

Midnight Blue Game and Engine

Generated by Doxygen 1.8.12

Contents

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	7
3.1 Class List	7
4 Namespace Documentation	11
4.1 MB2D Namespace Reference	11
4.1.1 Enumeration Type Documentation	12
4.1.1.1 TileFlag	12
4.1.1.2 Token	12
4.2 MB2D.Collision Namespace Reference	12
4.3 MB2D.EntityComponent Namespace Reference	13
4.4 MB2D.Geometry Namespace Reference	14
4.5 MB2D.IO Namespace Reference	14
4.5.1 Enumeration Type Documentation	14
4.5.1.1 CommandType	14
4.6 MB2D.Scenes Namespace Reference	15
4.6.1 Enumeration Type Documentation	15
4.6.1.1 TransitionState	15
4.7 MB2D.Testing Namespace Reference	15
4.8 MB2D.Tiles Namespace Reference	16

4.9 MB2D.UI Namespace Reference	16
4.9.1 Enumeration Type Documentation	16
4.9.1.1 UIState	16
4.10 MidnightBlue Namespace Reference	17
4.10.1 Enumeration Type Documentation	19
4.10.1.1 Biome	19
4.10.1.2 HeightLevel	19
4.10.1.3 MoistureLevel	19
4.10.1.4 PlanetType	20
4.10.1.5 ShipState	20
4.10.1.6 TemperatureLevel	20
4.11 MidnightBlue.Testing Namespace Reference	20
5 Class Documentation	21
5.1 MB2D.UI.Button Class Reference	21
5.1.1 Detailed Description	24
5.1.2 Constructor & Destructor Documentation	24
5.1.2.1 Button() [1/2]	24
5.1.2.2 Button() [2/2]	24
5.1.3 Member Function Documentation	24
5.1.3.1 Update()	24
5.1.4 Property Documentation	25
5.1.4.1 PressedSound	25
5.1.5 Event Documentation	25
5.1.5.1 OnPress	25
5.2 MB2D.Collectable Class Reference	25
5.2.1 Detailed Description	26
5.2.2 Constructor & Destructor Documentation	26
5.2.2.1 Collectable()	26
5.2.3 Member Function Documentation	27
5.2.3.1 Add()	27

5.2.3.2	Consume()	27
5.2.3.3	Effect()	27
5.2.4	Property Documentation	28
5.2.4.1	Count	28
5.2.4.2	Name	28
5.2.4.3	Tag	28
5.3	MB2D.Collision.CollisionCell Class Reference	29
5.3.1	Detailed Description	29
5.3.2	Constructor & Destructor Documentation	29
5.3.2.1	CollisionCell()	29
5.3.3	Member Function Documentation	30
5.3.3.1	Add()	30
5.3.3.2	Clear()	30
5.3.3.3	Contains()	30
5.3.3.4	Remove()	30
5.3.4	Property Documentation	31
5.3.4.1	Items	31
5.4	MB2D.EntityComponent.CollisionComponent Class Reference	31
5.4.1	Detailed Description	32
5.4.2	Constructor & Destructor Documentation	32
5.4.2.1	CollisionComponent()	32
5.4.3	Property Documentation	33
5.4.3.1	Boxes	33
5.4.3.2	Collider	33
5.4.3.3	Event	33
5.5	MB2D.Collision.CollisionMap Class Reference	34
5.5.1	Detailed Description	35
5.5.2	Constructor & Destructor Documentation	35
5.5.2.1	CollisionMap()	35
5.5.3	Member Function Documentation	35

5.5.3.1	Clear()	35
5.5.3.2	GetCollisions()	36
5.5.3.3	IndexExists() [1/2]	36
5.5.3.4	IndexExists() [2/2]	36
5.5.3.5	IndexOf()	37
5.5.3.6	Insert()	37
5.5.3.7	UpdatePosition()	38
5.5.4	Property Documentation	38
5.5.4.1	Grid	38
5.5.4.2	Max	39
5.5.4.3	Position	39
5.6	MB2D.Testing.CollisionRenderSystem Class Reference	39
5.6.1	Member Function Documentation	40
5.6.1.1	Process()	40
5.7	MB2D.EntityComponent.CollisionSystem Class Reference	41
5.7.1	Detailed Description	43
5.7.2	Constructor & Destructor Documentation	43
5.7.2.1	CollisionSystem()	43
5.7.3	Member Function Documentation	43
5.7.3.1	PreProcess()	43
5.7.3.2	Process()	43
5.7.3.3	ProcessingLoop()	44
5.7.3.4	ResetGrid()	44
5.7.3.5	SetTileMap()	45
5.7.4	Property Documentation	45
5.7.4.1	CurrentMap	45
5.7.4.2	NumberOfChecks	45
5.8	MB2D.IO.Command Class Reference	46
5.8.1	Detailed Description	47
5.8.2	Constructor & Destructor Documentation	47

5.8.2.1	Command()	47
5.8.3	Member Function Documentation	47
5.8.3.1	Execute()	47
5.8.3.2	OnKeyPress()	47
5.8.4	Property Documentation	48
5.8.4.1	Key	48
5.8.4.2	Type	48
5.9	MB2D.IO.ConsoleCommand Class Reference	48
5.9.1	Detailed Description	49
5.9.2	Constructor & Destructor Documentation	49
5.9.2.1	ConsoleCommand()	49
5.9.3	Member Function Documentation	50
5.9.3.1	OnKeyPress()	50
5.10	MB2D.EntityComponent.Depth Class Reference	50
5.10.1	Detailed Description	51
5.11	MB2D.EntityComponent.DepthSystem Class Reference	52
5.11.1	Detailed Description	53
5.11.2	Constructor & Destructor Documentation	54
5.11.2.1	DepthSystem()	54
5.11.3	Member Function Documentation	54
5.11.3.1	Process()	54
5.12	MidnightBlue.EnterStarSystem Class Reference	54
5.12.1	Detailed Description	56
5.12.2	Constructor & Destructor Documentation	56
5.12.2.1	EnterStarSystem()	56
5.12.3	Member Function Documentation	56
5.12.3.1	OnKeyPress()	56
5.13	MB2D.EntityComponent.Entity Class Reference	57
5.13.1	Detailed Description	58
5.13.2	Constructor & Destructor Documentation	58

5.13.2.1	Entity()	58
5.13.3	Member Function Documentation	58
5.13.3.1	Attach()	58
5.13.3.2	Attach< T >()	59
5.13.3.3	Detach< T >()	60
5.13.3.4	DetachAll()	60
5.13.3.5	GetComponent< T >()	61
5.13.3.6	HasComponent< T >()	61
5.13.4	Property Documentation	62
5.13.4.1	Active	62
5.13.4.2	ComponentList	62
5.13.4.3	ComponentTypeList	62
5.13.4.4	ID	62
5.13.4.5	Mask	63
5.13.4.6	Persistent	63
5.13.4.7	Tag	63
5.14	MB2D.Testing.EntityContainerTests Struct Reference	63
5.15	MB2D.EntityComponent.EntityMap Class Reference	64
5.15.1	Detailed Description	65
5.15.2	Constructor & Destructor Documentation	65
5.15.2.1	EntityMap() [1/2]	65
5.15.2.2	EntityMap() [2/2]	65
5.15.3	Member Function Documentation	66
5.15.3.1	AddComponent()	66
5.15.3.2	AddComponent< T >()	66
5.15.3.3	AddEntity()	66
5.15.3.4	AddSystem< T >()	67
5.15.3.5	Clear()	67
5.15.3.6	CreateEntity()	67
5.15.3.7	GetComponentID< T >()	68

5.15.3.8	GetSystem< T >()	68
5.15.3.9	UpdateEntityMask()	69
5.15.3.10	UpdateSystems()	69
5.15.4	Property Documentation	70
5.15.4.1	EntityCount	70
5.15.4.2	NextID	70
5.15.4.3	this[string key]	70
5.16	MB2D.EntityComponent.EntitySystem Class Reference	71
5.16.1	Detailed Description	72
5.16.2	Constructor & Destructor Documentation	73
5.16.2.1	EntitySystem()	73
5.16.3	Member Function Documentation	73
5.16.3.1	AssociateEntity()	73
5.16.3.2	Destroy()	73
5.16.3.3	PostAssociate()	73
5.16.3.4	PostProcess()	74
5.16.3.5	PreProcess()	74
5.16.3.6	Process()	74
5.16.3.7	ProcessingLoop()	74
5.16.3.8	RemoveAssociation()	74
5.16.3.9	Run()	75
5.16.4	Member Data Documentation	75
5.16.4.1	_associationType	75
5.16.4.2	_idEntityMap	75
5.16.5	Property Documentation	75
5.16.5.1	AssociatedEntities	75
5.16.5.2	Association	76
5.16.5.3	DestroyList	76
5.16.5.4	ID	76
5.16.5.5	ValidComponents	76

5.17	MidnightBlue.Fuel Class Reference	77
5.17.1	Detailed Description	78
5.17.2	Constructor & Destructor Documentation	78
5.17.2.1	Fuel()	78
5.17.3	Member Function Documentation	78
5.17.3.1	Effect()	78
5.18	MidnightBlue.GalaxyBuilder Class Reference	79
5.18.1	Constructor & Destructor Documentation	79
5.18.1.1	GalaxyBuilder()	79
5.18.2	Member Function Documentation	80
5.18.2.1	Generate()	80
5.18.3	Property Documentation	80
5.18.3.1	Bounds	80
5.18.3.2	Done	80
5.18.3.3	Size	81
5.18.3.4	StarSystems	81
5.19	MidnightBlue.GalaxyHud Class Reference	81
5.19.1	Detailed Description	82
5.19.2	Constructor & Destructor Documentation	82
5.19.2.1	GalaxyHud()	82
5.19.3	Member Function Documentation	83
5.19.3.1	Refresh()	83
5.20	MidnightBlue.GalaxyRenderSystem Class Reference	83
5.20.1	Detailed Description	86
5.20.2	Constructor & Destructor Documentation	86
5.20.2.1	GalaxyRenderSystem()	86
5.20.3	Member Function Documentation	86
5.20.3.1	PreProcess()	86
5.20.3.2	Process()	86
5.20.4	Property Documentation	87

5.20.4.1	InfoList	87
5.21	MidnightBlue.GalaxyScene Class Reference	87
5.21.1	Detailed Description	89
5.21.2	Constructor & Destructor Documentation	89
5.21.2.1	GalaxyScene()	89
5.21.3	Member Function Documentation	89
5.21.3.1	Draw()	89
5.21.3.2	Exit()	90
5.21.3.3	HandleInput()	90
5.21.3.4	Initialize()	90
5.21.3.5	Pause()	91
5.21.3.6	Resume()	91
5.21.3.7	Update()	92
5.22	MidnightBlue.Testing.GenTest Class Reference	92
5.23	MB2D.Geometry.Grid Class Reference	93
5.23.1	Detailed Description	93
5.23.2	Constructor & Destructor Documentation	93
5.23.2.1	Grid()	93
5.23.3	Property Documentation	94
5.23.3.1	CellSize	94
5.23.3.2	ColCount	94
5.23.3.3	ColSize	94
5.23.3.4	RowCount	94
5.23.3.5	RowSize	94
5.24	MB2D.EntityComponent.IComponent Interface Reference	95
5.24.1	Detailed Description	95
5.25	MidnightBlue.InitScene Class Reference	95
5.25.1	Detailed Description	98
5.25.2	Constructor & Destructor Documentation	98
5.25.2.1	InitScene()	98

5.25.3 Member Function Documentation	98
5.25.3.1 Draw()	98
5.25.3.2 Exit()	98
5.25.3.3 HandleInput()	99
5.25.3.4 Initialize()	99
5.25.3.5 Pause()	99
5.25.3.6 Resume()	100
5.25.3.7 Update()	100
5.26 MB2D.IO.InputMap Class Reference	100
5.26.1 Detailed Description	101
5.26.2 Constructor & Destructor Documentation	101
5.26.2.1 InputMap()	101
5.26.3 Member Function Documentation	101
5.26.3.1 Assign< T >()	101
5.26.3.2 Search< T >()	102
5.26.4 Property Documentation	102
5.26.4.1 Collection	102
5.26.4.2 this[Keys key]	102
5.27 MB2D.EntityComponent.InputSystem Class Reference	103
5.27.1 Detailed Description	104
5.27.2 Member Function Documentation	105
5.27.2.1 Process()	105
5.28 MB2D.EntityComponent.Inventory Class Reference	106
5.28.1 Detailed Description	107
5.28.2 Property Documentation	107
5.28.2.1 Items	107
5.29 MB2D.UI.Label Class Reference	108
5.29.1 Detailed Description	109
5.29.2 Constructor & Destructor Documentation	110
5.29.2.1 Label()	110

5.29.3 Member Function Documentation	110
5.29.3.1 Draw()	110
5.29.3.2 Update()	110
5.30 MidnightBlue.LandCommand Class Reference	110
5.30.1 Detailed Description	112
5.30.2 Constructor & Destructor Documentation	112
5.30.2.1 LandCommand()	112
5.30.3 Member Function Documentation	112
5.30.3.1 OnKeyPress()	112
5.31 MidnightBlue.LaunchCommand Class Reference	113
5.31.1 Detailed Description	114
5.31.2 Constructor & Destructor Documentation	114
5.31.2.1 LaunchCommand()	114
5.31.3 Member Function Documentation	114
5.31.3.1 OnKeyPress()	114
5.32 MB2D.UI.Layout Class Reference	115
5.32.1 Detailed Description	117
5.32.2 Constructor & Destructor Documentation	117
5.32.2.1 Layout()	117
5.32.3 Member Function Documentation	117
5.32.3.1 Add()	117
5.32.3.2 Draw()	118
5.32.3.3 Update()	118
5.33 MidnightBlue.LeaveStarSystem Class Reference	119
5.33.1 Constructor & Destructor Documentation	120
5.33.1.1 LeaveStarSystem()	120
5.33.2 Member Function Documentation	120
5.33.2.1 OnKeyPress()	120
5.34 MidnightBlue.Length Class Reference	121
5.34.1 Detailed Description	121

5.34.2 Constructor & Destructor Documentation	122
5.34.2.1 Length()	122
5.34.3 Member Data Documentation	122
5.34.3.1 AstronomicalUnit	122
5.34.4 Property Documentation	122
5.34.4.1 AU	122
5.34.4.2 Centimeters	122
5.34.4.3 Kilometers	122
5.34.4.4 Meters	123
5.34.4.5 RelativeKilometers	123
5.35 MB2D.Geometry.Line Class Reference	123
5.35.1 Detailed Description	124
5.35.2 Constructor & Destructor Documentation	124
5.35.2.1 Line() [1/2]	124
5.35.2.2 Line() [2/2]	124
5.35.3 Member Function Documentation	124
5.35.3.1 PointDistance() [1/2]	124
5.35.3.2 PointDistance() [2/2]	125
5.35.4 Property Documentation	125
5.35.4.1 End	125
5.35.4.2 Start	125
5.36 MB2D.UI.ListControl Class Reference	126
5.36.1 Detailed Description	128
5.36.2 Constructor & Destructor Documentation	128
5.36.2.1 ListControl() [1/2]	128
5.36.2.2 ListControl() [2/2]	128
5.36.3 Member Function Documentation	129
5.36.3.1 Draw()	129
5.36.3.2 Update()	129
5.36.4 Property Documentation	129

5.36.4.1	ControlSize	129
5.36.4.2	Count	129
5.36.4.3	Elements	130
5.36.4.4	ItemSpan	130
5.36.4.5	SeperatorColor	130
5.37	MidnightBlue.Testing.MapTest Class Reference	131
5.37.1	Member Function Documentation	133
5.37.1.1	Draw()	133
5.37.1.2	Exit()	133
5.37.1.3	HandleInput()	133
5.37.1.4	Initialize()	133
5.37.1.5	Pause()	134
5.37.1.6	Resume()	134
5.37.1.7	Update()	134
5.38	MB2D.MBConsole Class Reference	134
5.38.1	Detailed Description	135
5.38.2	Constructor & Destructor Documentation	136
5.38.2.1	MBConsole()	136
5.38.3	Member Function Documentation	136
5.38.3.1	AddFunc()	136
5.38.3.2	AddVar()	136
5.38.3.3	Debug() [1/4]	138
5.38.3.4	Debug() [2/4]	139
5.38.3.5	Debug() [3/4]	139
5.38.3.6	Debug() [4/4]	140
5.38.3.7	Draw()	140
5.38.3.8	InitWindow()	141
5.38.3.9	Toggle()	141
5.38.3.10	Update()	142
5.38.3.11	Write() [1/3]	142

5.38.3.12 Write() [2/3]	142
5.38.3.13 Write() [3/3]	143
5.38.4 Property Documentation	143
5.38.4.1 BGColor	143
5.38.4.2 Display	144
5.38.4.3 Funcs	144
5.38.4.4 TextColor	144
5.38.4.5 Vars	144
5.39 MB2D.MBConsoleASTNode Class Reference	145
5.39.1 Detailed Description	145
5.39.2 Member Function Documentation	145
5.39.2.1 Handle()	145
5.40 MB2D.MBConsoleLexer Class Reference	146
5.40.1 Detailed Description	147
5.40.2 Constructor & Destructor Documentation	147
5.40.2.1 MBConsoleLexer()	147
5.40.3 Member Function Documentation	147
5.40.3.1 Lex()	147
5.40.3.2 NextToken()	147
5.40.3.3 RawToken()	148
5.40.4 Property Documentation	148
5.40.4.1 CurrentPos	148
5.40.4.2 NextPos	148
5.40.4.3 NumTokens	149
5.41 MB2D.MBConsoleParser Class Reference	149
5.41.1 Detailed Description	149
5.41.2 Constructor & Destructor Documentation	149
5.41.2.1 MBConsoleParser()	149
5.41.3 Member Function Documentation	150
5.41.3.1 Parse()	150

5.42 MB2D.MBGame Class Reference	151
5.42.1 Detailed Description	154
5.42.2 Constructor & Destructor Documentation	154
5.42.2.1 MBGame()	154
5.42.3 Member Function Documentation	154
5.42.3.1 Draw()	154
5.42.3.2 Initialize()	155
5.42.3.3 LoadContent()	155
5.42.3.4 RegisterSystems()	156
5.42.3.5 Update()	156
5.42.4 Property Documentation	156
5.42.4.1 Camera	156
5.42.4.2 Console	157
5.42.4.3 DeltaTime	157
5.42.4.4 ForceQuit	157
5.42.4.5 FPS	157
5.42.4.6 Graphics	157
5.43 MidnightBlue.MenuCommand Class Reference	158
5.43.1 Member Function Documentation	159
5.43.1.1 OnKeyPress()	159
5.44 MidnightBlue.MenuScene Class Reference	160
5.44.1 Member Function Documentation	162
5.44.1.1 Draw()	162
5.44.1.2 Exit()	162
5.44.1.3 HandleInput()	162
5.44.1.4 Initialize()	163
5.44.1.5 Pause()	163
5.44.1.6 Resume()	163
5.44.1.7 Update()	163
5.45 MidnightBlue.MenuView Class Reference	164

5.46 MB2D.IO.MoveBackward Class Reference	165
5.46.1 Detailed Description	166
5.46.2 Constructor & Destructor Documentation	166
5.46.2.1 MoveBackward()	166
5.46.3 Member Function Documentation	167
5.46.3.1 OnKeyPress()	167
5.47 MB2D.IO.MoveDown Class Reference	167
5.47.1 Detailed Description	168
5.47.2 Constructor & Destructor Documentation	168
5.47.2.1 MoveDown()	168
5.47.3 Member Function Documentation	169
5.47.3.1 OnKeyPress()	169
5.48 MB2D.IO.MoveForward Class Reference	169
5.48.1 Detailed Description	170
5.48.2 Constructor & Destructor Documentation	170
5.48.2.1 MoveForward()	170
5.48.3 Member Function Documentation	171
5.48.3.1 OnKeyPress()	171
5.49 MB2D.IO.MoveLeft Class Reference	171
5.49.1 Detailed Description	172
5.49.2 Constructor & Destructor Documentation	172
5.49.2.1 MoveLeft()	172
5.49.3 Member Function Documentation	173
5.49.3.1 OnKeyPress()	173
5.50 MB2D.EntityComponent.Movement Class Reference	173
5.50.1 Detailed Description	174
5.50.2 Constructor & Destructor Documentation	175
5.50.2.1 Movement()	175
5.50.3 Property Documentation	175
5.50.3.1 Angle	175

5.50.3.2 Heading	175
5.50.3.3 LastPosition	175
5.50.3.4 Position	175
5.50.3.5 RotationSpeed	176
5.50.3.6 Speed	176
5.51 MB2D.EntityComponent.MovementSystem Class Reference	176
5.51.1 Detailed Description	177
5.51.2 Constructor & Destructor Documentation	178
5.51.2.1 MovementSystem()	178
5.51.3 Member Function Documentation	178
5.51.3.1 Process()	178
5.52 MB2D.IO.MoveRight Class Reference	178
5.52.1 Detailed Description	180
5.52.2 Constructor & Destructor Documentation	180
5.52.2.1 MoveRight()	180
5.52.3 Member Function Documentation	180
5.52.3.1 OnKeyPress()	180
5.53 MidnightBlue.MoveShip Class Reference	181
5.53.1 Detailed Description	182
5.53.2 Constructor & Destructor Documentation	182
5.53.2.1 MoveShip()	182
5.53.3 Member Function Documentation	182
5.53.3.1 OnKeyPress()	182
5.54 MB2D.IO.MoveUp Class Reference	183
5.54.1 Detailed Description	184
5.54.2 Constructor & Destructor Documentation	184
5.54.2.1 MoveUp()	184
5.54.3 Member Function Documentation	185
5.54.3.1 OnKeyPress()	185
5.55 MidnightBlue.NoiseMap Class Reference	185

5.55.1	Detailed Description	186
5.55.2	Constructor & Destructor Documentation	186
5.55.2.1	NoiseMap() [1/2]	186
5.55.2.2	NoiseMap() [2/2]	187
5.55.3	Member Function Documentation	187
5.55.3.1	GetValue()	187
5.55.3.2	SetValue()	187
5.55.4	Property Documentation	188
5.55.4.1	Height	188
5.55.4.2	Map	188
5.55.4.3	MaxValue	188
5.55.4.4	MinValue	189
5.55.4.5	Width	189
5.56	MB2D.EntityComponent.PhysicsComponent Class Reference	189
5.56.1	Detailed Description	190
5.56.2	Property Documentation	190
5.56.2.1	Acceleration	190
5.56.2.2	Power	191
5.56.2.3	RotationAcceleration	191
5.56.2.4	Velocity	191
5.57	MB2D.EntityComponent.PhysicsEnvironment Class Reference	191
5.57.1	Detailed Description	192
5.57.2	Property Documentation	192
5.57.2.1	Inertia	192
5.57.2.2	RotationInertia	192
5.58	MB2D.EntityComponent.PhysicsSystem Class Reference	193
5.58.1	Detailed Description	195
5.58.2	Constructor & Destructor Documentation	195
5.58.2.1	PhysicsSystem()	195
5.58.3	Member Function Documentation	195

5.58.3.1 Process()	195
5.58.4 Property Documentation	195
5.58.4.1 Environment	195
5.59 MidnightBlue.Planet Class Reference	196
5.59.1 Detailed Description	197
5.59.2 Constructor & Destructor Documentation	197
5.59.2.1 Planet()	197
5.59.3 Member Function Documentation	197
5.59.3.1 CreateMapTexture()	197
5.59.3.2 Generate()	197
5.59.3.3 GetMapLayer()	198
5.59.4 Property Documentation	199
5.59.4.1 Generated	199
5.59.4.2 Meta	199
5.59.4.3 Name	199
5.59.4.4 Position	199
5.59.4.5 Size	199
5.59.4.6 Tiles	200
5.60 MidnightBlue.PlanetComponent Class Reference	200
5.60.1 Detailed Description	201
5.60.2 Property Documentation	201
5.60.2.1 Data	201
5.61 MidnightBlue.PlanetMetadata Class Reference	202
5.61.1 Detailed Description	204
5.61.2 Property Documentation	204
5.61.2.1 Carbon	204
5.61.2.2 Density	204
5.61.2.3 Gas	204
5.61.2.4 Habitable	204
5.61.2.5 Name	204

5.61.2.6 Radius	205
5.61.2.7 StarDistance	205
5.61.2.8 SurfaceArea	205
5.61.2.9 SurfaceTemperature	205
5.61.2.10 Type	205
5.61.2.11 Water	205
5.62 MidnightBlue.PlanetScene Class Reference	206
5.62.1 Detailed Description	208
5.62.2 Constructor & Destructor Documentation	208
5.62.2.1 PlanetScene()	208
5.62.3 Member Function Documentation	208
5.62.3.1 Draw()	208
5.62.3.2 Exit()	209
5.62.3.3 HandleInput()	209
5.62.3.4 Initialize()	210
5.62.3.5 Pause()	210
5.62.3.6 Resume()	210
5.62.3.7 Update()	210
5.63 MidnightBlue.PlanetTile Class Reference	211
5.63.1 Detailed Description	212
5.63.2 Constructor & Destructor Documentation	213
5.63.2.1 PlanetTile()	213
5.63.3 Member Function Documentation	213
5.63.3.1 GetColor()	213
5.63.3.2 GetTextureID()	214
5.63.4 Property Documentation	214
5.63.4.1 Biome	215
5.63.4.2 Height	215
5.63.4.3 Moisture	215
5.63.4.4 Temperature	215

5.64 MB2D.EntityComponent.PlayerController Class Reference	216
5.64.1 Detailed Description	217
5.64.2 Constructor & Destructor Documentation	217
5.64.2.1 PlayerController()	217
5.64.3 Property Documentation	217
5.64.3.1 InputMap	217
5.65 MB2D.Testing.Position Class Reference	217
5.66 MB2D.PrintASTNode Class Reference	218
5.66.1 Detailed Description	220
5.66.2 Constructor & Destructor Documentation	220
5.66.2.1 PrintASTNode()	220
5.66.3 Member Function Documentation	220
5.66.3.1 Handle()	220
5.67 MB2D.QuitASTNode Class Reference	221
5.67.1 Detailed Description	222
5.67.2 Member Function Documentation	222
5.67.2.1 Handle()	222
5.68 MB2D.EntityComponent.RenderSystem Class Reference	222
5.68.1 Detailed Description	224
5.68.2 Constructor & Destructor Documentation	225
5.68.2.1 RenderSystem()	225
5.68.3 Member Function Documentation	225
5.68.3.1 PreProcess()	225
5.68.3.2 Process()	225
5.69 MB2D.RootASTNode Class Reference	225
5.69.1 Detailed Description	227
5.69.2 Constructor & Destructor Documentation	227
5.69.2.1 RootASTNode()	227
5.69.3 Member Function Documentation	227
5.69.3.1 Handle()	227

5.70 MB2D.IO.RotateLeft Class Reference	228
5.70.1 Detailed Description	229
5.70.2 Constructor & Destructor Documentation	229
5.70.2.1 RotateLeft()	229
5.70.3 Member Function Documentation	230
5.70.3.1 OnKeyPress()	230
5.71 MB2D.IO.RotateRight Class Reference	230
5.71.1 Detailed Description	231
5.71.2 Constructor & Destructor Documentation	231
5.71.2.1 RotateRight()	231
5.71.3 Member Function Documentation	232
5.71.3.1 OnKeyPress()	232
5.72 MB2D.RunASTNode Class Reference	232
5.72.1 Detailed Description	233
5.72.2 Constructor & Destructor Documentation	233
5.72.2.1 RunASTNode()	233
5.72.3 Member Function Documentation	234
5.72.3.1 Handle()	234
5.73 MB2D.Scenes.Scene Class Reference	234
5.73.1 Detailed Description	236
5.73.2 Constructor & Destructor Documentation	236
5.73.2.1 Scene()	236
5.73.3 Member Function Documentation	236
5.73.3.1 Cleanup()	237
5.73.3.2 Draw()	237
5.73.3.3 Exit()	237
5.73.3.4 HandleInput()	238
5.73.3.5 Initialize()	238
5.73.3.6 Pause()	239
5.73.3.7 Resume()	239

5.73.3.8 <code>Update()</code>	239
5.73.4 Property Documentation	240
5.73.4.1 <code>Content</code>	240
5.73.4.2 <code>DeltaTime</code>	240
5.73.4.3 <code>GameObjects</code>	240
5.73.4.4 <code>PreviousTransitionState</code>	240
5.73.4.5 <code>SceneController</code>	240
5.73.4.6 <code>TransitionState</code>	240
5.73.4.7 <code>WindowBackgroundColor</code>	241
5.74 MB2D.Scenes.SceneStack Class Reference	241
5.74.1 Detailed Description	242
5.74.2 Constructor & Destructor Documentation	242
5.74.2.1 <code>SceneStack()</code>	242
5.74.3 Member Function Documentation	242
5.74.3.1 <code>Pop()</code>	242
5.74.3.2 <code>Push()</code>	243
5.74.3.3 <code>ResetTo()</code>	244
5.74.3.4 <code>SceneAt()</code>	244
5.74.3.5 <code>Update()</code>	245
5.74.4 Property Documentation	246
5.74.4.1 <code>Bottom</code>	246
5.74.4.2 <code>LastIndex</code>	246
5.74.4.3 <code>Size</code>	246
5.74.4.4 <code>Top</code>	246
5.75 MB2D.SetASTNode Class Reference	247
5.75.1 Detailed Description	248
5.75.2 Constructor & Destructor Documentation	248
5.75.2.1 <code>SetASTNode()</code>	248
5.75.3 Member Function Documentation	248
5.75.3.1 <code>Handle()</code>	248

5.76	MidnightBlue.ShipController Class Reference	249
5.76.1	Detailed Description	250
5.76.2	Constructor & Destructor Documentation	250
5.76.2.1	ShipController()	250
5.76.3	Property Documentation	250
5.76.3.1	InputMap	250
5.76.3.2	IsLandable	250
5.76.3.3	State	251
5.77	MidnightBlue.ShipInputSystem Class Reference	251
5.77.1	Detailed Description	252
5.77.2	Constructor & Destructor Documentation	253
5.77.2.1	ShipInputSystem()	253
5.77.3	Member Function Documentation	253
5.77.3.1	Process()	253
5.78	MB2D.SoundTrigger Class Reference	253
5.78.1	Detailed Description	254
5.78.2	Constructor & Destructor Documentation	254
5.78.2.1	SoundTrigger()	254
5.78.3	Member Function Documentation	254
5.78.3.1	Cut()	254
5.78.3.2	FadeDown()	255
5.78.3.3	FadeUp()	255
5.78.3.4	Trigger()	256
5.78.4	Property Documentation	256
5.78.4.1	FadeSpeed	256
5.78.4.2	IsLooped	256
5.78.4.3	MaxVolume	256
5.79	MB2D.EntityComponent.SpriteTransform Class Reference	257
5.79.1	Detailed Description	258
5.79.2	Constructor & Destructor Documentation	258

5.79.2.1	SpriteTransform() [1/2]	258
5.79.2.2	SpriteTransform() [2/2]	258
5.79.3	Property Documentation	259
5.79.3.1	Bounds	259
5.79.3.2	Direction	259
5.79.3.3	Rotation	259
5.79.3.4	Target	259
5.79.3.5	Z	260
5.80	MidnightBlue.StarSystem Class Reference	260
5.80.1	Detailed Description	262
5.80.2	Constructor & Destructor Documentation	262
5.80.2.1	StarSystem()	262
5.80.3	Property Documentation	262
5.80.3.1	Bounds	262
5.80.3.2	Color	262
5.80.3.3	Draw	262
5.80.3.4	Name	262
5.80.3.5	PlanetList	263
5.80.3.6	Planets	263
5.80.3.7	Radius	263
5.80.3.8	Scanned	263
5.81	MidnightBlue.StarSystemHud Class Reference	264
5.81.1	Detailed Description	265
5.81.2	Constructor & Destructor Documentation	265
5.81.2.1	StarSystemHud()	265
5.82	MidnightBlue.StarSystemScene Class Reference	266
5.82.1	Detailed Description	269
5.82.2	Constructor & Destructor Documentation	269
5.82.2.1	StarSystemScene()	269
5.82.3	Member Function Documentation	270

5.82.3.1	Draw()	270
5.82.3.2	DrawMap()	270
5.82.3.3	Exit()	271
5.82.3.4	HandleInput()	271
5.82.3.5	Initialize()	271
5.82.3.6	Pause()	272
5.82.3.7	Resume()	272
5.82.3.8	Update()	272
5.83	MB2D.Testing.Test Class Reference	273
5.84	MB2D.Testing.TestSystem Class Reference	274
5.84.1	Member Function Documentation	275
5.84.1.1	Process()	275
5.85	MB2D.Testing.TestSystem2 Class Reference	276
5.85.1	Member Function Documentation	277
5.85.1.1	Process()	277
5.86	MB2D.Testing.TestUIView Class Reference	278
5.87	MB2D.IO.TextInputHandler Class Reference	279
5.87.1	Property Documentation	280
5.87.1.1	LastChar	280
5.88	MB2D.Tile Class Reference	280
5.88.1	Detailed Description	282
5.88.2	Constructor & Destructor Documentation	282
5.88.2.1	Tile() [1/2]	282
5.88.2.2	Tile() [2/2]	282
5.88.3	Property Documentation	282
5.88.3.1	Flag	282
5.88.3.2	ID	283
5.88.3.3	TintColor	283
5.89	MB2D.Tiles.TileMap Class Reference	283
5.89.1	Detailed Description	284

5.89.2 Constructor & Destructor Documentation	284
5.89.2.1 TileMap()	284
5.89.3 Member Function Documentation	285
5.89.3.1 Draw()	285
5.89.3.2 Fill()	285
5.89.3.3 GetTile()	286
5.89.3.4 HandleWrapping()	286
5.89.4 Property Documentation	286
5.89.4.1 MapSize	286
5.89.4.2 Texture	287
5.89.4.3 this[int x, int y]	287
5.89.4.4 TileSize	287
5.90 MidnightBlue.TitleScene Class Reference	287
5.90.1 Detailed Description	290
5.90.2 Constructor & Destructor Documentation	290
5.90.2.1 TitleScene()	290
5.90.3 Member Function Documentation	290
5.90.3.1 Draw()	290
5.90.3.2 Exit()	291
5.90.3.3 HandleInput()	291
5.90.3.4 Initialize()	291
5.90.3.5 Pause()	291
5.90.3.6 Resume()	291
5.90.3.7 Update()	292
5.91 MidnightBlue.TitleView Class Reference	292
5.91.1 Detailed Description	293
5.91.2 Constructor & Destructor Documentation	293
5.91.2.1 TitleView()	293
5.92 MB2D.UI.UIContent Class Reference	294
5.92.1 Detailed Description	295

5.92.2 Constructor & Destructor Documentation	295
5.92.2.1 UIContent()	295
5.92.3 Property Documentation	295
5.92.3.1 Elements	295
5.92.3.2 Grid	295
5.92.3.3 Rect	296
5.93 MB2D.UI.UIControlElement Class Reference	296
5.93.1 Detailed Description	299
5.93.2 Constructor & Destructor Documentation	299
5.93.2.1 UIControlElement() [1/2]	299
5.93.2.2 UIControlElement() [2/2]	300
5.93.3 Member Function Documentation	300
5.93.3.1 Draw()	300
5.93.3.2 Update()	300
5.93.4 Member Data Documentation	300
5.93.4.1 _currentState	300
5.93.4.2 _previousState	300
5.93.5 Property Documentation	301
5.93.5.1 HighlightedSound	301
5.93.5.2 HighlightedTextColor	301
5.93.5.3 HighlightedTexture	301
5.93.5.4 NormalTextColor	301
5.93.5.5 NormalTexture	301
5.93.5.6 PressedTexture	301
5.94 MB2D.UI.UIElement Class Reference	302
5.94.1 Detailed Description	304
5.94.2 Constructor & Destructor Documentation	304
5.94.2.1 UIElement()	304
5.94.3 Member Function Documentation	305
5.94.3.1 Draw()	305

5.94.3.2	DrawBorder()	305
5.94.3.3	SetRelativeSize() [1/2]	305
5.94.3.4	SetRelativeSize() [2/2]	306
5.94.3.5	Update()	306
5.94.4	Property Documentation	307
5.94.4.1	BackgroundColor	307
5.94.4.2	BorderBottomColor	307
5.94.4.3	BorderColor	307
5.94.4.4	BorderDisplayed	307
5.94.4.5	BorderLeftColor	308
5.94.4.6	BorderRightColor	308
5.94.4.7	BorderTopColor	308
5.94.4.8	BorderWidth	308
5.94.4.9	BoundingBox	308
5.94.4.10	Content	308
5.94.4.11	Fill	309
5.94.4.12	Font	309
5.94.4.13	Size	309
5.94.4.14	Tag	309
5.94.4.15	TextColor	309
5.94.4.16	TextContent	309
5.95	MB2D.Testing.UITest Class Reference	310
5.95.1	Member Function Documentation	312
5.95.1.1	Draw()	312
5.95.1.2	Exit()	312
5.95.1.3	HandleInput()	312
5.95.1.4	Initialize()	313
5.95.1.5	Pause()	313
5.95.1.6	Resume()	313
5.95.1.7	Update()	313

5.96 MB2D.UI.UIView Class Reference	314
5.96.1 Detailed Description	315
5.96.2 Constructor & Destructor Documentation	315
5.96.2.1 UIView()	315
5.96.3 Member Function Documentation	315
5.96.3.1 Add()	315
5.96.3.2 AddToLookup()	316
5.96.3.3 Draw()	317
5.96.3.4 Update()	317
5.96.4 Property Documentation	318
5.96.4.1 BackgroundTexture	318
5.96.4.2 Content	318
5.96.4.3 StretchBackground	319
5.96.4.4 this[string key]	319
5.97 MB2D.Testing.Unregistered Class Reference	319
5.98 MB2D.EntityComponent.UtilityController Class Reference	320
5.98.1 Detailed Description	322
5.98.2 Constructor & Destructor Documentation	322
5.98.2.1 UtilityController()	322
5.98.3 Property Documentation	322
5.98.3.1 InputMap	322
5.99 MB2D.VariableASTNode Class Reference	322
5.99.1 Detailed Description	323
5.99.2 Constructor & Destructor Documentation	323
5.99.2.1 VariableASTNode()	323
5.99.3 Property Documentation	323
5.99.3.1 Type	323
5.99.3.2 Value	323
5.100MB2D.Testing.Velocity Class Reference	324

Chapter 1

Namespace Index

1.1 Namespace List

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

MB2D	11
MB2D.Collision	12
MB2D.EntityComponent	13
MB2D.Geometry	14
MB2D.IO	14
MB2D.Scenes	15
MB2D.Testing	15
MB2D.Tiles	16
MB2D.UI	16
MidnightBlue	17
MidnightBlue.Testing	20

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

MB2D.Collectable	25
MidnightBlue.Fuel	77
MB2D.Collision.CollisionCell	29
MB2D.Collision.CollisionMap	34
MB2D.IO.Command	46
MB2D.IO.ConsoleCommand	48
MB2D.IO.MoveBackward	165
MB2D.IO.MoveDown	167
MB2D.IO.MoveForward	169
MB2D.IO.MoveLeft	171
MB2D.IO.MoveRight	178
MB2D.IO.MoveUp	183
MB2D.IO.RotateLeft	228
MB2D.IO.RotateRight	230
MidnightBlue.EnterStarSystem	54
MidnightBlue.LandCommand	110
MidnightBlue.LaunchCommand	113
MidnightBlue.LeaveStarSystem	119
MidnightBlue.MenuCommand	158
MidnightBlue.MoveShip	181
MB2D.EntityComponent.Entity	57
MB2D.Testing.EntityContainerTests	63
MB2D.EntityComponent.EntityMap	64
MB2D.EntityComponent.EntitySystem	71
MB2D.EntityComponent.CollisionSystem	41
MB2D.EntityComponent.DepthSystem	52
MB2D.EntityComponent.InputSystem	103
MB2D.EntityComponent.MovementSystem	176
MB2D.EntityComponent.PhysicsSystem	193
MB2D.EntityComponent.RenderSystem	222
MB2D.Testing.CollisionRenderSystem	39
MB2D.Testing.TestSystem	274
MB2D.Testing.TestSystem2	276
MidnightBlue.GalaxyRenderSystem	83
MidnightBlue.ShipInputSystem	251

MidnightBlue.GalaxyBuilder	79
Game	
MB2D.MBGame	151
MidnightBlue.Testing.GenTest	92
MB2D.Geometry.Grid	93
MB2D.EntityComponent.IComponent	95
MB2D.EntityComponent.CollisionComponent	31
MB2D.EntityComponent.Depth	50
MB2D.EntityComponent.Inventory	106
MB2D.EntityComponent.Movement	173
MB2D.EntityComponent.PhysicsComponent	189
MB2D.EntityComponent.PlayerController	216
MB2D.EntityComponent.SpriteTransform	257
MB2D.EntityComponent.UtilityController	320
MB2D.Testing.Position	217
MB2D.Testing.Test	273
MB2D.Testing.Unregistered	319
MB2D.Testing.Velocity	324
MidnightBlue.PlanetComponent	200
MidnightBlue.PlanetMetadata	202
MidnightBlue.ShipController	249
MidnightBlue.StarSystem	260
MB2D.IO.InputMap	100
MidnightBlue.Length	121
MB2D.Geometry.Line	123
MB2D.MBConsole	134
MB2D.MBConsoleASTNode	145
MB2D.PrintASTNode	218
MB2D.QuitASTNode	221
MB2D.RootASTNode	225
MB2D.RunASTNode	232
MB2D.SetASTNode	247
MB2D.MBConsoleLexer	146
MB2D.MBConsoleParser	149
MidnightBlue.NoiseMap	185
MB2D.EntityComponent.PhysicsEnvironment	191
MidnightBlue.Planet	196
MB2D.Scenes.Scene	234
MB2D.Testing.UITest	310
MidnightBlue.GalaxyScene	87
MidnightBlue.InitScene	95
MidnightBlue.MenuScene	160
MidnightBlue.PlanetScene	206
MidnightBlue.StarSystemScene	266
MidnightBlue.Testing.MapTest	131
MidnightBlue.TitleScene	287
MB2D.Scenes.SceneStack	241
MB2D.SoundTrigger	253
MB2D.IO.TextInputHandler	279
MB2D.Tile	280
MidnightBlue.PlanetTile	211
MB2D.Tiles.TileMap	283
MB2D.UI.UIContent	294
MB2D.UI.UIElement	302
MB2D.UI.Label	108
MB2D.UI.Layout	115
MB2D.UI.UIControlElement	296

MB2D.UI.Button	21
MB2D.UI.ListControl	126
MB2D.UI.UIView	314
MB2D.Testing.TestUIView	278
MidnightBlue.GalaxyHud	81
MidnightBlue.MenuView	164
MidnightBlue.StarSystemHud	264
MidnightBlue.TitleView	292
MB2D.VariableASTNode	322

Chapter 3

Class Index

3.1 Class List

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

MB2D.UI.Button	A pressable ui element with a single OnPress event	21
MB2D.Collectable	Defines an object that can be contained within components and systems that operate on collectible items, such as Inventory.	25
MB2D.Collision.CollisionCell	A cell used in a collision map to hold a linked list of all its contained entities	29
MB2D.EntityComponent.CollisionComponent	Used for running collision detection on an Entity	31
MB2D.Collision.CollisionMap	A 2D Grid that represents a particular space in the game world to check for collisions. Uses spatial indexing to determine where an entity will be located at any given time. For best results, the cellsize and overall size of the map should be tweaked for each individual game screen and environment.	34
MB2D.Testing.CollisionRenderSystem	39
MB2D.EntityComponent.CollisionSystem	Checks collisions. Uses a spatial indexing grid for broad-phase collision checking and AABB checks for narrow phase	41
MB2D.IO.Command	Executes an action associated with a specific key	46
MB2D.IO.ConsoleCommand	Shows or hides the debug console	48
MB2D.EntityComponent.Depth	A tag class used to define an entity that should be draw sorted according to its current z-index	50
MB2D.EntityComponent.DepthSystem	Changes an entities z-index based on the y coordinate of the top of their sprite	52
MidnightBlue.EnterStarSystem	Enters a star system scene from the galaxy view	54
MB2D.EntityComponent.Entity	Represents a tagged and id'd container for components that can be operated on by systems.	57
MB2D.Testing.EntityContainerTests	63
MB2D.EntityComponent.EntityMap	Maps entities, systems and components to one another and provides querying and updating access to all elements	64
MB2D.EntityComponent.EntitySystem	Performs logic on an entity	71

MidnightBlue.Fuel	
Fuel used in the ships normal thruster drive	77
MidnightBlue.GalaxyBuilder	
.	79
MidnightBlue.GalaxyHud	
HUD to show in the galaxy view.	81
MidnightBlue.GalaxyRenderSystem	
Renders all information into the main HUD's list box on the hovered star system. Also displays the name of all the star systems planets.	83
MidnightBlue.GalaxyScene	
The scene displayed at the galaxy view - handles the control over all systems, loading, and content management for the scene.	87
MidnightBlue.Testing.GenTest	
.	92
MB2D.Geometry.Grid	
Represents a grid structure. Can be drawn via a SpriteBatch	93
MB2D.EntityComponent.IComponent	
Tags any class as a valid component for use in the EntityMap . Derived classes should contain no logic, only data fields.	95
MidnightBlue.InitScene	
The scene shown at the title screen.	95
MB2D.IO.InputMap	
Maps commands to keys and trigger types	100
MB2D.EntityComponent.InputSystem	
Processes input for PlayerController and UtilityController entities. Can operate on an entity with either or both components	103
MB2D.EntityComponent.Inventory	
Defines a dictionary of Collectable types used for entities	106
MB2D.UI.Label	
A static UIElement with a TextContent, border and optional texture	108
MidnightBlue.LandCommand	
Lands the ship	110
MidnightBlue.LaunchCommand	
Launches the ship from a landed state	113
MB2D.UI.Layout	
A container for UIElements used within a UIView . Used to divide the View into smaller segments and to move around a group of elements easily	115
MidnightBlue.LeaveStarSystem	
.	119
MidnightBlue.Length	
Defines a measurement of length in meters able to be converted to other measurements.	121
MB2D.Geometry.Line	
A line structure, can be drawn via SpriteBatch	123
MB2D.UI.ListControl	
A scrollable list box. Items can be added and interacted with.	126
MidnightBlue.Testing.MapTest	
.	131
MB2D.MBConsole	
Midnight Blue debug console class. Executes attached methods and changes attached variables.	134
MB2D.MBConsoleASTNode	
Class all AST nodes inherit from	145
MB2D.MBConsoleLexer	
Breaks a string into a series of tokens to use for parsing the debug consoles command language	146
MB2D.MBConsoleParser	
Parses command string input and executes it using a debug console.	149
MB2D.MBGame	
This is the main type for your game.	151
MidnightBlue.MenuCommand	
.	158
MidnightBlue.MenuScene	
.	160
MidnightBlue.MenuView	
.	164

MB2D.IO.MoveBackward	Moves an entity backward. Only runs on entities with a physics component	165
MB2D.IO.MoveDown	Moves an entity down	167
MB2D.IO.MoveForward	Moves an entity forward. Only runs on entities with a physics component	169
MB2D.IO.MoveLeft	Moves an entity left.	171
MB2D.EntityComponent.Movement	Defines position, rotation and speed related data for moving an entity.	173
MB2D.EntityComponent.MovementSystem	Processes the change in position, rotation, and sprite transform for an entity	176
MB2D.IO.MoveRight	Moves an entity right	178
MidnightBlue.MoveShip	Performs logic aside from movement required to execute when moving the ship such as consuming fuel.	181
MB2D.IO.MoveUp	Moves a player controller up	183
MidnightBlue.NoiseMap	Generates a fractal 2D map using Simplex Noise	185
MB2D.EntityComponent.PhysicsComponent	Physics component used to define acceleration and velocity.	189
MB2D.EntityComponent.PhysicsEnvironment	Defines a new environment to feed into the physics system to alter the impact it has on an entity	191
MB2D.EntityComponent.PhysicsSystem	Processes physics changes for a given entity	193
MidnightBlue.Planet	A fully-generated planet in a star system with associated texture maps.	196
MidnightBlue.PlanetComponent	Represents a planet entity with pre-generated metadata	200
MidnightBlue.PlanetMetadata	Planet metadata used as information and arguments for generating the actual biome map of a planet. Required for an entity to be treated as a planet.	202
MidnightBlue.PlanetScene	Scene active when the player is exploring a given planet.	206
MidnightBlue.PlanetTile	A tile type used in planet tilemaps	211
MB2D.EntityComponent.PlayerController	Defines the attached entity as controllable	216
MB2D.Testing.Position	217
MB2D.PrintASTNode	Prints a variable to the console.	218
MB2D.QuitASTNode	Handles quitting the game	221
MB2D.EntityComponent.RenderSystem	Renders culled entities with a SpriteTransform to the window	222
MB2D.RootASTNode	The entry point for command execution with a single child.	225
MB2D.IO.RotateLeft	Rotates an entity left	228
MB2D.IO.RotateRight	Rotates an entity right	230
MB2D.RunASTNode	AST node entry point for executing a run command	232
MB2D.Scenes.Scene	Holds all logic and data for a single game screen	234

MB2D.Scenes.SceneStack	Holds the games scenes in a stack structure running the top scene every frame. Handles switching state for scenes and popping/pushing new scenes on top of one another. Allows the current scene to access other scenes.	241
MB2D.SetASTNode	AST node representing the entry point for a 'set' command with an identifier and a value child	247
MidnightBlue.ShipController	Controls a ships movement and actions	249
MidnightBlue.ShipInputSystem	Handles moving the ship forward and backwards.	251
MB2D.SoundTrigger	Triggers a sound effect	253
MB2D.EntityComponent.SpriteTransform	Defines a sprite component with control over its size, rotation, and scale	257
MidnightBlue.StarSystem	Represents an star system entity to be used in the galaxy view	260
MidnightBlue.StarSystemHud	Star system hud with minimap.	264
MidnightBlue.StarSystemScene	Scene to display a star system with planets and a star.	266
MB2D.Testing.Test	273
MB2D.Testing.TestSystem	274
MB2D.Testing.TestSystem2	276
MB2D.Testing.TestUIView	278
MB2D.IO.TextInputHandler	279
MB2D.Tile	Represents a single tile in a tile map.	280
MB2D.Tiles.TileMap	A grid of tiles with collision. Wraps coordinates when they fall out of bounds. Allows accessing tiles by index.	283
MidnightBlue.TitleScene	The scene shown at the title screen.	287
MidnightBlue.TitleView	The title screens UI view	292
MB2D.UI.UIContent	Holds content in a grid structure for a UIContext or Layout	294
MB2D.UI.UIControlElement	An interactive and controllable UIElement	296
MB2D.UI.UIElement	Defines a UI object that can be contained within Views and Layouts, drawn, updated, and moved about	302
MB2D.Testing.UITest	310
MB2D.UI.UIView	A single context for all UI elements and layouts.	314
MB2D.Testing.Unregistered	319
MB2D.EntityComponent.UtilityController	Declares the attached entity as able to control utility commands such as opening the debug console	320
MB2D.VariableASTNode	Represents a variable with a type and a value	322
MB2D.Testing.Velocity	324

Chapter 4

Namespace Documentation

4.1 MB2D Namespace Reference

Namespaces

Classes

- class **Collectable**
Defines an object that can be contained within components and systems that operate on collectible items, such as Inventory.
- class **MBConsole**
Midnight Blue debug console class. Executes attached methods and changes attached variables.
- class **MBConsoleASTNode**
Class all AST nodes inherit from
- class **MBConsoleLexer**
Breaks a string into a series of tokens to use for parsing the debug consoles command language
- class **MBConsoleParser**
Parses command string input and executes it using a debug console.
- class **MBGame**
This is the main type for your game.
- class **MBMath**
Math helper class
- class **PrintASTNode**
Prints a variable to the console.
- class **QuitASTNode**
Handles quitting the game
- class **RootASTNode**
The entry point for command execution with a single child.
- class **RunASTNode**
AST node entry point for executing a run command
- class **SetASTNode**
AST node representing the entry point for a 'set' command with an identifier and a value child
- class **SoundTrigger**
Triggers a sound effect
- class **SpriteBatchExtensions**
Extends SpriteBatch with [MidnightBlue](#) data structures.
- class **Tile**
Represents a single tile in a tile map.
- class **VariableASTNode**
Represents a variable with a type and a value
- class **Vector2Extensions**

Enumerations

- enum [Token](#) {
 [Token.Unknown](#), [Token.String](#), [Token.Set](#), [Token.Run](#),
[Token.Print](#), [Token.Quit](#)

Category of tokens created by the lexer

- enum [TileFlag](#) { [TileFlag.Passable](#), [TileFlag.Impassable](#) }

Flags a tile as passable or impassable, used in collision checking.

