.Core.FieldElements.GridPoint	296
Mergelt.Game.Field.Elements.Generator.FieldElementGeneratorPresenter	222		
Mergelt.Game.Managers.HintsManager	300		
Mergelt.Game.Services.GeneratorService	284		
Mergelt.Game.Services.Saves.GameSaveEveryIntervalService	264		
UpdateCell			
Mergelt.Editor.LevelEditor.EvolutionSelection	193		