4.1.1 Enumeration Type Documentation

4.1.1.1 TileFlag

enum [MB2D.TileFlag](#) [strong]

Flags a tile as passable or impassable, used in collision checking.

Enumerator

Passable	Flags a tile as walkable
Impassable	Flags a tile as collidable and unable to be walked on.

4.1.1.2 Token

enum [MB2D.Token](#) [strong]

Category of tokens created by the lexer

Enumerator

Unknown	A value or an identifier
String	A string sequence
Set	A set command statement
Run	A run command statement
Print	A print command statement
Quit	A quit command statement

4.2 MB2D.Collision Namespace Reference

Classes

- class [CollisionCell](#)
A cell used in a collision map to hold a linked list of all its contained entities
- class [CollisionMap](#)
A 2D Grid that represents a particular space in the game world to check for collisions. Uses spatial indexing to determine where an entity will be located at any given time. For best results, the cellsize and overall size of the map should be tweaked for each individual game screen and environment.

4.3 MB2D.EntityComponent Namespace Reference

Classes

- class [CollisionComponent](#)
Used for running collision detection on an Entity
- class [CollisionSystem](#)
Checks collisions. Uses a spatial indexing grid for broad-phase collision checking and AABB checks for narrow phase
- class [Depth](#)
A tag class used to define an entity that should be draw sorted according to its current z-index
- class [DepthSystem](#)
Changes an entities z-index based on the y coordinate of the top of their sprite
- class [Entity](#)
Represents a tagged and id'd container for components that can be operated on by systems.
- class [EntityMap](#)
Maps entities, systems and components to one another and provides querying and updating access to all elements
- class [EntitySystem](#)
Performs logic on an entity.
- interface [IComponent](#)
Tags any class as a valid component for use in the EntityMap. Derived classes should contain no logic, only data fields.
- class [InputSystem](#)
Processes input for PlayerController and UtilityController entities. Can operate on an entity with either or both components
- class [Inventory](#)
Defines a dictionary of Collectable types used for entities
- class [Movement](#)
Defines position, rotation and speed related data for moving an entity.
- class [MovementSystem](#)
Processes the change in position, rotation, and sprite transform for an entity
- class [PhysicsComponent](#)
Physics component used to define acceleration and velocity.
- class [PhysicsEnvironment](#)
Defines a new environment to feed into the physics system to alter the impact it has on an entity
- class [PhysicsSystem](#)
Processes physics changes for a given entity
- class [PlayerController](#)
Defines the attached entity as controllable
- class [RenderSystem](#)
Renders culled entities with a SpriteTransform to the window
- class [SpriteTransform](#)
Defines a sprite component with control over its size, rotation, and scale
- class [UtilityController](#)
Declares the attached entity as able to control utility commands such as opening the debug console

Enumerations

- enum [EntityAssociation](#) { Strict, Loose }

4.4 MB2D.Geometry Namespace Reference

Classes

- class [Grid](#)
Represents a grid structure. Can be drawn via a SpriteBatch
- class [Line](#)
A line structure, can be drawn via SpriteBatch

4.5 MB2D.IO Namespace Reference

Classes

- class [Command](#)
Executes an action associated with a specific key
- class [ConsoleCommand](#)
Shows or hides the debug console
- class [InputMap](#)
Maps commands to keys and trigger types
- class [IOUtil](#)
Utility methods for working with the keyboard and mouse
- class [MoveBackward](#)
Moves an entity backward. Only runs on entities with a physics component
- class [MoveDown](#)
Moves an entity down
- class [MoveForward](#)
Moves an entity forward. Only runs on entities with a physics component
- class [MoveLeft](#)
Moves an entity left.
- class [MoveRight](#)
Moves an entity right
- class [MoveUp](#)
Moves a player controller up
- class [RotateLeft](#)
Rotates an entity left
- class [RotateRight](#)
Rotates an entity right
- class [TextInputHandler](#)

Enumerations

- enum [CommandType](#) { [CommandType.Hold](#), [CommandType.Trigger](#) }
Represents either a trigger or hold command type

4.5.1 Enumeration Type Documentation

4.5.1.1 CommandType

```
enum MB2D.IO.CommandType [strong]
```

Represents either a trigger or hold command type

Enumerator

Hold	Execute command every frame its associated input key/button is detected
Trigger	Execute command only on the first frame its associated input key/button is detected and don't execute again until it's released and pressed again

4.6 MB2D.Scenes Namespace Reference

Classes

- class [Scene](#)
Holds all logic and data for a single game screen
- class [SceneStack](#)
Holds the games scenes in a stack structure running the top scene every frame. Handles switching state for scenes and popping/pushing new scenes on top of one another. Allows the current scene to access other scenes.

Enumerations

- enum [TransitionState](#) {
 [TransitionState.Null](#), [TransitionState.None](#), [TransitionState.Pausing](#), [TransitionState.Resuming](#),
 [TransitionState.Exiting](#), [TransitionState.Initializing](#) }
Defines a valid transition state to move into. Once set, the scene stack will automatically move the current scene into that state the next frame.

4.6.1 Enumeration Type Documentation

4.6.1.1 TransitionState

```
enum MB2D.Scenes.TransitionState [strong]
```

Defines a valid transition state to move into. Once set, the scene stack will automatically move the current scene into that state the next frame.

Enumerator

Null	The scene hasn't initialized yet
None	The normal state
Pausing	The scene is currently pausing. Set state to None to end transition.
Resuming	The scene is resuming from the paused state. Set state to None to end transition.
Exiting	The scene is exiting to be destroyed. Set state to Null to end transition.
Initializing	The scene is initializing from the an unconstructed state. Set state to None to end.

4.7 MB2D.Testing Namespace Reference

Classes

- class [CollisionRenderSystem](#)
- struct [EntityContainerTests](#)
- class [Position](#)
- class [Test](#)
- class [TestSystem](#)
- class [TestSystem2](#)
- class [TestUIView](#)
- class [UITest](#)
- class [Unregistered](#)
- class [Velocity](#)

4.8 MB2D.Tiles Namespace Reference

Classes

- class [TileMap](#)

A grid of tiles with collision. Wraps coordinates when they fall out of bounds. Allows accessing tiles by index.

4.9 MB2D.UI Namespace Reference

Classes

- class [Button](#)

A pressable ui element with a single OnPress event
- class [Label](#)

A static [UIElement](#) with a [TextContent](#), border and optional texture
- class [Layout](#)

A container for [UIElements](#) used within a [UIView](#). Used to divide the View into smaller segments and to move around a group of elements easily
- class [ListControl](#)

A scrollable list box. Items can be added and interacted with.
- class [UIContent](#)

Holds content in a grid structure for a [UIContext](#) or [Layout](#)
- class [UIControlElement](#)

An interactive and controllable [UIElement](#)
- class [UIElement](#)

Defines a [UI](#) object that can be contained within Views and Layouts, drawn, updated, and moved about
- class [UIView](#)

A single context for all [UI](#) elements and layouts.

Enumerations

- enum [UIState](#) { [UIState.Normal](#), [UIState.Selected](#), [UIState.Pressed](#) }

Represents the current state of a controllable [UIElement](#)

4.9.1 Enumeration Type Documentation

4.9.1.1 UIState

```
enum MB2D.UI.UIState [strong]
```

Represents the current state of a controllable [UIElement](#)

Enumerator

Normal	Unselected, unpressed state
Selected	Hovered or highlighted state
Pressed	Clicked or pressed state

4.10 MidnightBlue Namespace Reference

Namespaces

Classes

- class [EcosystemTool](#)
Takes numeric inputs and produces temperature, height, and moisture categories. Also produces biomes based on the three outputs.
- class [EnterStarSystem](#)
Enters a star system scene from the galaxy view
- class [Fuel](#)
Fuel used in the ships normal thruster drive
- class [GalaxyBuilder](#)
- class [GalaxyHud](#)
HUD to show in the galaxy view.
- class [GalaxyRenderSystem](#)
Renders all information into the main HUD's list box on the hovered star system. Also displays the name of all the star systems planets.
- class [GalaxyScene](#)
The scene displayed at the galaxy view - handles the control over all systems, loading, and content management for the scene.
- class [InitScene](#)
The scene shown at the title screen.
- class [LandCommand](#)
Lands the ship
- class [LaunchCommand](#)
Launches the ship from a landed state
- class [LeaveStarSystem](#)
- class [Length](#)
Defines a measurement of length in meters able to be converted to other measurements.
- class [MenuCommand](#)
- class [MenuScene](#)
- class [MenuView](#)
- class [MoveShip](#)
Performs logic aside from movement required to execute when moving the ship such as consuming fuel.
- class [NoiseMap](#)
Generates a fractal 2D map using Simplex Noise
- class [Planet](#)
A fully-generated planet in a star system with associated texture maps.
- class [PlanetComponent](#)
Represents a planet entity with pre-generated metadata

- class [PlanetMetadata](#)
Planet metadata used as information and arguments for generating the actual biome map of a planet. Required for an entity to be treated as a planet.
- class [PlanetScene](#)
Scene active when the player is exploring a given planet.
- class [PlanetTile](#)
A tile type used in planet tilemaps
- class [ShipController](#)
Controls a ships movement and actions
- class [ShipInputSystem](#)
Handles moving the ship forward and backwards.
- class [StarSystem](#)
Represents an star system entity to be used in the galaxy view
- class [StarSystemHud](#)
Star system hud with minimap.
- class [StarSystemScene](#)
Scene to display a star system with planets and a star.
- class [TitleScene](#)
The scene shown at the title screen.
- class [TitleView](#)
The title screens UI view
- class [UIColors](#)
The main color scheme for the UI

Enumerations

- enum [ShipState](#) {
Normal, Landing, Launching, LeavingScreen, Warping }
Represents the current travelling state of the ship
- enum [Biome](#) {
Biome.Tundra, Biome.Taiga, Biome.Woodland, Biome.Shrubland,
Biome.TemperateGrassland, Biome.Desert, Biome.SubtropicalDesert, Biome.Savana,
Biome.TropicalSeasonalForest, Biome.TemperateSeasonalForest, Biome.TemperateRainforest, Biome.TropicalRainforest,
Biome.Barren, Biome.ShallowOcean, Biome.Ocean, Biome.Ice }
Represents a biome type in a planets tile map.
- enum [HeightLevel](#) {
HeightLevel.Depths = 0, HeightLevel.SeaLevel, HeightLevel.Lowland, HeightLevel.Mountainous,
HeightLevel.Alpine, HeightLevel.Snow }
Represents height categories for biomes.
- enum [MoistureLevel](#) {
Arid = 0, Dry, SemiDry, SemiMoist, Moist, Wet }
Represents moisture categories for biomes used in generation.
- enum [TemperatureLevel](#) {
TemperatureLevel.Freezing = 0, TemperatureLevel.Polar, TemperatureLevel.Tundra, TemperatureLevel.Taiga,
TemperatureLevel.Temperate, TemperatureLevel.SubTropical, TemperatureLevel.Tropical, TemperatureLevel.Hot,
TemperatureLevel.Harsh, TemperatureLevel.SuperHot, TemperatureLevel.Scorching }
Represents categories of temperature used in generating biomes and player interactions.
- enum [PlanetType](#) { **Water, Terrestrial, Gas** }
Planet type used for information.

4.10.1 Enumeration Type Documentation

4.10.1.1 Biome

```
enum MidnightBlue.Biome [strong]
```

Represents a biome type in a planets tile map.

Enumerator

Tundra	Super cold environment.
Taiga	Most common cold environment.
Woodland	Cold forest biome.
Shrubland	Shrubland - can be cold or hot.
TemperateGrassland	Most common temperate biome.
Desert	Desert biome.
SubtropicalDesert	Desert biome akin to southern californian deserts.
Savana	Savana biome - flatlands with grass.
TropicalSeasonalForest	Biome similar to Australian outback forests.
TemperateSeasonalForest	Biome similar to canadian redwood forests.
TemperateRainforest	Biome like southern Australian rainforests - colder environment rainforest.
TropicalRainforest	Classic rainforest biome.
Barren	Completely cracked, barren environment most present in inhospitable location.
ShallowOcean	Shallow water biome - lighter in color than deep water.
Ocean	Deep ocean biome.
Ice	Ice biome - mostly present in inhospitable cold planets.

4.10.1.2 HeightLevel

```
enum MidnightBlue.HeightLevel [strong]
```

Represents height categories for biomes.

Enumerator

Depths	Will always be ocean biome unless super cold where it will be ice.
SeaLevel	Will always be shallow ocean biome unless super cold where it will be ice.
Lowland	Will contain all forest and grassland biomes.
Mountainous	Will contain all mountainous biomes - will be colder than other biomes, too.
Alpine	Will contain all mountainous biomes - will be much colder than other biomes, too.
Snow	Highest level of elevation, super cold and only mountainous biomes.

4.10.1.3 MoistureLevel

```
enum MidnightBlue.MoistureLevel [strong]
```

Represents moisture categories for biomes used in generation.

4.10.1.4 PlanetType

enum [MidnightBlue.PlanetType](#) [strong]

[Planet](#) type used for information.

4.10.1.5 ShipState

enum [MidnightBlue.ShipState](#) [strong]

Represents the current travelling state of the ship

4.10.1.6 TemperatureLevel

enum [MidnightBlue.TemperatureLevel](#) [strong]

Represents categories of temperature used in generating biomes and player interactions.

Enumerator

Freezing	Super cold - only on uninhabitable planets
Polar	Coldest temperature found on inhabitable planets.
Tundra	Coldest temperature to produce varied biomes.
Taiga	Most common cold temperature across inhabitable planets.
Temperate	Most inhabitable temperature that produces the most varied biomes.
SubTropical	Temperature most commonly found for 'hot' biomes.
Tropical	Produces simultaneously very moist, green biomes, and very dry, arid biomes.
Hot	Hottest inhabitable temperature.
Harsh	Hottest temperature to produce varied biomes.
SuperHot	Will only produce barren or desert biomes - no water
Scorching	Will only produce barren biomes - no water. This is like 200+ degrees

4.11 MidnightBlue.Testing Namespace Reference

Classes

- class [GenTest](#)
- class [MapTest](#)

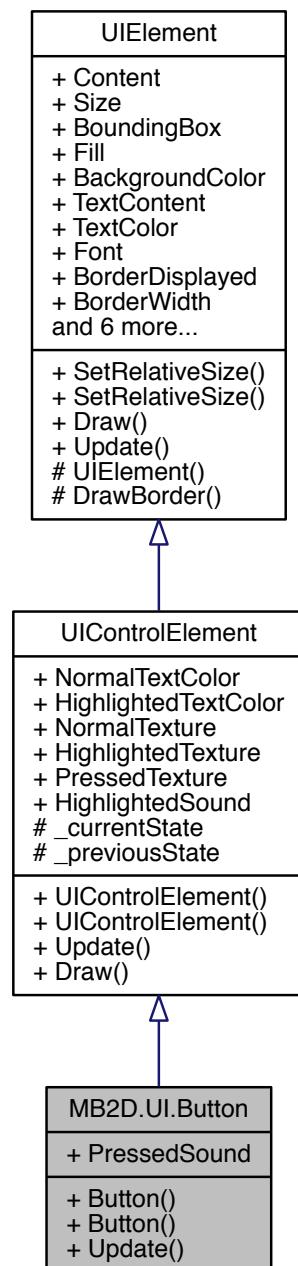
Chapter 5

Class Documentation

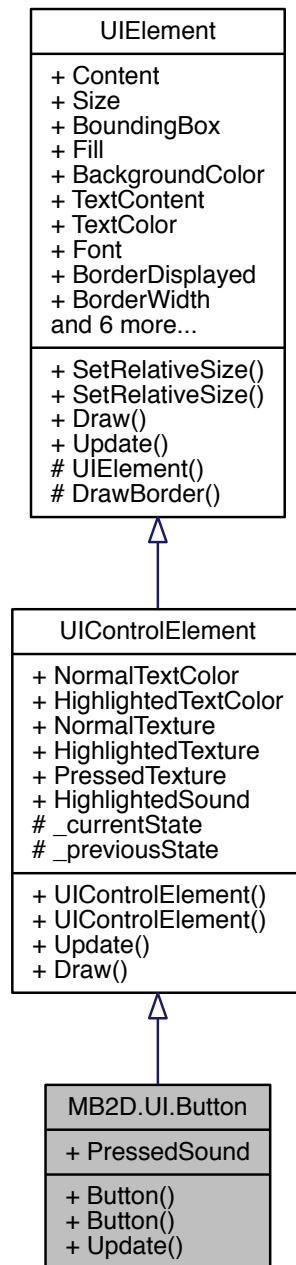
5.1 MB2D.UI.Button Class Reference

A pressable ui element with a single OnPress event

Inheritance diagram for MB2D.UI.Button:



Collaboration diagram for MB2D.UI.Button:



Public Member Functions

- **Button** (Texture2D normal, Texture2D selected, Texture2D pressed)
Initializes a new instance of the T:MB2D.UI.Button class.
- **Button ()**
Initializes a new instance of the T:MB2D.UI.Button class with no associated textures
- **override void Update ()**
Updates the button state.

Properties

- SoundEffectInstance [PressedSound](#) [get, set]
Gets or sets the sound fired when transitioning to the pressed state.

Events

- EventHandler [OnPress](#)
Occurs when the button has been clicked or pressed.

Additional Inherited Members

5.1.1 Detailed Description

A pressable ui element with a single OnPress event

5.1.2 Constructor & Destructor Documentation

5.1.2.1 [Button\(\)](#) [1/2]

```
MB2D.UI.Button.Button (
    Texture2D normal,
    Texture2D selected,
    Texture2D pressed ) [inline]
```

Initializes a new instance of the T:MB2D.UI.Button class.

Parameters

<i>normal</i>	Normal state texture
<i>selected</i>	Selected state texture.
<i>pressed</i>	Pressed state texture.

5.1.2.2 [Button\(\)](#) [2/2]

```
MB2D.UI.Button.Button ( ) [inline]
```

Initializes a new instance of the T:MB2D.UI.Button class with no associated textures

5.1.3 Member Function Documentation

5.1.3.1 [Update\(\)](#)

```
override void MB2D.UI.Button.Update ( ) [inline], [virtual]
```

Updates the button state.

Implements [MB2D.UI.UIElement](#).

5.1.4 Property Documentation

5.1.4.1 PressedSound

```
SoundEffectInstance MB2D.UI.Button.PressedSound [get], [set]
```

Gets or sets the sound fired when transitioning to the pressed state.

The pressed state sound.

5.1.5 Event Documentation

5.1.5.1 OnPress

```
EventHandler MB2D.UI.Button.OnPress
```

Occurs when the button has been clicked or pressed.

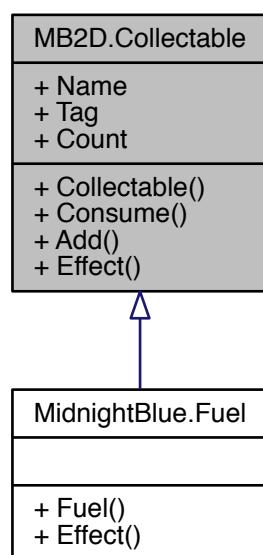
The documentation for this class was generated from the following file:

- MB2D/src/UI/Button.cs

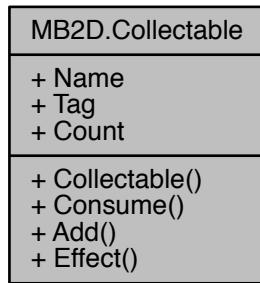
5.2 MB2D.Collectable Class Reference

Defines an object that can be contained within components and systems that operate on collectible items, such as Inventory.

Inheritance diagram for MB2D.Collectable:



Collaboration diagram for MB2D.Collectable:



Public Member Functions

- **Collectable** (string name, string tag, int initialCount)
Initializes a new instance of the T:MB2D.Collectable class.
- void **Consume** (int amount=1)
Consumes a number of instances of the item
- void **Add** (int amount=1)
Adds a number of instances of this item to the container
- abstract void **Effect** (Entity entity)
The action to enact when the item is consumed or used

Properties

- string **Name** [get]
Gets the name of the item.
- string **Tag** [get]
Gets the items tag.
- int **Count** [get]
Gets the count of available instances of the item.

5.2.1 Detailed Description

Defines an object that can be contained within components and systems that operate on collectible items, such as Inventory.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 Collectable()

```
MB2D.Collectable.Collectable (
    string name,
    string tag,
    int initialCount ) [inline]
```

Initializes a new instance of the T:MB2D.Collectable class.

Parameters

<i>name</i>	Name to give to the item.
<i>tag</i>	Short tag to give to the item.
<i>initialCount</i>	Initial count to add to the container.

5.2.3 Member Function Documentation**5.2.3.1 Add()**

```
void MB2D.Collectable.Add (
    int amount = 1 ) [inline]
```

Adds a number of instances of this item to the container

Parameters

<i>amount</i>	Amount to add.
---------------	----------------

Here is the call graph for this function:

**5.2.3.2 Consume()**

```
void MB2D.Collectable.Consume (
    int amount = 1 ) [inline]
```

Consumes a number of instances of the item

Parameters

<i>amount</i>	Amount to consume.
---------------	--------------------

5.2.3.3 Effect()

```
abstract void MB2D.Collectable.Effect (
    Entity entity ) [pure virtual]
```

The action to enact when the item is consumed or used

Parameters

<i>entity</i>	Entity to operate on.
---------------	-----------------------

Implemented in [MidnightBlue.Fuel](#).

Here is the caller graph for this function:



5.2.4 Property Documentation

5.2.4.1 Count

```
int MB2D.Collectable.Count [get]
```

Gets the count of available instances of the item.

The count.

5.2.4.2 Name

```
string MB2D.Collectable.Name [get]
```

Gets the name of the item.

The name.

5.2.4.3 Tag

```
string MB2D.Collectable.Tag [get]
```

Gets the items tag.

The tag.

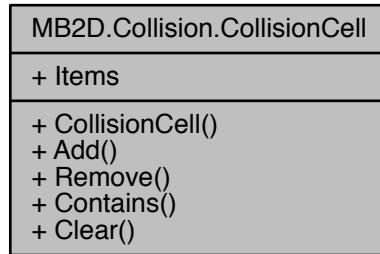
The documentation for this class was generated from the following file:

- MB2D/src/Inventory/Collectable.cs

5.3 MB2D.Collision.CollisionCell Class Reference

A cell used in a collision map to hold a linked list of all its contained entities

Collaboration diagram for MB2D.Collision.CollisionCell:



Public Member Functions

- [CollisionCell \(\)](#)
Initializes a new instance of the T:MB2D.Collision.CollisionCell class.
- void [Add \(Entity entity\)](#)
Adds an entity to the cell
- void [Remove \(Entity entity\)](#)
Removes a specific entity from the cell
- bool [Contains \(Entity entity\)](#)
Checks if the entity is inside the cell already
- void [Clear \(\)](#)
Clear the cell of all entities.

Properties

- [LinkedList< Entity > Items \[get\]](#)
Gets the list of this cells entities

5.3.1 Detailed Description

A cell used in a collision map to hold a linked list of all its contained entities

5.3.2 Constructor & Destructor Documentation

5.3.2.1 CollisionCell()

```
MB2D.Collision.CollisionCell.CollisionCell ( ) [inline]
```

Initializes a new instance of the T:MB2D.Collision.CollisionCell class.

5.3.3 Member Function Documentation

5.3.3.1 Add()

```
void MB2D.Collision.CollisionCell.Add (
    Entity entity ) [inline]
```

Adds an entity to the cell

Parameters

<i>entity</i>	Entity to add.
---------------	----------------

Here is the caller graph for this function:



5.3.3.2 Clear()

```
void MB2D.Collision.CollisionCell.Clear ( ) [inline]
```

Clear the cell of all entities.

5.3.3.3 Contains()

```
bool MB2D.Collision.CollisionCell.Contains (
    Entity entity ) [inline]
```

Checks if the entity is inside the cell already

Parameters

<i>entity</i>	Entity.
---------------	---------

5.3.3.4 Remove()

```
void MB2D.Collision.CollisionCell.Remove (
    Entity entity ) [inline]
```

Removes a specific entity from the cell

Parameters

<code>entity</code>	Entity to remove.
---------------------	-------------------

5.3.4 Property Documentation

5.3.4.1 Items

`LinkedList<Entity> MB2D.Collision.CollisionCell.Items [get]`

Gets the list of this cells entities

The entities.

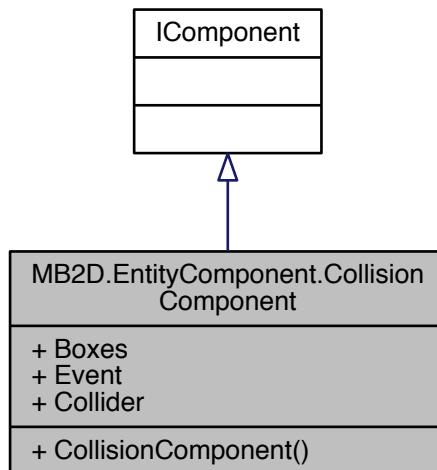
The documentation for this class was generated from the following file:

- MB2D/src/Collision/CollisionCell.cs

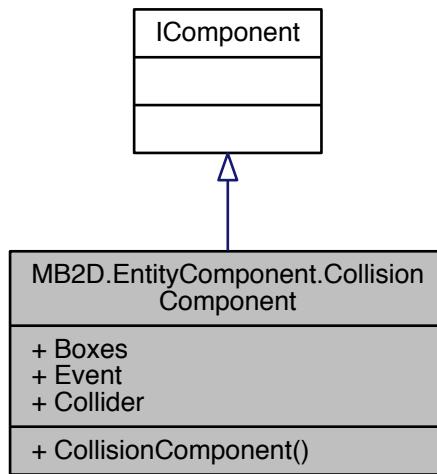
5.4 MB2D.EntityComponent.CollisionComponent Class Reference

Used for running collision detection on an [Entity](#)

Inheritance diagram for MB2D.EntityComponent.CollisionComponent:



Collaboration diagram for MB2D.EntityComponent.CollisionComponent:



Public Member Functions

- [CollisionComponent](#) (params RectangleF[] boxes)
Initializes a new instance of the T:MB2D.EntityComponent.CollisionComponent class with an array of its associated AABB's

Properties

- List< RectangleF > [Boxes](#) [get, set]
Gets or sets the list of bounding boxes used for collision detection.
- bool [Event](#) [get, set]
Gets or sets a value indicating whether this T:MB2D.EntityComponent.CollisionComponent has had a collision event this frame.
- Entity [Collider](#) [get, set]
Gets or sets the collider entity associated with the collision event.

5.4.1 Detailed Description

Used for running collision detection on an [Entity](#)

5.4.2 Constructor & Destructor Documentation

5.4.2.1 CollisionComponent()

```
MB2D.EntityComponent.CollisionComponent.CollisionComponent (
    params RectangleF [ ] boxes ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.CollisionComponent class with an array of its associated AABB's

Parameters

<i>boxes</i>	The bounding boxes used for detecting collisions.
--------------	---

5.4.3 Property Documentation

5.4.3.1 Boxes

```
List<RectangleF> MB2D.EntityComponent.CollisionComponent.Boxes [get], [set]
```

Gets or sets the list of bounding boxes used for collision detection.

The boxes.

5.4.3.2 Collider

```
Entity MB2D.EntityComponent.CollisionComponent.Collider [get], [set]
```

Gets or sets the collider entity associated with the collision event.

The collider.

5.4.3.3 Event

```
bool MB2D.EntityComponent.CollisionComponent.Event [get], [set]
```

Gets or sets a value indicating whether this T:MB2D.EntityComponent.CollisionComponent has had a collision event this frame.

true if an event occurred; otherwise, false.

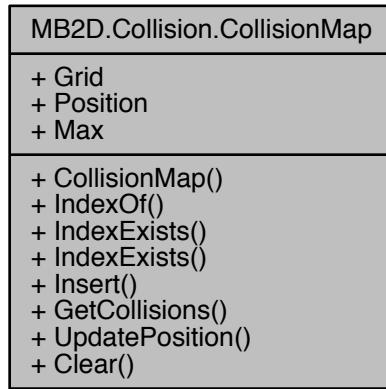
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Components/CollisionComponent.cs

5.5 MB2D.Collision.CollisionMap Class Reference

A 2D Grid that represents a particular space in the game world to check for collisions. Uses spatial indexing to determine where an entity will be located at any given time. For best results, the cellsize and overall size of the map should be tweaked for each individual game screen and environment.

Collaboration diagram for MB2D.Collision.CollisionMap:



Public Member Functions

- **CollisionMap** (int xMin, int xMax, int yMin, int yMax, int cellSize)
Initializes a new instance of the T:MB2D.Collision.CollisionMap class.
- **Point IndexOf** (Point position)
Indexes a world-based coordinate into the collision grid
- **bool IndexExists** (int x, int y)
Checks if a particular index exists in the grid
- **bool IndexExists** (Point index)
Checks if a particular index exists in the grid
- **void Insert** (Entity entity, CollisionComponent collision)
Inserts an entity and its associated collision component into the grid
- **List< Entity > GetCollisions** (Entity entity, CollisionComponent collision)
Gets a list of all entities located in the same cell/s as a specific single entity
- **void UpdatePosition** (int x, int y)
Updates the position of the collision grid.
- **void Clear** ()
Clears all non-empty cells of the grid from their entities

Properties

- **Grid Grid** [get]
Gets the geometric representation of the grid
- **Vector2 Position** [get]
Gets the current position of the grid.
- **Vector2 Max** [get]
Gets the upper bounds of the x and y coordinates in the grid

5.5.1 Detailed Description

A 2D Grid that represents a particular space in the game world to check for collisions. Uses spatial indexing to determine where an entity will be located at any given time. For best results, the cellsize and overall size of the map should be tweaked for each individual game screen and environment.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 CollisionMap()

```
MB2D.Collision.CollisionMap.CollisionMap (
    int xMin,
    int xMax,
    int yMin,
    int yMax,
    int cellSize ) [inline]
```

Initializes a new instance of the T:MB2D.Collision.CollisionMap class.

Parameters

<i>xMin</i>	The grids left most x coordinate.
<i>xMax</i>	Right most x coordinate.
<i>yMin</i>	Top most y coordinate.
<i>yMax</i>	Bottom most y coordinate.
<i>cellSize</i>	The size of each cell in the grid.

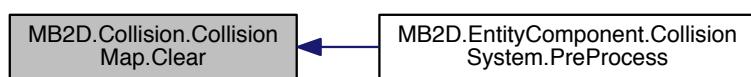
5.5.3 Member Function Documentation

5.5.3.1 Clear()

```
void MB2D.Collision.CollisionMap.Clear ( ) [inline]
```

Clears all non-empty cells of the grid from their entities

Here is the caller graph for this function:



5.5.3.2 GetCollisions()

```
List<Entity> MB2D.Collision.CollisionMap.GetCollisions (
    Entity entity,
    CollisionComponent collision ) [inline]
```

Gets a list of all entities located in the same cell/s as a specific single entity

Returns

The entities neighbours.

Parameters

<i>entity</i>	Entity to get collisions for.
<i>collision</i>	Collision component to use in checking.

Here is the caller graph for this function:



5.5.3.3 IndexExists() [1/2]

```
bool MB2D.Collision.CollisionMap.IndexExists (
    int x,
    int y ) [inline]
```

Checks if a particular index exists in the grid

Returns

true, if index exists, false otherwise.

Parameters

<i>x</i>	The x coordinate.
<i>y</i>	The y coordinate.

5.5.3.4 IndexExists() [2/2]

```
bool MB2D.Collision.CollisionMap.IndexExists (
```

```
Point index ) [inline]
```

Checks if a particular index exists in the grid

Returns

true, if index exists, false otherwise.

Parameters

<i>index</i>	Index to check.
--------------	-----------------

5.5.3.5 IndexOf()

```
Point MB2D.Collision.CollisionMap.IndexOf (
    Point position ) [inline]
```

Indexes a world-based coordinate into the collision grid

Returns

The grid-based position.

Parameters

<i>position</i>	World-based position to index.
-----------------	--------------------------------

5.5.3.6 Insert()

```
void MB2D.Collision.CollisionMap.Insert (
    Entity entity,
    CollisionComponent collision ) [inline]
```

Inserts an entity and its associated collision component into the grid

Parameters

<i>entity</i>	Entity to insert.
<i>collision</i>	The entities collision component.

Here is the call graph for this function:



Here is the caller graph for this function:



5.5.3.7 UpdatePosition()

```
void MB2D.Collision.CollisionMap.UpdatePosition (
    int x,
    int y ) [inline]
```

Updates the position of the collision grid.

Parameters

x	The x coordinate.
y	The y coordinate.

5.5.4 Property Documentation

5.5.4.1 Grid

`Grid` `MB2D.Collision.CollisionMap.Grid` [get]

Gets the geometric representation of the grid

The grid.

5.5.4.2 Max

```
Vector2 MB2D.Collision.CollisionMap.Max [get]
```

Gets the upper bounds of the x and y coordinates in the grid

The max coordinates.

5.5.4.3 Position

```
Vector2 MB2D.Collision.CollisionMap.Position [get]
```

Gets the current position of the grid.

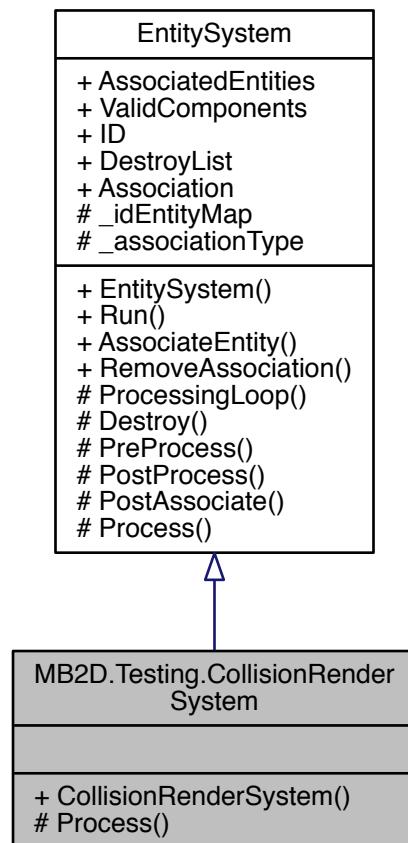
The position.

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

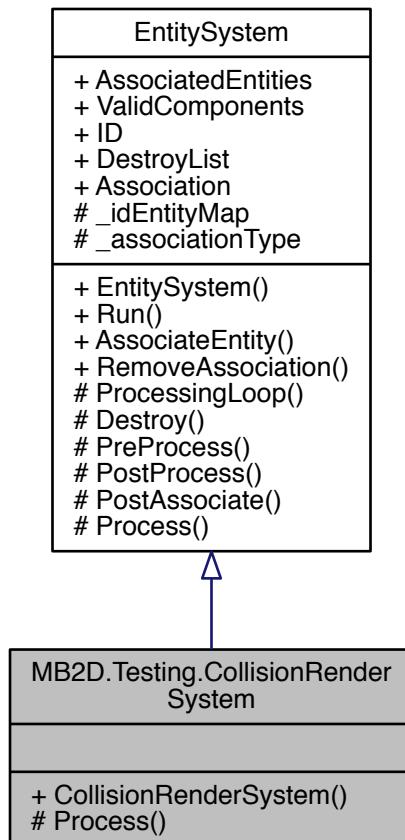
- MB2D/src/Collision/CollisionMap.cs

5.6 MB2D.Testing.CollisionRenderSystem Class Reference

Inheritance diagram for MB2D.Testing.CollisionRenderSystem:



Collaboration diagram for MB2D.Testing.CollisionRenderSystem:



Public Member Functions

- **CollisionRenderSystem** (SpriteBatch spriteBatch)

Protected Member Functions

- override void **Process** (Entity entity)
Executes this systems logic on a single entity

Additional Inherited Members

5.6.1 Member Function Documentation

5.6.1.1 Process()

```
override void MB2D.Testing.CollisionRenderSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
```

Executes this systems logic on a single entity

Parameters

<code>entity</code>	Entity to operate on
---------------------	----------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

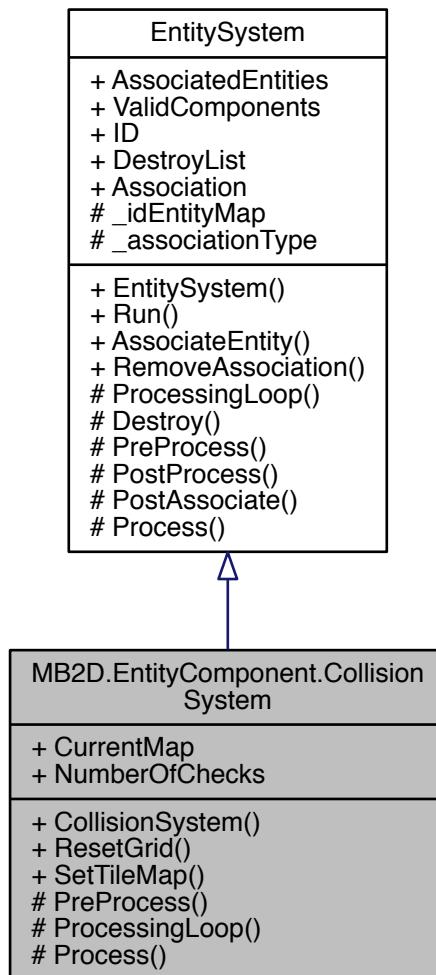
The documentation for this class was generated from the following file:

- MB2D/src/Test/CollisionRenderSystem.cs

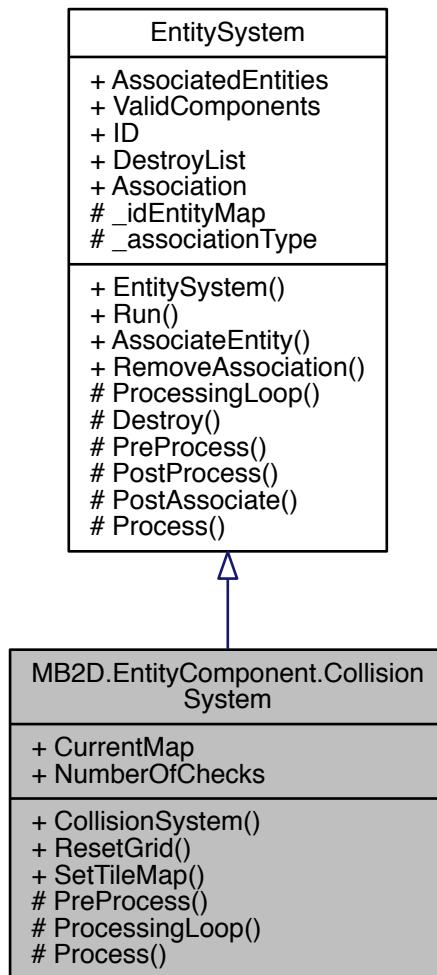
5.7 MB2D.EntityComponent.CollisionSystem Class Reference

Checks collisions. Uses a spatial indexing grid for broad-phase collision checking and AABB checks for narrow phase

Inheritance diagram for MB2D.EntityComponent.CollisionSystem:



Collaboration diagram for MB2D.EntityComponent.CollisionSystem:



Public Member Functions

- **[CollisionSystem \(\)](#)**
Initializes a new instance of the T:MB2D.EntityComponent.CollisionSystem class.
- void **[ResetGrid \(int xMin, int xMax, int yMin, int yMax, int cellSize\)](#)**
Resets the grid position in the world.
- void **[SetTileMap \(TileMap tileMap\)](#)**
Sets the current tile map to check for collisions

Protected Member Functions

- override void **[PreProcess \(\)](#)**
Clears the collision grid and inserts all entities before checking collisions

- override void [ProcessingLoop \(\)](#)
Override. Only processes entities with movement components. Still considers static entities, but only as possible neighbours.
- override void [Process \(Entity entity\)](#)
Checks all collisions within the entities known collision cells

Properties

- [CollisionMap CurrentMap \[get\]](#)
Gets the current collision map.
- int [NumberOfChecks \[get\]](#)
Gets the number of collision checks made last frame. Used for debugging.

Additional Inherited Members

5.7.1 Detailed Description

Checks collisions. Uses a spatial indexing grid for broad-phase collision checking and AABB checks for narrow phase

5.7.2 Constructor & Destructor Documentation

5.7.2.1 CollisionSystem()

```
MB2D.EntityComponent.CollisionSystem.CollisionSystem ( ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.CollisionSystem class.

5.7.3 Member Function Documentation

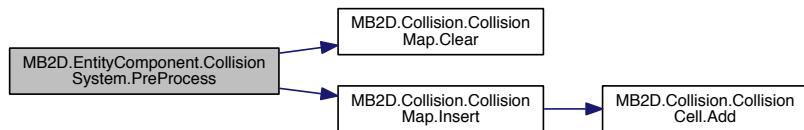
5.7.3.1 PreProcess()

```
override void MB2D.EntityComponent.CollisionSystem.PreProcess ( ) [inline], [protected], [virtual]
```

Clears the collision grid and inserts all entities before checking collisions

Reimplemented from [MB2D.EntityComponent.EntitySystem](#).

Here is the call graph for this function:



5.7.3.2 Process()

```
override void MB2D.EntityComponent.CollisionSystem.Process (
    Entity entity) [inline], [protected], [virtual]
```

Checks all collisions within the entities known collision cells

Parameters

<i>entity</i>	Entity to check.
---------------	------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

Here is the call graph for this function:

**5.7.3.3 ProcessingLoop()**

```
override void MB2D.EntityComponent.CollisionSystem.ProcessingLoop () [inline], [protected], [virtual]
```

Override. Only processes entities with movement components. Still considers static entities, but only as possible neighbours.

Reimplemented from [MB2D.EntityComponent.EntitySystem](#).

5.7.3.4 ResetGrid()

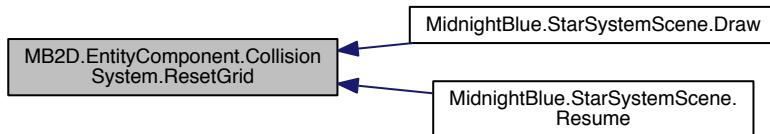
```
void MB2D.EntityComponent.CollisionSystem.ResetGrid (
    int xMin,
    int xMax,
    int yMin,
    int yMax,
    int cellSize ) [inline]
```

Resets the grid position in the world.

Parameters

<i>xMin</i>	The grids left most x coordinate.
<i>xMax</i>	Right most x coordinae.
<i>yMin</i>	Top most y coordinate.
<i>yMax</i>	Bottom most y coordinate.
<i>cellSize</i>	The size of each cell in the grid.

Here is the caller graph for this function:



5.7.3.5 SetTileMap()

```
void MB2D.EntityComponent.CollisionSystem.SetTileMap (
    TileMap tileMap ) [inline]
```

Sets the current tile map to check for collisions

Parameters

<code>tileMap</code>	Tile map.
----------------------	-----------

Here is the caller graph for this function:



5.7.4 Property Documentation

5.7.4.1 CurrentMap

```
CollisionMap MB2D.EntityComponent.CollisionSystem.CurrentMap [get]
```

Gets the current collision map.

The current map.

5.7.4.2 NumberOfChecks

```
int MB2D.EntityComponent.CollisionSystem.NumberOfChecks [get]
```

Gets the number of collision checks made last frame. Used for debugging.

The number of collision checks.

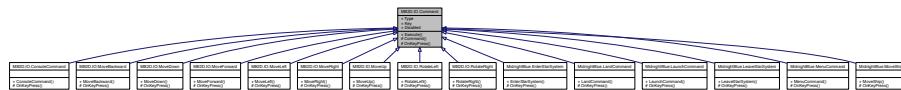
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/CollisionSystem.cs

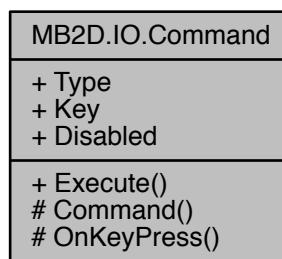
5.8 MB2D.IO.Command Class Reference

Executes an action associated with a specific key

Inheritance diagram for MB2D.IO.Command:



Collaboration diagram for MB2D.IO.Command:



Public Member Functions

- bool Execute (Entity e=null)
Executes the specific command on the entity parameter

Protected Member Functions

- **Command** (`Keys key, CommandType commandType`)
Initializes a new instance of the T:MidnightBlue.Command class.
 - abstract void **OnKeyPress** (`Entity e=null`)
Defines the logic to perform when operating on a given entity

Properties

- **CommandType** `Type` [get, set]
Gets or sets the trigger type of the command.
 - **Keys** `Key` [get]
Gets the keycode associated with the command.
 - `bool Disabled` [get, set]

5.8.1 Detailed Description

Executes an action associated with a specific key

5.8.2 Constructor & Destructor Documentation

5.8.2.1 Command()

```
MB2D.IO.Command.Command (
    Keys key,
    CommandType commandType ) [inline], [protected]
```

Initializes a new instance of the T:MidnightBlue.Command class.

Parameters

key	Key to associate with the command
commandType	Trigger or hold command

5.8.3 Member Function Documentation

5.8.3.1 Execute()

```
bool MB2D.IO.Command.Execute (
    Entity e = null ) [inline]
```

Executes the specific command on the entity parameter

Parameters

e	Entity to operate on. Optional
---	--------------------------------

5.8.3.2 OnKeyPress()

```
abstract void MB2D.IO.Command.OnKeyPress (
    Entity e = null ) [protected], [pure virtual]
```

Defines the logic to perform when operating on a given entity

Parameters

e	Entity to operate on
---	----------------------

Implemented in [MB2D.IO.RotateLeft](#), [MB2D.IO.RotateRight](#), [MB2D.IO.MoveBackward](#), [MidnightBlue.LeaveStarSystem](#), [MB2D.IO.MoveForward](#), [MidnightBlue.EnterStarSystem](#), [MB2D.IO.MoveLeft](#), [MidnightBlue.Launch](#)

[Command](#), [MB2D.IO.MoveDown](#), [MidnightBlue.LandCommand](#), [MB2D.IO.MoveRight](#), [MidnightBlue.MoveShip](#), [MB2D.IO.MoveUp](#), [MidnightBlue.MenuCommand](#), and [MB2D.IO.ConsoleCommand](#).

5.8.4 Property Documentation

5.8.4.1 Key

`Keys MB2D.IO.Command.Key [get]`

Gets the keycode associated with the command.

The key code

5.8.4.2 Type

`CommandType MB2D.IO.Command.Type [get], [set]`

Gets or sets the trigger type of the command.

The command type

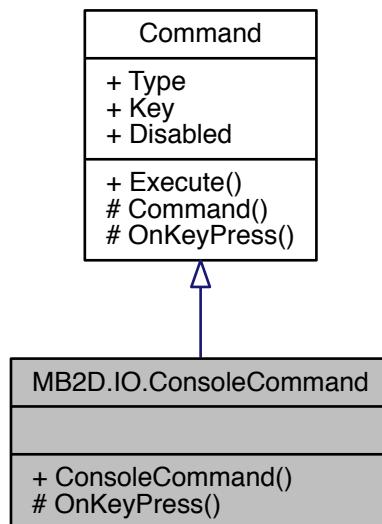
The documentation for this class was generated from the following file:

- [MB2D/src/Input/Command.cs](#)

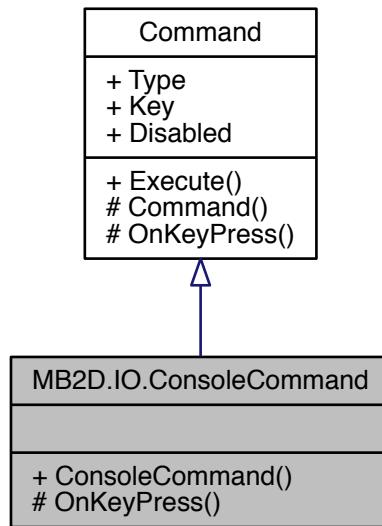
5.9 MB2D.IO.ConsoleCommand Class Reference

Shows or hides the debug console

Inheritance diagram for MB2D.IO.ConsoleCommand:



Collaboration diagram for MB2D.IO.ConsoleCommand:



Public Member Functions

- `ConsoleCommand (Keys key, CommandType type)`
Initializes a new instance of the T:MB2D.IO.ConsoleCommand class.

Protected Member Functions

- `override void OnKeyPress (Entity e=null)`
Toggles the debug console open/closed

Additional Inherited Members

5.9.1 Detailed Description

Shows or hides the debug console

5.9.2 Constructor & Destructor Documentation

5.9.2.1 ConsoleCommand()

```
MB2D.IO.ConsoleCommand.ConsoleCommand (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MB2D.IO.ConsoleCommand class.

Parameters

<i>key</i>	Key to assign the command to.
<i>type</i>	Type of command trigger.

5.9.3 Member Function Documentation**5.9.3.1 OnKeyPress()**

```
override void MB2D.IO.ConsoleCommand.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

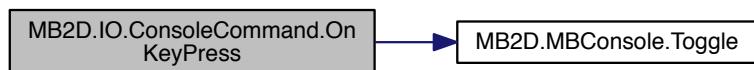
Toggles the debug console open/closed

Parameters

<i>e</i>	Entity with the controller component. Unused.
----------	---

Implements [MB2D.IO.Command](#).

Here is the call graph for this function:



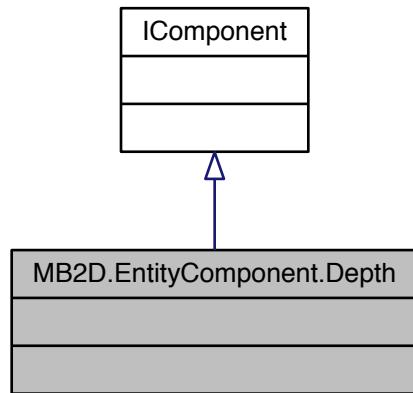
The documentation for this class was generated from the following file:

- MB2D/src/Input/ConsoleCommand.cs

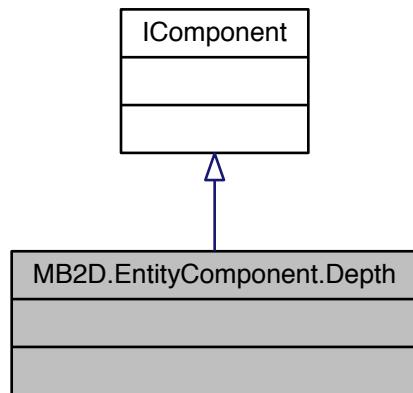
5.10 MB2D.EntityComponent.Depth Class Reference

A tag class used to define an entity that should be draw sorted according to its current z-index

Inheritance diagram for MB2D.EntityComponent.Depth:



Collaboration diagram for MB2D.EntityComponent.Depth:



5.10.1 Detailed Description

A tag class used to define an entity that should be draw sorted according to its current z-index

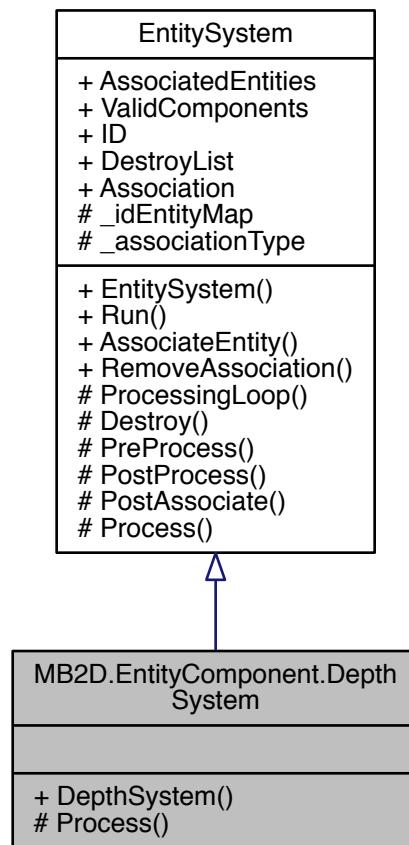
The documentation for this class was generated from the following file:

- `MB2D/src/EntityComponent/Components/Depth.cs`

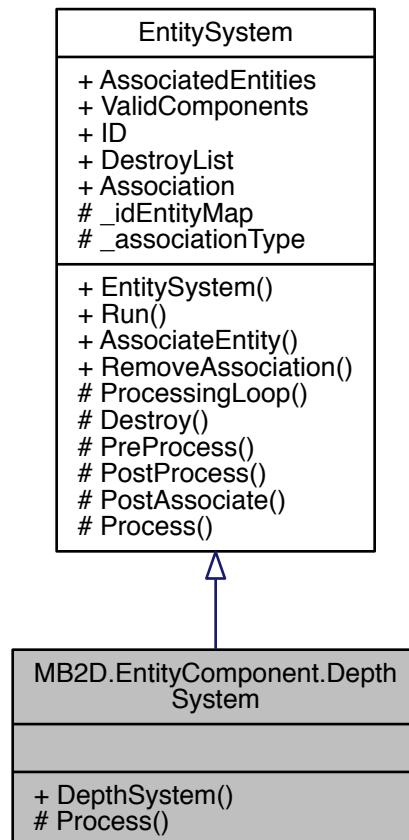
5.11 MB2D.EntityComponent.DepthSystem Class Reference

Changes an entities z-index based on the y coordinate of the top of their sprite

Inheritance diagram for MB2D.EntityComponent.DepthSystem:



Collaboration diagram for MB2D.EntityComponent.DepthSystem:



Public Member Functions

- [DepthSystem \(\)](#)
Initializes a new instance of the T:MB2D.EntityComponent.DepthSystem class.

Protected Member Functions

- `override void Process (Entity entity)`
Changes the z index based on the y coordinate of the entities bounds top

Additional Inherited Members

5.11.1 Detailed Description

Changes an entities z-index based on the y coordinate of the top of their sprite

5.11.2 Constructor & Destructor Documentation

5.11.2.1 DepthSystem()

```
MB2D.EntityComponent.DepthSystem.DepthSystem ( ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.DepthSystem class.

5.11.3 Member Function Documentation

5.11.3.1 Process()

```
override void MB2D.EntityComponent.DepthSystem.Process (
    Entity entity) [inline], [protected], [virtual]
```

Changes the z index based on the y coordinate of the entities bounds top

Parameters

entity	Entity to process.
--------	--------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

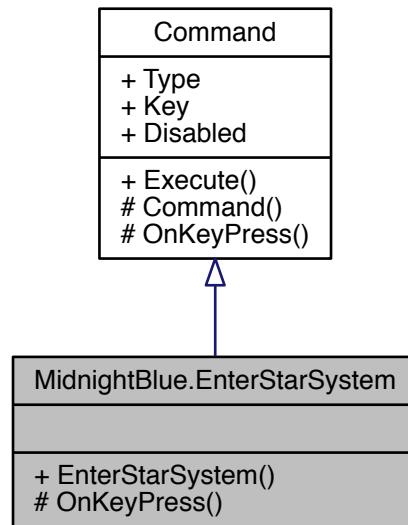
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/DepthSystem.cs

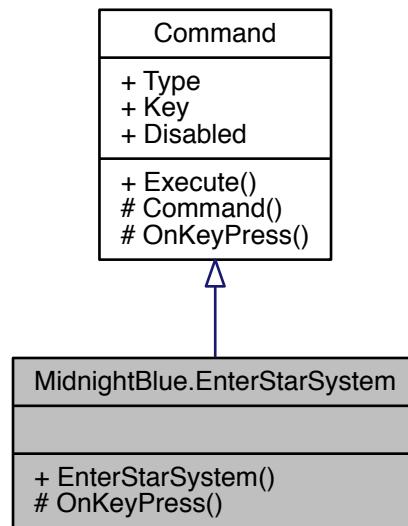
5.12 MidnightBlue.EnterStarSystem Class Reference

Enters a star system scene from the galaxy view

Inheritance diagram for MidnightBlue.EnterStarSystem:



Collaboration diagram for `MidnightBlue.EnterStarSystem`:



Public Member Functions

- `EnterStarSystem (Keys key, CommandType type)`

Initializes a new instance of the T:MidnightBlue.EnterStarSystem class.

Protected Member Functions

- override void [OnKeyPress \(Entity e\)](#)
Enters the collided with star system on keypress

Additional Inherited Members

5.12.1 Detailed Description

Enters a star system scene from the galaxy view

5.12.2 Constructor & Destructor Documentation

5.12.2.1 EnterStarSystem()

```
MidnightBlue.EnterStarSystem.EnterStarSystem (
    Keys key,
    CommandType type )  [inline]
```

Initializes a new instance of the T:MidnightBlue.EnterStarSystem class.

Parameters

key	Key to assign to.
type	Trigger type.

5.12.3 Member Function Documentation

5.12.3.1 OnKeyPress()

```
override void MidnightBlue.EnterStarSystem.OnKeyPress (
    Entity e )  [inline], [protected], [virtual]
```

Enters the collided with star system on keypress

Parameters

e	E.
---	----

Implements [MB2D.IO.Command](#).

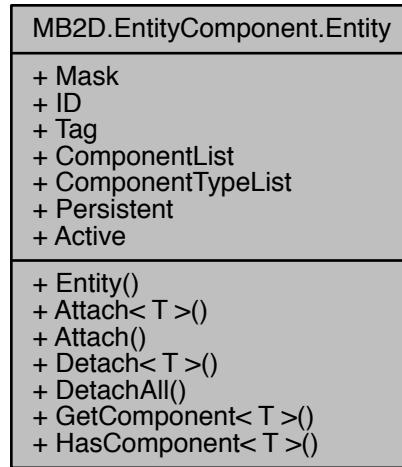
The documentation for this class was generated from the following file:

- Shared/src/Game/Commands/ShipCommands.cs

5.13 MB2D.EntityComponent.Entity Class Reference

Represents a tagged and id'd container for components that can be operated on by systems.

Collaboration diagram for MB2D.EntityComponent.Entity:



Public Member Functions

- **Entity** ([EntityMap](#) container, string tag="")

Initializes a new instance of the T:MB2D.EntityComponent.Entity class.
- **IComponent Attach< T >** (params object[] args)

Attaches a new component to the entity.
- void **Attach** ([IComponent](#) component)

Attaches a new component to the entity.
- void **Detach< T >** ()

Detaches a specific component from the entity
- void **DetachAll** ()

Detaches all of the entities attached components.
- T **GetComponent< T >** ()

Queries the entity to see if it has a component attached and returns it if it does
- bool **HasComponent< T >** ()

Checks if an entity has a specific component attached

Properties

- ulong **Mask** [get, set]

Gets and sets the entities component mask.
- ulong **ID** [get, set]

Gets and sets the entities Globally Unique ID in the EntityMap

- string **Tag** [get]
Gets this entities tagname
- Dictionary< Type, **IComponent** >.ValueCollection **ComponentList** [get]
Gets the list if components attached to this entity.
- Dictionary< Type, **IComponent** >.KeyCollection **ComponentTypeList** [get]
Gets the types of components this entity has attached
- bool **Persistent** [get, set]
Gets or sets a value indicating whether this T:MB2D.EntityComponent.Entity is persistant in its parent T:MB2D.EntityComponent.EntityMap.
- bool **Active** [get, set]
Gets or sets a value indicating whether this T:MB2D.EntityComponent.Entity is active. Inactive entities are skipped over in each EntitySystems Process() method but aren't destroyed. Allowing semi-persistant entities.

5.13.1 Detailed Description

Represents a tagged and id'd container for components that can be operated on by systems.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 Entity()

```
MB2D.EntityComponent.Entity.Entity (
    EntityMap container,
    string tag = "" ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.Entity class.

Parameters

<i>container</i>	The entities parent EntityMap
<i>tag</i>	Tagname to give the entity

5.13.3 Member Function Documentation

5.13.3.1 Attach()

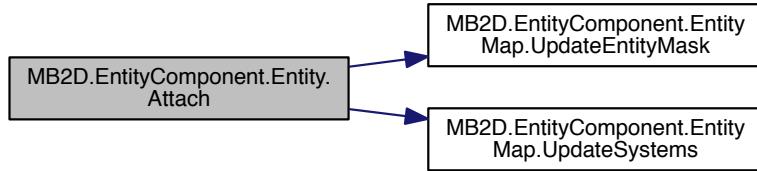
```
void MB2D.EntityComponent.Entity.Attach (
    IComponent component ) [inline]
```

Attaches a new component to the entity.

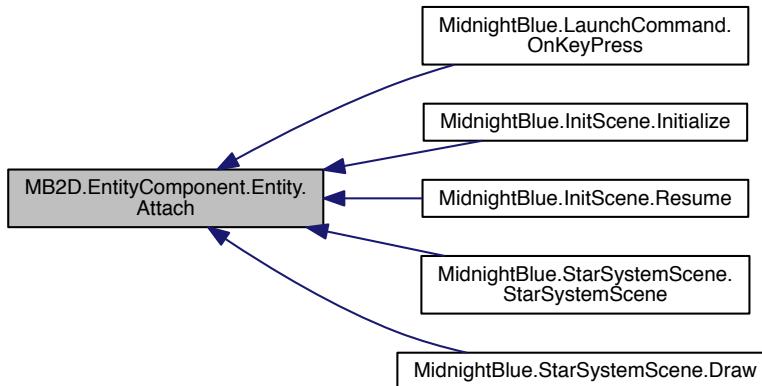
Parameters

<i>component</i>	Pre constructed component to add
------------------	----------------------------------

Here is the call graph for this function:



Here is the caller graph for this function:



5.13.3.2 Attach< T >()

```
IComponent MB2D.EntityComponent.Entity.Attach< T > (
    params object[] args) [inline]
```

Attaches a new component to the entity.

Parameters

<code>args</code>	The components constructor arguments
-------------------	--------------------------------------

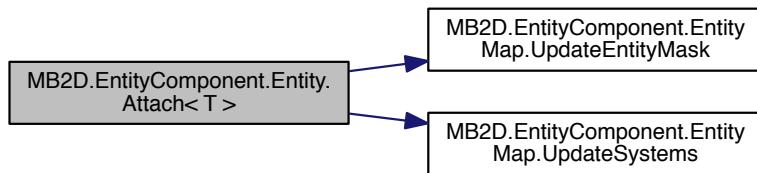
Template Parameters

<code>T</code>	Type of component to attach
----------------	-----------------------------

Type Constraints

T : IComponent

Here is the call graph for this function:



5.13.3.3 Detach< T >()

```
void MB2D.EntityComponent.Entity.Detach< T > ( ) [inline]
```

Detaches a specific component from the entity

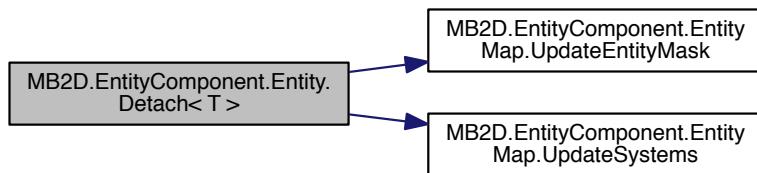
Template Parameters

<i>T</i>	The type of component to detach.
----------	----------------------------------

Type Constraints

T : IComponent

Here is the call graph for this function:

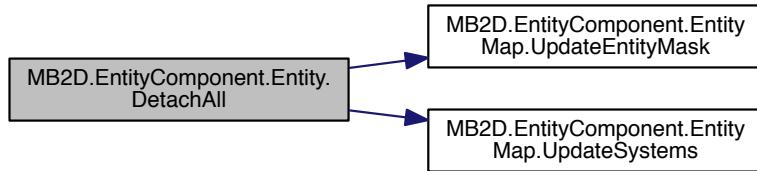


5.13.3.4 DetachAll()

```
void MB2D.EntityComponent.Entity.DetachAll ( ) [inline]
```

Detaches all of the entities attached components.

Here is the call graph for this function:



Here is the caller graph for this function:



5.13.3.5 GetComponent< T >()

`T MB2D.EntityComponent.Entity.GetComponent< T > () [inline]`

Queries the entity to see if it has a component attached and returns it if it does

Returns

The component if the entity has it attached, null otherwise

Template Parameters

<code>T</code>	Component to query the entity for.
----------------	------------------------------------

Type Constraints

`T : IComponent`

5.13.3.6 HasComponent< T >()

`bool MB2D.EntityComponent.Entity.HasComponent< T > () [inline]`

Checks if an entity has a specific component attached

Returns

true, if component is attached, false otherwise.

Template Parameters

<i>T</i>	The type of component to check.
----------	---------------------------------

Type Constraints

T : *IComponent*

5.13.4 Property Documentation

5.13.4.1 Active

```
bool MB2D.EntityComponent.Entity.Active [get], [set]
```

Gets or sets a value indicating whether this T:MB2D.EntityComponent.Entity is active. Inactive entities are skipped over in each EntitySystems Process() method but aren't destroyed. Allowing semi-persistent entities.

true if the entity is active; otherwise, false.

5.13.4.2 ComponentList

```
Dictionary<Type, IComponent>.ValueCollection MB2D.EntityComponent.Entity.ComponentList [get]
```

Gets the list of components attached to this entity.

The component list.

5.13.4.3 ComponentTypeList

```
Dictionary<Type, IComponent>.KeyCollection MB2D.EntityComponent.Entity.ComponentTypeList  
[get]
```

Gets the types of components this entity has attached

The component type list.

5.13.4.4 ID

```
ulong MB2D.EntityComponent.Entity.ID [get], [set]
```

Gets and sets the entity's Globally Unique ID in the [EntityMap](#)

GUID

5.13.4.5 Mask

```
ulong MB2D.EntityComponent.Entity.Mask [get], [set]
```

Gets and sets the entities component mask.

The component mask.

5.13.4.6 Persistent

```
bool MB2D.EntityComponent.Entity.Persistent [get], [set]
```

Gets or sets a value indicating whether this T:MB2D.EntityComponent.Entity is persistant in its parent T:MB2D.EntityComponent.EntityMap.

true if persistant; otherwise, false.

5.13.4.7 Tag

```
string MB2D.EntityComponent.Entity.Tag [get]
```

Gets this entities tagname

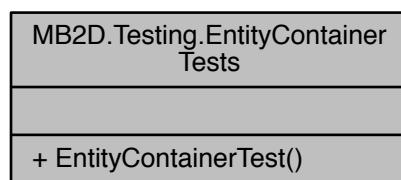
The tagname

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

- MB2D/src/EntityComponent/Entity.cs

5.14 MB2D.Testing.EntityContainerTests Struct Reference

Collaboration diagram for MB2D.Testing.EntityContainerTests:



Static Public Member Functions

- static void **EntityContainerTest** (params string[] args)

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

- MB2D/src/Test/EntityContainerTests.cs

5.15 MB2D.EntityComponent.EntityMap Class Reference

Maps entities, systems and components to one another and provides querying and updating access to all elements

Collaboration diagram for MB2D.EntityComponent.EntityMap:

MB2D.EntityComponent.EntityMap
+ this[string key] + EntityCount + NextID + NextMask
+ EntityMap() + EntityMap() + AddComponent< T >() + AddComponent() + AddSystem< T >() + AddEntity() + UpdateEntityMask() + UpdateSystems() + CreateEntity() + GetComponentID< T >() and 6 more...

Public Member Functions

- **EntityMap ()**
Initializes a new instance of the T:MB2D.EntityComponent.EntityMap class.
- **EntityMap (EntityMap map)**
Initializes a new instance of the T:MB2D.EntityComponent.EntityMap class. Uses an existing EntityMap to copy all registered systems and components as well as any persistant Entities.
- void **AddComponent< T > ()**
Registers a new component type to the EntityMap
- void **AddComponent (Type componentType)**
Registers a new component type to the EntityMap
- void **AddSystem< T > (params object[] args)**
Registers a new EntitySystem to the map

- void [AddEntity \(Entity entity\)](#)
Adds a created entity to this map
- void [UpdateEntityMask \(Entity entity\)](#)
Updates a specific entities component mask. Use after registering new components or systems.
- void [UpdateSystems \(Entity entity\)](#)
Updates each systems associated entity list, adding the specified Entity. Use after creating a new Entity and adding it manually
- [Entity CreateEntity \(string tag=""\)](#)
Creates a new Entity with the given tag in this map. Auto-Registers the entity with all systems and updates its mask.
- ulong [GetComponentID< T > \(\)](#)
Gets the id of a specified component type if it exists.
- [EntitySystem GetSystem< T > \(\)](#)
Retrieves a pre-registered system from the map
- List< [Entity](#) > [EntitiesWithComponent< T > \(\)](#)
- void [Clear \(\)](#)
Clears all entities from this map except for any marked as persistant.
- void [Reset \(\)](#)
- void [MakeBlueprint \(string id, Action< Entity > buildFunction\)](#)
- void [UseBlueprint \(string name, Entity entity\)](#)

Properties

- [Entity this\[string key\] \[get\]](#)
Gets the T:MB2D.EntityComponent.Entity with the specified tag if it exists; null otherwise
- int [EntityCount \[get\]](#)
Gets the number of entities in the map.
- ulong [NextID \[get\]](#)
Auto-increments the last generated GUID and retrieves the result

5.15.1 Detailed Description

Maps entities, systems and components to one another and provides querying and updating access to all elements

5.15.2 Constructor & Destructor Documentation

5.15.2.1 EntityMap() [1/2]

```
MB2D.EntityComponent.EntityMap.EntityMap ( ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.EntityMap class.

5.15.2.2 EntityMap() [2/2]

```
MB2D.EntityComponent.EntityMap.EntityMap (
    EntityMap map ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.EntityMap class. Uses an existing EntityMap to copy all registered systems and components as well as any persistant Entities.

Parameters

<i>map</i>	EntityMap to copy from
------------	--

5.15.3 Member Function Documentation

5.15.3.1 AddComponent()

```
void MB2D.EntityComponent.EntityMap.AddComponent (
    Type componentType ) [inline]
```

Registers a new component type to the [EntityMap](#)

Parameters

<i>componentType</i>	Type of component to register
----------------------	-------------------------------

5.15.3.2 AddComponent< T >()

```
void MB2D.EntityComponent.EntityMap.AddComponent< T > ( ) [inline]
```

Registers a new component type to the [EntityMap](#)

Template Parameters

<i>T</i>	Type of component to register
----------	-------------------------------

Type Constraints

T : *IComponent*

5.15.3.3 AddEntity()

```
void MB2D.EntityComponent.EntityMap.AddEntity (
    Entity entity ) [inline]
```

Adds a created entity to this map

Parameters

<i>entity</i>	Entity to add
---------------	---------------

5.15.3.4 AddSystem< T >()

```
void MB2D.EntityComponent.EntityMap.AddSystem< T > (
    params object [] args ) [inline]
```

Registers a new [EntitySystem](#) to the map

Template Parameters

<i>T</i>	EntitySystem type to add
----------	--

Type Constraints

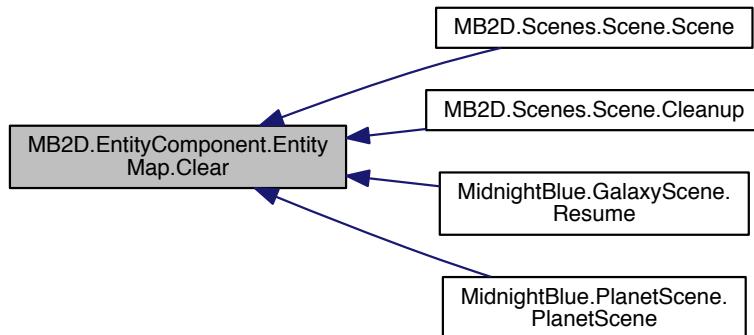
T : [EntitySystem](#)

5.15.3.5 Clear()

```
void MB2D.EntityComponent.EntityMap.Clear () [inline]
```

Clears all entities from this map except for any marked as persistant.

Here is the caller graph for this function:



5.15.3.6 CreateEntity()

```
Entity MB2D.EntityComponent.EntityMap.CreateEntity (
    string tag = "" ) [inline]
```

Creates a new [Entity](#) with the given tag in this map. Auto-Registers the entity with all systems and updates its mask.

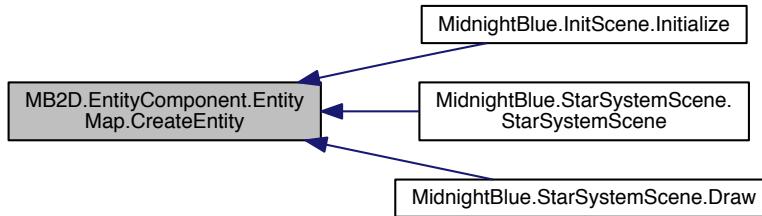
Returns

The created entity

Parameters

<i>tag</i>	Tagname to give the entity
------------	----------------------------

Here is the caller graph for this function:

**5.15.3.7 GetComponentID< T >()**

ulong MB2D.EntityComponent.EntityMap.GetComponentID< T > () [inline]

Gets the id of a specified component type if it exists.

Returns

The component id mask.

Template Parameters

<i>T</i>	Type of component to query for.
----------	---------------------------------

Type Constraints

T : *IComponent*

5.15.3.8 GetSystem< T >()

EntitySystem MB2D.EntityComponent.EntityMap.GetSystem< T > () [inline]

Retrieves a pre-registered system from the map

Returns

The system if it exists; null otherwise

Template Parameters

<i>T</i>	Type of system to retrieve.
----------	-----------------------------

Type Constraints

T : **EntitySystem**

5.15.3.9 UpdateEntityMask()

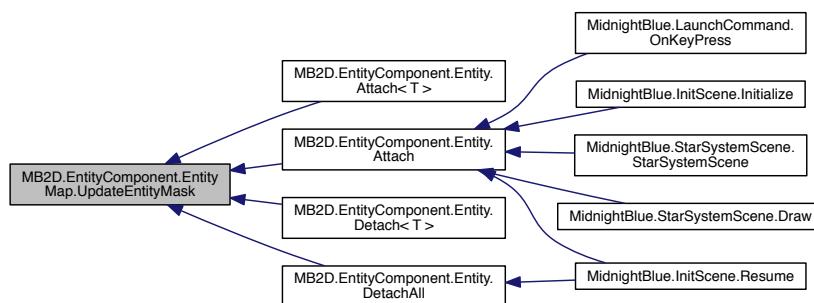
```
void MB2D.EntityComponent.EntityMap.UpdateEntityMask (
    Entity entity) [inline]
```

Updates a specific entities component mask. Use after registering new components or systems.

Parameters

<i>entity</i>	Entity to update
---------------	------------------

Here is the caller graph for this function:



5.15.3.10 UpdateSystems()

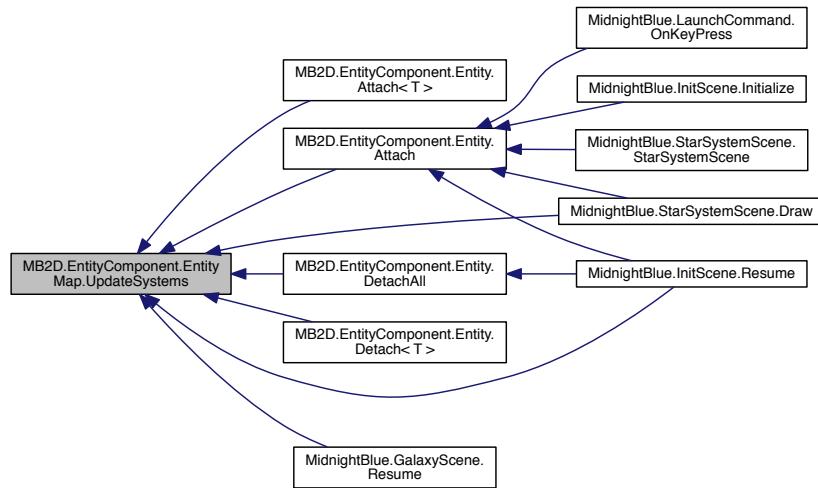
```
void MB2D.EntityComponent.EntityMap.UpdateSystems (
    Entity entity) [inline]
```

Updates each systems associated entity list, adding the specified Entity. Use after creating a new Entity and adding it manually

Parameters

<i>entity</i>	Entity to track in each system
---------------	--------------------------------

Here is the caller graph for this function:



5.15.4 Property Documentation

5.15.4.1 EntityCount

```
int MB2D.EntityComponent.EntityMap.EntityCount [get]
```

Gets the number of entities in the map.

The entity count.

5.15.4.2 NextID

```
ulong MB2D.EntityComponent.EntityMap.NextID [get]
```

Auto-increments the last generated GUID and retrieves the result

The next identifier.

5.15.4.3 this[string key]

```
Entity MB2D.EntityComponent.EntityMap.this[string key] [get]
```

Gets the T:MB2D.EntityComponent.Entity with the specified tag if it exists; null otherwise

Parameters

<i>key</i>	Tagname of the entity to retrieve.
------------	------------------------------------

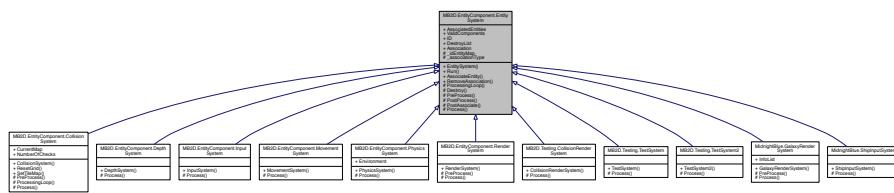
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/EntityMap.cs

5.16 MB2D.EntityComponent.EntitySystem Class Reference

Performs logic on an entity.

Inheritance diagram for MB2D.EntityComponent.EntitySystem:



Collaboration diagram for MB2D.EntityComponent.EntitySystem:



Public Member Functions

- [EntitySystem](#) (params Type[] components)

Initializes a new instance of the T:MB2D.EntityComponent.EntitySystem class. Checks if the passed in types are valid components and if not, all components are deregistered leaving a system that knows about nothing and can't operate on any entities.

- void [Run \(\)](#)
Runs this systems [Process\(\)](#) method on all entities
- void [AssociateEntity \(Entity entity\)](#)
Associates a new entity with this system.
- void [RemoveAssociation \(Entity entity\)](#)
Decouples an association with this system

Protected Member Functions

- virtual void [ProcessingLoop \(\)](#)
Executes [Process\(\)](#) on all AssociatedEntities.
- void [Destroy \(Entity entity\)](#)
Adds the specific entity to the destroy list to be cleaned up the next time a system is run
- virtual void [PreProcess \(\)](#)
Used to setup any data needed before processing all entities, such as sorting a list ahead of time
- virtual void [PostProcess \(\)](#)
Used to cleanup or execute any teardown logic needed
- virtual void [PostAssociate \(Entity entity\)](#)
Used to define any logic or extra data needed after an entity is associated with the system.
- abstract void [Process \(Entity entity\)](#)
Executes this systems logic on a single entity

Protected Attributes

- Dictionary< ulong, Entity > [_idEntityMap](#)
All GUID's of entities this system knows about
- EntityAssociation [_associationType](#)
Defines if this system should be Loose or Strict

Properties

- List< Entity > [AssociatedEntities](#) [get, set]
Gets or sets the associated entities list.
- List< Type > [ValidComponents](#) [get]
Gets the list of components this system is interested in
- ulong [ID](#) [get, set]
Gets or sets this systems ID mask
- List< Entity > [DestroyList](#) [get]
Gets the destroy list.
- EntityAssociation [Association](#) [get]
Gets the systems association level.

5.16.1 Detailed Description

Performs logic on an entity.

5.16.2 Constructor & Destructor Documentation

5.16.2.1 EntitySystem()

```
MB2D.EntityComponent.EntitySystem.EntitySystem (
    params Type [] components ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.EntitySystem class. Checks if the passed in types are valid components and if not, all components are deregistered leaving a system that knows about nothing and can't operate on any entities.

Parameters

<i>components</i>	Components this system is interested in
-------------------	---

5.16.3 Member Function Documentation

5.16.3.1 AssociateEntity()

```
void MB2D.EntityComponent.EntitySystem.AssociateEntity (
    Entity entity ) [inline]
```

Associates a new entity with this system.

Parameters

<i>entity</i>	Entity to associate
---------------	---------------------

5.16.3.2 Destroy()

```
void MB2D.EntityComponent.EntitySystem.Destroy (
    Entity entity ) [inline], [protected]
```

Adds the specific entity to the destroy list to be cleaned up the next time a system is run

Parameters

<i>entity</i>	Entity to destroy.
---------------	--------------------

5.16.3.3 PostAssociate()

```
virtual void MB2D.EntityComponent.EntitySystem.PostAssociate (
    Entity entity ) [inline], [protected], [virtual]
```

Used to define any logic or extra data needed after an entity is associated with the system.

Parameters

<code>entity</code>	<code>Entity</code> .
---------------------	-----------------------

5.16.3.4 PostProcess()

```
virtual void MB2D.EntityComponent.EntitySystem.PostProcess ( ) [inline], [protected], [virtual]
```

Used to cleanup or execute any teardown logic needed

5.16.3.5 PreProcess()

```
virtual void MB2D.EntityComponent.EntitySystem.PreProcess ( ) [inline], [protected], [virtual]
```

Used to setup any data needed before processing all entities, such as sorting a list ahead of time

Reimplemented in [MB2D.EntityComponent.CollisionSystem](#), [MidnightBlue.GalaxyRenderSystem](#), and [MB2D.EntityComponent.RenderSystem](#).

5.16.3.6 Process()

```
abstract void MB2D.EntityComponent.EntitySystem.Process (
    Entity entity) [protected], [pure virtual]
```

Executes this systems logic on a single entity

Parameters

<code>entity</code>	<code>Entity</code> to operate on
---------------------	-----------------------------------

Implemented in [MB2D.EntityComponent.CollisionSystem](#), [MidnightBlue.GalaxyRenderSystem](#), [MB2D.EntityComponent.RenderSystem](#), [MB2D.Testing.TestSystem2](#), [MB2D.EntityComponent.PhysicsSystem](#), [MB2D.Testing.TestSystem](#), [MidnightBlue.ShipInputSystem](#), [MB2D.EntityComponent.InputSystem](#), [MB2D.EntityComponent.MovementSystem](#), [MB2D.EntityComponent.DepthSystem](#), and [MB2D.Testing.CollisionRenderSystem](#).

5.16.3.7 ProcessingLoop()

```
virtual void MB2D.EntityComponent.EntitySystem.ProcessingLoop ( ) [inline], [protected], [virtual]
```

Executes [Process\(\)](#) on all AssociatedEntities.

Reimplemented in [MB2D.EntityComponent.CollisionSystem](#).

5.16.3.8 RemoveAssociation()

```
void MB2D.EntityComponent.EntitySystem.RemoveAssociation (
    Entity entity) [inline]
```

Decouples an association with this system

Parameters

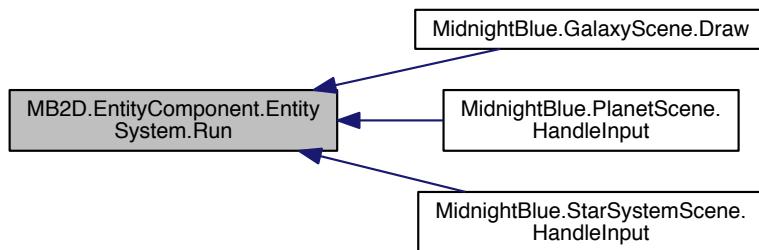
<code>entity</code>	<code>Entity</code> to decouple.
---------------------	----------------------------------

5.16.3.9 Run()

```
void MB2D.EntityComponent.EntitySystem.Run ( ) [inline]
```

Runs this systems [Process\(\)](#) method on all entities

Here is the caller graph for this function:

**5.16.4 Member Data Documentation****5.16.4.1 _associationType**

```
EntityAssociation MB2D.EntityComponent.EntitySystem._associationType [protected]
```

Defines if this system should be Loose or Strict

5.16.4.2 _idEntityMap

```
Dictionary<ulong, Entity> MB2D.EntityComponent.EntitySystem._idEntityMap [protected]
```

All GUID's of entities this system knows about

5.16.5 Property Documentation**5.16.5.1 AssociatedEntities**

```
List<Entity> MB2D.EntityComponent.EntitySystem.AssociatedEntities [get], [set]
```

Gets or sets the associated entities list.

The associated entities.

5.16.5.2 Association

```
EntityAssociation MB2D.EntityComponent.EntitySystem.Association [get]
```

Gets the systems association level.

The association level.

5.16.5.3 DestroyList

```
List<Entity> MB2D.EntityComponent.EntitySystem.DestroyList [get]
```

Gets the destroy list.

The destroy list.

5.16.5.4 ID

```
ulong MB2D.EntityComponent.EntitySystem.ID [get], [set]
```

Gets or sets this systems ID mask

The identifier mask.

5.16.5.5 ValidComponents

```
List<Type> MB2D.EntityComponent.EntitySystem.ValidComponents [get]
```

Gets the list of components this system is interested in

The valid components.

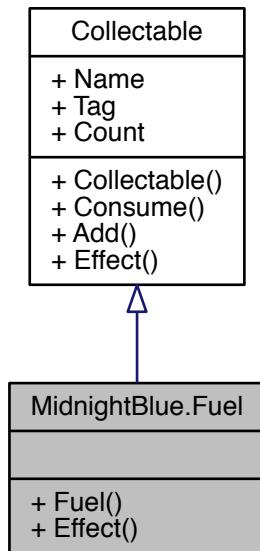
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/EntitySystem.cs

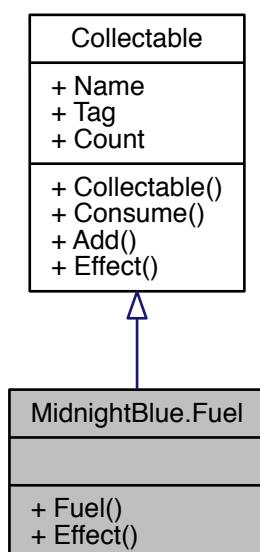
5.17 MidnightBlue.Fuel Class Reference

Fuel used in the ships normal thruster drive

Inheritance diagram for MidnightBlue.Fuel:



Collaboration diagram for MidnightBlue.Fuel:



Public Member Functions

- **Fuel** (int initialCount)
Initializes a new instance of the T:MidnightBlue.Fuel class.
- override void **Effect** (Entity entity)
Has no effect on the entity

Additional Inherited Members

5.17.1 Detailed Description

Fuel used in the ships normal thruster drive

5.17.2 Constructor & Destructor Documentation

5.17.2.1 Fuel()

```
MidnightBlue.Fuel.Fuel (
    int initialCount ) [inline]
```

Initializes a new instance of the T:MidnightBlue.Fuel class.

Parameters

<i>initialCount</i>	Initial amount of fuel.
---------------------	-------------------------

5.17.3 Member Function Documentation

5.17.3.1 Effect()

```
override void MidnightBlue.Fuel.Effect (
    Entity entity ) [inline], [virtual]
```

Has no effect on the entity

Parameters

<i>entity</i>	Entity.
---------------	---------

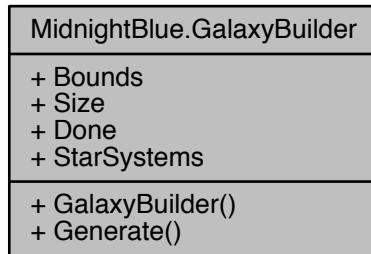
Implements [MB2D.Collectable](#).

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

- Shared/src/Game/Inventory/Fuel.cs

5.18 MidnightBlue.GalaxyBuilder Class Reference

Collaboration diagram for MidnightBlue.GalaxyBuilder:



Public Member Functions

- [GalaxyBuilder](#) (ContentManager content, int size, int seed=0)
Initializes a new instance of the T:MidnightBlue.GalaxyBuilder class. Does not actually generate the galaxy - that's done via [Generate\(\)](#)
- List< [StarSystem](#) > [Generate](#) (int maxDistance)
Generates the galaxy with a specified max distance between stars. Takes a while so should be called only once per gameplay session.

Properties

- Rectangle [Bounds](#) [get]
Gets the bounding rectangle of the galaxy.
- int [Size](#) [get]
Gets the number of star systems the galaxy has.
- bool [Done](#) [get]
Gets a value indicating whether this T:MidnightBlue.GalaxyBuilder is done generating.
- List< [StarSystem](#) > [StarSystems](#) [get]
Gets the star system list.

5.18.1 Constructor & Destructor Documentation

5.18.1.1 [GalaxyBuilder\(\)](#)

```

MidnightBlue.GalaxyBuilder.GalaxyBuilder (
    ContentManager content,
    int size,
    int seed = 0 ) [inline]
  
```

Initializes a new instance of the T:MidnightBlue.GalaxyBuilder class. Does not actually generate the galaxy - that's done via [Generate\(\)](#)

Parameters

<i>content</i>	Content manager to use for loading resources.
<i>size</i>	Number of star systems to generate.
<i>seed</i>	Seed to use for generation.

5.18.2 Member Function Documentation

5.18.2.1 Generate()

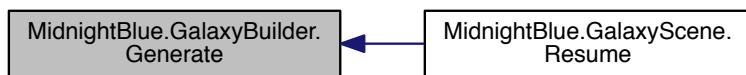
```
List<StarSystem> MidnightBlue.GalaxyBuilder.Generate (
    int maxDistance ) [inline]
```

Generates the galaxy with a specified max distance between stars. Takes a while so should be called only once per gameplay session.

Parameters

<i>maxDistance</i>	Max distance between generated star systems.
--------------------	--

Here is the caller graph for this function:



5.18.3 Property Documentation

5.18.3.1 Bounds

```
Rectangle MidnightBlue.GalaxyBuilder.Bounds [get]
```

Gets the bounding rectangle of the galaxy.

The bounds.

5.18.3.2 Done

```
bool MidnightBlue.GalaxyBuilder.Done [get]
```

Gets a value indicating whether this T:MidnightBlue.GalaxyBuilder is done generating.

true if done; otherwise, false.

5.18.3.3 Size

```
int MidnightBlue.GalaxyBuilder.Size [get]
```

Gets the number of star systems the galaxy has.

The number of star systems.

5.18.3.4 StarSystems

```
List<StarSystem> MidnightBlue.GalaxyBuilder.StarSystems [get]
```

Gets the star system list.

The star systems.

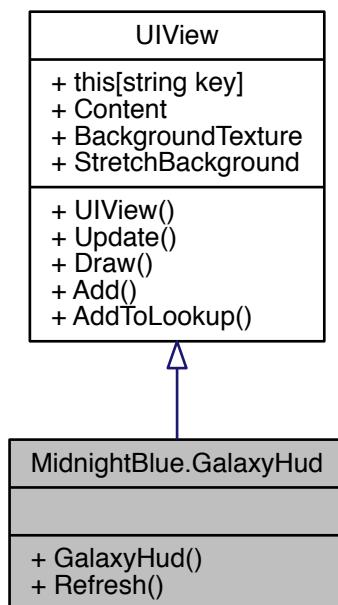
The documentation for this class was generated from the following file:

- Shared/src/Game/Environment/GalaxyBuilder.cs

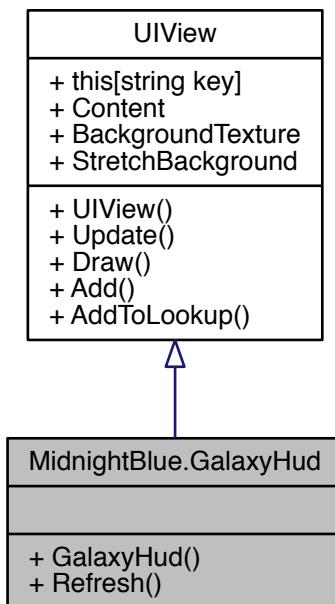
5.19 MidnightBlue.GalaxyHud Class Reference

HUD to show in the galaxy view.

Inheritance diagram for MidnightBlue.GalaxyHud:



Collaboration diagram for `MidnightBlue.GalaxyHud`:



Public Member Functions

- [GalaxyHud](#) (`ContentManager content`)
Initializes a new instance of the T:MB2D.GalaxyHud class.
- void [Refresh](#) (`Inventory inventory`)
Refreshed the HUD with the specified inventory values.

Additional Inherited Members

5.19.1 Detailed Description

HUD to show in the galaxy view.

5.19.2 Constructor & Destructor Documentation

5.19.2.1 GalaxyHud()

```
MidnightBlue.GalaxyHud.GalaxyHud (
    ContentManager content ) [inline]
```

Initializes a new instance of the T:MB2D.GalaxyHud class.

Parameters

<i>content</i>	Content to use in loading fonts and textures.
----------------	---

Here is the call graph for this function:



5.19.3 Member Function Documentation

5.19.3.1 Refresh()

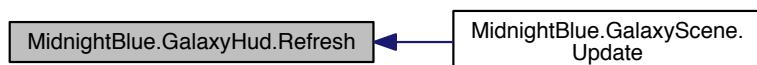
```
void MidnightBlue.GalaxyHud.Refresh (
    Inventory inventory ) [inline]
```

Refreshed the HUD with the specified inventory values.

Parameters

<i>inventory</i>	Inventory to use to refresh the display.
------------------	--

Here is the caller graph for this function:



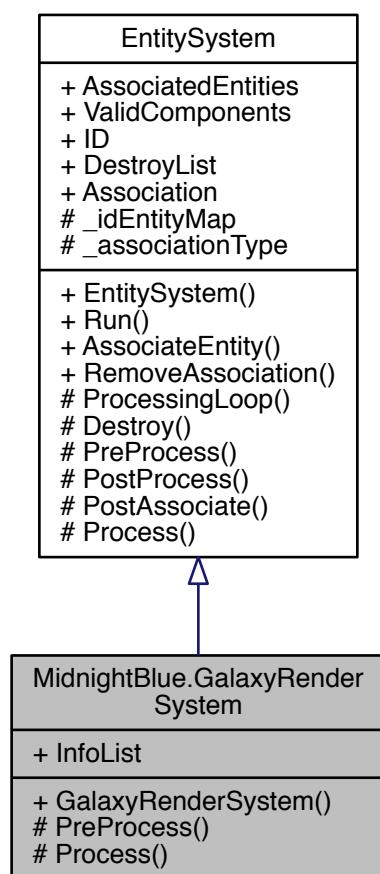
The documentation for this class was generated from the following file:

- Shared/src/Game/UIViews/GalaxyHud.cs

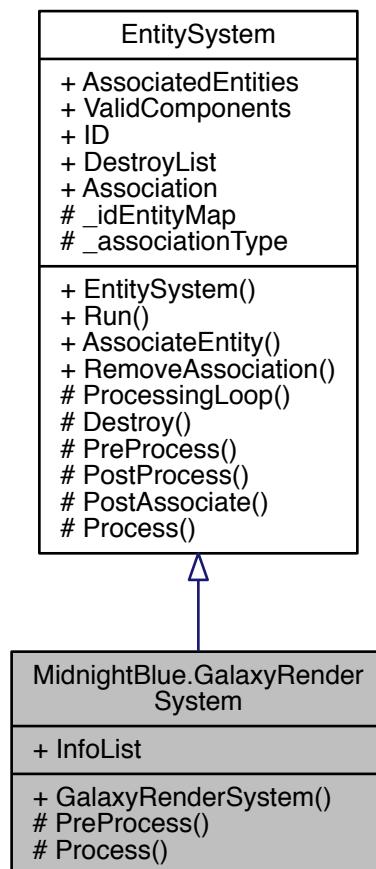
5.20 MidnightBlue.GalaxyRenderSystem Class Reference

Renders all information into the main HUD's list box on the hovered star system. Also displays the name of all the star systems planets.

Inheritance diagram for MidnightBlue.GalaxyRenderSystem:



Collaboration diagram for MidnightBlue.GalaxyRenderSystem:



Public Member Functions

- [GalaxyRenderSystem](#) (SpriteBatch spriteBatch, ContentManager content)
Initializes a new instance of the T:MidnightBlue.GalaxyRenderSystem class.

Protected Member Functions

- override void [PreProcess](#) ()
Clears the starsystems info list before processing all entities.
- override void [Process](#) (Entity entity)
Checks for collisions with a star system in the galaxy view and renders any information associated with that star re:planets.

Properties

- List< string > [InfoList](#) [get]
Gets the list of all planets in the star system's information.

Additional Inherited Members

5.20.1 Detailed Description

Renders all information into the main HUD's list box on the hovered star system. Also displays the name of all the star systems planets.

5.20.2 Constructor & Destructor Documentation

5.20.2.1 GalaxyRenderSystem()

```
MidnightBlue.GalaxyRenderSystem.GalaxyRenderSystem (
    SpriteBatch spriteBatch,
    ContentManager content ) [inline]
```

Initializes a new instance of the T:MidnightBlue.GalaxyRenderSystem class.

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
<i>content</i>	Content to load fonts from.

5.20.3 Member Function Documentation

5.20.3.1 PreProcess()

```
override void MidnightBlue.GalaxyRenderSystem.PreProcess ( ) [inline], [protected], [virtual]
```

Clears the starsystems info list before processing all entities.

Reimplemented from [MB2D.EntityComponent.EntitySystem](#).

5.20.3.2 Process()

```
override void MidnightBlue.GalaxyRenderSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
```

Checks for collisions with a star system in the galaxy view and renders any information associated with that star re:planets.

Parameters

<i>entity</i>	Entity to check collisions with.
---------------	----------------------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

5.20.4 Property Documentation

5.20.4.1 InfoList

```
List<string> MidnightBlue.GalaxyRenderSystem.InfoList [get]
```

Gets the list of all planets in the star system's information.

The info list.

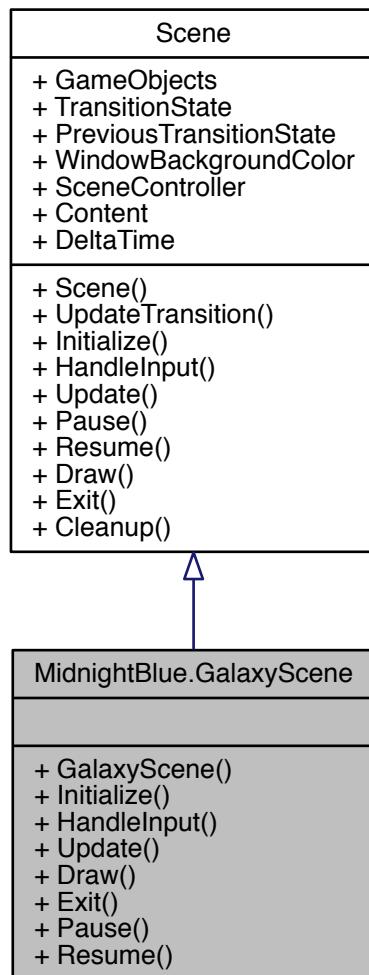
The documentation for this class was generated from the following file:

- Shared/src/Game/Systems/GalaxyRenderSystem.cs

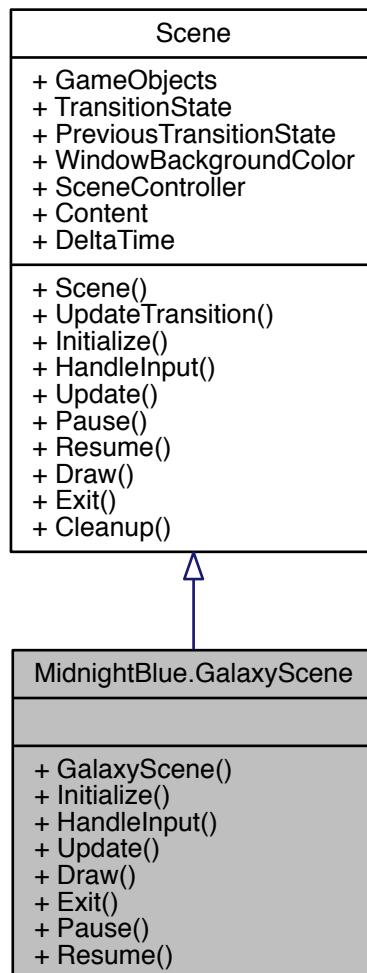
5.21 MidnightBlue.GalaxyScene Class Reference

The scene displayed at the galaxy view - handles the control over all systems, loading, and content management for the scene.

Inheritance diagram for MidnightBlue.GalaxyScene:



Collaboration diagram for MidnightBlue.GalaxyScene:



Public Member Functions

- **GalaxyScene** ([EntityMap](#) map, [ContentManager](#) content)

Initializes a new instance of the T:MB2D.GalaxyScene class. Loads all resources and sets up the galaxy for generation.

- **override void Initialize ()**

Initializes the galaxy generation and background music - sets up the players ship and the collision bounds.

- **override void HandleInput ()**

Handles all input for the player ship and moves between galaxy and star system scenes if the player enters a star system.

- **override void Update ()**

Updates the galaxy view and fuel consumption if not loading.

- **override void Draw (SpriteBatch spriteBatch, SpriteBatch uiSpriteBatch)**

Draw the game world and UI to the specified spriteBatch and uiSpriteBatch.

- `override void Exit ()`
Exit this scene instantly.
- `override void Pause ()`
Fades the sound out when transitioning to another scene unless the next scene is the star system scene.
- `override void Resume ()`
Resets the physics environment when returning to the scene and rebuilds the galaxy from its cache.

Additional Inherited Members

5.21.1 Detailed Description

The scene displayed at the galaxy view - handles the control over all systems, loading, and content management for the scene.

5.21.2 Constructor & Destructor Documentation

5.21.2.1 GalaxyScene()

```
MidnightBlue.GalaxyScene.GalaxyScene (
    EntityMap map,
    ContentManager content ) [inline]
```

Initializes a new instance of the T:MB2D.GalaxyScene class. Loads all resources and sets up the galaxy for generation.

Parameters

<code>map</code>	Game objects to use.
<code>content</code>	Content manager to use.

5.21.3 Member Function Documentation

5.21.3.1 Draw()

```
override void MidnightBlue.GalaxyScene.Draw (
    SpriteBatch spriteBatch,
    SpriteBatch uiSpriteBatch ) [inline], [virtual]
```

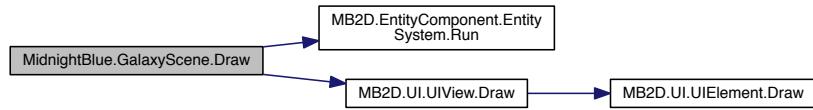
Draw the game world and UI to the specified spriteBatch and uiSpriteBatch.

Parameters

<code>spriteBatch</code>	Sprite batch for world-based entities.
<code>uiSpriteBatch</code>	User interface sprite batch.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.21.3.2 Exit()

```
override void MidnightBlue.GalaxyScene.Exit ( ) [inline], [virtual]
```

Exit this scene instantly.

Implements [MB2D.Scenes.Scene](#).

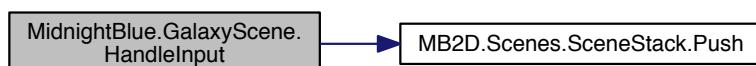
5.21.3.3 HandleInput()

```
override void MidnightBlue.GalaxyScene.HandleInput ( ) [inline], [virtual]
```

Handles all input for the player ship and moves between galaxy and star system scenes if the player enters a star system.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.21.3.4 Initialize()

```
override void MidnightBlue.GalaxyScene.Initialize ( ) [inline], [virtual]
```

Initializes the galaxy generation and background music - sets up the players ship and the collision bounds.

Implements [MB2D.Scenes.Scene](#).

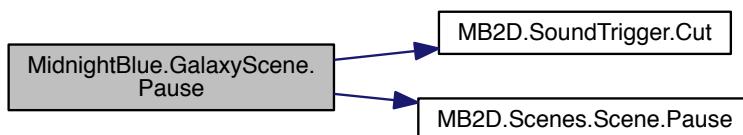
5.21.3.5 Pause()

```
override void MidnightBlue.GalaxyScene.Pause ( ) [inline], [virtual]
```

Fades the sound out when transitioning to another scene unless the next scene is the star system scene.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



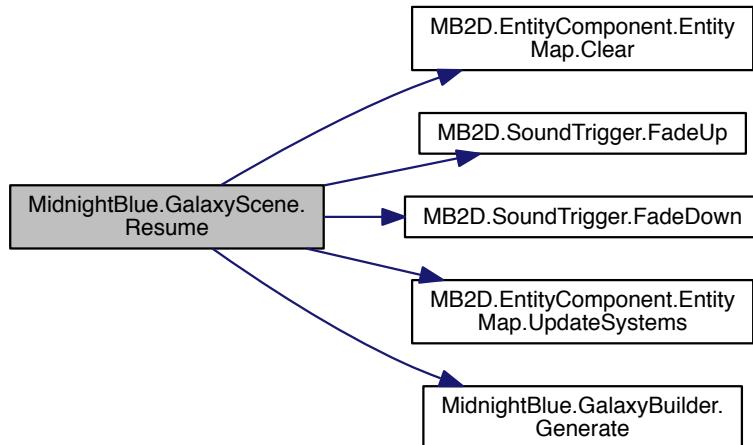
5.21.3.6 Resume()

```
override void MidnightBlue.GalaxyScene.Resume ( ) [inline], [virtual]
```

Resets the physics environment when returning to the scene and rebuilds the galaxy from its cache.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



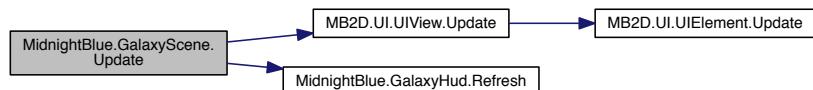
5.21.3.7 Update()

```
override void MidnightBlue.GalaxyScene.Update ( ) [inline], [virtual]
```

Updates the galaxy view and fuel consumption if not loading.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:

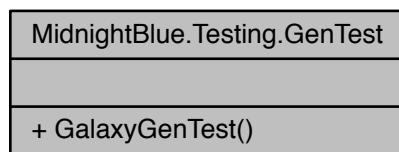


The documentation for this class was generated from the following file:

- Shared/src/Game/Scenes/GalaxyScene.cs

5.22 MidnightBlue.Testing.GenTest Class Reference

Collaboration diagram for `MidnightBlue.Testing.GenTest`:



Static Public Member Functions

- static void **GalaxyGenTest** (params string[] args)

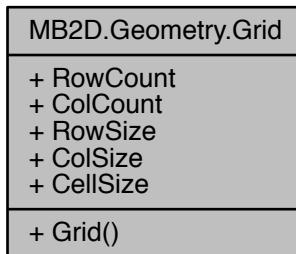
The documentation for this class was generated from the following file:

- Shared/src/Game/Tests/GenTest.cs

5.23 MB2D.Geometry.Grid Class Reference

Represents a grid structure. Can be drawn via a SpriteBatch

Collaboration diagram for MB2D.Geometry.Grid:



Public Member Functions

- [Grid](#) (int rows, int cols, int rowSize, int colSize)
Initializes a new instance of the T:MB2D.Geometry.Grid class.

Properties

- int [RowCount](#) [get, set]
Gets or sets the number of rows.
- int [ColCount](#) [get, set]
Gets or sets the number of columns.
- int [RowSize](#) [get, set]
Gets or sets the size of each row.
- int [ColSize](#) [get, set]
Gets or sets the size of each column.
- Vector2 [CellSize](#) [get]
Gets the size of each cell.

5.23.1 Detailed Description

Represents a grid structure. Can be drawn via a SpriteBatch

5.23.2 Constructor & Destructor Documentation

5.23.2.1 Grid()

```
MB2D.Geometry.Grid.Grid (
    int rows,
    int cols,
    int rowSize,
    int colSize ) [inline]
```

Initializes a new instance of the T:MB2D.Geometry.Grid class.

Parameters

<i>rows</i>	Number of rows
<i>cols</i>	Number of columns
<i>rowSize</i>	Size of each row
<i>colSize</i>	Size of each column

5.23.3 Property Documentation

5.23.3.1 CellSize

```
Vector2 MB2D.Geometry.Grid.CellSize [get]
```

Gets the size of each cell.

The size of each cell.

5.23.3.2 ColCount

```
int MB2D.Geometry.Grid.ColCount [get], [set]
```

Gets or sets the number of columns.

The column count.

5.23.3.3 ColSize

```
int MB2D.Geometry.Grid.ColSize [get], [set]
```

Gets or sets the size of each column.

The size of each column.

5.23.3.4 RowCount

```
int MB2D.Geometry.Grid.RowCount [get], [set]
```

Gets or sets the number of rows.

The row count.

5.23.3.5 RowSize

```
int MB2D.Geometry.Grid.RowSize [get], [set]
```

Gets or sets the size of each row.

The size of each row.

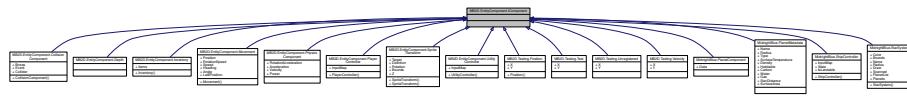
The documentation for this class was generated from the following file:

- MB2D/src/Geometry/Grid.cs

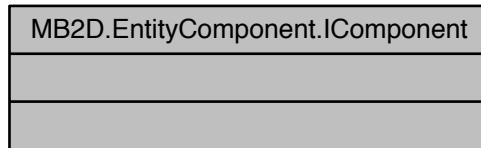
5.24 MB2D.EntityComponent.IComponent Interface Reference

Tags any class as a valid component for use in the [EntityMap](#). Derived classes should contain no logic, only data fields.

Inheritance diagram for MB2D.EntityComponent.IComponent:



Collaboration diagram for MB2D.EntityComponent.IComponent:



5.24.1 Detailed Description

Tags any class as a valid component for use in the [EntityMap](#). Derived classes should contain no logic, only data fields.

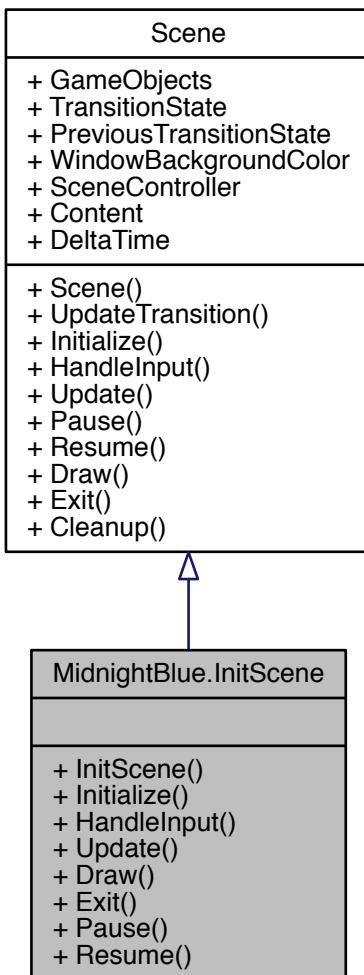
The documentation for this interface was generated from the following file:

- `MB2D/src/EntityComponent/IComponent.cs`

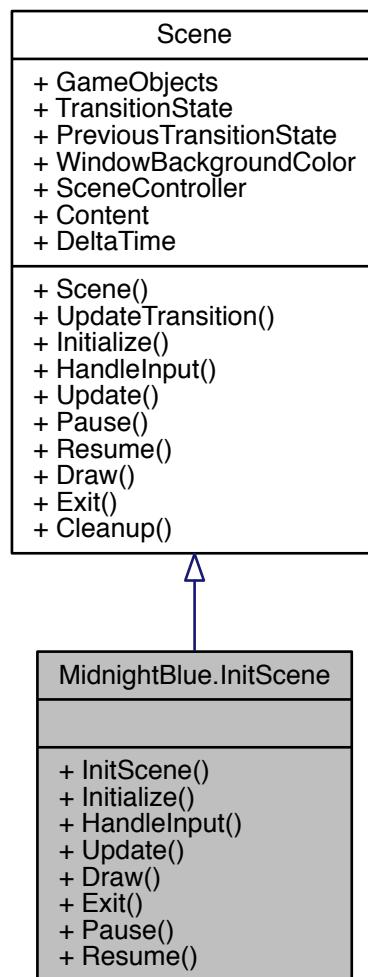
5.25 MidnightBlue.InitScene Class Reference

The scene shown at the title screen.

Inheritance diagram for `MidnightBlue.InitScene`:



Collaboration diagram for MidnightBlue.InitScene:



Public Member Functions

- `InitScene (EntityMap map, ContentManager content)`
Initializes a new instance of the T:MidnightBlue.InitScene class. Loads all blueprints and setup data.
- `override void Initialize ()`
Registers all blueprints to the EntityMap
- `override void HandleInput ()`
Handles the input for the scene.
- `override void Update ()`
Updates the scene.
- `override void Draw (SpriteBatch spriteBatch, SpriteBatch uiSpriteBatch)`
Draws the scene to the uiSpriteBatch
- `override void Exit ()`
Exits the scene.

- override void [Pause \(\)](#)
Pauses the scene.
- override void [Resume \(\)](#)
Resumes the scene.

Additional Inherited Members

5.25.1 Detailed Description

The scene shown at the title screen.

5.25.2 Constructor & Destructor Documentation

5.25.2.1 [InitScene\(\)](#)

```
MidnightBlue.InitScene.InitScene (
    EntityMap map,
    ContentManager content ) [inline]
```

Initializes a new instance of the T:MidnightBlue.InitScene class. Loads all blueprints and setup data.

Parameters

<i>map</i>	Game objects.
<i>content</i>	Content manager for loading textures and sounds.

5.25.3 Member Function Documentation

5.25.3.1 [Draw\(\)](#)

```
override void MidnightBlue.InitScene.Draw (
    SpriteBatch spriteBatch,
    SpriteBatch uiSpriteBatch ) [inline], [virtual]
```

Draws the scene to the uiSpriteBatch

Parameters

<i>spriteBatch</i>	Sprite batch for world-based entities.
<i>uiSpriteBatch</i>	User interface sprite batch.

Implements [MB2D.Scenes.Scene](#).

5.25.3.2 [Exit\(\)](#)

```
override void MidnightBlue.InitScene.Exit ( ) [inline], [virtual]
```

Exits the scene.

Implements [MB2D.Scenes.Scene](#).

5.25.3.3 HandleInput()

```
override void MidnightBlue.InitScene.HandleInput ( ) [inline], [virtual]
```

Handles the input for the scene.

Implements [MB2D.Scenes.Scene](#).

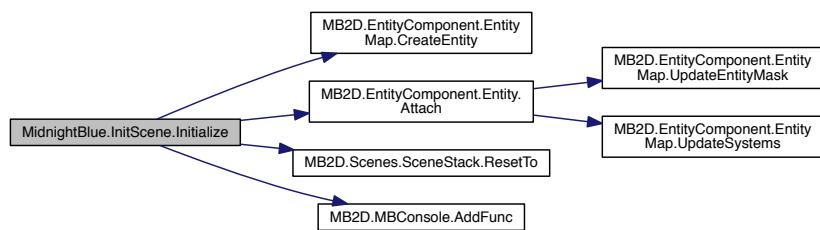
5.25.3.4 Initialize()

```
override void MidnightBlue.InitScene.Initialize ( ) [inline], [virtual]
```

Registers all blueprints to the EntityMap

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.25.3.5 Pause()

```
override void MidnightBlue.InitScene.Pause ( ) [inline], [virtual]
```

Pauses the scene.

Implements [MB2D.Scenes.Scene](#).

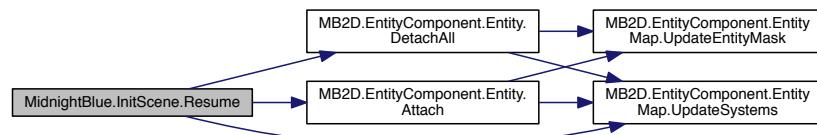
5.25.3.6 Resume()

```
override void MidnightBlue.InitScene.Resume ( ) [inline], [virtual]
```

Resumes the scene.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.25.3.7 Update()

```
override void MidnightBlue.InitScene.Update ( ) [inline], [virtual]
```

Updates the scene.

Implements [MB2D.Scenes.Scene](#).

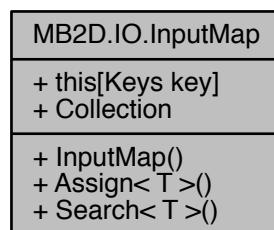
The documentation for this class was generated from the following file:

- Shared/src/Game/Scenes/InitScene.cs

5.26 MB2D.IO.InputMap Class Reference

Maps commands to keys and trigger types

Collaboration diagram for MB2D.IO.InputMap:



Public Member Functions

- `InputMap ()`
Initializes a new instance of the T:MB2D.IO.InputMap class.
- `Command Assign< T > (Keys key, CommandType type, params object[] args)`
Assign a `Command` to the specified key and `CommandType`.
- `T Search< T > ()`
Searches the map for a specific command

Properties

- `List< Command > this[Keys key] [get]`
Gets the T:MB2D.IO.Command mapped to the specified key.
- `Dictionary< Keys, List< Command > > Collection [get]`
Gets a key->`Command` dictionary

5.26.1 Detailed Description

Maps commands to keys and trigger types

5.26.2 Constructor & Destructor Documentation

5.26.2.1 InputMap()

```
MB2D.IO.InputMap ( ) [inline]
```

Initializes a new instance of the T:MB2D.IO.InputMap class.

5.26.3 Member Function Documentation

5.26.3.1 Assign< T >()

```
Command MB2D.IO.InputMap.Assign< T > (
    Keys key,
    CommandType type,
    params object[] args) [inline]
```

Assign a `Command` to the specified key and `CommandType`.

Parameters

<code>key</code>	Key to assign the command to
<code>type</code>	Type of trigger

Template Parameters

<i>T</i>	The command to assign
----------	-----------------------

Type Constraints

T : Command

5.26.3.2 Search< T >()

```
T MB2D.IO.InputMap.Search< T > ( ) [inline]
```

Searches the map for a specific command

Template Parameters

<i>T</i>	The 1st type parameter.
----------	-------------------------

Type Constraints

T : Command

5.26.4 Property Documentation

5.26.4.1 Collection

```
Dictionary<Keys, List<Command> > MB2D.IO.InputMap.Collection [get]
```

Gets a key->[Command](#) dictionary

The collection of commands.

5.26.4.2 this[Keys key]

```
List<Command> MB2D.IO.InputMap.this[Keys key] [get]
```

Gets the T:MB2D.IO.Commnad mapped to the specified key.

Parameters

<i>key</i>	Key to query
------------	--------------

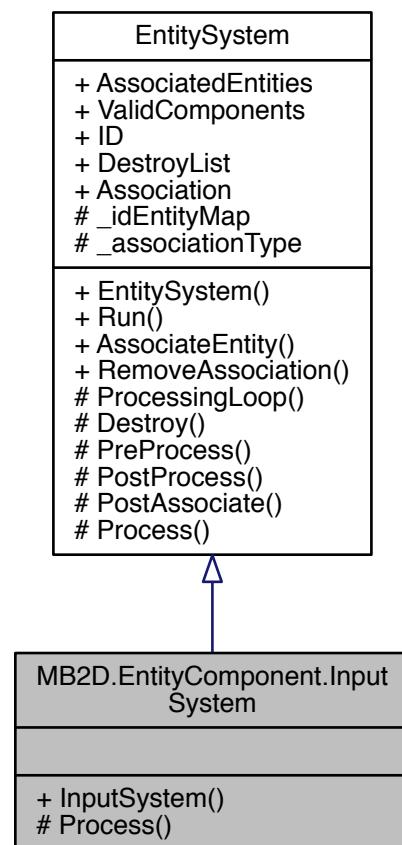
The documentation for this class was generated from the following file:

- MB2D/src/Input/InputMap.cs

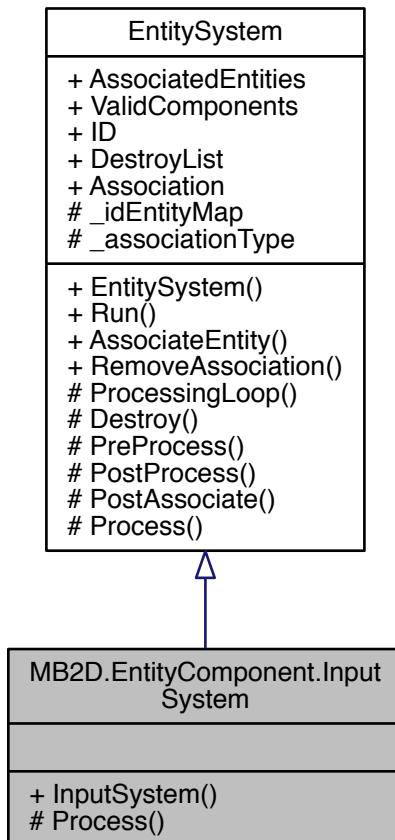
5.27 MB2D.EntityComponent.InputSystem Class Reference

Processes input for [PlayerController](#) and [UtilityController](#) entities. Can operate on an entity with either or both components

Inheritance diagram for MB2D.EntityComponent.InputSystem:



Collaboration diagram for MB2D.EntityComponent.InputSystem:



Protected Member Functions

- override void `Process (Entity entity)`

Processes the controllers inputs

Additional Inherited Members

5.27.1 Detailed Description

Processes input for `PlayerController` and `UtilityController` entities. Can operate on an entity with either or both components

5.27.2 Member Function Documentation

5.27.2.1 Process()

```
override void MB2D.EntityComponent.InputSystem.Process (
    Entity entity) [inline], [protected], [virtual]
```

Processes the controllers inputs

Parameters

<i>entity</i>	Entity to process.
---------------	--------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

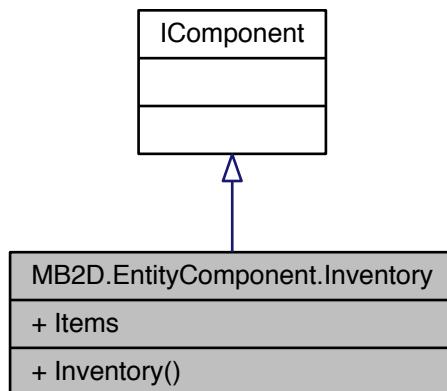
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/InputSystem.cs

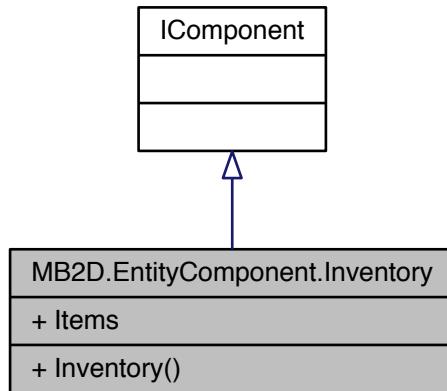
5.28 MB2D.EntityComponent.Inventory Class Reference

Defines a dictionary of [Collectable](#) types used for entities

Inheritance diagram for MB2D.EntityComponent.Inventory:



Collaboration diagram for MB2D.EntityComponent.Inventory:



Properties

- Dictionary< Type, Collectable > **Items** [get, set]
The items currently in the inventory

5.28.1 Detailed Description

Defines a dictionary of [Collectable](#) types used for entities

5.28.2 Property Documentation

5.28.2.1 Items

Dictionary<Type, [Collectable](#)> MB2D.EntityComponent.Inventory.Items [get], [set]

The items currently in the inventory

The items.

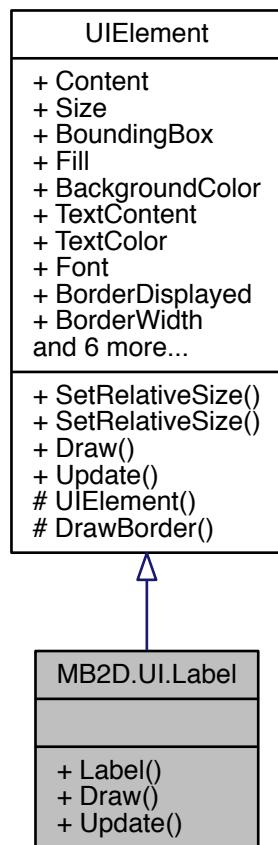
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Components/Inventory.cs

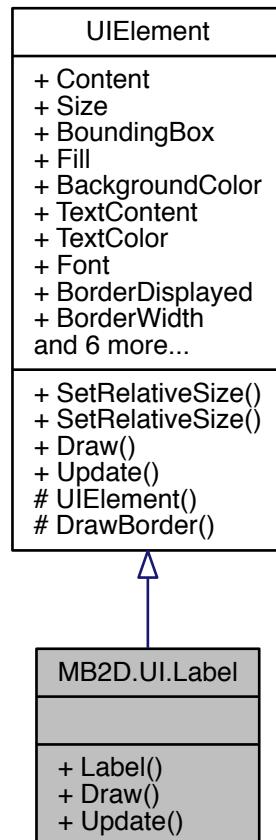
5.29 MB2D.UI.Label Class Reference

A static [UIElement](#) with a TextContent, border and optional texture

Inheritance diagram for MB2D.UI.Label:



Collaboration diagram for MB2D.UI.Label:



Public Member Functions

- [Label \(\)](#)
Initializes a new instance of the T:MB2D.UI.Label class.
- [override void Draw \(SpriteBatch spriteBatch\)](#)
Draw the label to the window.
- [override void Update \(\)](#)
Updates the labels state

Additional Inherited Members

5.29.1 Detailed Description

A static [UIElement](#) with a TextContent, border and optional texture

5.29.2 Constructor & Destructor Documentation

5.29.2.1 Label()

```
MB2D.UI.Label.Label ( ) [inline]
```

Initializes a new instance of the T:MB2D.UI.Label class.

5.29.3 Member Function Documentation

5.29.3.1 Draw()

```
override void MB2D.UI.Label.Draw (
    SpriteBatch spriteBatch) [inline], [virtual]
```

Draw the label to the window.

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
--------------------	--------------------------

Reimplemented from [MB2D.UI.UIElement](#).

5.29.3.2 Update()

```
override void MB2D.UI.Label.Update () [inline], [virtual]
```

Updates the labels state

Implements [MB2D.UI.UIElement](#).

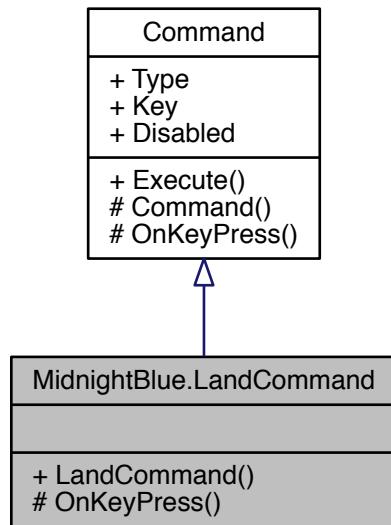
The documentation for this class was generated from the following file:

- MB2D/src/UI/Label.cs

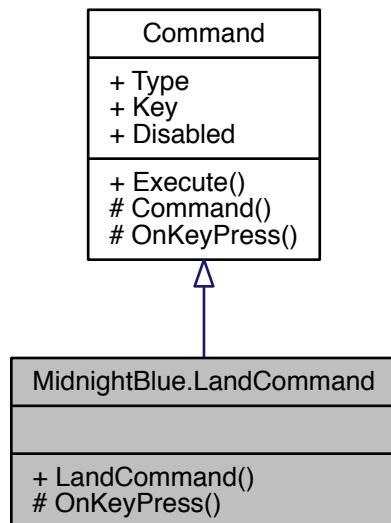
5.30 MidnightBlue.LandCommand Class Reference

Lands the ship

Inheritance diagram for MidnightBlue.LandCommand:



Collaboration diagram for `MidnightBlue.LandCommand`:



Public Member Functions

- [LandCommand](#) (`Keys key, CommandType type`)

Initializes a new instance of the T:MidnightBlue.LandCommand class.

Protected Member Functions

- override void [OnKeyPress \(Entity e\)](#)
Lands the ship on the key press if terrain is landable

Additional Inherited Members

5.30.1 Detailed Description

Lands the ship

5.30.2 Constructor & Destructor Documentation

5.30.2.1 LandCommand()

```
MidnightBlue.LandCommand.LandCommand (
    Keys key,
    CommandType type )  [inline]
```

Initializes a new instance of the T:MidnightBlue.LandCommand class.

Parameters

<code>key</code>	Key to assign to.
<code>type</code>	Trigger type.

5.30.3 Member Function Documentation

5.30.3.1 OnKeyPress()

```
override void MidnightBlue.LandCommand.OnKeyPress (
    Entity e )  [inline], [protected], [virtual]
```

Lands the ship on the key press if terrain is landable

Parameters

<code>e</code>	Entity with the ship controller to operate on.
----------------	--

Implements [MB2D.IO.Command](#).

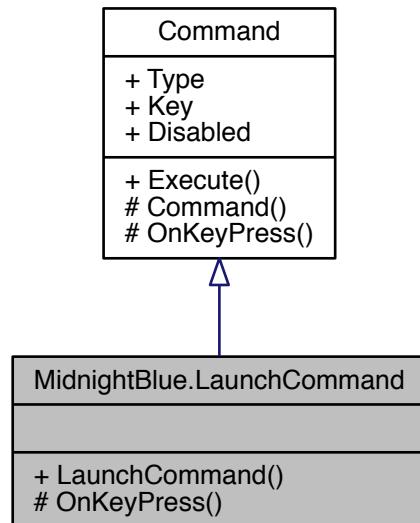
The documentation for this class was generated from the following file:

- Shared/src/Game/Commands/ShipCommands.cs

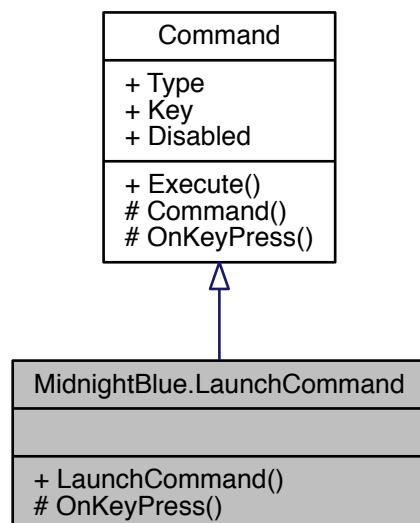
5.31 MidnightBlue.LaunchCommand Class Reference

Launches the ship from a landed state

Inheritance diagram for MidnightBlue.LaunchCommand:



Collaboration diagram for `MidnightBlue.LaunchCommand`:



Public Member Functions

- [LaunchCommand \(Keys key, CommandType type\)](#)

Initializes a new instance of the T:MidnightBlue.LaunchCommand class.

Protected Member Functions

- [override void OnKeyPress \(Entity e\)](#)

Launches the ship from landed on key press.

Additional Inherited Members

5.31.1 Detailed Description

Launches the ship from a landed state

5.31.2 Constructor & Destructor Documentation

5.31.2.1 LaunchCommand()

```
MidnightBlue.LaunchCommand.LaunchCommand (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MidnightBlue.LaunchCommand class.

Parameters

<code>key</code>	Key to assign to.
<code>type</code>	Trigger type.

5.31.3 Member Function Documentation

5.31.3.1 OnKeyPress()

```
override void MidnightBlue.LaunchCommand.OnKeyPress (
    Entity e ) [inline], [protected], [virtual]
```

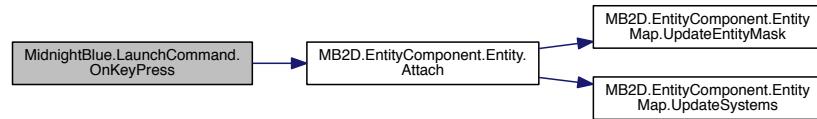
Launches the ship from landed on key press.

Parameters

<code>e</code>	Entity with ship controller to operate on.
----------------	--

Implements [MB2D.IO.Command](#).

Here is the call graph for this function:



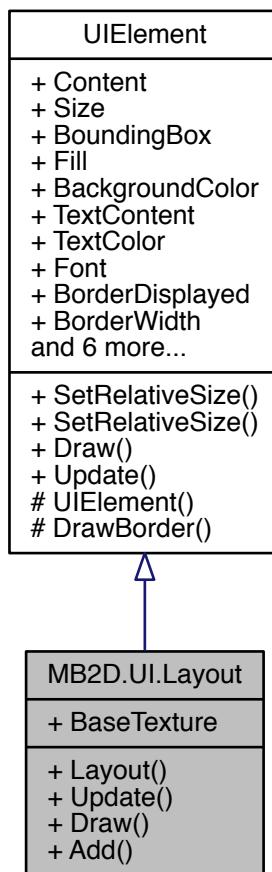
The documentation for this class was generated from the following file:

- Shared/src/Game/Commands/ShipCommands.cs

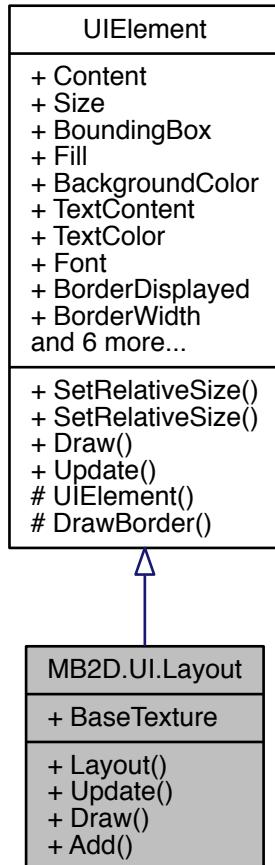
5.32 MB2D.UI.Layout Class Reference

A container for UIElements used within a [UIView](#). Used to divide the View into smaller segments and to move around a group of elements easily

Inheritance diagram for MB2D.UI.Layout:



Collaboration diagram for MB2D.UI.Layout:



Public Member Functions

- **Layout** ([UIView](#) parentView, int rows, int cols)

Initializes a new instance of the T:MB2D.UI.Layout class. Divides the layouts rows and columns into an even cell size based of its parents size.
- **override void Update ()**

Update all elements contained within the layout.
- **override void Draw** ([SpriteBatch](#) spriteBatch)

Draws all elements contained within the layout and the layouts border to the window.
- **void Add** ([UIElement](#) element, int atRow, int atCol, int rowSpan, int colSpan)

Adds a new [UIElement](#) to the layout, setting its size relative to this layouts cell size

Properties

- **Texture2D BaseTexture** [get, set]

Additional Inherited Members

5.32.1 Detailed Description

A container for UIElements used within a [UIView](#). Used to divide the View into smaller segments and to move around a group of elements easily

5.32.2 Constructor & Destructor Documentation

5.32.2.1 Layout()

```
MB2D.UI.Layout.Layout (
    UIView parentView,
    int rows,
    int cols ) [inline]
```

Initializes a new instance of the T:MB2D.UI.Layout class. Divides the layouts rows and columns into an even cell size based of its parents size.

Parameters

<i>rows</i>	Number of rows the layout has.
<i>cols</i>	Number of columns the layout has.

5.32.3 Member Function Documentation

5.32.3.1 Add()

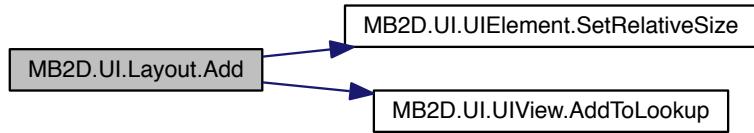
```
void MB2D.UI.Layout.Add (
    UIElement element,
    int atRow,
    int atCol,
    int rowSpan,
    int colSpan ) [inline]
```

Adds a new [UIElement](#) to the layout, setting its size relative to this layouts cell size

Parameters

<i>element</i>	Element to add.
<i>atRow</i>	Row position to add the element at.
<i>atCol</i>	Column position to add the element at.
<i>rowSpan</i>	Number of rows the element should span.
<i>colSpan</i>	Number of columns the element should span.

Here is the call graph for this function:



5.32.3.2 Draw()

```
override void MB2D.UI.Layout.Draw (
    SpriteBatch spriteBatch) [inline], [virtual]
```

Draws all elements contained within the layout and the layouts border to the window.

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
--------------------	--------------------------

Reimplemented from [MB2D.UI.UIElement](#).

5.32.3.3 Update()

```
override void MB2D.UI.Layout.Update () [inline], [virtual]
```

Update all elements contained within the layout.

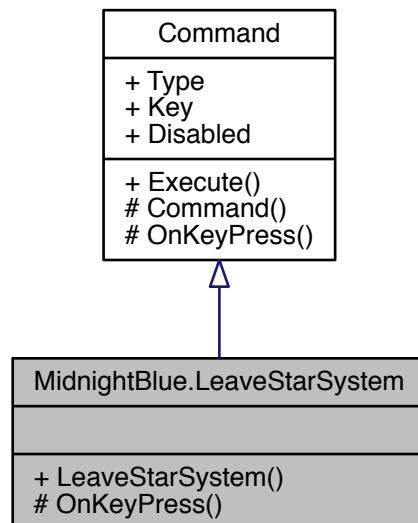
Implements [MB2D.UI.UIElement](#).

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

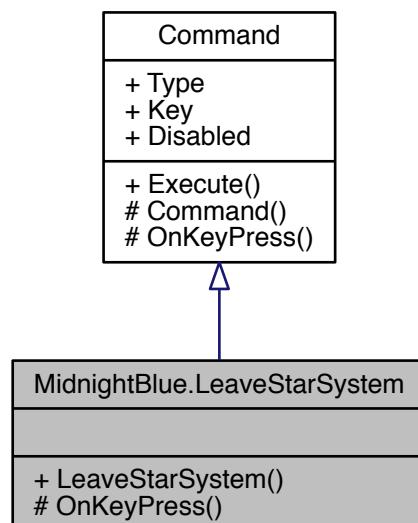
- MB2D/src/UI/Layout.cs

5.33 MidnightBlue.LeaveStarSystem Class Reference

Inheritance diagram for MidnightBlue.LeaveStarSystem:



Collaboration diagram for MidnightBlue.LeaveStarSystem:



Public Member Functions

- [LeaveStarSystem \(Keys key, CommandType type\)](#)

Initializes a new instance of the T:MidnightBlue.LeaveStarSystem class.

Protected Member Functions

- override void [OnKeyPress \(Entity e=null\)](#)

Defines the logic to perform when operating on a given entity

Additional Inherited Members

5.33.1 Constructor & Destructor Documentation

5.33.1.1 LeaveStarSystem()

```
MidnightBlue.LeaveStarSystem.LeaveStarSystem (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MidnightBlue.LeaveStarSystem class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.33.2 Member Function Documentation

5.33.2.1 OnKeyPress()

```
override void MidnightBlue.LeaveStarSystem.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Defines the logic to perform when operating on a given entity

Parameters

<i>e</i>	Entity to operate on
----------	----------------------

Implements [MB2D.IO.Command](#).

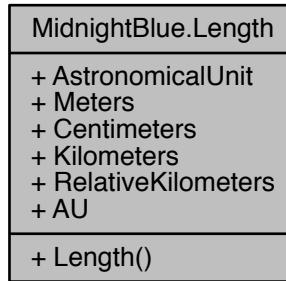
The documentation for this class was generated from the following file:

- Shared/src/Game/Commands/ShipCommands.cs

5.34 MidnightBlue.Length Class Reference

Defines a measurement of length in meters able to be converted to other measurements.

Collaboration diagram for MidnightBlue.Length:



Public Member Functions

- [Length \(ulong meters\)](#)
Initializes a new instance of the T:MidnightBlue.Length class.

Public Attributes

- const float [AstronomicalUnit = 149597870.7f](#)
A single Astronomical Unit in kilometers

Properties

- ulong [Meters](#) [get]
Gets the length in meters
- ulong [Centimeters](#) [get]
Gets the length in centimeters
- ulong [Kilometers](#) [get]
Gets the length in kilometers
- int [RelativeKilometers](#) [get]
Gets the length in kilometers represented as a smaller value used for calculations.
- float [AU](#) [get]
Gets the length in Astronomical Unites

5.34.1 Detailed Description

Defines a measurement of length in meters able to be converted to other measurements.

5.34.2 Constructor & Destructor Documentation

5.34.2.1 Length()

```
MidnightBlue.Length.Length (
    ulong meters ) [inline]
```

Initializes a new instance of the T:MidnightBlue.Length class.

Parameters

<i>meters</i>	Initial length to set in meters.
---------------	----------------------------------

5.34.3 Member Data Documentation

5.34.3.1 AstronomicalUnit

```
const float MidnightBlue.Length.AstronomicalUnit = 149597870.7f
```

A single Astronomical Unit in kilometers

5.34.4 Property Documentation

5.34.4.1 AU

```
float MidnightBlue.Length.AU [get]
```

Gets the length in Astronomical Units

The length in astronomical units.

5.34.4.2 Centimeters

```
ulong MidnightBlue.Length.Centimeters [get]
```

Gets the length in centimeters

The length in centimeters.

5.34.4.3 Kilometers

```
ulong MidnightBlue.Length.Kilometers [get]
```

Gets the length in kilometers

The length in kilometers.

5.34.4.4 Meters

```
ulong MidnightBlue.Length.Meters [get]
```

Gets the length in meters

The length in meters.

5.34.4.5 RelativeKilometers

```
int MidnightBlue.Length.RelativeKilometers [get]
```

Gets the length in kilometers represented as a smaller value used for calculations.

The length in relative kilometers.

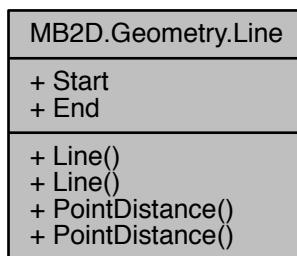
The documentation for this class was generated from the following file:

- Shared/src/Game/Environment/Length.cs

5.35 MB2D.Geometry.Line Class Reference

A line structure, can be drawn via SpriteBatch

Collaboration diagram for MB2D.Geometry.Line:



Public Member Functions

- [Line \(Vector2 start, Vector2 end\)](#)
Initializes a new instance of the T:MB2D.Geometry.Line class.
- [Line \(Point start, Point end\)](#)
Initializes a new instance of the T:MB2D.Geometry.Line class.
- float [PointDistance \(Vector2 point\)](#)
Gets the distance from this line a given point is
- float [PointDistance \(Point point\)](#)
Gets the distance from this line a given point is

Properties

- Vector2 **Start** [get, set]
Gets or sets the start point.
- Vector2 **End** [get, set]
Gets or sets the end point.

5.35.1 Detailed Description

A line structure, can be drawn via SpriteBatch

5.35.2 Constructor & Destructor Documentation

5.35.2.1 Line() [1/2]

```
MB2D.Geometry.Line.Line (
    Vector2 start,
    Vector2 end ) [inline]
```

Initializes a new instance of the T:MB2D.Geometry.Line class.

Parameters

<i>start</i>	Start point
<i>end</i>	End point

5.35.2.2 Line() [2/2]

```
MB2D.Geometry.Line.Line (
    Point start,
    Point end ) [inline]
```

Initializes a new instance of the T:MB2D.Geometry.Line class.

Parameters

<i>start</i>	Start point
<i>end</i>	End point

5.35.3 Member Function Documentation

5.35.3.1 PointDistance() [1/2]

```
float MB2D.Geometry.Line.PointDistance (
    Vector2 point ) [inline]
```

Gets the distance from this line a given point is

Returns

The distance.

Parameters

<i>point</i>	Point to calculate.
--------------	---------------------

5.35.3.2 PointDistance() [2/2]

```
float MB2D.Geometry.Line.PointDistance (
    Point point ) [inline]
```

Gets the distance from this line a given point is

Returns

The distance.

Parameters

<i>point</i>	Point to calculate.
--------------	---------------------

5.35.4 Property Documentation**5.35.4.1 End**

```
Vector2 MB2D.Geometry.Line.End [get], [set]
```

Gets or sets the end point.

The end point.

5.35.4.2 Start

```
Vector2 MB2D.Geometry.Line.Start [get], [set]
```

Gets or sets the start point.

The start point

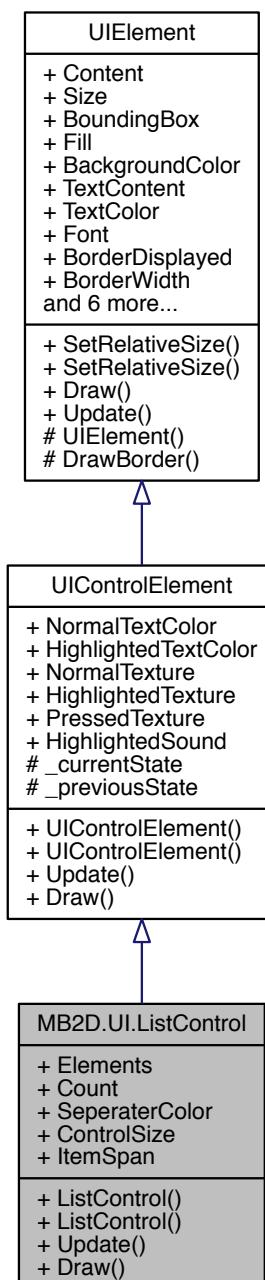
The documentation for this class was generated from the following file:

- MB2D/src/Geometry/Line.cs

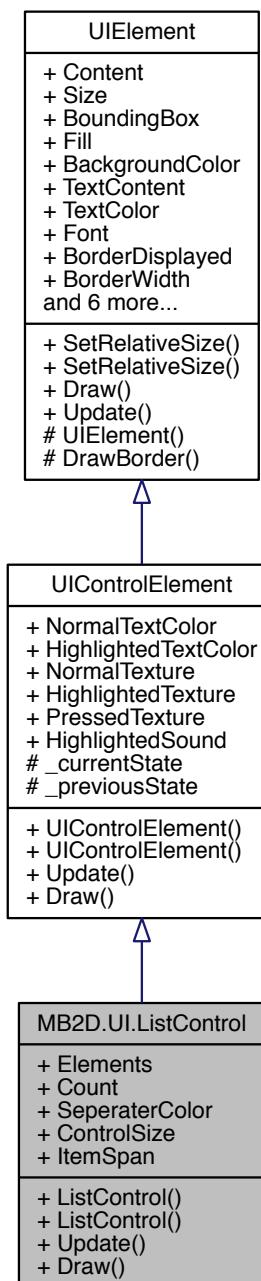
5.36 MB2D.UI.ListControl Class Reference

A scrollable list box. Items can be added and interacted with.

Inheritance diagram for MB2D.UI.ListControl:



Collaboration diagram for MB2D.UI.ListControl:



Public Member Functions

- **ListControl** (SpriteFont font)

Initializes a new instance of the T:MB2D.UI.ListControl class using the specified font to draw elements.

- **ListControl** (SpriteFont font, Texture2D normal, Texture2D selected, Texture2D pressed)

Initializes a new instance of the T:MB2D.UI.ListControl class with a font to use when drawing elements alongside state textures.

- override void [Update \(\)](#)
Updates the element to handle scrolling and clicking. Must be called once per frame.
- override void [Draw \(SpriteBatch spriteBatch\)](#)
Draws the list control to the window

Properties

- List< string > [Elements](#) [get, set]
Gets or sets a list of all of the list control elements
- int [Count](#) [get]
Gets the count of list control elements.
- Color [SeperatorColor](#) [get, set]
Gets or sets the color of the seperater between list elements.
- int [ControlSize](#) [get, set]
Gets or sets the size of the up and down arrow controls.
- int [ItemSpan](#) [get, set]
Gets or sets the span of each item vertically in px.

Additional Inherited Members

5.36.1 Detailed Description

A scrollable list box. Items can be added and interacted with.

5.36.2 Constructor & Destructor Documentation

5.36.2.1 ListControl() [1/2]

```
MB2D.UI.ListControl.ListControl (
    SpriteFont font ) [inline]
```

Initializes a new instance of the T:MB2D.UI.ListControl class using the specified font to draw elements.

Parameters

<i>font</i>	Font to use.
-------------	--------------

5.36.2.2 ListControl() [2/2]

```
MB2D.UI.ListControl.ListControl (
    SpriteFont font,
    Texture2D normal,
    Texture2D selected,
    Texture2D pressed ) [inline]
```

Initializes a new instance of the T:MB2D.UI.ListControl class with a font to use when drawing elements alongside state textures.

Parameters

<i>font</i>	Font to use.
<i>normal</i>	Normal state texture.
<i>selected</i>	Selected stae texture.
<i>pressed</i>	Pressed state texture.

5.36.3 Member Function Documentation

5.36.3.1 Draw()

```
override void MB2D.UI.ListControl.Draw (
    SpriteBatch spriteBatch) [inline], [virtual]
```

Draws the list control to the window

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
--------------------	--------------------------

Reimplemented from [MB2D.UI.UIElement](#).

5.36.3.2 Update()

```
override void MB2D.UI.ListControl.Update () [inline], [virtual]
```

Updates the element to handle scrolling and clicking. Must be called once per frame.

Implements [MB2D.UI.UIElement](#).

5.36.4 Property Documentation

5.36.4.1 ControlSize

```
int MB2D.UI.ListControl.ControlSize [get], [set]
```

Gets or sets the size of the up and down arrow controls.

The size of the controls.

5.36.4.2 Count

```
int MB2D.UI.ListControl.Count [get]
```

Gets the count of list control elements.

The count.

5.36.4.3 Elements

```
List<string> MB2D.UI.ListControl.Elements [get], [set]
```

Gets or sets a list of all of the list control elements

The elements.

5.36.4.4 ItemSpan

```
int MB2D.UI.ListControl.ItemSpan [get], [set]
```

Gets or sets the span of each item vertically in px.

The item span.

5.36.4.5 SeperatorColor

```
Color MB2D.UI.ListControl.SeperatorColor [get], [set]
```

Gets or sets the color of the seperator between list elements.

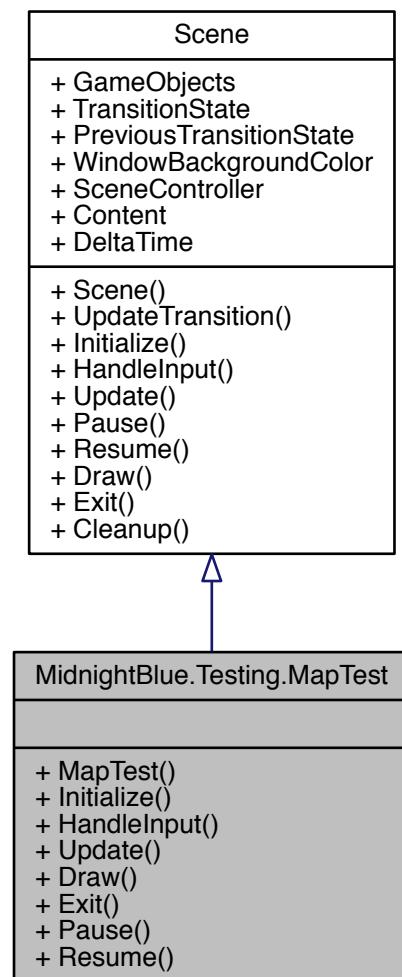
The color of the seperator.

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

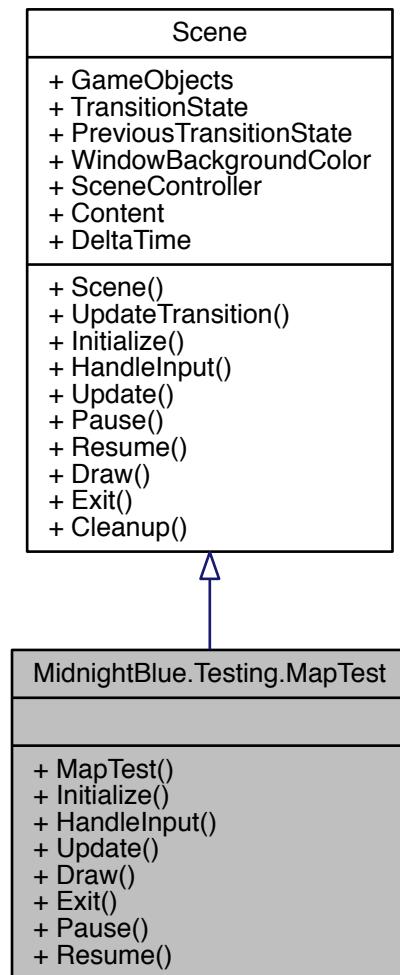
- MB2D/src/UI/ListControl.cs

5.37 MidnightBlue.Testing.MapTest Class Reference

Inheritance diagram for MidnightBlue.Testing.MapTest:



Collaboration diagram for `MidnightBlue.Testing.MapTest`:



Public Member Functions

- **MapTest** (`EntityMap` map, `ContentManager` content)
- override void `Initialize ()`

Initialize this scene and loads all resources. Runs logic to execute during the Initializing state. Set state to None to end.

- override void `HandleInput ()`

Handles all input for the scene

- override void `Update ()`

Updates game logic and changes scene state.

- override void `Draw` (`SpriteBatch` spriteBatch, `SpriteBatch` uiSpriteBatch)

Draws entities and UI elements to the specified SpriteBatches

- override void `Exit ()`

Runs logic to execute while the scene is in the Exiting state. Set state to Null to end.

- `override void Pause ()`
Runs logic to execute while the scene is in the Pausing state. Set state to None to end.
- `override void Resume ()`
Runs logic to execute while the scene is in the Resuming state. Set state to None to end.

Additional Inherited Members

5.37.1 Member Function Documentation

5.37.1.1 Draw()

```
override void MidnightBlue.Testing.MapTest.Draw (
    SpriteBatch spriteBatch,
    SpriteBatch uiSpriteBatch ) [inline], [virtual]
```

Draws entities and UI elements to the specified SpriteBatches

Parameters

<code>spriteBatch</code>	World-coordinate based sprite batch.
<code>uiSpriteBatch</code>	Camera-based User Interface sprite batch.

Implements [MB2D.Scenes.Scene](#).

5.37.1.2 Exit()

```
override void MidnightBlue.Testing.MapTest.Exit () [inline], [virtual]
```

Runs logic to execute while the scene is in the Exiting state. Set state to Null to end.

Implements [MB2D.Scenes.Scene](#).

5.37.1.3 HandleInput()

```
override void MidnightBlue.Testing.MapTest.HandleInput () [inline], [virtual]
```

Handles all input for the scene

Implements [MB2D.Scenes.Scene](#).

5.37.1.4 Initialize()

```
override void MidnightBlue.Testing.MapTest.Initialize () [inline], [virtual]
```

Initialize this scene and loads all resources. Runs logic to execute during the Initializing state. Set state to None to end.

Implements [MB2D.Scenes.Scene](#).

5.37.1.5 Pause()

```
override void MidnightBlue.Testing.MapTest.Pause ( ) [inline], [virtual]
```

Runs logic to execute while the scene is in the Pausing state. Set state to None to end.

Implements [MB2D.Scenes.Scene](#).

5.37.1.6 Resume()

```
override void MidnightBlue.Testing.MapTest.Resume ( ) [inline], [virtual]
```

Runs logic to execute while the scene is in the Resuming state. Set state to None to end.

Implements [MB2D.Scenes.Scene](#).

5.37.1.7 Update()

```
override void MidnightBlue.Testing.MapTest.Update ( ) [inline], [virtual]
```

Updates game logic and changes scene state.

Implements [MB2D.Scenes.Scene](#).

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

- Shared/src/Game/Tests/MapTest.cs

5.38 MB2D.MBConsole Class Reference

Midnight Blue debug console class. Executes attached methods and changes attached variables.

Collaboration diagram for MB2D.MBConsole:

MB2D.MBConsole
+ Display + BGColor + TextColor + Vars + Funcs + LastOutput
+ MBConsole() + InitWindow() + AddFunc() + AddVar() + Update() + Draw() + Write() + Write() + Write() + Debug() + Debug() + Debug() + Debug() + Toggle()

Public Member Functions

- **MBConsole** (Color bgColor, Color txtColor, SpriteFont font)
Initializes a new instance of the T:MidnightBlue.MBConsole class.
- void **InitWindow** (GraphicsDevice graphics)
Initializes a graphics target to render the console to
- void **AddFunc** (string name, Action< string[]> func)
Adds a new function to the console for executing in game
- void **AddVar** (string name, object variable)
Adds a new variable to the console for altering in game
- void **Update** ()
Updates any animation until no longer in an animation state. Otherwise calls ProcessInput()
- void **Draw** (SpriteBatch spriteBatch)
Draws the console and associated text to the attached window
- void **Write** (string line)
Writes a line to the console to display
- void **Write** (string line, params string[] args)
Writes a line to the console to display with specified string format information.
- void **Write** (string line, params object[] args)
Writes a line to the console to display with specified string format information.
- void **Debug** (string line, params object[] args)
Writes a debug line to the console with the specified string format information
- void **Debug** (int line, params object[] args)
Writes a debug line to the console with the specified string format information
- void **Debug** (uint line, params object[] args)
Writes a debug line to the console with the specified string format information
- void **Debug** (float line, params object[] args)
Writes a debug line to the console with the specified string format information
- void **Toggle** ()
Toggles the display/hide state of the console

Properties

- bool **Display** [get, set]
Determines if the console is currently shown or hidden
- Color **BGColor** [get]
Gets the background color of the console
- Color **TextColor** [get]
Gets the color of the console text.
- Dictionary< string, object > **Vars** [get]
Gets the consoles game variables.
- Dictionary< string, Action< string[]> > **Funcs** [get]
Gets the consoles game functions.
- string **LastOutput** [get]

5.38.1 Detailed Description

Midnight Blue debug console class. Executes attached methods and changes attached variables.

5.38.2 Constructor & Destructor Documentation

5.38.2.1 MBConsole()

```
MB2D.MBConsole.MBConsole (
    Color bgColor,
    Color txtColor,
    SpriteFont font ) [inline]
```

Initializes a new instance of the T:MidnightBlue.MBConsole class.

Parameters

<i>bgColor</i>	Background color for rendering the console
<i>txtColor</i>	Text color

5.38.3 Member Function Documentation

5.38.3.1 AddFunc()

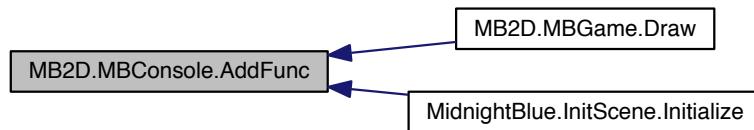
```
void MB2D.MBConsole.AddFunc (
    string name,
    Action< string[]> func ) [inline]
```

Adds a new function to the console for executing in game

Parameters

<i>name</i>	Name to use when calling the function in game
<i>func</i>	Function to attach

Here is the caller graph for this function:



5.38.3.2 AddVar()

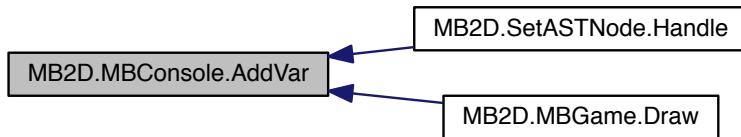
```
void MB2D.MBConsole.AddVar (
    string name,
    object variable ) [inline]
```

Adds a new variable to the console for altering in game

Parameters

<i>name</i>	Name to use when altering the variable in game
<i>variable</i>	Variable to attach

Here is the caller graph for this function:

**5.38.3.3 Debug() [1/4]**

```
void MB2D.MBConsole.Debug (
    string line,
    params object [] args ) [inline]
```

Writes a debug line to the console with the specified string format information

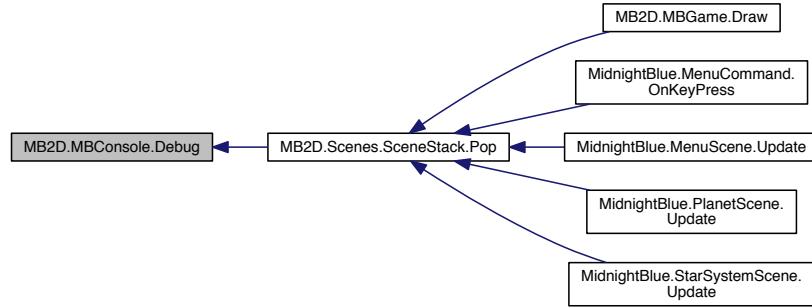
Parameters

<i>line</i>	Line to write
<i>args</i>	Arguments to format into string

Here is the call graph for this function:



Here is the caller graph for this function:



5.38.3.4 Debug() [2/4]

```
void MB2D.MBConsole.Debug (
    int line,
    params object [] args ) [inline]
```

Writes a debug line to the console with the specified string format information

Parameters

<i>line</i>	Line to write
<i>args</i>	Arguments to format into string

Here is the call graph for this function:



5.38.3.5 Debug() [3/4]

```
void MB2D.MBConsole.Debug (
    uint line,
    params object [] args ) [inline]
```

Writes a debug line to the console with the specified string format information

Parameters

<i>line</i>	Line to write
<i>args</i>	Arguments to format into string

Here is the call graph for this function:

**5.38.3.6 Debug() [4/4]**

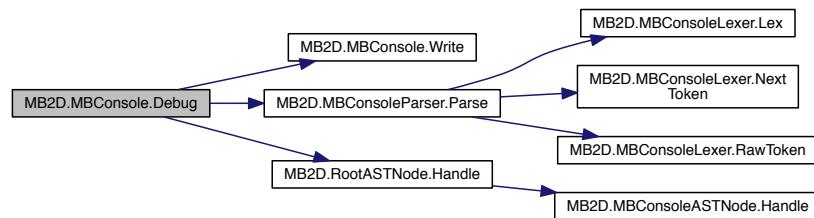
```
void MB2D.MBConsole.Debug (
    float line,
    params object [] args ) [inline]
```

Writes a debug line to the console with the specified string format information

Parameters

<i>line</i>	Line to write
<i>args</i>	Arguments to format into string

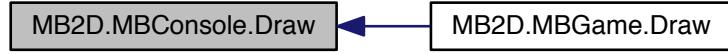
Here is the call graph for this function:

**5.38.3.7 Draw()**

```
void MB2D.MBConsole.Draw (
    SpriteBatch spriteBatch ) [inline]
```

Draws the console and associated text to the attached window

Here is the caller graph for this function:



5.38.3.8 InitWindow()

```
void MB2D.MBConsole.InitWindow (
    GraphicsDevice graphics ) [inline]
```

Initializes a graphics target to render the console to

Parameters

<i>graphics</i>	GraphicsDevice to use for rendering
-----------------	-------------------------------------

Here is the caller graph for this function:



5.38.3.9 Toggle()

```
void MB2D.MBConsole.Toggle ( ) [inline]
```

Toggles the display/hide state of the console

Here is the caller graph for this function:



5.38.3.10 Update()

```
void MB2D.MBConsole.Update ( ) [inline]
```

Updates any animation until no longer in an animation state. Otherwise calls ProcessInput()

Here is the caller graph for this function:



5.38.3.11 Write() [1/3]

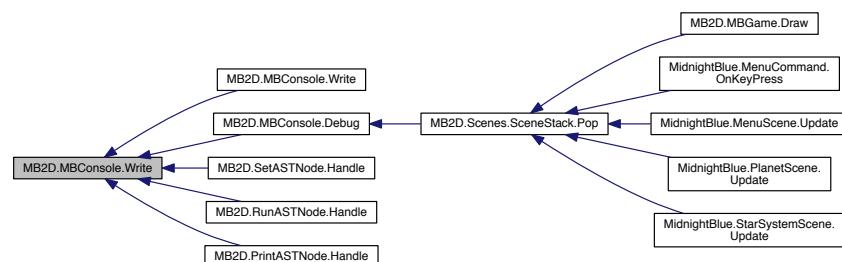
```
void MB2D.MBConsole.Write (
    string line ) [inline]
```

Writes a line to the console to display

Parameters

<i>line</i>	Line to write
-------------	---------------

Here is the caller graph for this function:



5.38.3.12 Write() [2/3]

```
void MB2D.MBConsole.Write (
    string line,
    params string [] args ) [inline]
```

Writes a line to the console to display with specified string format information.

Parameters

<i>line</i>	Line to write
<i>args</i>	Arguments to add to string format

Here is the call graph for this function:

**5.38.3.13 Write() [3/3]**

```
void MB2D.MBConsole.Write (
    string line,
    params object [] args ) [inline]
```

Writes a line to the console to display with specified string format information.

Parameters

<i>line</i>	Line to write
<i>args</i>	Arguments to format into a string

Here is the call graph for this function:

**5.38.4 Property Documentation****5.38.4.1 BGColor**

```
Color MB2D.MBConsole.BGColor [get]
```

Gets the background color of the console

The background color.

5.38.4.2 Display

```
bool MB2D.MBConsole.Display [get], [set]
```

Determines if the console is currently shown or hidden

true if shown; otherwise, false.

5.38.4.3 Funcs

```
Dictionary<string, Action<string[]>> MB2D.MBConsole.Funcs [get]
```

Gets the consoles game functions.

The functions.

5.38.4.4 TextColor

```
Color MB2D.MBConsole.TextColor [get]
```

Gets the color of the console text.

The color of the text.

5.38.4.5 Vars

```
Dictionary<string, object> MB2D.MBConsole.Vars [get]
```

Gets the consoles game variables.

The variables.

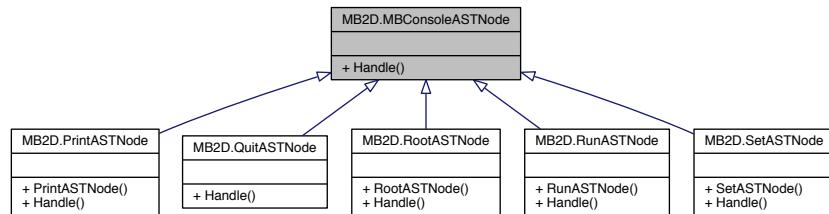
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsole.cs

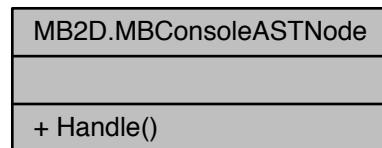
5.39 MB2D.MBConsoleASTNode Class Reference

Class all AST nodes inherit from

Inheritance diagram for MB2D.MBConsoleASTNode:



Collaboration diagram for MB2D.MBConsoleASTNode:



Public Member Functions

- abstract void [Handle \(MBConsole console\)](#)
Executes specific logic on the console

5.39.1 Detailed Description

Class all AST nodes inherit from

5.39.2 Member Function Documentation

5.39.2.1 Handle()

```
abstract void MB2D.MBConsoleASTNode.Handle (
    MBConsole console ) [pure virtual]
```

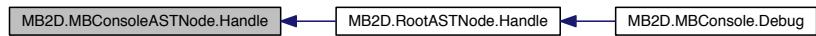
Executes specific logic on the console

Parameters

<code>console</code>	Game console.
----------------------	---------------

Implemented in [MB2D.QuitASTNode](#), [MB2D.PrintASTNode](#), [MB2D.RunASTNode](#), [MB2D.SetASTNode](#), and [MB2D.RootASTNode](#).

Here is the caller graph for this function:



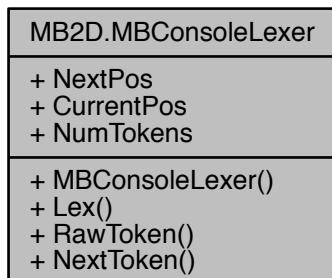
The documentation for this class was generated from the following file:

- [MB2D/src/MBConsole/MBConsoleAST.cs](#)

5.40 MB2D.MBConsoleLexer Class Reference

Breaks a string into a series of tokens to use for parsing the debug consoles command language

Collaboration diagram for MB2D.MBConsoleLexer:



Public Member Functions

- **MBConsoleLexer ()**
Initializes a new instance of the T:MidnightBlue.MBConsoleLexer class.
- void **Lex (string command)**
Lexes the command string breaking it up into token representation and a second raw string array for retrieving values.
- string **RawToken (int index)**
Gets an untokenized representation of a string at a specific index
- Token **NextToken ()**
Gets the next token in the tokenized representation of the command

Properties

- int **NextPos** [get]
Gets the next token index in the lexer
- int **CurrentPos** [get]
Gets the current token index in the lexer
- int **NumTokens** [get]
Gets the number of tokens scanned.

5.40.1 Detailed Description

Breaks a string into a series of tokens to use for parsing the debug consoles command language

5.40.2 Constructor & Destructor Documentation

5.40.2.1 MBConsoleLexer()

```
MB2D.MBConsoleLexer.MBConsoleLexer ( ) [inline]
```

Initializes a new instance of the T:MidnightBlue.MBConsoleLexer class.

5.40.3 Member Function Documentation

5.40.3.1 Lex()

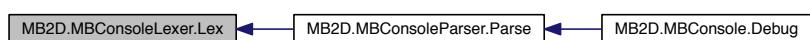
```
void MB2D.MBConsoleLexer.Lex (
    string command ) [inline]
```

Lexes the command string breaking it up into token representation and a second raw string array for retrieving values.

Parameters

<i>command</i>	Command string to scan.
----------------	-------------------------

Here is the caller graph for this function:

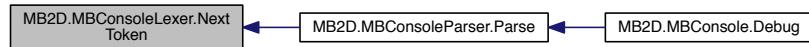


5.40.3.2 NextToken()

```
Token MB2D.MBConsoleLexer.NextToken ( ) [inline]
```

Gets the next token in the tokenized representation of the command

The next token. Here is the caller graph for this function:



5.40.3.3 RawToken()

```
string MB2D.MBConsoleLexer.RawToken (
    int index ) [inline]
```

Gets an untokenized representation of a string at a specific index

Returns

The string representation.

Parameters

<i>index</i>	Index.
--------------	--------

Here is the caller graph for this function:



5.40.4 Property Documentation

5.40.4.1 CurrentPos

```
int MB2D.MBConsoleLexer.CurrentPos [get]
```

Gets the current token index in the lexer

The current position.

5.40.4.2 NextPos

```
int MB2D.MBConsoleLexer.NextPos [get]
```

Gets the next token index in the lexer

The next position.

5.40.4.3 NumTokens

```
int MB2D.MBConsoleLexer.NumTokens [get]
```

Gets the number of tokens scanned.

The number of tokens.

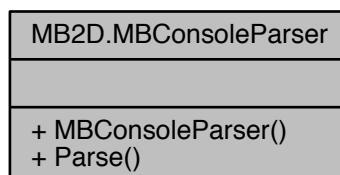
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleLexer.cs

5.41 MB2D.MBConsoleParser Class Reference

Parses command string input and executes it using a debug console.

Collaboration diagram for MB2D.MBConsoleParser:



Public Member Functions

- [MBConsoleParser \(\)](#)
Initializes a new instance of the T:MB2D.MBConsoleParser class.
- [RootASTNode Parse \(string command\)](#)
Processes the command string and executes to the given console.

5.41.1 Detailed Description

Parses command string input and executes it using a debug console.

5.41.2 Constructor & Destructor Documentation

5.41.2.1 MBConsoleParser()

```
MB2D.MBConsoleParser.MBConsoleParser ( ) [inline]
```

Initializes a new instance of the T:MB2D.MBConsoleParser class.

5.41.3 Member Function Documentation

5.41.3.1 Parse()

```
RootASTNode MB2D.MBConsoleParser.Parse (
    string command ) [inline]
```

Processes the command string and executes to the given console.

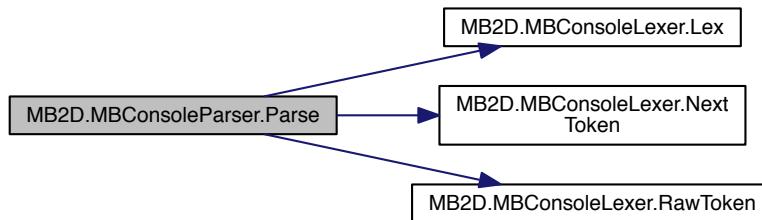
Parameters

<i>console</i>	Console to execute on.
<i>command</i>	Command to parse.

Returns

The entry point for the AST

Here is the call graph for this function:



Here is the caller graph for this function:



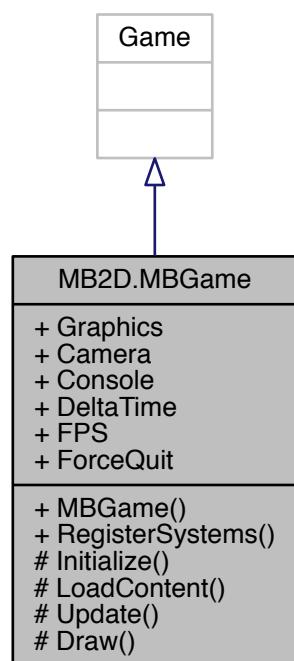
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleParser.cs

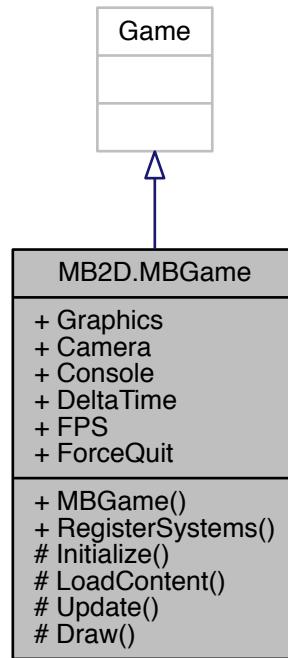
5.42 MB2D.MBGame Class Reference

This is the main type for your game.

Inheritance diagram for MB2D.MBGame:



Collaboration diagram for MB2D.MBGame:



Public Member Functions

- [MBGame](#) (Type initSceneType)

Initializes a new instance of the T:MB2D.MBGame class and defines essential graphics settings.

- void [RegisterSystems](#) ()

Registers all EntitySystems used in the engine

Protected Member Functions

- override void [Initialize](#) ()

Allows the game to perform any initialization it needs to before starting to run. This is where it can query for any required services and load any non-graphic related content. Calling base.Initialize will enumerate through any components and initialize them as well.

- override void [LoadContent](#) ()

Loads content at the beginning of the game

- override void [Update](#) (GameTime gameTime)

Allows the game to run logic such as updating the world, checking for collisions, gathering input, and playing audio.

- override void [Draw](#) (GameTime gameTime)

Draws the current scene to the window

Properties

- static GraphicsDevice [Graphics](#) [get]
Gets the main graphics device.
- static Camera2D [Camera](#) [get]
Gets the main camera.
- static [MBConsole](#) [Console](#) [get]
Gets the debug console for reading and writing to. There should only ever be one of these
- static float [DeltaTime](#) [get]
Gets time it took to complete the last frame
- static float [FPS](#) [get]
Gets the current average frames per second
- static bool [ForceQuit](#) [get, set]
Gets or sets a value indicating whether this T:MB2D.MBGame should quit.

5.42.1 Detailed Description

This is the main type for your game.

5.42.2 Constructor & Destructor Documentation

5.42.2.1 [MBGame\(\)](#)

```
MB2D.MBGame.MBGame (
    Type initSceneType ) [inline]
```

Initializes a new instance of the T:MB2D.MBGame class and defines essential graphics settings.

5.42.3 Member Function Documentation

5.42.3.1 [Draw\(\)](#)

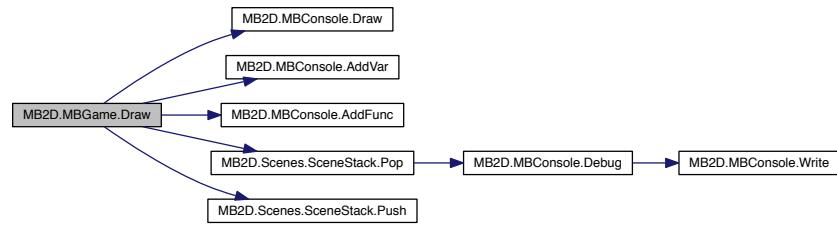
```
override void MB2D.MBGame.Draw (
    GameTime gameTime ) [inline], [protected]
```

Draws the current scene to the window

Parameters

<i>gameTime</i>	Provides a snapshot of timing values.
-----------------	---------------------------------------

Here is the call graph for this function:

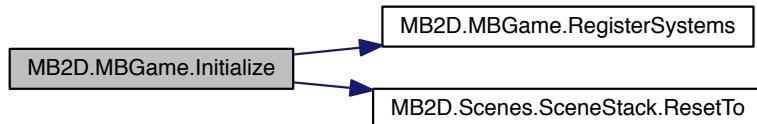


5.42.3.2 Initialize()

```
override void MB2D.MBGame.Initialize ( ) [inline], [protected]
```

Allows the game to perform any initialization it needs to before starting to run. This is where it can query for any required services and load any non-graphic related content. Calling base.Initialize will enumerate through any components and initialize them as well.

Here is the call graph for this function:



5.42.3.3 LoadContent()

```
override void MB2D.MBGame.LoadContent ( ) [inline], [protected]
```

Loads content at the beginning of the game

Here is the call graph for this function:



5.42.3.4 RegisterSystems()

```
void MB2D.MBGame.RegisterSystems ( ) [inline]
```

Registers all EntitySystems used in the engine

Here is the caller graph for this function:



5.42.3.5 Update()

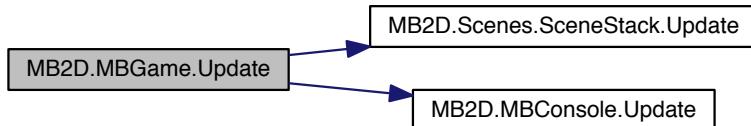
```
override void MB2D.MBGame.Update (
    GameTime gameTime) [inline], [protected]
```

Allows the game to run logic such as updating the world, checking for collisions, gathering input, and playing audio.

Parameters

<i>gameTime</i>	Provides a snapshot of timing values.
-----------------	---------------------------------------

Here is the call graph for this function:



5.42.4 Property Documentation

5.42.4.1 Camera

```
Camera2D MB2D.MBGame.Camera [static], [get]
```

Gets the main camera.

The main camera.

5.42.4.2 Console

```
MBConsole MB2D.MBGame.Console [static], [get]
```

Gets the debug console for reading and writing to. There should only ever be one of these

The debug console.

5.42.4.3 DeltaTime

```
float MB2D.MBGame.deltaTime [static], [get]
```

Gets time it took to complete the last frame

The delta time.

5.42.4.4 ForceQuit

```
bool MB2D.MBGame.ForceQuit [static], [get], [set]
```

Gets or sets a value indicating whether this T:MB2D.MBGame should quit.

`true` if the game should quit; otherwise, `false`.

5.42.4.5 FPS

```
float MB2D.MBGame.FPS [static], [get]
```

Gets the current average frames per second

The fps.

5.42.4.6 Graphics

```
GraphicsDevice MB2D.MBGame.Graphics [static], [get]
```

Gets the main graphics device.

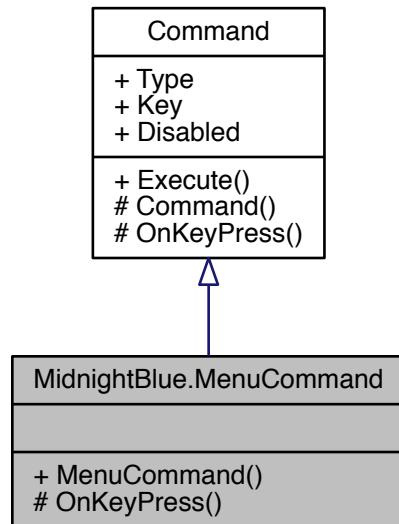
The graphics device.

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

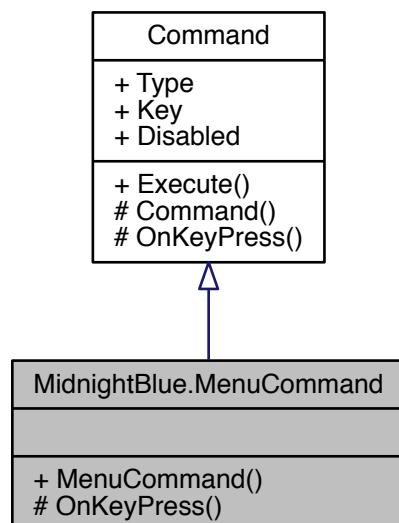
- MB2D/src/MBGame.cs

5.43 MidnightBlue.MenuCommand Class Reference

Inheritance diagram for MidnightBlue.MenuCommand:



Collaboration diagram for MidnightBlue.MenuCommand:



Public Member Functions

- **MenuCommand** (Keys key, CommandType commandType, SceneStack sceneController, ContentManager content)

Protected Member Functions

- override void **OnKeyPress** (Entity e=null)
Defines the logic to perform when operating on a given entity

Additional Inherited Members

5.43.1 Member Function Documentation

5.43.1.1 OnKeyPress()

```
override void MidnightBlue.MenuCommand.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

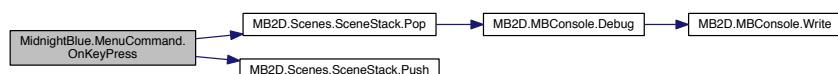
Defines the logic to perform when operating on a given entity

Parameters

e	Entity to operate on
---	----------------------

Implements [MB2D.IO.Command](#).

Here is the call graph for this function:

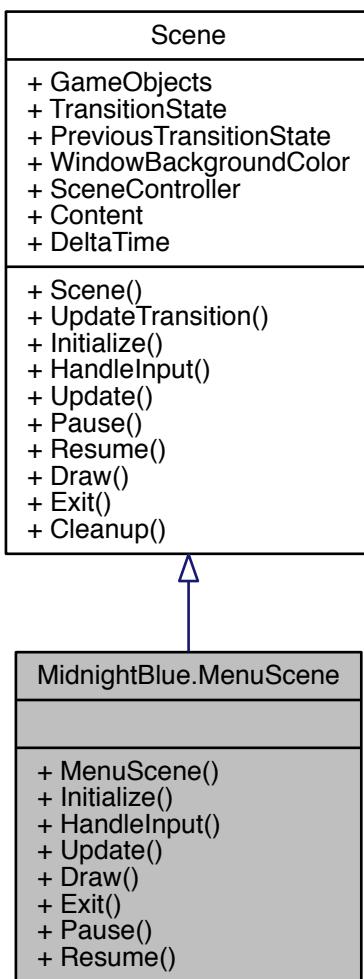


The documentation for this class was generated from the following file:

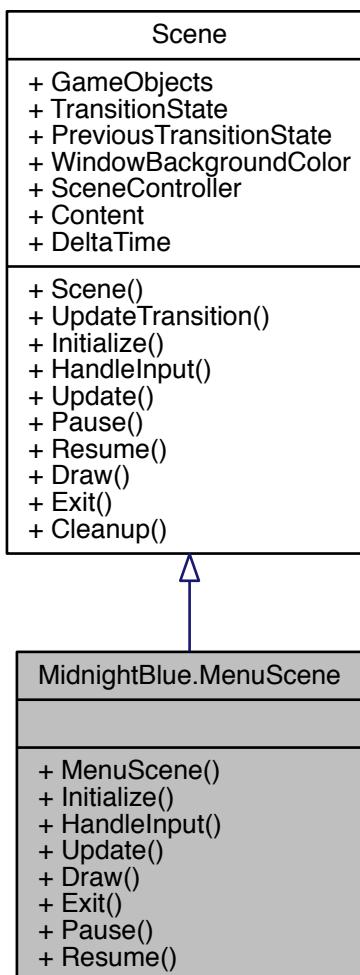
- Shared/src/Game/Commands/MenuCommand.cs

5.44 MidnightBlue.MenuScene Class Reference

Inheritance diagram for MidnightBlue.MenuScene:



Collaboration diagram for MidnightBlue.MenuScene:



Public Member Functions

- **MenuScene** (ContentManager content)
- override void **Initialize** ()
Creates the UI View and starts the background music.
- override void **HandleInput** ()
Handles the input for the menu.
- override void **Update** ()
Updates the UI
- override void **Draw** (SpriteBatch spriteBatch, SpriteBatch uiSpriteBatch)
Draws the UI to the uiSpriteBatch
- override void **Exit** ()
Exits the menu
- override void **Pause** ()

- Pauses the scene*
- `override void Resume ()`
- Resumes the scene*

Additional Inherited Members

5.44.1 Member Function Documentation

5.44.1.1 Draw()

```
override void MidnightBlue.MenuScene.Draw (
    SpriteBatch spriteBatch,
    SpriteBatch uiSpriteBatch ) [inline], [virtual]
```

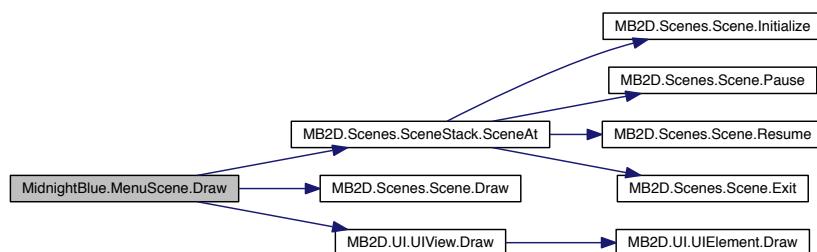
Draws the UI to the uiSpriteBatch

Parameters

<code>spriteBatch</code>	Sprite batch for world-based entities.
<code>uiSpriteBatch</code>	User interface sprite batch.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.44.1.2 Exit()

```
override void MidnightBlue.MenuScene.Exit ( ) [inline], [virtual]
```

Exits the menu

Implements [MB2D.Scenes.Scene](#).

5.44.1.3 HandleInput()

```
override void MidnightBlue.MenuScene.HandleInput ( ) [inline], [virtual]
```

Handles the input for the menu.

Implements [MB2D.Scenes.Scene](#).

5.44.1.4 Initialize()

```
override void MidnightBlue.MenuScene.Initialize ( ) [inline], [virtual]
```

Creates the UIView and starts the background music.

Implements [MB2D.Scenes.Scene](#).

5.44.1.5 Pause()

```
override void MidnightBlue.MenuScene.Pause ( ) [inline], [virtual]
```

Pauses the scene

Implements [MB2D.Scenes.Scene](#).

5.44.1.6 Resume()

```
override void MidnightBlue.MenuScene.Resume ( ) [inline], [virtual]
```

Resumes the scene

Implements [MB2D.Scenes.Scene](#).

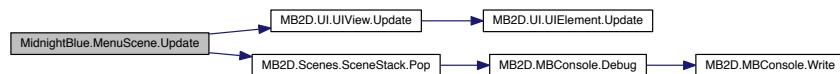
5.44.1.7 Update()

```
override void MidnightBlue.MenuScene.Update ( ) [inline], [virtual]
```

Updates the UI

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:

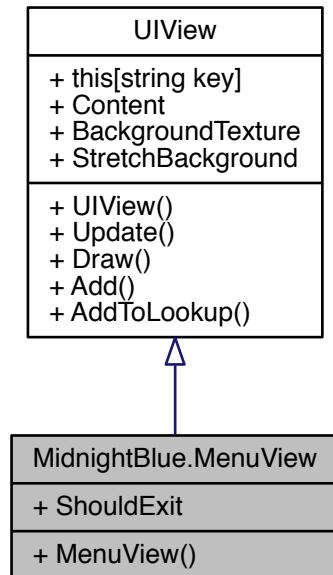


The documentation for this class was generated from the following file:

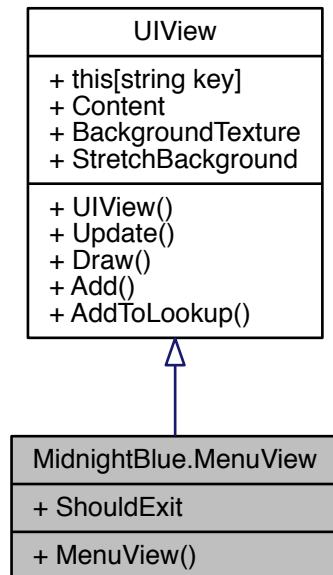
- Shared/src/Game/Scenes/MenuScene.cs

5.45 MidnightBlue.MenuView Class Reference

Inheritance diagram for MidnightBlue.MenuView:



Collaboration diagram for `MidnightBlue.MenuView`:



Public Member Functions

- **MenuView** (ContentManager content)

Properties

- bool **ShouldExit** [get]

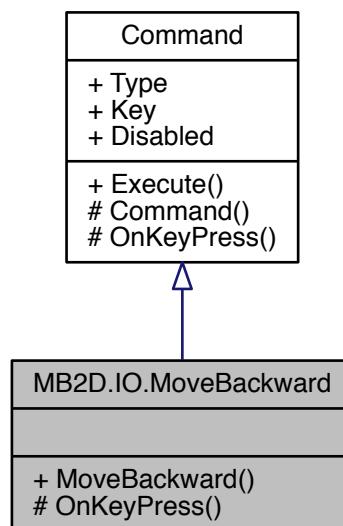
The documentation for this class was generated from the following file:

- Shared/src/Game/UIViews/MenuView.cs

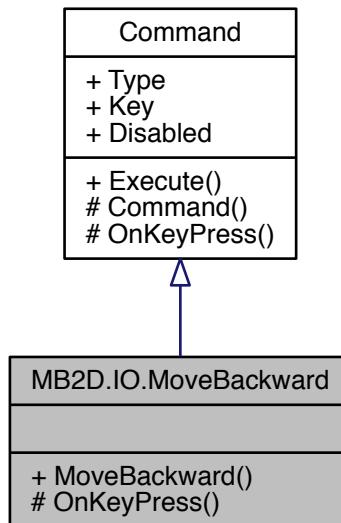
5.46 MB2D.IO.MoveBackward Class Reference

Moves an entity backward. Only runs on entities with a physics component

Inheritance diagram for MB2D.IO.MoveBackward:



Collaboration diagram for MB2D.IO.MoveBackward:



Public Member Functions

- **MoveBackward** ([Keys key](#), [CommandType type](#))
Initializes a new instance of the T:MB2D.IO.MoveBackward class.

Protected Member Functions

- [override void OnKeyPress \(Entity e=null\)](#)
Move an entity forward based on their velocity

Additional Inherited Members

5.46.1 Detailed Description

Moves an entity backward. Only runs on entities with a physics component

5.46.2 Constructor & Destructor Documentation

5.46.2.1 MoveBackward()

```
MB2D.IO.MoveBackward.MoveBackward (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MB2D.IO.MoveBackward class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.46.3 Member Function Documentation**5.46.3.1 OnKeyPress()**

```
override void MB2D.IO.MoveBackward.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Move an entity forward based on their velocity

Parameters

<i>e</i>	Entity to move.
----------	-----------------

Implements [MB2D.IO.Command](#).

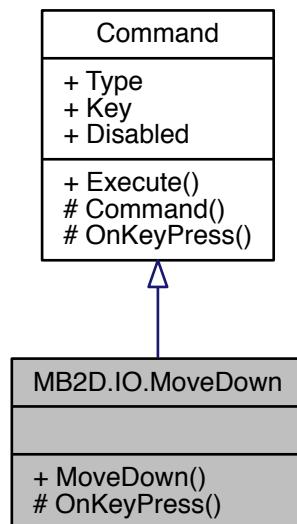
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

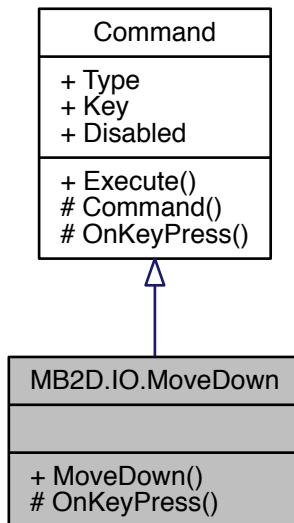
5.47 MB2D.IO.MoveDown Class Reference

Moves an entity down

Inheritance diagram for MB2D.IO.MoveDown:



Collaboration diagram for MB2D.IO.MoveDown:



Public Member Functions

- `MoveDown (Keys key, CommandType type)`
Initializes a new instance of the T:MB2D.IO.MoveDown class.

Protected Member Functions

- `override void OnKeyPress (Entity e=null)`
Move an entity down

Additional Inherited Members

5.47.1 Detailed Description

Moves an entity down

5.47.2 Constructor & Destructor Documentation

5.47.2.1 MoveDown()

```
MB2D.IO.MoveDown.MoveDown (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MB2D.IO.MoveDown class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.47.3 Member Function Documentation**5.47.3.1 OnKeyPress()**

```
override void MB2D.IO.MoveDown.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Move an entity down

Parameters

<i>e</i>	Entity to move.
----------	-----------------

Implements [MB2D.IO.Command](#).

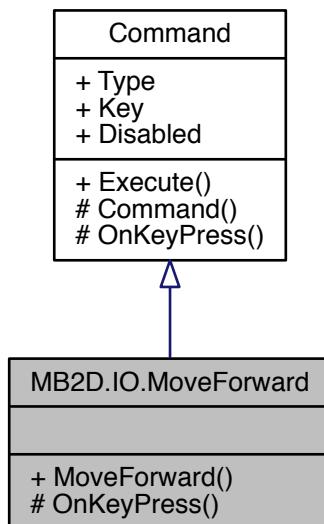
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

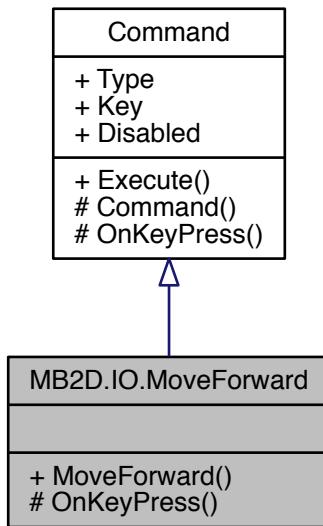
5.48 MB2D.IO.MoveForward Class Reference

Moves an entity forward. Only runs on entities with a physics component

Inheritance diagram for MB2D.IO.MoveForward:



Collaboration diagram for MB2D.IO.MoveForward:



Public Member Functions

- [MoveForward](#) ([Keys key](#), [CommandType type](#))
Initializes a new instance of the T:MB2D.IO.MoveForward class.

Protected Member Functions

- [override void OnKeyPress](#) ([Entity e=null](#))
Move an entity forward based on their velocity

Additional Inherited Members

5.48.1 Detailed Description

Moves an entity forward. Only runs on entities with a physics component

5.48.2 Constructor & Destructor Documentation

5.48.2.1 MoveForward()

```

MB2D.IO.MoveForward.MoveForward (
    Keys key,
    CommandType type ) [inline]
  
```

Initializes a new instance of the T:MB2D.IO.MoveForward class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.48.3 Member Function Documentation**5.48.3.1 OnKeyPress()**

```
override void MB2D.IO.MoveForward.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Move an entity forward based on their velocity

Parameters

<i>e</i>	Entity to move.
----------	-----------------

Implements [MB2D.IO.Command](#).

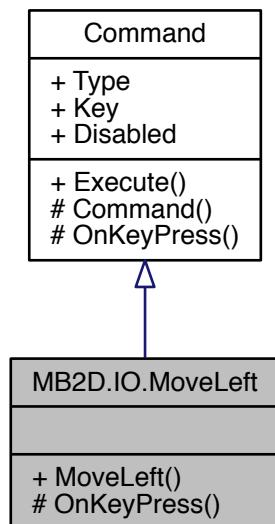
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

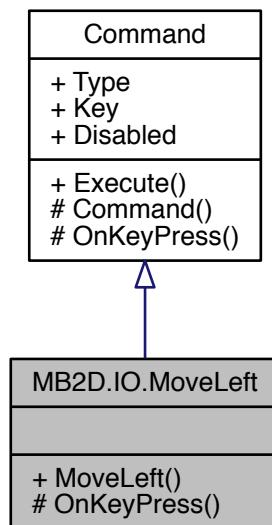
5.49 MB2D.IO.MoveLeft Class Reference

Moves an entity left.

Inheritance diagram for MB2D.IO.MoveLeft:



Collaboration diagram for MB2D.IO.MoveLeft:



Public Member Functions

- `MoveLeft (Keys key, CommandType type)`
Initializes a new instance of the T:MB2D.IO.MoveLeft class.

Protected Member Functions

- `override void OnKeyPress (Entity e=null)`
Move an entity up

Additional Inherited Members

5.49.1 Detailed Description

Moves an entity left.

5.49.2 Constructor & Destructor Documentation

5.49.2.1 MoveLeft()

```

MB2D.IO.MoveLeft.MoveLeft (
    Keys key,
    CommandType type ) [inline]
  
```

Initializes a new instance of the T:MB2D.IO.MoveLeft class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.49.3 Member Function Documentation**5.49.3.1 OnKeyPress()**

```
override void MB2D.IO.MoveLeft.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Move an entity up

Parameters

<i>e</i>	Entity to move.
----------	-----------------

Implements [MB2D.IO.Command](#).

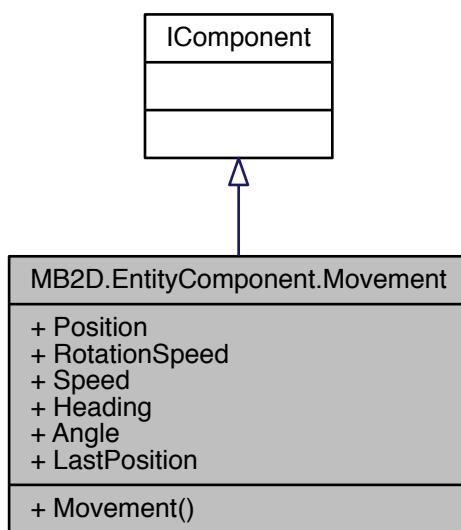
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

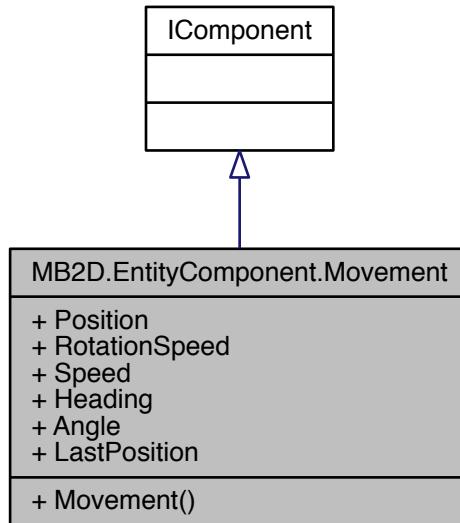
5.50 MB2D.EntityComponent.Movement Class Reference

Defines position, rotation and speed related data for moving an entity.

Inheritance diagram for MB2D.EntityComponent.Movement:



Collaboration diagram for MB2D.EntityComponent.Movement:



Public Member Functions

- **Movement** (float speed=0.0f, float rotationSpeed=0.0f)
Initializes a new instance of the T:MB2D.EntityComponent.Movement class.

Properties

- Vector2 **Position** [get, set]
Gets or sets the world position.
- float **RotationSpeed** [get, set]
Gets or sets the rotation speed.
- float **Speed** [get, set]
Gets or sets the movement speed.
- Vector2 **Heading** [get, set]
Gets or sets the current heading.
- float **Angle** [get, set]
Gets or sets the angle.
- Vector2 **LastPosition** [get, set]
Gets or sets the last known position.

5.50.1 Detailed Description

Defines position, rotation and speed related data for moving an entity.

5.50.2 Constructor & Destructor Documentation

5.50.2.1 Movement()

```
MB2D.EntityComponent.Movement.Movement (
    float speed = 0.0f,
    float rotationSpeed = 0.0f ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.Movement class.

Parameters

<i>speed</i>	Initial speed value.
<i>rotationSpeed</i>	Initial rotation speed value.

5.50.3 Property Documentation

5.50.3.1 Angle

```
float MB2D.EntityComponent.Movement.Angle [get], [set]
```

Gets or sets the angle.

The angle in radians.

5.50.3.2 Heading

```
Vector2 MB2D.EntityComponent.Movement.Heading [get], [set]
```

Gets or sets the current heading.

The heading.

5.50.3.3 LastPosition

```
Vector2 MB2D.EntityComponent.Movement.LastPosition [get], [set]
```

Gets or sets the last known position.

The last position.

5.50.3.4 Position

```
Vector2 MB2D.EntityComponent.Movement.Position [get], [set]
```

Gets or sets the world position.

The position.

5.50.3.5 RotationSpeed

```
float MB2D.EntityComponent.Movement.RotationSpeed [get], [set]
```

Gets or sets the rotation speed.

The rotation speed.

5.50.3.6 Speed

```
float MB2D.EntityComponent.Movement.Speed [get], [set]
```

Gets or sets the movement speed.

The speed.

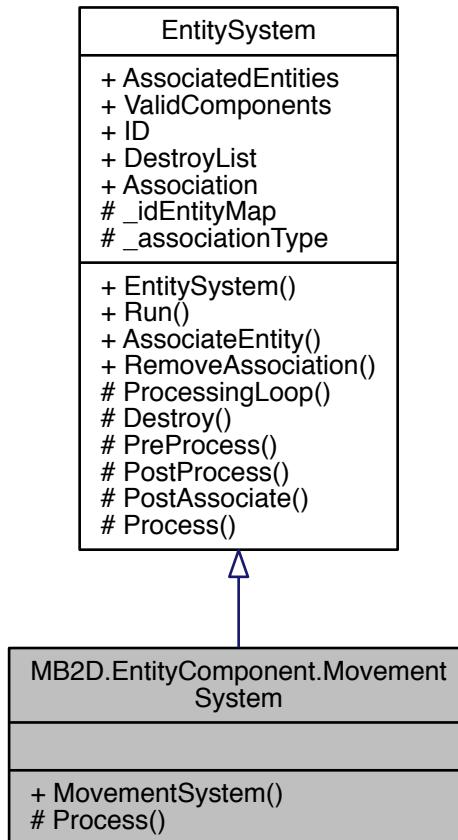
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Components/Movement.cs

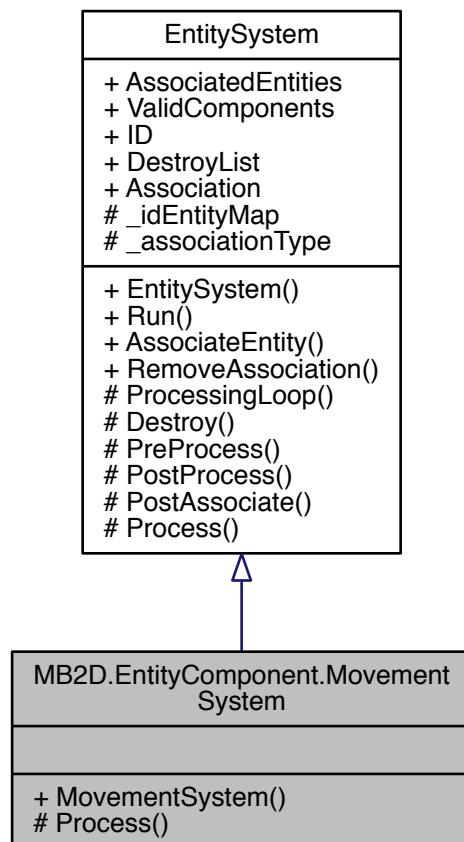
5.51 MB2D.EntityComponent.MovementSystem Class Reference

Processes the change in position, rotation, and sprite transform for an entity

Inheritance diagram for MB2D.EntityComponent.MovementSystem:



Collaboration diagram for MB2D.EntityComponent.MovementSystem:



Public Member Functions

- [MovementSystem \(\)](#)
Initializes a new instance of the T:MB2D.EntityComponent.MovementSystem class.

Protected Member Functions

- `override void Process (Entity entity)`
Processes the movement for the specific entity

Additional Inherited Members

5.51.1 Detailed Description

Processes the change in position, rotation, and sprite transform for an entity

5.51.2 Constructor & Destructor Documentation

5.51.2.1 MovementSystem()

```
MB2D.EntityComponent.MovementSystem.MovementSystem ( ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.MovementSystem class.

5.51.3 Member Function Documentation

5.51.3.1 Process()

```
override void MB2D.EntityComponent.MovementSystem.Process (
    Entity entity) [inline], [protected], [virtual]
```

Processes the movement for the specific entity

Parameters

entity	Entity to operate on.
--------	-----------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

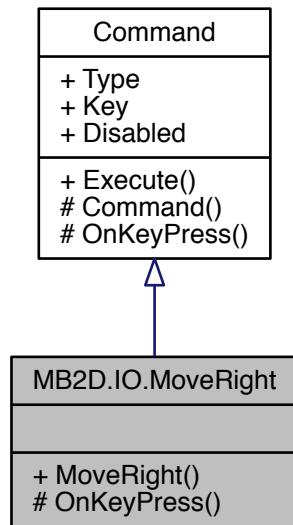
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/MovementSystem.cs

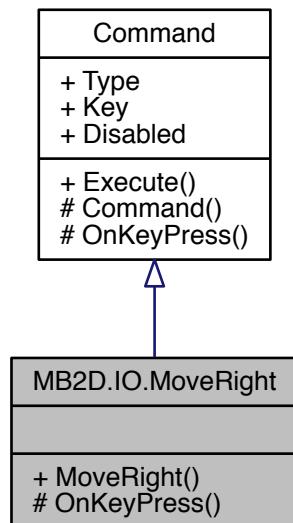
5.52 MB2D.IO.MoveRight Class Reference

Moves an entity right

Inheritance diagram for MB2D.IO.MoveRight:



Collaboration diagram for MB2D.IO.MoveRight:



Public Member Functions

- [MoveRight](#) (`Keys key, CommandType type`)

Initializes a new instance of the T:MB2D.IO.MoveRight class.

Protected Member Functions

- override void [OnKeyPress \(Entity e=null\)](#)
Move an entity right

Additional Inherited Members

5.52.1 Detailed Description

Moves an entity right

5.52.2 Constructor & Destructor Documentation

5.52.2.1 MoveRight()

```
MB2D.IO.MoveRight.MoveRight (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MB2D.IO.MoveRight class.

Parameters

key	Key to assign to.
type	Trigger type.

5.52.3 Member Function Documentation

5.52.3.1 OnKeyPress()

```
override void MB2D.IO.MoveRight.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Move an entity right

Parameters

e	Entity to move.
---	-----------------

Implements [MB2D.IO.Command](#).

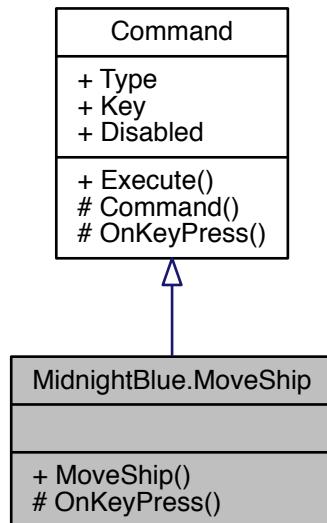
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

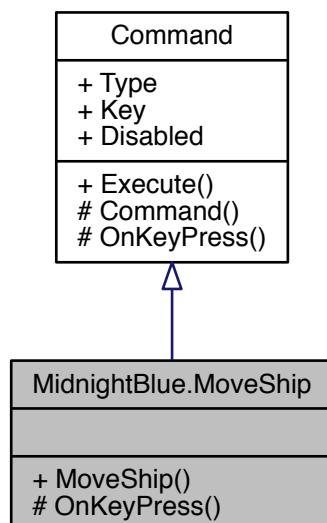
5.53 MidnightBlue.MoveShip Class Reference

Performs logic aside from movement required to execute when moving the ship such as consuming fuel.

Inheritance diagram for MidnightBlue.MoveShip:



Collaboration diagram for `MidnightBlue.MoveShip`:



Public Member Functions

- [MoveShip \(Keys key, CommandType type\)](#)

Initializes a new instance of the T:MidnightBlue.MoveShip class.

Protected Member Functions

- [override void OnKeyPress \(Entity e\)](#)

Consumes fuel, stopping the ship if there's none remaining

Additional Inherited Members

5.53.1 Detailed Description

Performs logic aside from movement required to execute when moving the ship such as consuming fuel.

5.53.2 Constructor & Destructor Documentation

5.53.2.1 MoveShip()

```
MidnightBlue.MoveShip.MoveShip (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MidnightBlue.MoveShip class.

Parameters

<code>key</code>	Key to assign to.
<code>type</code>	Trigger type.

5.53.3 Member Function Documentation

5.53.3.1 OnKeyPress()

```
override void MidnightBlue.MoveShip.OnKeyPress (
    Entity e ) [inline], [protected], [virtual]
```

Consumes fuel, stopping the ship if there's none remaining

Parameters

<code>e</code>	Entity with inventory to operate on.
----------------	--------------------------------------

Implements [MB2D.IO.Command](#).

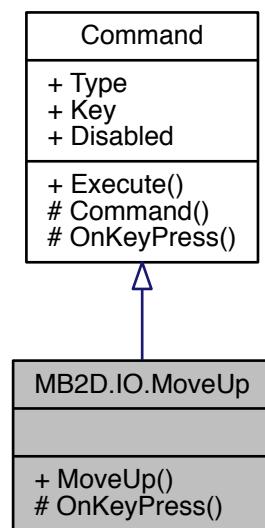
The documentation for this class was generated from the following file:

- Shared/src/Game/Commands/ShipCommands.cs

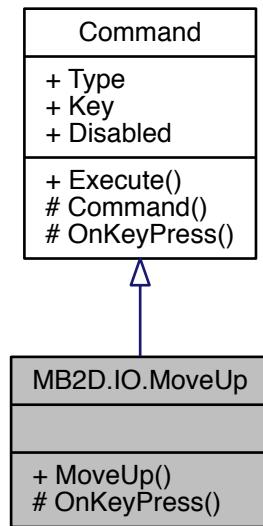
5.54 MB2D.IO.MoveUp Class Reference

Moves a player controller up

Inheritance diagram for MB2D.IO.MoveUp:



Collaboration diagram for MB2D.IO.MoveUp:



Public Member Functions

- **MoveUp** (`Keys key, CommandType type`)
Initializes a new instance of the T:MB2D.IO.MoveUp class.

Protected Member Functions

- `override void OnKeyPress (Entity e=null)`
Move an entity up

Additional Inherited Members

5.54.1 Detailed Description

Moves a player controller up

5.54.2 Constructor & Destructor Documentation

5.54.2.1 MoveUp()

```
MB2D.IO.MoveUp.MoveUp (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MB2D.IO.MoveUp class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.54.3 Member Function Documentation**5.54.3.1 OnKeyPress()**

```
override void MB2D.IO.MoveUp.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Move an entity up

Parameters

<i>e</i>	Entity to move.
----------	-----------------

Implements [MB2D.IO.Command](#).

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

- MB2D/src/Input/MoveCommands.cs

5.55 MidnightBlue.NoiseMap Class Reference

Generates a fractal 2D map using Simplex Noise

Collaboration diagram for `MidnightBlue.NoiseMap`:



Public Member Functions

- **NoiseMap** (`ImplicitModuleBase fractal, int width, int height, int seed`)
Initializes a new instance of the T:MidnightBlue.NoiseMap class. Initializes the fractal generator to use Simplex Noise
- **NoiseMap** (`ImplicitModuleBase fractal, int width, int height`)
Initializes a new instance of the T:MidnightBlue.NoiseMap class. Initializes the fractal generator to use Simplex Noise
- double **GetValue** (`int x, int y`)
Gets a noise value at the specified x and y coordinates. Returned as a normalized value in the range of 0 - 1
- void **SetValue** (`int x, int y, double value`)
Sets a noise value at the specified x and y coordinates. Assigned as a normalized value in the range of 0 - 1

Properties

- int **Width** [get]
Gets the width of the noise map.
- int **Height** [get]
Gets the height of the noise map.
- double **MaxValue** [get]
Gets the maximum value found in the currently generated noise map.
- double **MinValue** [get]
Gets the minimum value found in the currently generated noise map.
- `ImplicitModuleBase Map` [get]
Gets the internal map.

5.55.1 Detailed Description

Generates a fractal 2D map using Simplex Noise

5.55.2 Constructor & Destructor Documentation

5.55.2.1 NoiseMap() [1/2]

```
MidnightBlue.NoiseMap.NoiseMap (
    ImplicitModuleBase fractal,
    int width,
    int height,
    int seed ) [inline]
```

Initializes a new instance of the T:MidnightBlue.NoiseMap class. Initializes the fractal generator to use Simplex Noise

Parameters

<i>width</i>	Width of the noise map.
<i>height</i>	Height of the noise map.
<i>seed</i>	Seed to use in generating the noise map.

5.55.2.2 NoiseMap() [2/2]

```
MidnightBlue.NoiseMap.NoiseMap (
    ImplicitModuleBase fractal,
    int width,
    int height ) [inline]
```

Initializes a new instance of the T:MidnightBlue.NoiseMap class. Initializes the fractal generator to use Simplex Noise

Parameters

<i>width</i>	Width of the noise map.
<i>height</i>	Height of the noise map.

5.55.3 Member Function Documentation

5.55.3.1 GetValue()

```
double MidnightBlue.NoiseMap.GetValue (
    int x,
    int y ) [inline]
```

Gets a noise value at the specified x and y coordinates. Returned as a normalized value in the range of 0 - 1

Returns

The noise value.

Parameters

<i>x</i>	The x coordinate.
<i>y</i>	The y coordinate.

Here is the caller graph for this function:



5.55.3.2 SetValue()

```
void MidnightBlue.NoiseMap.SetValue (
    int x,
    int y,
    double value ) [inline]
```

Sets a noise value at the specified x and y coordinates. Assigned as a normalized value in the range of 0 - 1

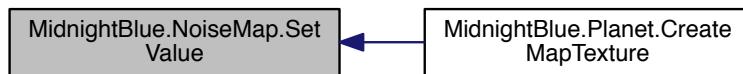
Returns

The noise value.

Parameters

x	The x coordinate.
y	The y coordinate.

Here is the caller graph for this function:



5.55.4 Property Documentation

5.55.4.1 Height

```
int MidnightBlue.NoiseMap.Height [get]
```

Gets the height of the noise map.

The height.

5.55.4.2 Map

```
ImplicitModuleBase MidnightBlue.NoiseMap.Map [get]
```

Gets the internal map.

The map.

5.55.4.3 MaxValue

```
double MidnightBlue.NoiseMap.MaxValue [get]
```

Gets the maximum value found in the currently generated noise map.

The max value.

5.55.4.4 MinValue

```
double MidnightBlue.NoiseMap.MinValue [get]
```

Gets the minimum value found in the currently generated noise map.

The max value.

5.55.4.5 Width

```
int MidnightBlue.NoiseMap.Width [get]
```

Gets the width of the noise map.

The width.

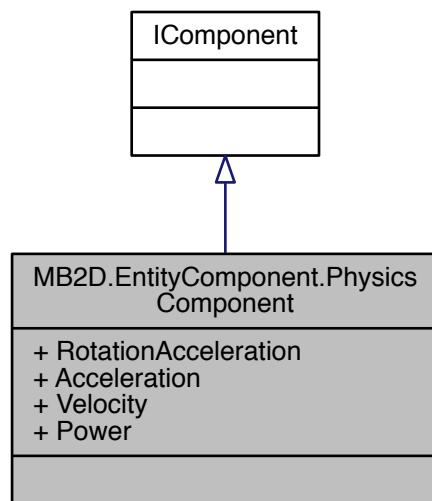
The documentation for this class was generated from the following file:

- Shared/src/Game/Environment/NoiseMap.cs

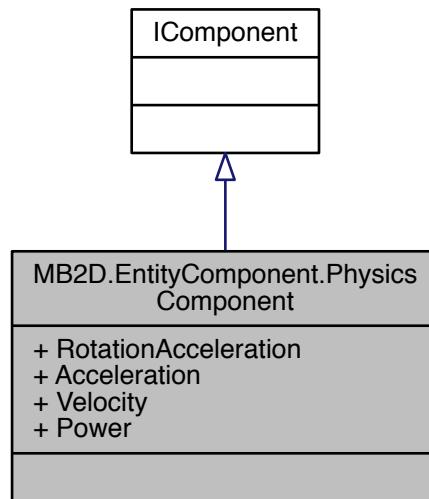
5.56 MB2D.EntityComponent.PhysicsComponent Class Reference

Physics component used to define acceleration and velocity.

Inheritance diagram for MB2D.EntityComponent.PhysicsComponent:



Collaboration diagram for MB2D.EntityComponent.PhysicsComponent:



Properties

- float [RotationAcceleration](#) [get, set]
Gets or sets the current acceleration of the rotation.
- Vector2 [Acceleration](#) [get, set]
Gets or sets the current acceleration.
- Vector2 [Velocity](#) [get, set]
Gets or sets the current positional velocity.
- float [Power](#) [get, set]
Gets or sets the current power applied.

5.56.1 Detailed Description

Physics component used to define acceleration and velocity.

5.56.2 Property Documentation

5.56.2.1 Acceleration

`Vector2 MB2D.EntityComponent.PhysicsComponent.Acceleration [get], [set]`

Gets or sets the current acceleration.

The acceleration.

5.56.2.2 Power

```
float MB2D.EntityComponent.PhysicsComponent.Power [get], [set]
```

Gets or sets the current power applied.

The power applied.

5.56.2.3 RotationAcceleration

```
float MB2D.EntityComponent.PhysicsComponent.RotationAcceleration [get], [set]
```

Gets or sets the current acceleration of the rotation.

The rotation acceleration.

5.56.2.4 Velocity

```
Vector2 MB2D.EntityComponent.PhysicsComponent.Velocity [get], [set]
```

Gets or sets the current positional velocity.

The velocity.

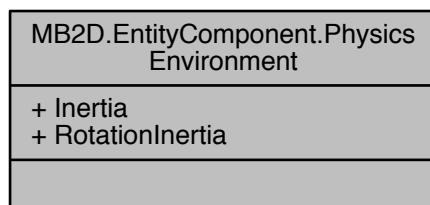
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Components/PhysicsComponent.cs

5.57 MB2D.EntityComponent.PhysicsEnvironment Class Reference

Defines a new environment to feed into the physics system to alter the impact it has on an entity

Collaboration diagram for MB2D.EntityComponent.PhysicsEnvironment:



Properties

- float **Inertia** [get, set]
Gets or sets the inertia of the environment.
- float **RotationInertia** [get, set]
Gets or sets the rotation inertia of the environment.

5.57.1 Detailed Description

Defines a new environment to feed into the physics system to alter the impact it has on an entity

5.57.2 Property Documentation

5.57.2.1 Inertia

```
float MB2D.EntityComponent.PhysicsEnvironment.Inertia [get], [set]
```

Gets or sets the inertia of the environment.

The inertia.

5.57.2.2 RotationInertia

```
float MB2D.EntityComponent.PhysicsEnvironment.RotationInertia [get], [set]
```

Gets or sets the rotation inertia of the environment.

The rotation inertia.

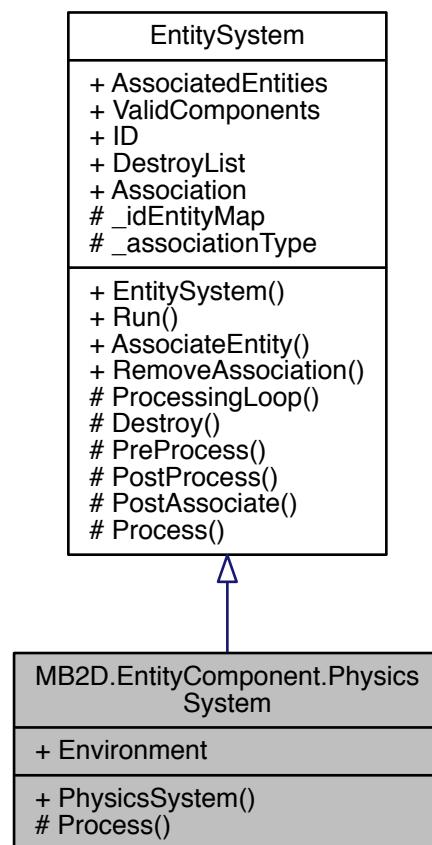
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/PhysicsSystem.cs

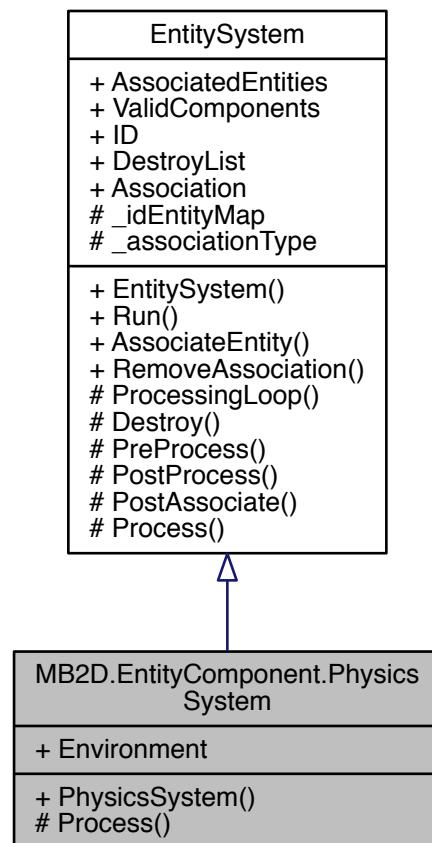
5.58 MB2D.EntityComponent.PhysicsSystem Class Reference

Processes physics changes for a given entity

Inheritance diagram for MB2D.EntityComponent.PhysicsSystem:



Collaboration diagram for MB2D.EntityComponent.PhysicsSystem:



Public Member Functions

- [PhysicsSystem \(\)](#)

Initializes a new instance of the T:MB2D.EntityComponent.PhysicsSystem class.

Protected Member Functions

- `override void Process (Entity entity)`

Updates the entities movement and velocity values based on the current physics environment

Properties

- [PhysicsEnvironment Environment \[get, set\]](#)

Gets or sets the current physics environment.

Additional Inherited Members

5.58.1 Detailed Description

Processes physics changes for a given entity

5.58.2 Constructor & Destructor Documentation

5.58.2.1 PhysicsSystem()

```
MB2D.EntityComponent.PhysicsSystem.PhysicsSystem ( ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.PhysicsSystem class.

5.58.3 Member Function Documentation

5.58.3.1 Process()

```
override void MB2D.EntityComponent.PhysicsSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
```

Updates the entities movement and velocity values based on the current physics environment

Parameters

<i>entity</i>	Entity to process.
---------------	--------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

5.58.4 Property Documentation

5.58.4.1 Environment

```
PhysicsEnvironment MB2D.EntityComponent.PhysicsSystem.Environment [get], [set]
```

Gets or sets the current physics environment.

The environment.

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

- MB2D/src/EntityComponent/Systems/PhysicsSystem.cs

5.59 MidnightBlue.Planet Class Reference

A fully-generated planet in a star system with associated texture maps.

Collaboration diagram for MidnightBlue.Planet:



Public Member Functions

- **Planet** ([PlanetMetadata](#) meta, int seed)
Initializes a new instance of the T:MidnightBlue.Planet class and sets up all noise maps ready for generation.
- void **Generate** (Random rand)
Generates the planet after setting up with pre-defined metadata parameters.
- void **CreateMapTexture** (ContentManager content)
Creates the biome map texture and planet mask texture to use for rendering to star system view and to use as maps.
- Texture2D **GetMapLayer** (string layerName)
Gets one of the planets generated noise map textures

Properties

- **PlanetTile [,] Tiles [get]**
Gets all the tiles in the generated planet
- Point **Size [get]**
Gets the rectangular size of the planets tile map
- string **Name [get]**
Gets the name of the planet
- [PlanetMetadata](#) **Meta [get]**
Gets the planets assigned metadata parameters
- Vector2 **Position [get, set]**
Gets or sets the planets position in the star system scene
- bool **Generated [get]**
Gets a value indicating whether this T:MidnightBlue.Planet is generated or only setup ready to be generated.

5.59.1 Detailed Description

A fully-generated planet in a star system with associated texture maps.

5.59.2 Constructor & Destructor Documentation

5.59.2.1 Planet()

```
MidnightBlue.Planet.Planet (
    PlanetMetadata meta,
    int seed ) [inline]
```

Initializes a new instance of the T:MidnightBlue.Planet class and sets up all noise maps ready for generation.

Parameters

<i>meta</i>	Metadata received from the planet view to act as parameters for generation.
<i>seed</i>	Seed to use in generating the map.

5.59.3 Member Function Documentation

5.59.3.1 CreateMapTexture()

```
void MidnightBlue.Planet.CreateMapTexture (
    ContentManager content ) [inline]
```

Creates the biome map texture and planet mask texture to use for rendering to star system view and to use as maps.

Parameters

<i>content</i>	Content manager for loading textures.
----------------	---------------------------------------

Here is the call graph for this function:



5.59.3.2 Generate()

```
void MidnightBlue.Planet.Generate (
```

```
Random rand ) [inline]
```

Generates the planet after setting up with pre-defined metadata parameters.

Parameters

<i>rand</i>	Random number generator from galaxy view to use in generating the planet.
-------------	---

Here is the call graph for this function:



Here is the caller graph for this function:



5.59.3.3 GetMapLayer()

```
Texture2D MidnightBlue.Planet.GetMapLayer (
    string layerName ) [inline]
```

Gets one of the planets generated noise map textures

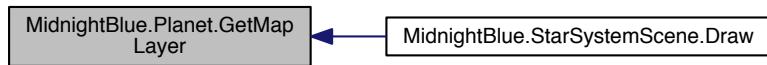
Returns

The map layer.

Parameters

<i>layerName</i>	Layer name.
------------------	-------------

Here is the caller graph for this function:



5.59.4 Property Documentation

5.59.4.1 Generated

```
bool MidnightBlue.Planet.Generated [get]
```

Gets a value indicating whether this T:MidnightBlue.Planet is generated or only setup ready to be generated.

true if generated; otherwise, false.

5.59.4.2 Meta

```
PlanetMetadata MidnightBlue.Planet.Meta [get]
```

Gets the planets assigned metadata parameters

The metadata.

5.59.4.3 Name

```
string MidnightBlue.Planet.Name [get]
```

Gets the name of the planet

The name.

5.59.4.4 Position

```
Vector2 MidnightBlue.Planet.Position [get], [set]
```

Gets or sets the planets position in the star system scene

The position.

5.59.4.5 Size

```
Point MidnightBlue.Planet.Size [get]
```

Gets the rectangular size of the planets tile map

The size of the planet.

5.59.4.6 Tiles

```
PlanetTile [,] MidnightBlue.Planet.Tiles [get]
```

Gets all the tiles in the generated planet

The tiles.

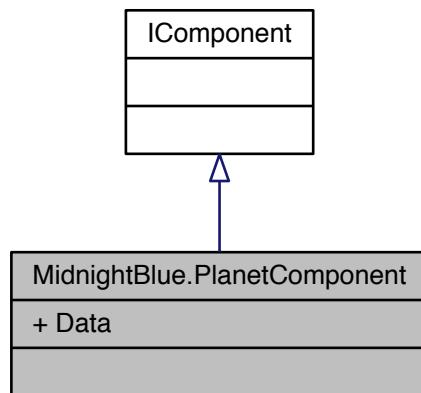
The documentation for this class was generated from the following file:

- Shared/src/Game/Environment/Planet.cs

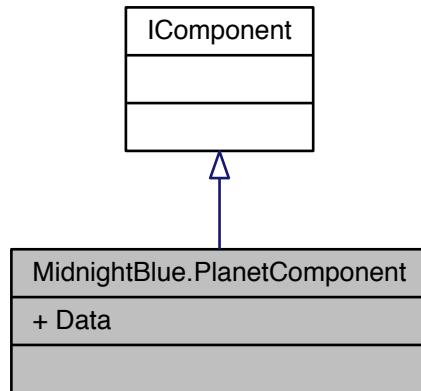
5.60 MidnightBlue.PlanetComponent Class Reference

Represents a planet entity with pre-generated metadata

Inheritance diagram for MidnightBlue.PlanetComponent:



Collaboration diagram for MidnightBlue.PlanetComponent:



Properties

- [Planet Data](#) [get, set]
All pre-generated arguments used when generating a planets map

5.60.1 Detailed Description

Represents a planet entity with pre-generated metadata

5.60.2 Property Documentation

5.60.2.1 Data

[Planet](#) `MidnightBlue.PlanetComponent.Data` [get], [set]

All pre-generated arguments used when generating a planets map

The data.

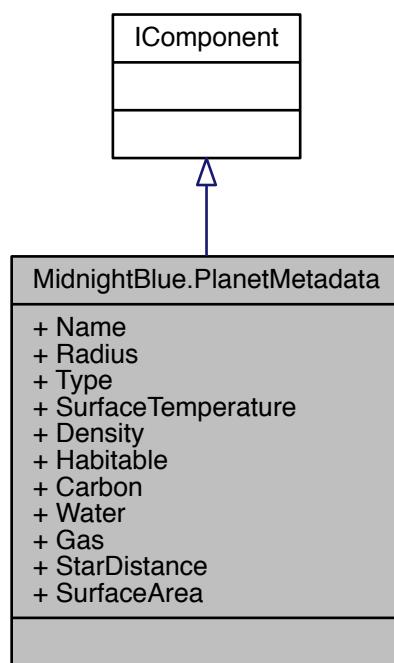
The documentation for this class was generated from the following file:

- Shared/src/Game/Components/PlanetComponent.cs

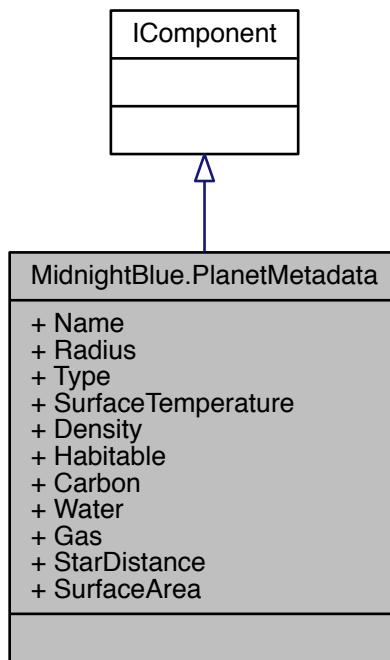
5.61 MidnightBlue.PlanetMetadata Class Reference

[Planet](#) metadata used as information and arguments for generating the actual biome map of a planet. Required for an entity to be treated as a planet.

Inheritance diagram for `MidnightBlue.PlanetMetadata`:



Collaboration diagram for MidnightBlue.PlanetMetadata:



Properties

- string `Name` [get, set]

Gets or sets the name of the planet.
- int `Radius` [get, set]

Gets or sets the radius of the planet.
- `PlanetType Type` [get, set]

Gets or sets the type of the planet.
- float `SurfaceTemperature` [get, set]

Gets or sets the surface temperature.
- int `Density` [get, set]

Gets or sets the density.
- float `Habitable` [get, set]

Gets or sets the score indicating the planets ability to support life.
- int `Carbon` [get, set]

Gets or sets the amount of carbon on the planet.
- int `Water` [get, set]

Gets or sets the amount of water on the planet.
- int `Gas` [get, set]

Gets or sets the amount of gas on the planet.
- `Length StarDistance` [get, set]

Gets or sets the distance of this planet to its star
- float `SurfaceArea` [get]

Gets the surface area of the planet. Used mostly for information displays - not very useful for anything else.

5.61.1 Detailed Description

[Planet](#) metadata used as information and arguments for generating the actual biome map of a planet. Required for an entity to be treated as a planet.

5.61.2 Property Documentation

5.61.2.1 Carbon

```
int MidnightBlue.PlanetMetadata.Carbon [get], [set]
```

Gets or sets the amount of carbon on the planet.

The carbon amount.

5.61.2.2 Density

```
int MidnightBlue.PlanetMetadata.Density [get], [set]
```

Gets or sets the density.

The density.

5.61.2.3 Gas

```
int MidnightBlue.PlanetMetadata.Gas [get], [set]
```

Gets or sets the amount of gas on the planet.

The gas amount.

5.61.2.4 Habitable

```
float MidnightBlue.PlanetMetadata.Habitable [get], [set]
```

Gets or sets the score indicating the planets ability to support life.

The life score.

5.61.2.5 Name

```
string MidnightBlue.PlanetMetadata.Name [get], [set]
```

Gets or sets the name of the planet.

The name.

5.61.2.6 Radius

```
int MidnightBlue.PlanetMetadata.Radius [get], [set]
```

Gets or sets the radius of the planet.

The radius.

5.61.2.7 StarDistance

```
Length MidnightBlue.PlanetMetadata.StarDistance [get], [set]
```

Gets or sets the distance of this planet to its star

The star distance.

5.61.2.8 SurfaceArea

```
float MidnightBlue.PlanetMetadata.SurfaceArea [get]
```

Gets the surface area of the planet. Used mostly for information displays - not very useful for anything else.

The surface area.

5.61.2.9 SurfaceTemperature

```
float MidnightBlue.PlanetMetadata.SurfaceTemperature [get], [set]
```

Gets or sets the surface temperature.

The surface temperature.

5.61.2.10 Type

```
PlanetType MidnightBlue.PlanetMetadata.Type [get], [set]
```

Gets or sets the type of the planet.

The type.

5.61.2.11 Water

```
int MidnightBlue.PlanetMetadata.Water [get], [set]
```

Gets or sets the amount of water on the planet.

The water amount.

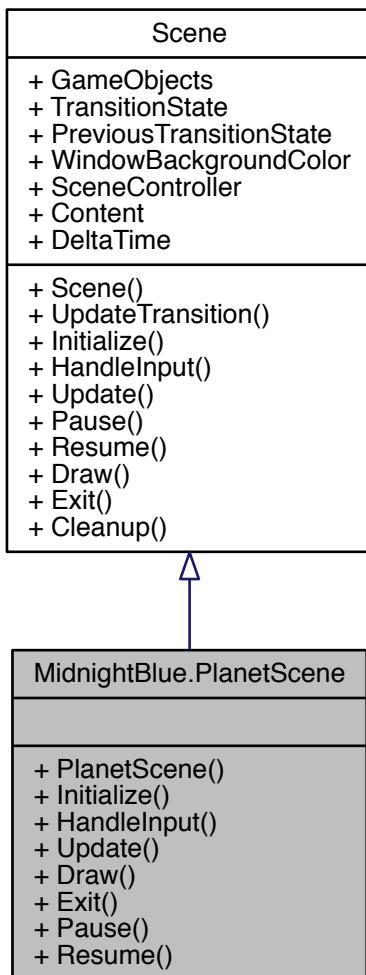
The documentation for this class was generated from the following file:

- Shared/src/Game/Environment/PlanetMetadata.cs

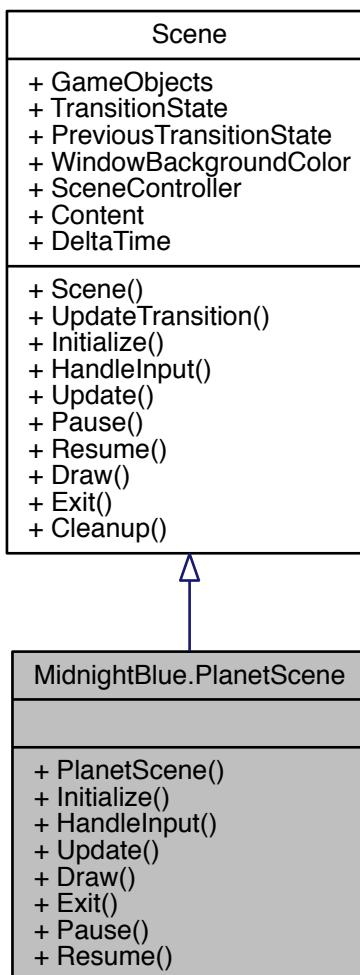
5.62 MidnightBlue.PlanetScene Class Reference

Scene active when the player is exploring a given planet.

Inheritance diagram for `MidnightBlue.PlanetScene`:



Collaboration diagram for MidnightBlue.PlanetScene:



Public Member Functions

- **PlanetScene** ([EntityMap](#) map, [ContentManager](#) content, [Planet](#) planet)
Initializes a new instance of the T:MidnightBlue.PlanetScene class.
- **override void Initialize ()**
Sets up the player and physics environment for this planet
- **override void HandleInput ()**
Handles the input for the scene.
- **override void Update ()**
Updates the players position and state alongside the current biome the player is located at.
- **override void Draw (SpriteBatch spriteBatch, SpriteBatch uiSpriteBatch)**
Draw the tilemap to the specified spriteBatch and uiSpriteBatch.
- **override void Exit ()**
Exit this scene.

- override void [Pause \(\)](#)
Instantly pause the scene
- override void [Resume \(\)](#)
Instantly resume the scene

Additional Inherited Members

5.62.1 Detailed Description

Scene active when the player is exploring a given planet.

5.62.2 Constructor & Destructor Documentation

5.62.2.1 [PlanetScene\(\)](#)

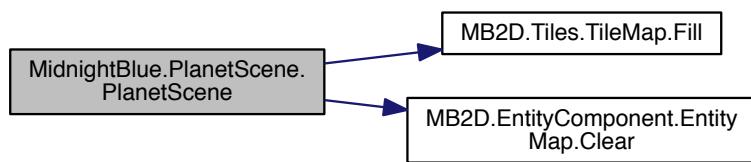
```
MidnightBlue.PlanetScene.PlanetScene (
    EntityMap map,
    ContentManager content,
    Planet planet ) [inline]
```

Initializes a new instance of the T:MidnightBlue.PlanetScene class.

Parameters

<i>map</i>	Game object map.
<i>content</i>	Content manager for loading resources.
<i>planet</i>	Planet to use in this scene.

Here is the call graph for this function:



5.62.3 Member Function Documentation

5.62.3.1 [Draw\(\)](#)

```
override void MidnightBlue.PlanetScene.Draw (
    SpriteBatch spriteBatch,
    SpriteBatch uiSpriteBatch ) [inline], [virtual]
```

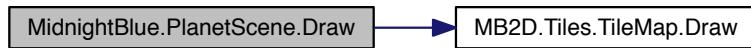
Draw the tilemap to the specified spriteBatch and uiSpriteBatch.

Parameters

<i>spriteBatch</i>	Sprite batch to draw world-based entities to.
<i>uiSpriteBatch</i>	User interface sprite batch.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.62.3.2 Exit()

```
override void MidnightBlue.PlanetScene.Exit ( ) [inline], [virtual]
```

Exit this scene.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.62.3.3 HandleInput()

```
override void MidnightBlue.PlanetScene.HandleInput ( ) [inline], [virtual]
```

Handles the input for the scene.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.62.3.4 Initialize()

```
override void MidnightBlue.PlanetScene.Initialize ( ) [inline], [virtual]
```

Sets up the player and physics environment for this planet

Implements [MB2D.Scenes.Scene](#).

5.62.3.5 Pause()

```
override void MidnightBlue.PlanetScene.Pause ( ) [inline], [virtual]
```

Instantly pause the scene

Implements [MB2D.Scenes.Scene](#).

5.62.3.6 Resume()

```
override void MidnightBlue.PlanetScene.Resume ( ) [inline], [virtual]
```

Instantly resume the scene

Implements [MB2D.Scenes.Scene](#).

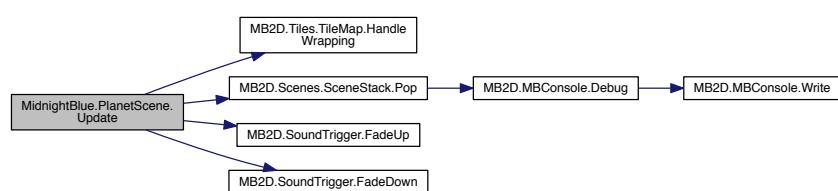
5.62.3.7 Update()

```
override void MidnightBlue.PlanetScene.Update ( ) [inline], [virtual]
```

Updates the players position and state alongside the current biome the player is located at.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



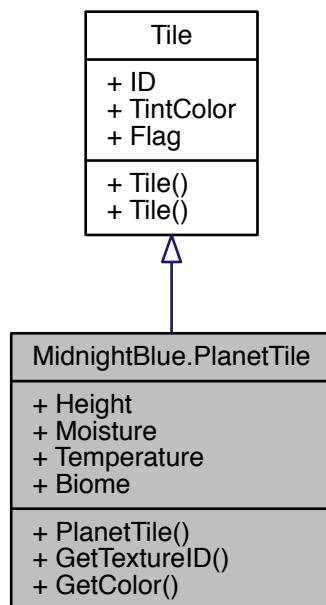
The documentation for this class was generated from the following file:

- Shared/src/Game/Scenes/PlanetScene.cs

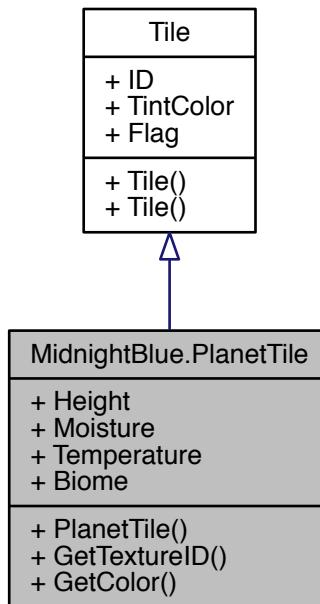
5.63 MidnightBlue.PlanetTile Class Reference

A tile type used in planet tilemaps

Inheritance diagram for MidnightBlue.PlanetTile:



Collaboration diagram for `MidnightBlue.PlanetTile`:



Public Member Functions

- `PlanetTile` (double height, double moisture, double temperature, Random rand)
Initializes a new instance of the T:MidnightBlue.PlanetTile class, generates a biome and specifies collision data based on its biome type.
- int `GetTextureID` (Biome biome, Random rand)
Gets the ID of a biomes texture region for use in a tilemap
- Color `GetColor` (Biome biome)
Gets the color of a tile in the minimap based on its biome.

Properties

- HeightLevel `Height` [get]
Gets the height category of the tile.
- MoistureLevel `Moisture` [get]
Gets the moisture category of the tile.
- TemperatureLevel `Temperature` [get]
Gets the temperature category of the tile.
- Biome `Biome` [get]
Gets the biome category of the tile.

5.63.1 Detailed Description

A tile type used in planet tilemaps

5.63.2 Constructor & Destructor Documentation

5.63.2.1 PlanetTile()

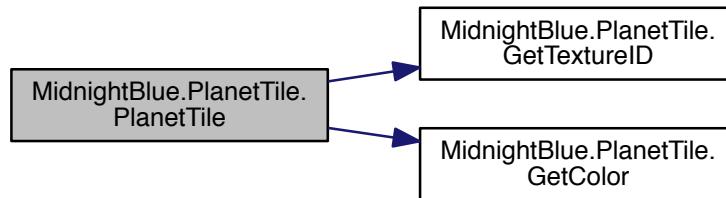
```
MidnightBlue.PlanetTile.PlanetTile (
    double height,
    double moisture,
    double temperature,
    Random rand ) [inline]
```

Initializes a new instance of the T:MidnightBlue.PlanetTile class, generates a biome and specifies collision data based on its biome type.

Parameters

<i>height</i>	Height value to use in generation.
<i>moisture</i>	Moisture value to use in generation.
<i>temperature</i>	Temperature value to use in generation.
<i>rand</i>	Random number generator used in biome generation.

Here is the call graph for this function:



5.63.3 Member Function Documentation

5.63.3.1 GetColor()

```
Color MidnightBlue.PlanetTile.GetColor (
    Biome biome ) [inline]
```

Gets the color of a tile in the minimap based on its biome.

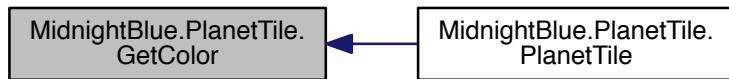
Returns

The color.

Parameters

<i>biome</i>	Biome to get.
--------------	---------------

Here is the caller graph for this function:

**5.63.3.2 GetTextureID()**

```
int MidnightBlue.PlanetTile.GetTextureID (
    Biome biome,
    Random rand ) [inline]
```

Gets the ID of a biomes texture region for use in a tilemap

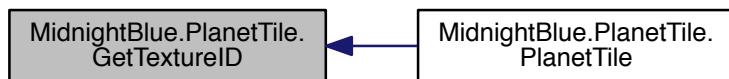
Returns

The texture region identifier.

Parameters

<i>biome</i>	Biome to get.
<i>rand</i>	Random number generator to use for varied region id's.

Here is the caller graph for this function:

**5.63.4 Property Documentation**

5.63.4.1 Biome

```
Biome MidnightBlue.PlanetTile.Biome [get]
```

Gets the biome category of the tile.

The biome category.

5.63.4.2 Height

```
HeightLevel MidnightBlue.PlanetTile.Height [get]
```

Gets the height category of the tile.

The height category.

5.63.4.3 Moisture

```
MoistureLevel MidnightBlue.PlanetTile.Moisture [get]
```

Gets the moisture category of the tile.

The moisture category.

5.63.4.4 Temperature

```
TemperatureLevel MidnightBlue.PlanetTile.Temperature [get]
```

Gets the temperature category of the tile.

The temperature category.

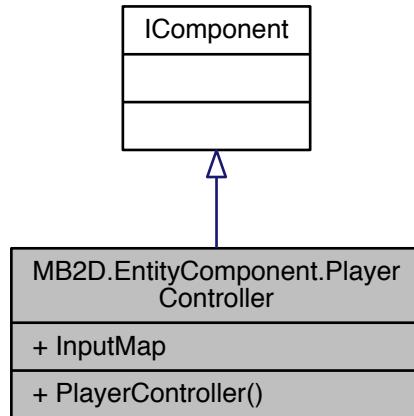
The documentation for this class was generated from the following file:

-
- Shared/src/Game/Environment/PlanetTile.cs

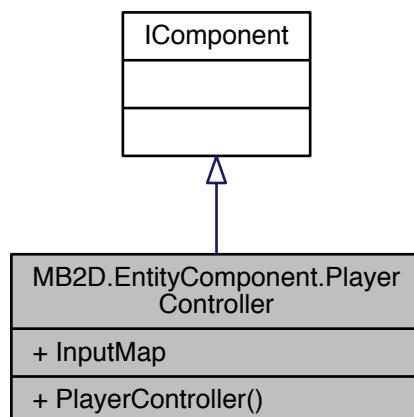
5.64 MB2D.EntityComponent.PlayerController Class Reference

Defines the attached entity as controllable

Inheritance diagram for MB2D.EntityComponent.PlayerController:



Collaboration diagram for MB2D.EntityComponent.PlayerController:



Public Member Functions

- [PlayerController \(\)](#)

Initializes a new instance of the T:MidnightBlue.PlayerController component with default input assignment

Properties

- `InputMap InputMap [get]`
Gets the input map.

5.64.1 Detailed Description

Defines the attached entity as controllable

5.64.2 Constructor & Destructor Documentation

5.64.2.1 PlayerController()

```
MB2D.EntityComponent.PlayerController.PlayerController ( ) [inline]
```

Initializes a new instance of the T:MidnightBlue.PlayerController component with default input assignment

5.64.3 Property Documentation

5.64.3.1 InputMap

```
InputMap MB2D.EntityComponent.PlayerController.InputMap [get]
```

Gets the input map.

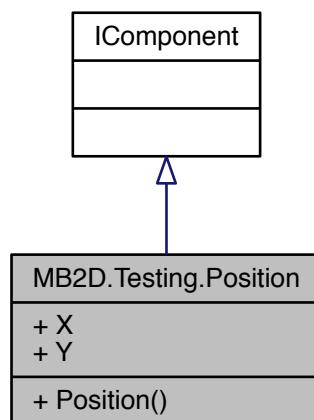
The input map.

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

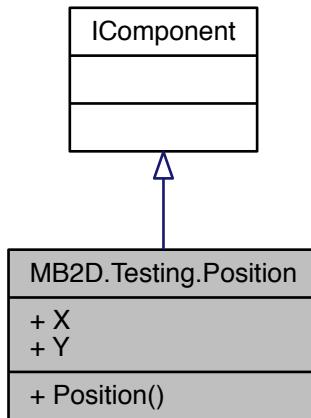
- MB2D/src/EntityComponent/Components/PlayerController.cs

5.65 MB2D.Testing.Position Class Reference

Inheritance diagram for MB2D.Testing.Position:



Collaboration diagram for MB2D.Testing.Position:



Public Member Functions

- **Position** (int x=0, int y=0)

Properties

- int **X** [get, set]
- int **Y** [get, set]

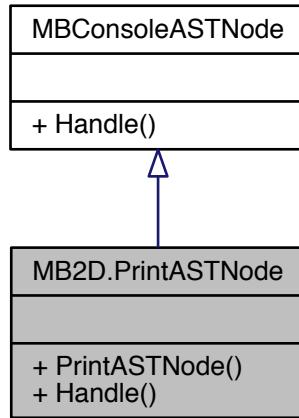
The documentation for this class was generated from the following file:

- MB2D/src/Test/TestSystem.cs

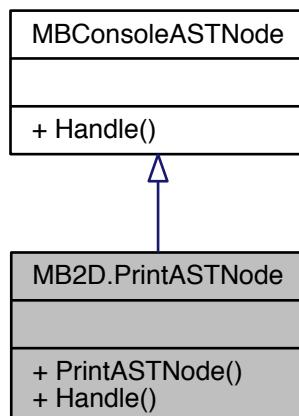
5.66 MB2D.PrintASTNode Class Reference

Prints a variable to the console.

Inheritance diagram for MB2D.PrintASTNode:



Collaboration diagram for MB2D.PrintASTNode:



Public Member Functions

- `PrintASTNode (string var, Token type)`
Initializes a new instance of the T:MB2D.PrintASTNode class.
- `override void Handle (MBConsole console)`
Prints the variable to the console. If it's not a previously assigned variable, it will print the variable name itself as if it's an immediate value.

5.66.1 Detailed Description

Prints a variable to the console.

5.66.2 Constructor & Destructor Documentation

5.66.2.1 PrintASTNode()

```
MB2D.PrintASTNode.PrintASTNode (
    string var,
    Token type ) [inline]
```

Initializes a new instance of the T:MB2D.PrintASTNode class.

Parameters

<i>var</i>	The variables identifier or, if not found, the print statements argument.
<i>type</i>	The variables token type

5.66.3 Member Function Documentation

5.66.3.1 Handle()

```
override void MB2D.PrintASTNode.Handle (
    MBConsole console ) [inline], [virtual]
```

Prints the variable to the console. If it's not a previously assigned variable, it will print the variable name itself as if it's an immediate value.

Parameters

<i>console</i>	Console to print to.
----------------	----------------------

Implements [MB2D.MBConsoleASTNode](#).

Here is the call graph for this function:



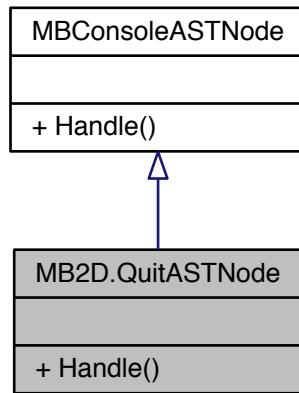
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleAST.cs

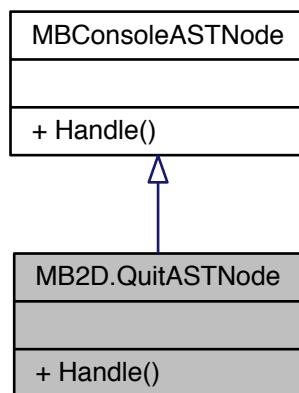
5.67 MB2D.QuitASTNode Class Reference

Handles quitting the game

Inheritance diagram for MB2D.QuitASTNode:



Collaboration diagram for MB2D.QuitASTNode:



Public Member Functions

- `override void Handle (MBConsole console)`
Quits the game

5.67.1 Detailed Description

Handles quitting the game

5.67.2 Member Function Documentation

5.67.2.1 Handle()

```
override void MB2D.QuitASTNode.Handle (
    MBConsole console) [inline], [virtual]
```

Quits the game

Parameters

<i>console</i>	Console to handle.
----------------	--------------------

Implements [MB2D.MBConsoleASTNode](#).

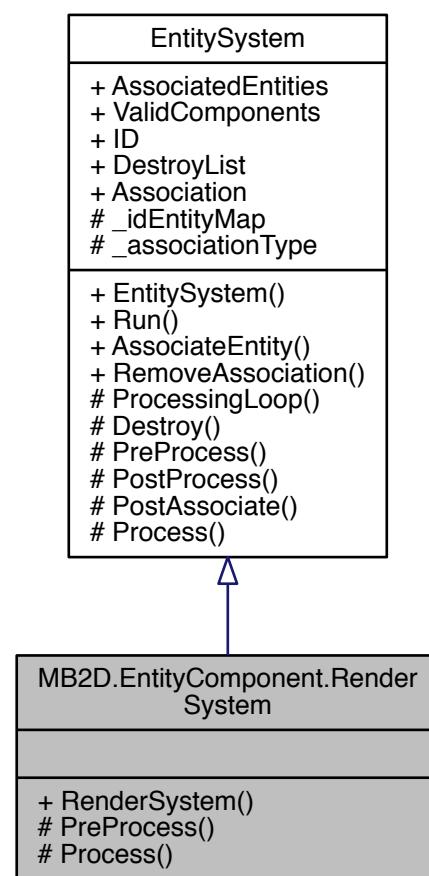
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleAST.cs

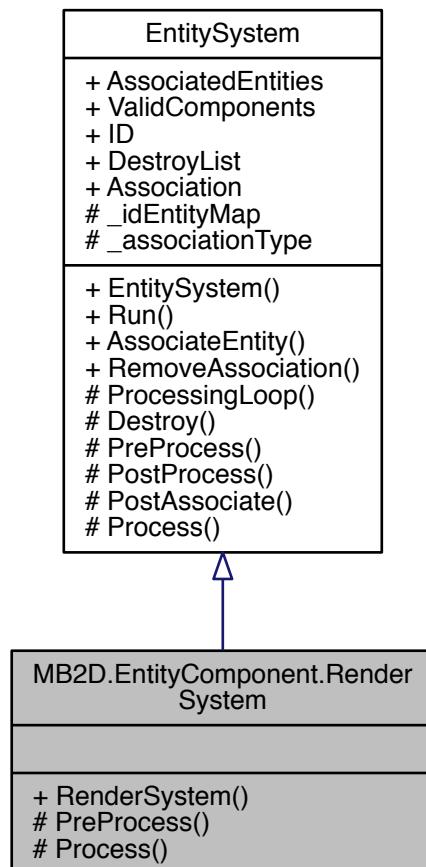
5.68 MB2D.EntityComponent.RenderSystem Class Reference

Renders culled entities with a [SpriteTransform](#) to the window

Inheritance diagram for MB2D.EntityComponent.RenderSystem:



Collaboration diagram for MB2D.EntityComponent.RenderSystem:



Public Member Functions

- [RenderSystem](#) (SpriteBatch spriteBatch)
Initializes a new instance of the T:MB2D.EntityComponent.RenderSystem class.

Protected Member Functions

- override void [PreProcess](#) ()
Re-orders the list of AssociatedEntities based on their current z-index
- override void [Process](#) (Entity entity)
Culls and then draws an entity to the window

Additional Inherited Members

5.68.1 Detailed Description

Renders culled entities with a [SpriteTransform](#) to the window

5.68.2 Constructor & Destructor Documentation

5.68.2.1 RenderSystem()

```
MB2D.EntityComponent.RenderSystem.RenderSystem (
    SpriteBatch spriteBatch ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.RenderSystem class.

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
--------------------	--------------------------

5.68.3 Member Function Documentation

5.68.3.1 PreProcess()

```
override void MB2D.EntityComponent.RenderSystem.PreProcess ( ) [inline], [protected], [virtual]
```

Re-orders the list of AssociatedEntities based on their current z-index

Reimplemented from [MB2D.EntityComponent.EntitySystem](#).

5.68.3.2 Process()

```
override void MB2D.EntityComponent.RenderSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
```

Culls and then draws an entity to the window

Parameters

<i>entity</i>	Entity .
---------------	--------------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

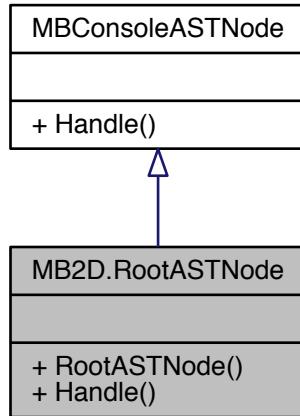
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Systems/RenderSystem.cs

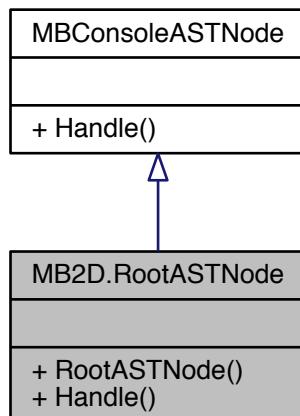
5.69 MB2D.RootASTNode Class Reference

The entry point for command execution with a single child.

Inheritance diagram for MB2D.RootASTNode:



Collaboration diagram for MB2D.RootASTNode:



Public Member Functions

- `RootASTNode (MBConsoleASTNode child)`

Initializes a new instance of the T:MB2D.RootASTNode class.

- `override void Handle (MBConsole console)`

Calls the child commands handle method

5.69.1 Detailed Description

The entry point for command execution with a single child.

5.69.2 Constructor & Destructor Documentation

5.69.2.1 RootASTNode()

```
MB2D.RootASTNode.RootASTNode (
    MBConsoleASTNode child ) [inline]
```

Initializes a new instance of the T:MB2D.RootASTNode class.

Parameters

<i>child</i>	Command AST node to handle.
--------------	-----------------------------

5.69.3 Member Function Documentation

5.69.3.1 Handle()

```
override void MB2D.RootASTNode.Handle (
    MBConsole console ) [inline], [virtual]
```

Calls the child commands handle method

Parameters

<i>console</i>	Console to handle.
----------------	--------------------

Implements [MB2D.MBConsoleASTNode](#).

Here is the call graph for this function:



Here is the caller graph for this function:



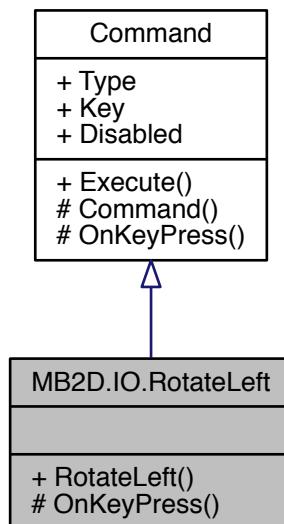
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleAST.cs

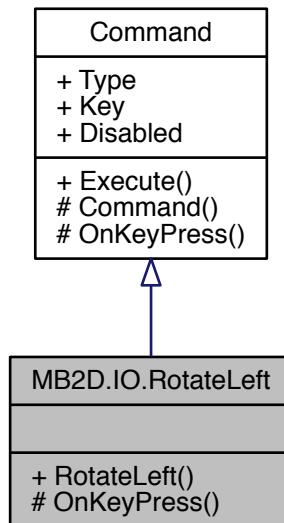
5.70 MB2D.IO.RotateLeft Class Reference

Rotates an entity left

Inheritance diagram for MB2D.IO.RotateLeft:



Collaboration diagram for MB2D.IO.RotateLeft:



Public Member Functions

- `RotateLeft (Keys key, CommandType type)`
Initializes a new instance of the T:MB2D.IO.RotateLeft class.

Protected Member Functions

- `override void OnKeyPress (Entity e=null)`
Rotates an entity left

Additional Inherited Members

5.70.1 Detailed Description

`Rotates an entity left`

5.70.2 Constructor & Destructor Documentation

5.70.2.1 RotateLeft()

```

MB2D.IO.RotateLeft.RotateLeft (
    Keys key,
    CommandType type ) [inline]
  
```

`Initializes a new instance of the T:MB2D.IO.RotateLeft class.`

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.70.3 Member Function Documentation**5.70.3.1 OnKeyPress()**

```
override void MB2D.IO.RotateLeft.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Rotates an entity left

Parameters

<i>e</i>	Entity to rotate.
----------	-------------------

Implements [MB2D.IO.Command](#).

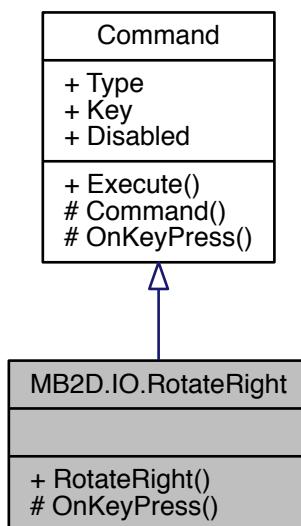
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

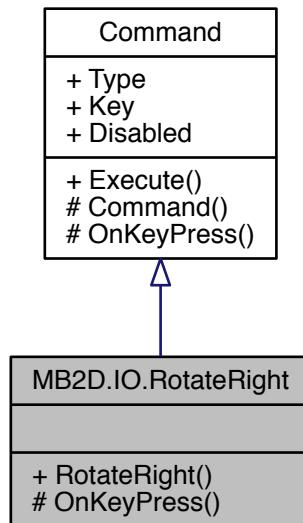
5.71 MB2D.IO.RotateRight Class Reference

Rotates an entity right

Inheritance diagram for MB2D.IO.RotateRight:



Collaboration diagram for MB2D.IO.RotateRight:



Public Member Functions

- [RotateRight](#) (`Keys key`, `CommandType type`)
Initializes a new instance of the T:MB2D.IO.RotateRight class.

Protected Member Functions

- `override void OnKeyPress (Entity e=null)`
Rotates an entity right

Additional Inherited Members

5.71.1 Detailed Description

Rotates an entity right

5.71.2 Constructor & Destructor Documentation

5.71.2.1 RotateRight()

```
MB2D.IO.RotateRight.RotateRight (
    Keys key,
    CommandType type ) [inline]
```

Initializes a new instance of the T:MB2D.IO.RotateRight class.

Parameters

<i>key</i>	Key to assign to.
<i>type</i>	Trigger type.

5.71.3 Member Function Documentation**5.71.3.1 OnKeyPress()**

```
override void MB2D.IO.RotateRight.OnKeyPress (
    Entity e = null ) [inline], [protected], [virtual]
```

Rotates an entity right

Parameters

<i>e</i>	Entity to rotate.
----------	-------------------

Implements [MB2D.IO.Command](#).

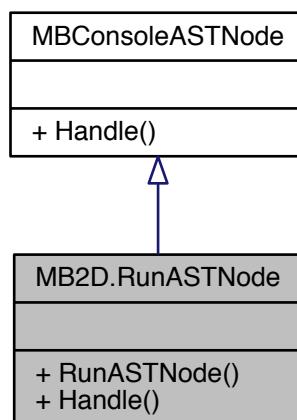
The documentation for this class was generated from the following file:

- MB2D/src/Input/MoveCommands.cs

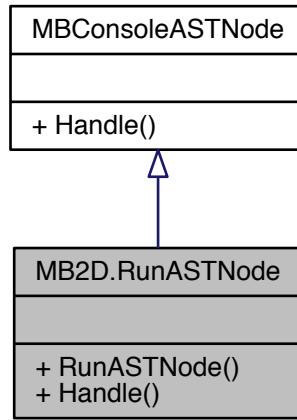
5.72 MB2D.RunASTNode Class Reference

AST node entry point for executing a run command

Inheritance diagram for MB2D.RunASTNode:



Collaboration diagram for MB2D.RunASTNode:



Public Member Functions

- [RunASTNode](#) (string ident, params string[] args)
Initializes a new instance of the T:MB2D.RunASTNode class.
- override void [Handle](#) ([MBConsole](#) console)
Checks for valid identifier and executes the given function from the console if correctly defined.

5.72.1 Detailed Description

AST node entry point for executing a run command

5.72.2 Constructor & Destructor Documentation

5.72.2.1 RunASTNode()

```
MB2D.RunASTNode.RunASTNode (
    string ident,
    params string [ ] args ) [inline]
```

Initializes a new instance of the T:MB2D.RunASTNode class.

Parameters

<i>ident</i>	Identifier of the function.
<i>args</i>	Arguments to pass to the function.

5.72.3 Member Function Documentation

5.72.3.1 Handle()

```
override void MB2D.RunASTNode.Handle (
    MBConsole console) [inline], [virtual]
```

Checks for valid identifier and executes the given function from the console if correctly defined.

Parameters

<code>console</code>	Console to handle.
----------------------	--------------------

Implements [MB2D.MBConsoleASTNode](#).

Here is the call graph for this function:



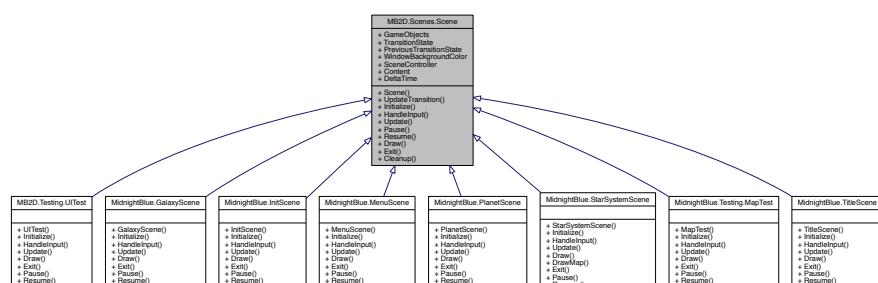
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleAST.cs

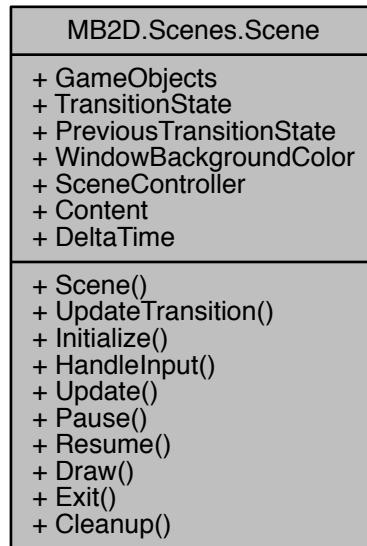
5.73 MB2D.Scenes.Scene Class Reference

Holds all logic and data for a single game screen

Inheritance diagram for MB2D.Scenes.Scene:



Collaboration diagram for MB2D.Scenes.Scene:



Public Member Functions

- [Scene \(EntityMap gameObjects, ContentManager content\)](#)

Initializes a new instance of the T:MB2D.Scenes.Scene class with a pre-existing EntityMap

- void [UpdateTransition \(\)](#)

- abstract void [Initialize \(\)](#)

Initialize this scene and loads all resources. Runs logic to execute during the Initializing state. Set state to None to end.

- abstract void [HandleInput \(\)](#)

Handles all input for the scene

- abstract void [Update \(\)](#)

Updates game logic and changes scene state.

- abstract void [Pause \(\)](#)

Runs logic to execute while the scene is in the Pausing state. Set state to None to end.

- abstract void [Resume \(\)](#)

Runs logic to execute while the scene is in the Resuming state. Set state to None to end.

- abstract void [Draw \(SpriteBatch spriteBatch, SpriteBatch uiSpriteBatch\)](#)

Draws entities and UI elements to the specified SpriteBatches

- abstract void [Exit \(\)](#)

Runs logic to execute while the scene is in the Exiting state. Set state to Null to end.

- void [Cleanup \(\)](#)

Cleans up the scene and unloads content.

Properties

- EntityMap `GameObjects` [get]
Gets all entities allocated to the scene
- TransitionState `TransitionState` [get, set]
Gets or sets the current transition state of the scene. This causes the scene stack to change the scenes state on the next frame.
- TransitionState `PreviousTransitionState` [get]
Gets the state the scene was in during the last frame.
- Color `WindowBackgroundColor` [get, set]
Gets or sets the color of the window background for this scene.
- SceneStack `SceneController` [get, set]
Gets or sets the scene controller.
- ContentManager `Content` [get]
Gets the content manager for loading and unloading resources.
- float `DeltaTime` [get, set]
Gets or sets the delta time value.

5.73.1 Detailed Description

Holds all logic and data for a single game screen

5.73.2 Constructor & Destructor Documentation

5.73.2.1 `Scene()`

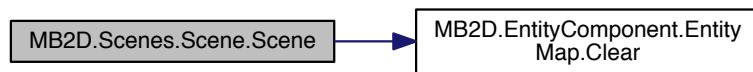
```
MB2D.Scenes.Scene (
    EntityMap gameObjects,
    ContentManager content ) [inline]
```

Initializes a new instance of the T:MB2D.Scenes.Scene class with a pre-existing EntityMap

Parameters

<code>gameObjects</code>	EntityMap to assign to the scene.
--------------------------	-----------------------------------

Here is the call graph for this function:



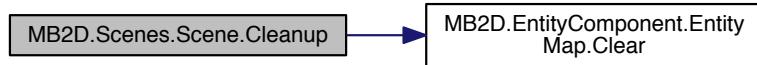
5.73.3 Member Function Documentation

5.73.3.1 Cleanup()

```
void MB2D.Scenes.Scene.Cleanup ( ) [inline]
```

Cleans up the scene and unloads content.

Here is the call graph for this function:



5.73.3.2 Draw()

```
abstract void MB2D.Scenes.Scene.Draw (
    SpriteBatch spriteBatch,
    SpriteBatch uiSpriteBatch ) [pure virtual]
```

Draws entities and [UI](#) elements to the specified SpriteBatches

Parameters

<i>spriteBatch</i>	World-coordinate based sprite batch.
<i>uiSpriteBatch</i>	Camera-based User Interface sprite batch.

Implemented in [MidnightBlue.GalaxyScene](#), [MidnightBlue.PlanetScene](#), [MidnightBlue.StarSystemScene](#), [MidnightBlue.InitScene](#), [MidnightBlue.TitleScene](#), [MidnightBlue.MenuScene](#), [MidnightBlue.Testing.MapTest](#), and [MB2D.Testing.UITest](#).

Here is the caller graph for this function:



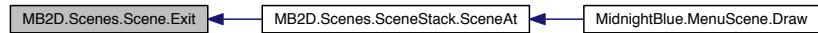
5.73.3.3 Exit()

```
abstract void MB2D.Scenes.Scene.Exit ( ) [pure virtual]
```

Runs logic to execute while the scene is in the Exiting state. Set state to Null to end.

Implemented in [MidnightBlue.StarSystemScene](#), [MidnightBlue.GalaxyScene](#), [MidnightBlue.PlanetScene](#), [MidnightBlue.InitScene](#), [MidnightBlue.TitleScene](#), [MidnightBlue.MenuScene](#), [MidnightBlue.Testing.MapTest](#), and [MB2D.Testing.UITest](#).

Here is the caller graph for this function:



5.73.3.4 HandleInput()

```
abstract void MB2D.Scenes.Scene.HandleInput ( ) [pure virtual]
```

Handles all input for the scene

Implemented in [MidnightBlue.GalaxyScene](#), [MidnightBlue.StarSystemScene](#), [MidnightBlue.PlanetScene](#), [MidnightBlue.InitScene](#), [MidnightBlue.TitleScene](#), [MidnightBlue.MenuScene](#), [MidnightBlue.Testing.MapTest](#), and [MB2D.Testing.UITest](#).

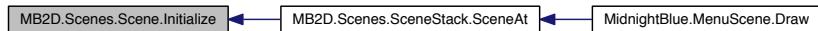
5.73.3.5 Initialize()

```
abstract void MB2D.Scenes.Scene.Initialize ( ) [pure virtual]
```

Initialize this scene and loads all resources. Runs logic to execute during the Initializing state. Set state to None to end.

Implemented in [MidnightBlue.StarSystemScene](#), [MidnightBlue.GalaxyScene](#), [MidnightBlue.PlanetScene](#), [MidnightBlue.TitleScene](#), [MidnightBlue.InitScene](#), [MidnightBlue.MenuScene](#), [MidnightBlue.Testing.MapTest](#), and [MB2D.Testing.UITest](#).

Here is the caller graph for this function:



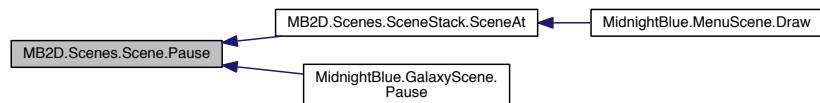
5.73.3.6 Pause()

```
abstract void MB2D.Scenes.Scene.Pause ( ) [pure virtual]
```

Runs logic to execute while the scene is in the Pausing state. Set state to None to end.

Implemented in [MidnightBlue.StarSystemScene](#), [MidnightBlue.GalaxyScene](#), [MidnightBlue.PlanetScene](#), [MidnightBlue.InitScene](#), [MidnightBlue.TitleScene](#), [MidnightBlue.MenuScene](#), [MidnightBlue.Testing.MapTest](#), and [MB2D.Testing.UITest](#).

Here is the caller graph for this function:



5.73.3.7 Resume()

```
abstract void MB2D.Scenes.Scene.Resume ( ) [pure virtual]
```

Runs logic to execute while the scene is in the Resuming state. Set state to None to end.

Implemented in [MidnightBlue.StarSystemScene](#), [MidnightBlue.GalaxyScene](#), [MidnightBlue.PlanetScene](#), [MidnightBlue.TitleScene](#), [MidnightBlue.InitScene](#), [MidnightBlue.MenuScene](#), [MidnightBlue.Testing.MapTest](#), and [MB2D.Testing.UITest](#).

Here is the caller graph for this function:



5.73.3.8 Update()

```
abstract void MB2D.Scenes.Scene.Update ( ) [pure virtual]
```

Updates game logic and changes scene state.

Implemented in [MidnightBlue.GalaxyScene](#), [MidnightBlue.StarSystemScene](#), [MidnightBlue.PlanetScene](#), [MidnightBlue.InitScene](#), [MidnightBlue.TitleScene](#), [MidnightBlue.MenuScene](#), [MidnightBlue.Testing.MapTest](#), and [MB2D.Testing.UITest](#).

5.73.4 Property Documentation

5.73.4.1 Content

```
ContentManager MB2D.Scenes.Scene.Content [get], [protected]
```

Gets the content manager for loading and unloading resources.

The content manager.

5.73.4.2 DeltaTime

```
float MB2D.Scenes.Scene.deltaTime [get], [set]
```

Gets or sets the delta time value.

The delta time.

5.73.4.3 GameObjects

```
EntityMap MB2D.Scenes.Scene.GameObjects [get]
```

Gets all entities allocated to the scene

The game objects.

5.73.4.4 PreviousTransitionState

```
TransitionState MB2D.Scenes.Scene.PreviousTransitionState [get]
```

Gets the state the scene was in during the last frame.

The state of the previous transition.

5.73.4.5 SceneController

```
SceneStack MB2D.Scenes.Scene.SceneController [get], [set]
```

Gets or sets the scene controller.

The scene controller.

5.73.4.6 TransitionState

```
TransitionState MB2D.Scenes.Scene.TransitionState [get], [set]
```

Gets or sets the current transition state of the scene. This causes the scene stack to change the scenes state on the next frame.

The transition state.

5.73.4.7 WindowBackgroundColor

`Color MB2D.Scenes.Scene.WindowBackgroundColor [get], [set]`

Gets or sets the color of the window background for this scene.

The color of the window background.

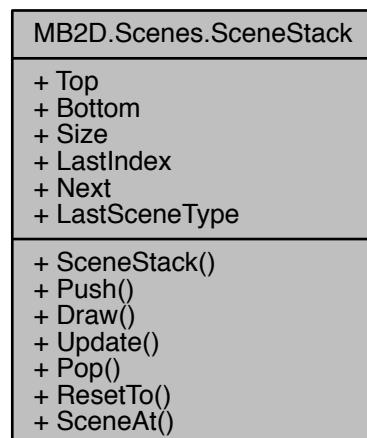
The documentation for this class was generated from the following file:

- `MB2D/src/Scene/Scene.cs`

5.74 MB2D.Scenes.SceneStack Class Reference

Holds the games scenes in a stack structure running the top scene every frame. Handles switching state for scenes and popping/pushing new scenes on top of one another. Allows the current scene to access other scenes.

Collaboration diagram for `MB2D.Scenes.SceneStack`:



Public Member Functions

- **`SceneStack ()`**
Initializes a new instance of the T:MB2D.Scenes.SceneStack class.
- **`void Push (Scene scene)`**
Pushes a new scene to the top of the stack. Calls the new scenes Initialize method and the previous scenes Pause method
- **`void Draw (SpriteBatch spriteBatch, SpriteBatch uiSpriteBatch)`**
- **`void Update ()`**
Updates the scene at the top of the stack and handles any state transitions if they've been called. For any transitions or scene logic to function correctly this must be called once per frame.

- void [Pop \(\)](#)
Pops the top scene off the stack, calling its Exit method and calls the Resume method of the next scene on the stack if it exists.
- void [ResetTo \(Scene scene\)](#)
Resets the scene stack to the specified scene, clearing all other scenes from the stack. Use this in most scenarios instead of Push to save memory by not keeping scenes allocated if unnecessary.
- Scene [SceneAt \(int index\)](#)
Gets the scene located at the specific index in the stack

Properties

- Scene [Top \[get\]](#)
Gets the scene at the top of the stack.
- Scene [Bottom \[get\]](#)
Gets the scene at the bottom of the stack.
- int [Size \[get\]](#)
Gets the current size of the stack
- int [LastIndex \[get\]](#)
Gets the upper bounds of the indexes of the stack
- Scene [Next \[get\]](#)
- Type [LastSceneType \[get\]](#)

5.74.1 Detailed Description

Holds the games scenes in a stack structure running the top scene every frame. Handles switching state for scenes and popping/pushing new scenes on top of one another. Allows the current scene to access other scenes.

5.74.2 Constructor & Destructor Documentation

5.74.2.1 [SceneStack\(\)](#)

```
MB2D.Scenes.SceneStack.SceneStack ( ) [inline]
```

Initializes a new instance of the T:MB2D.Scenes.SceneStack class.

5.74.3 Member Function Documentation

5.74.3.1 [Pop\(\)](#)

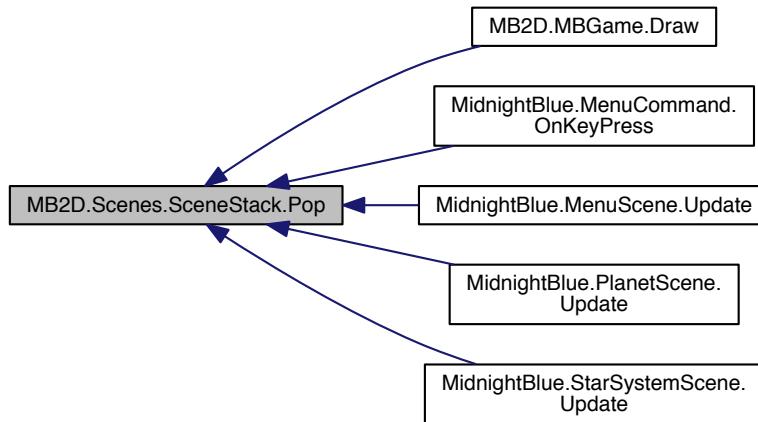
```
void MB2D.Scenes.SceneStack.Pop ( ) [inline]
```

Pops the top scene off the stack, calling its Exit method and calls the Resume method of the next scene on the stack if it exists.

Here is the call graph for this function:



Here is the caller graph for this function:



5.74.3.2 Push()

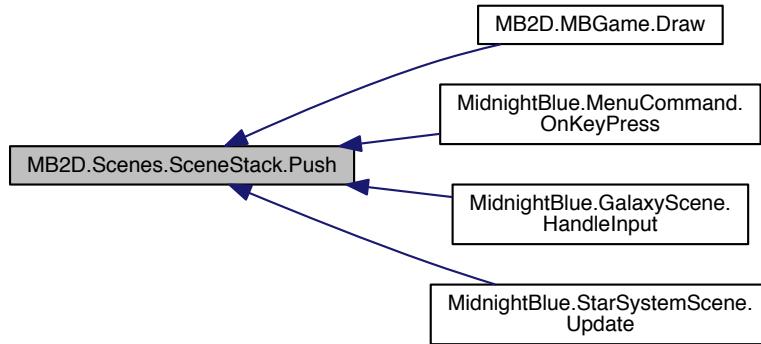
```
void MB2D.Scenes.SceneStack.Push ( Scene scene ) [inline]
```

Pushes a new scene to the top of the stack. Calls the new scenes Initialize method and the previous scenes Pause method

Parameters

<code>scene</code>	Scene to push.
--------------------	----------------

Here is the caller graph for this function:



5.74.3.3 ResetTo()

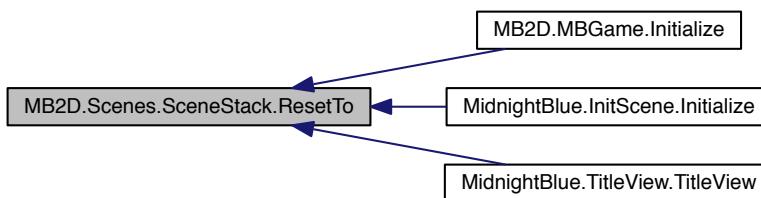
```
void MB2D.Scenes.SceneStack.ResetTo (
    Scene scene ) [inline]
```

Resets the scene stack to the specified scene, clearing all other scenes from the stack. Use this in most scenarios instead of Push to save memory by not keeping scenes allocated if unnecessary.

Parameters

<code>scene</code>	<code>Scene</code> to reset to.
--------------------	---------------------------------

Here is the caller graph for this function:



5.74.3.4 SceneAt()

```
Scene MB2D.Scenes.SceneStack.SceneAt (
    int index ) [inline]
```

Gets the scene located at the specific index in the stack

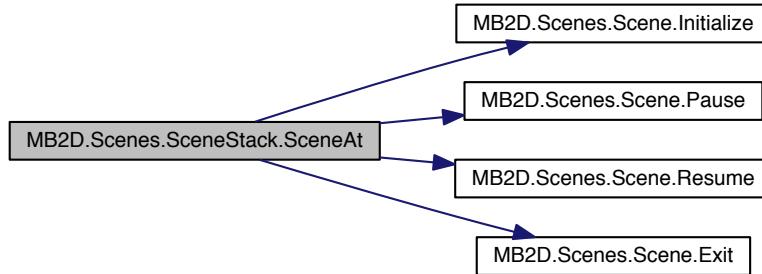
Returns

The T:MB2D.Scenes.Scene.

Parameters

<i>index</i>	Index to get.
--------------	---------------

Here is the call graph for this function:



Here is the caller graph for this function:

**5.74.3.5 Update()**

```
void MB2D.Scenes.SceneStack.Update( ) [inline]
```

Updates the scene at the top of the stack and handles any state transitions if they've been called. For any transitions or scene logic to function correctly this must be called once per frame.

Here is the caller graph for this function:



5.74.4 Property Documentation

5.74.4.1 Bottom

```
Scene MB2D.Scenes.SceneStack.Bottom [get]
```

Gets the scene at the bottom of the stack.

The bottom scene.

5.74.4.2 LastIndex

```
int MB2D.Scenes.SceneStack.LastIndex [get]
```

Gets the upper bounds of the indexes of the stack

The last index.

5.74.4.3 Size

```
int MB2D.Scenes.SceneStack.Size [get]
```

Gets the current size of the stack

The size.

5.74.4.4 Top

```
Scene MB2D.Scenes.SceneStack.Top [get]
```

Gets the scene at the top of the stack.

The scene at the top of the stack.

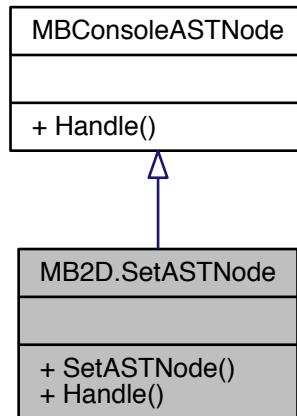
The documentation for this class was generated from the following file:

- MB2D/src/Scene/SceneStack.cs

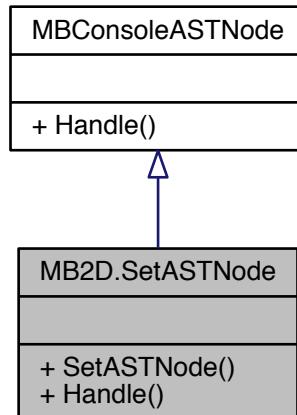
5.75 MB2D.SetASTNode Class Reference

AST node representing the entry point for a 'set' command with an identifier and a value child

Inheritance diagram for MB2D.SetASTNode:



Collaboration diagram for MB2D.SetASTNode:



Public Member Functions

- **`SetASTNode`** (string ident, `VariableASTNode` val)

Initializes a new instance of the T:MB2D.SetASTNode class.
- **override void `Handle`** (`MBConsole` console)

Handles setting the variable using the consoles Vars property. Checks if the identifier only starts with an alpha or underscore and handles any type checking or parse errors.

5.75.1 Detailed Description

AST node representing the entry point for a 'set' command with an identifier and a value child

5.75.2 Constructor & Destructor Documentation

5.75.2.1 SetASTNode()

```
MB2D.SetASTNode.SetASTNode (
    string ident,
    VariableASTNode val ) [inline]
```

Initializes a new instance of the T:MB2D.SetASTNode class.

Parameters

<i>ident</i>	Identifier of the variable.
<i>val</i>	Value to assign.

5.75.3 Member Function Documentation

5.75.3.1 Handle()

```
override void MB2D.SetASTNode.Handle (
    MBConsole console ) [inline], [virtual]
```

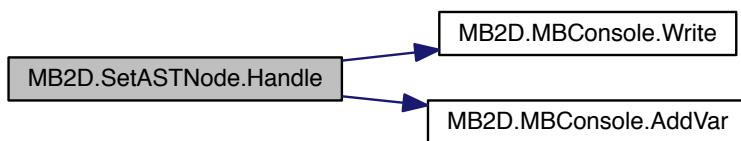
Handles setting the variable using the consoles Vars property. Checks if the identifier only starts with an alpha or underscore and handles any type checking or parse errors.

Parameters

<i>console</i>	Console to handle.
----------------	--------------------

Implements [MB2D.MBConsoleASTNode](#).

Here is the call graph for this function:



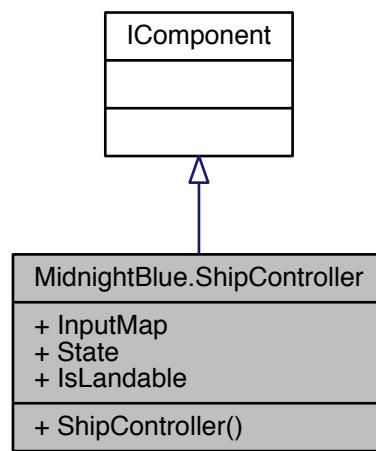
The documentation for this class was generated from the following file:

- MB2D/src/MBConsole/MBConsoleAST.cs

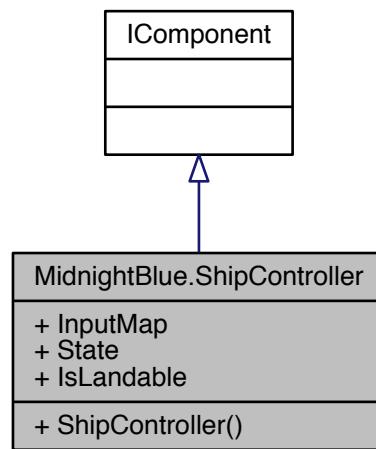
5.76 MidnightBlue.ShipController Class Reference

Controls a ships movement and actions

Inheritance diagram for MidnightBlue.ShipController:



Collaboration diagram for `MidnightBlue.ShipController`:



Public Member Functions

- [ShipController \(\)](#)

Initializes a new instance of the T:MidnightBlue.ShipController class and assigns all default input and key mappings.

Properties

- [InputMap InputMap \[get\]](#)

Gets the input map.

- [ShipState State \[get, set\]](#)

Gets or sets the current travel state of the ship.

- [bool IsLandable \[get, set\]](#)

Gets or sets a value indicating whether this T:MidnightBlue.ShipController is able to be landed when the entity calls their LandCommand.

5.76.1 Detailed Description

Controls a ships movement and actions

5.76.2 Constructor & Destructor Documentation

5.76.2.1 ShipController()

`MidnightBlue.ShipController.ShipController () [inline]`

Initializes a new instance of the T:MidnightBlue.ShipController class and assigns all default input and key mappings.

5.76.3 Property Documentation

5.76.3.1 InputMap

`InputMap MidnightBlue.ShipController.InputMap [get]`

Gets the input map.

The input map.

5.76.3.2 IsLandable

`bool MidnightBlue.ShipController.IsLandable [get], [set]`

Gets or sets a value indicating whether this T:MidnightBlue.ShipController is able to be landed when the entity calls their LandCommand.

`true` if is landable; otherwise, `false`.

5.76.3.3 State

```
ShipState MidnightBlue.ShipController.State [get], [set]
```

Gets or sets the current travel state of the ship.

The ships travelling state.

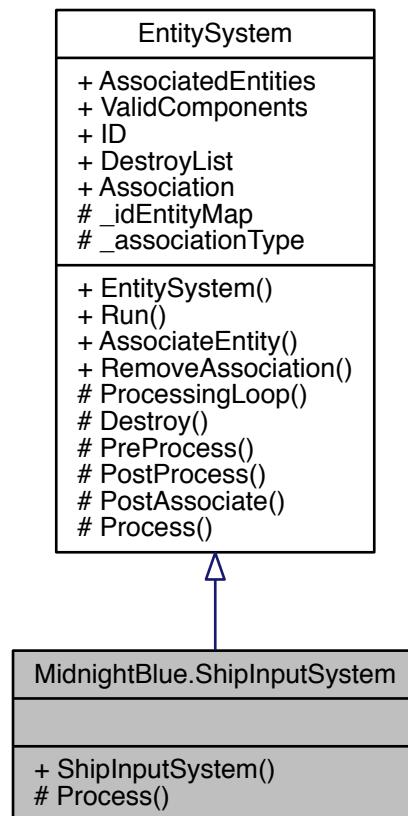
The documentation for this class was generated from the following file:

- Shared/src/Game/Components/ShipController.cs

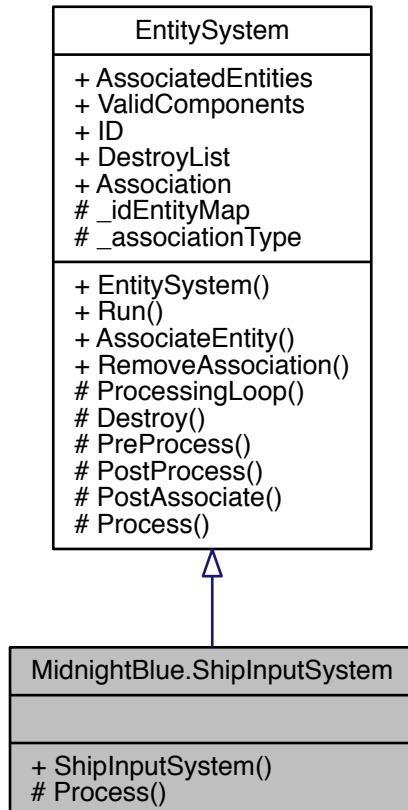
5.77 MidnightBlue.ShipInputSystem Class Reference

Handles moving the ship forward and backwards.

Inheritance diagram for MidnightBlue.ShipInputSystem:



Collaboration diagram for `MidnightBlue.ShipInputSystem`:



Public Member Functions

- [ShipInputSystem \(\)](#)
Initializes a new instance of the T:MidnightBlue.ShipInputSystem class.

Protected Member Functions

- [override void Process \(Entity entity\)](#)
Uses the entities ship controller to move forward and backward

Additional Inherited Members

5.77.1 Detailed Description

Handles moving the ship forward and backwards.

5.77.2 Constructor & Destructor Documentation

5.77.2.1 ShipInputSystem()

```
MidnightBlue.ShipInputSystem.ShipInputSystem () [inline]
```

Initializes a new instance of the T:MidnightBlue.ShipInputSystem class.

5.77.3 Member Function Documentation

5.77.3.1 Process()

```
override void MidnightBlue.ShipInputSystem.Process (
    Entity entity) [inline], [protected], [virtual]
```

Uses the entities ship controller to move forward and backward

Parameters

entity	Entity to process.
--------	--------------------

Implements [MB2D.EntityComponent.EntitySystem](#).

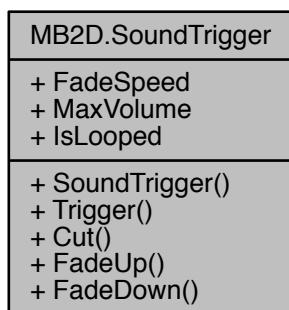
The documentation for this class was generated from the following file:

- Shared/src/Game/Systems/ShipInputSystem.cs

5.78 MB2D.SoundTrigger Class Reference

Triggers a sound effect

Collaboration diagram for MB2D.SoundTrigger:



Public Member Functions

- **SoundTrigger** (*SoundEffect sound*)

Initializes a new instance of the T:MB2D.SoundTrigger class using the specified SoundEffect
- **void Trigger ()**

Plays the sound if it's not already playing
- **void Cut ()**

Stops the sound if it's playing
- **void FadeUp ()**

Increases the sounds volume one step based on the specified FadeSpeed
- **void FadeDown ()**

Decreases the volume based on the specified FadeSpeed. Stops the sound once the volume reaches 0

Properties

- **float FadeSpeed [get, set]**

Gets or sets the fade speed.
- **float MaxVolume [get, set]**

Determines the maximum volume to FadeUp
- **bool IsLooped [get, set]**

Gets or sets a value indicating whether this T:MB2D.SoundTrigger is looped or one-shot.

5.78.1 Detailed Description

Triggers a sound effect

5.78.2 Constructor & Destructor Documentation

5.78.2.1 SoundTrigger()

```
MB2D.SoundTrigger.SoundTrigger (
    SoundEffect sound ) [inline]
```

Initializes a new instance of the T:MB2D.SoundTrigger class using the specified SoundEffect

Parameters

<i>sound</i>	Sound to use.
--------------	---------------

5.78.3 Member Function Documentation

5.78.3.1 Cut()

```
void MB2D.SoundTrigger.Cut ( ) [inline]
```

Stops the sound if it's playing

Here is the caller graph for this function:

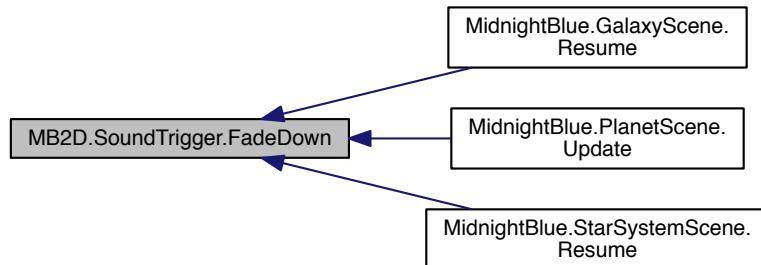


5.78.3.2 FadeDown()

```
void MB2D.SoundTrigger.FadeDown ( ) [inline]
```

Decreases the volume based on the specified FadeSpeed. Stops the sound once the volume reaches 0

Here is the caller graph for this function:

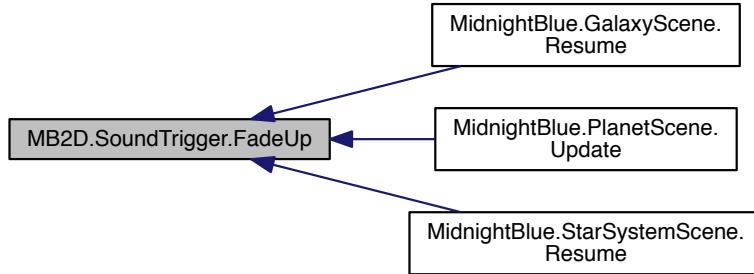


5.78.3.3 FadeUp()

```
void MB2D.SoundTrigger.FadeUp ( ) [inline]
```

Increases the sounds volume one step based on the specified FadeSpeed

Here is the caller graph for this function:



5.78.3.4 Trigger()

```
void MB2D.SoundTrigger.Trigger () [inline]
```

Plays the sound if it's not already playing

5.78.4 Property Documentation

5.78.4.1 FadeSpeed

```
float MB2D.SoundTrigger.FadeSpeed [get], [set]
```

Gets or sets the fade speed.

The fade speed.

5.78.4.2 IsLooped

```
bool MB2D.SoundTrigger.IsLooped [get], [set]
```

Gets or sets a value indicating whether this T:MB2D.SoundTrigger is looped or one-shot.

true if is looped; otherwise, false.

5.78.4.3 MaxVolume

```
float MB2D.SoundTrigger.MaxVolume [get], [set]
```

Determines the maximum volume to FadeUp

The max volume.

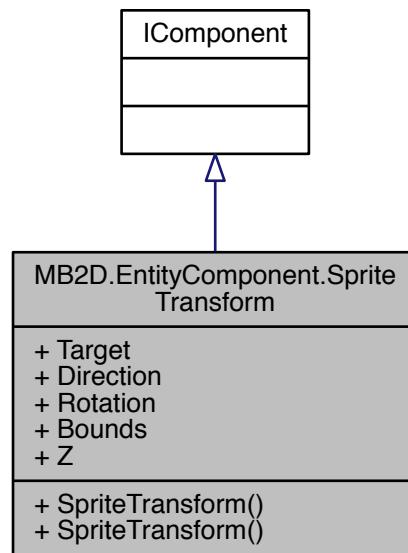
The documentation for this class was generated from the following file:

- MB2D/src/Audio/SoundTrigger.cs

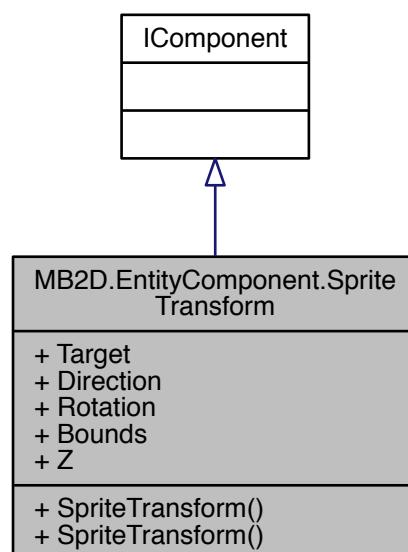
5.79 MB2D.EntityComponent.SpriteTransform Class Reference

Defines a sprite component with control over its size, rotation, and scale

Inheritance diagram for MB2D.EntityComponent.SpriteTransform:



Collaboration diagram for MB2D.EntityComponent.SpriteTransform:



Public Member Functions

- [SpriteTransform](#) (Texture2D texture, Vector2 position, Vector2 scale)
Initializes a new instance of the T:MB2D.EntityComponent.SpriteTransform class.
- [SpriteTransform](#) (TextureRegion2D texture, Vector2 position, Vector2 scale)
Initializes a new instance of the T:MB2D.EntityComponent.SpriteTransform class.

Properties

- [Sprite Target](#) [get, set]
Gets or sets the sprites target containing all of the applied data and logicd.
- [Vector2 Direction](#) [get]
Gets the sprites direction.
- [float Rotation](#) [get, set]
Gets or sets the rotation in radians.
- [RectangleF Bounds](#) [get, set]
Gets or sets the sprites bounding box.
- [float Z](#) [get, set]
Gets or sets the sprites z index. Used in depth systems.

5.79.1 Detailed Description

Defines a sprite component with control over its size, rotation, and scale

5.79.2 Constructor & Destructor Documentation

5.79.2.1 [SpriteTransform\(\)](#) [1/2]

```
MB2D.EntityComponent.SpriteTransform.SpriteTransform (
    Texture2D texture,
    Vector2 position,
    Vector2 scale ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.SpriteTransform class.

Parameters

<i>texture</i>	Texture to assign to the sprite.
<i>position</i>	Initial position of the sprite. Should be the entities position for best practice.
<i>scale</i>	Initial scale of the sprite.

5.79.2.2 [SpriteTransform\(\)](#) [2/2]

```
MB2D.EntityComponent.SpriteTransform.SpriteTransform (
    TextureRegion2D texture,
```

```
Vector2 position,  
Vector2 scale ) [inline]
```

Initializes a new instance of the T:MB2D.EntityComponent.SpriteTransform class.

Parameters

<i>texture</i>	Texture region in a texture atlas to assign to the sprite.
<i>position</i>	Initial position of the sprite. Should be the entities position for best practice.
<i>scale</i>	Initial scale of the sprite.

5.79.3 Property Documentation

5.79.3.1 Bounds

```
RectangleF MB2D.EntityComponent.SpriteTransform.Bounds [get], [set]
```

Gets or sets the sprites bounding box.

The bounds.

5.79.3.2 Direction

```
Vector2 MB2D.EntityComponent.SpriteTransform.Direction [get]
```

Gets the sprites direction.

The direction.

5.79.3.3 Rotation

```
float MB2D.EntityComponent.SpriteTransform.Rotation [get], [set]
```

Gets or sets the rotation in radians.

The rotation in radians.

5.79.3.4 Target

```
Sprite MB2D.EntityComponent.SpriteTransform.Target [get], [set]
```

Gets or sets the sprites target containing all of the applied data and logicd.

The sprite target.

5.79.3.5 Z

```
float MB2D.EntityComponent.SpriteTransform.Z [get], [set]
```

Gets or sets the sprites z index. Used in depth systems.

The z index.

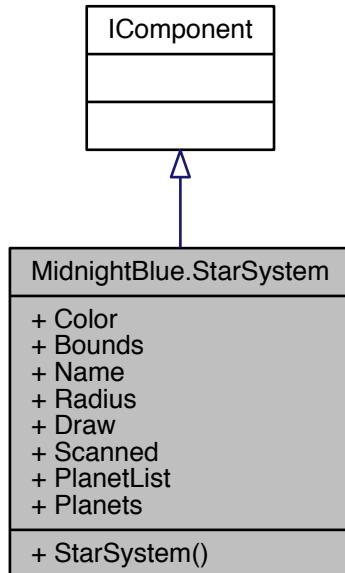
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Components/SpriteTransform.cs

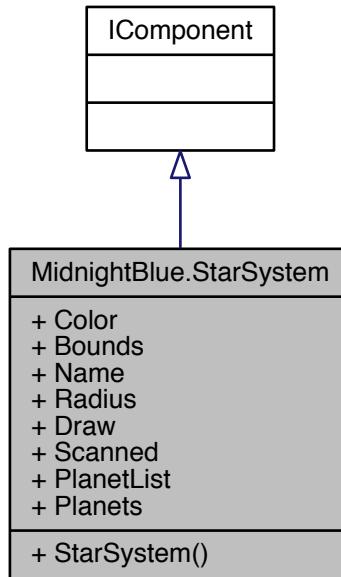
5.80 MidnightBlue.StarSystem Class Reference

Represents an star system entity to be used in the galaxy view

Inheritance diagram for MidnightBlue.StarSystem:



Collaboration diagram for MidnightBlue.StarSystem:



Public Member Functions

- **StarSystem ()**
Initializes a new instance of the T:MidnightBlue.StarSystem class.

Properties

- Color **Color** [get, set]
Gets or sets the color of the star rendered in the galaxy view.
- Rectangle **Bounds** [get, set]
Gets or sets the bounding rectangle of the star system in the galaxy view.
- string **Name** [get, set]
Gets or sets the name of the star system.
- int **Radius** [get, set]
Gets or sets the radius of the star at the center of the system.
- bool **Draw** [get, set]
Gets or sets a value indicating whether this T:MidnightBlue.StarSystem is drawn or not.
- bool **Scanned** [get, set]
Gets or sets a value indicating whether this T:MidnightBlue.StarSystem has been scanned by the player.
- string **PlanetList** [get]
Gets a string representation of the list of planets in the star system and all their information.
- List< PlanetMetadata > **Planets** [get, set]
Gets or sets the list of all planets.

5.80.1 Detailed Description

Represents an star system entity to be used in the galaxy view

5.80.2 Constructor & Destructor Documentation

5.80.2.1 StarSystem()

```
MidnightBlue.StarSystem.StarSystem () [inline]
```

Initializes a new instance of the T:MidnightBlue.StarSystem class.

5.80.3 Property Documentation

5.80.3.1 Bounds

```
Rectangle MidnightBlue.StarSystem.Bounds [get], [set]
```

Gets or sets the bounding rectangle of the star system in the galaxy view.

The bounds.

5.80.3.2 Color

```
Color MidnightBlue.StarSystem.Color [get], [set]
```

Gets or sets the color of the star rendered in the galaxy view.

The color.

5.80.3.3 Draw

```
bool MidnightBlue.StarSystem.Draw [get], [set]
```

Gets or sets a value indicating whether this T:MidnightBlue.StarSystem is drawn or not.

true if should be drawn; otherwise, false.

5.80.3.4 Name

```
string MidnightBlue.StarSystem.Name [get], [set]
```

Gets or sets the name of the star system.

The name.

5.80.3.5 PlanetList

```
string MidnightBlue.StarSystem.PlanetList [get]
```

Gets a string representation of the list of planets in the star system and all their information.

The planet list.

5.80.3.6 Planets

```
List<PlanetMetadata> MidnightBlue.StarSystem.Planets [get], [set]
```

Gets or sets the list of all planets.

The planets.

5.80.3.7 Radius

```
int MidnightBlue.StarSystem.Radius [get], [set]
```

Gets or sets the radius of the star at the center of the system.

The radius.

5.80.3.8 Scanned

```
bool MidnightBlue.StarSystem.Scanned [get], [set]
```

Gets or sets a value indicating whether this T:MidnightBlue.StarSystem has been scanned by the player.

true if scanned; otherwise, false.

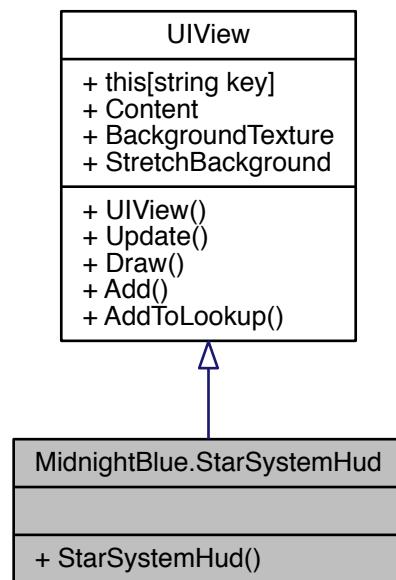
The documentation for this class was generated from the following file:

- Shared/src/Game/Components/StarSystem.cs

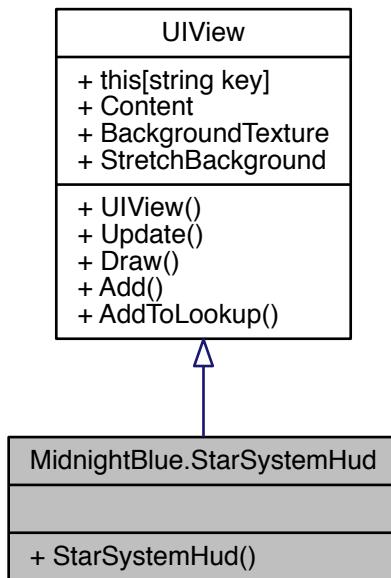
5.81 MidnightBlue.StarSystemHud Class Reference

Star system hud with minimap.

Inheritance diagram for `MidnightBlue.StarSystemHud`:



Collaboration diagram for MidnightBlue.StarSystemHud:



Public Member Functions

- **StarSystemHud** (`ContentManager content`, `EntityMap gameObjects`, `SceneStack scenes`)
Initializes a new instance of the T:MidnightBlue.StarSystemHud class.

Additional Inherited Members

5.81.1 Detailed Description

Star system hud with minimap.

5.81.2 Constructor & Destructor Documentation

5.81.2.1 StarSystemHud()

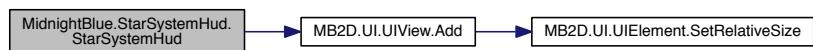
```
MidnightBlue.StarSystemHud.StarSystemHud (
    ContentManager content,
    EntityMap gameObjects,
    SceneStack scenes ) [inline]
```

Initializes a new instance of the T:MidnightBlue.StarSystemHud class.

Parameters

<i>content</i>	Content to load fonts and textures with.
<i>gameObjects</i>	Game objects to track in the minimap.
<i>scenes</i>	Scene stack to use in UI interactions.

Here is the call graph for this function:



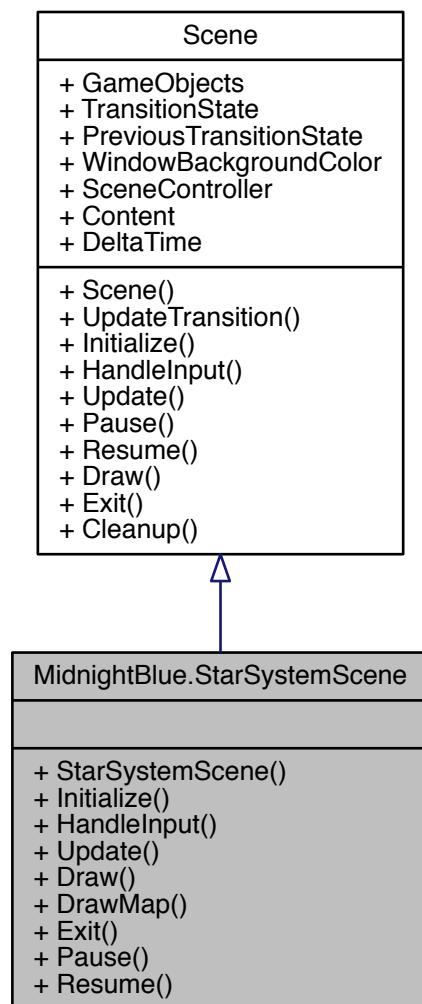
The documentation for this class was generated from the following file:

- Shared/src/Game/UIViews/StarSystemHud.cs

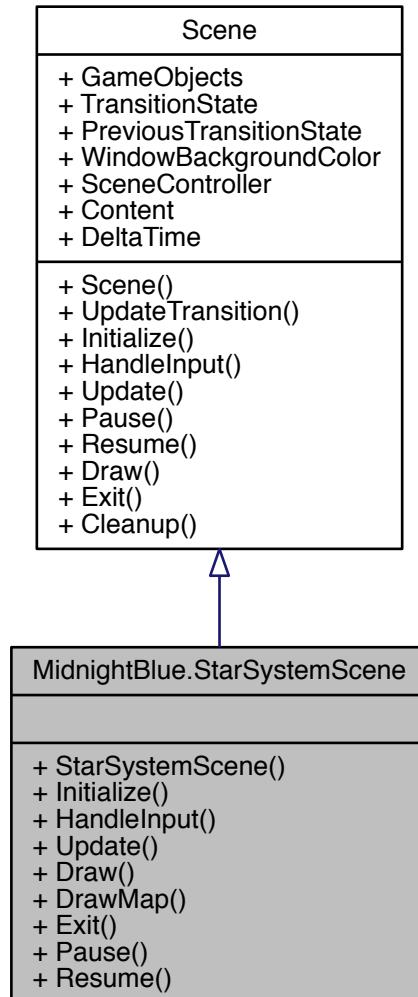
5.82 MidnightBlue.StarSystemScene Class Reference

Scene to display a star system with planets and a star.

Inheritance diagram for MidnightBlue.StarSystemScene:



Collaboration diagram for `MidnightBlue.StarSystemScene`:



Public Member Functions

- `StarSystemScene (EntityMap map, ContentManager content, StarSystem starSystem, Dictionary< string, Planet > cache, int seed)`

Initializes a new instance of the T:MidnightBlue.StarSystemScene class.
- `override void Initialize ()`

Ends initializing instantly.
- `override void HandleInput ()`

Handles moving the players ship.
- `override void Update ()`

Updates all systems in the game and handles the occurrence of the player entering a planet.
- `override void Draw (SpriteBatch spriteBatch, SpriteBatch uiSpriteBatch)`

Draws the star system to the sprite batch and the HUD to the UI spritebatch.

- void [DrawMap](#) (SpriteBatch uiSpriteBatch)
Draws the minimap to the window.
- override void [Exit](#) ()
Exits the scene instantly.
- override void [Pause](#) ()
Pauses the scene instantly.
- override void [Resume](#) ()
Resumes the star system scene after leaving a planet. Handles resetting the physics environment, players ship settings and reactivates all planets and the star.

Additional Inherited Members

5.82.1 Detailed Description

Scene to display a star system with planets and a star.

5.82.2 Constructor & Destructor Documentation

5.82.2.1 StarSystemScene()

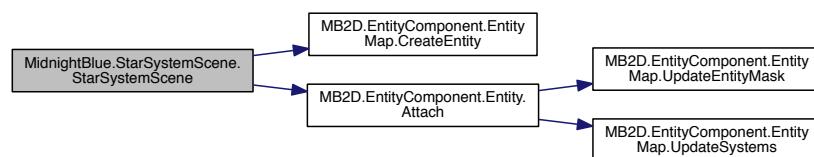
```
MidnightBlue.StarSystemScene.StarSystemScene (
    EntityMap map,
    ContentManager content,
    StarSystem starSystem,
    Dictionary< string, Planet > cache,
    int seed ) [inline]
```

Initializes a new instance of the T:MidnightBlue.StarSystemScene class.

Parameters

<i>map</i>	Entity map to load entities into.
<i>content</i>	Content manager for loading resources.
<i>starSystem</i>	Star system information to use for the scene.
<i>cache</i>	Planet cache used for quickly loading recently-visited star systems.
<i>seed</i>	Seed to use in random generation.

Here is the call graph for this function:



5.82.3 Member Function Documentation

5.82.3.1 Draw()

```
override void MidnightBlue.StarSystemScene.Draw (
    SpriteBatch spriteBatch,
    SpriteBatch uiSpriteBatch ) [inline], [virtual]
```

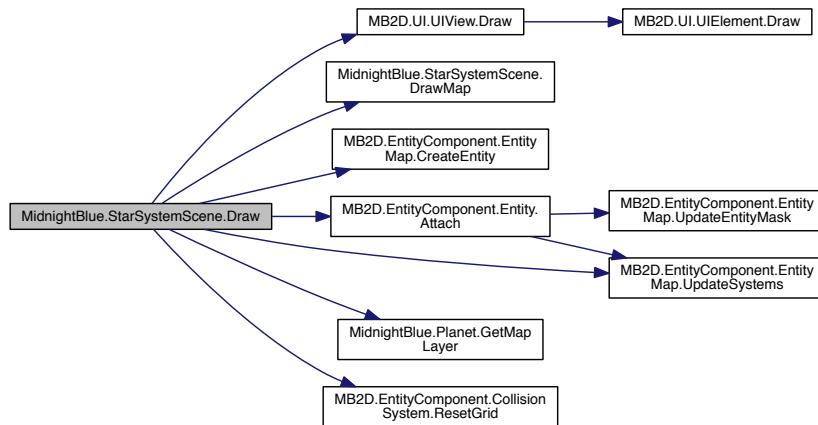
Draws the star system to the sprite batch and the HUD to the UI spritebatch.

Parameters

<i>spriteBatch</i>	Sprite batch to draw world-based entities to.
<i>uiSpriteBatch</i>	User interface sprite batch.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.82.3.2 DrawMap()

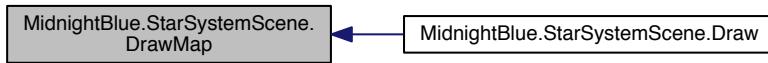
```
void MidnightBlue.StarSystemScene.DrawMap (
    SpriteBatch uiSpriteBatch ) [inline]
```

Draws the minimap to the window.

Parameters

<i>uiSpriteBatch</i>	User interface sprite batch to draw to.
----------------------	---

Here is the caller graph for this function:



5.82.3.3 Exit()

```
override void MidnightBlue.StarSystemScene.Exit () [inline], [virtual]
```

Exits the scene instantly.

Implements [MB2D.Scenes.Scene](#).

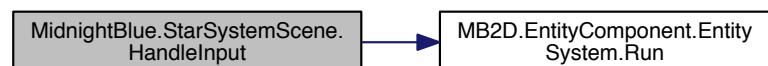
5.82.3.4 HandleInput()

```
override void MidnightBlue.StarSystemScene.HandleInput () [inline], [virtual]
```

Handles moving the players ship.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.82.3.5 Initialize()

```
override void MidnightBlue.StarSystemScene.Initialize () [inline], [virtual]
```

Ends initializing instantly.

Implements [MB2D.Scenes.Scene](#).

5.82.3.6 Pause()

```
override void MidnightBlue.StarSystemScene.Pause ( ) [inline], [virtual]
```

Pauses the scene instantly.

Implements [MB2D.Scenes.Scene](#).

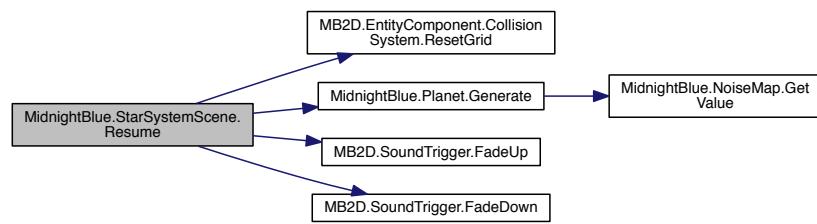
5.82.3.7 Resume()

```
override void MidnightBlue.StarSystemScene.Resume ( ) [inline], [virtual]
```

Resumes the star system scene after leaving a planet. Handles resetting the physics environment, players ship settings and reactivates all planets and the star.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



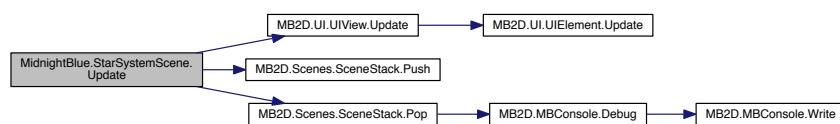
5.82.3.8 Update()

```
override void MidnightBlue.StarSystemScene.Update ( ) [inline], [virtual]
```

Updates all systems in the game and handles the occurrence of the player entering a planet.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:

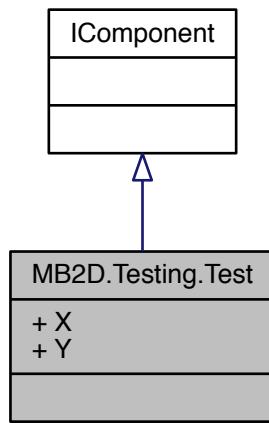


The documentation for this class was generated from the following file:

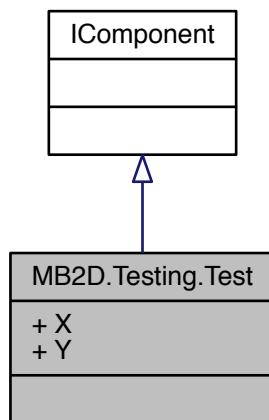
- Shared/src/Game/Scenes/StarSystemScene.cs

5.83 MB2D.Testing.Test Class Reference

Inheritance diagram for MB2D.Testing.Test:



Collaboration diagram for MB2D.Testing.Test:



Properties

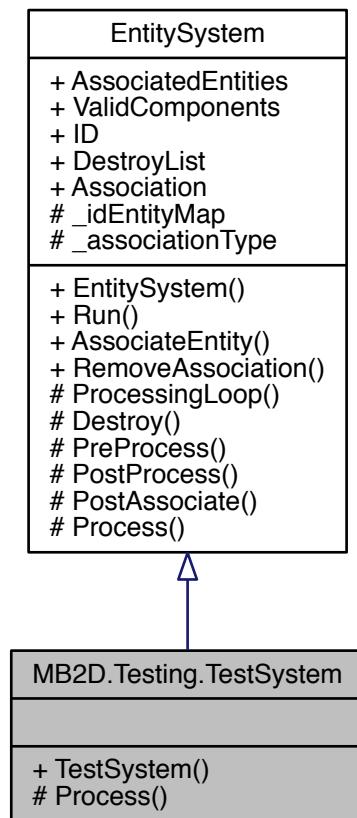
- int `X` [get, set]
- int `Y` [get, set]

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

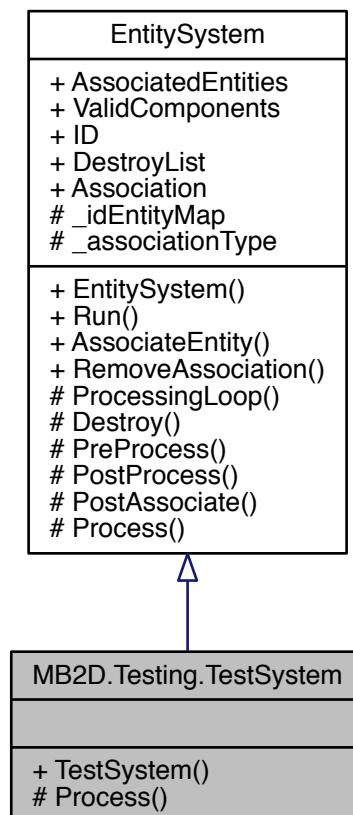
- MB2D/src/Test/TestSystem.cs

5.84 MB2D.Testing.TestSystem Class Reference

Inheritance diagram for MB2D.Testing.TestSystem:



Collaboration diagram for MB2D.Testing.TestSystem:



Protected Member Functions

- `override void Process (Entity entity)`
Executes this systems logic on a single entity

Additional Inherited Members

5.84.1 Member Function Documentation

5.84.1.1 Process()

```
override void MB2D.Testing.TestSystem.Process (
    Entity entity ) [inline], [protected], [virtual]
```

Executes this systems logic on a single entity

Parameters

<i>entity</i>	Entity to operate on
---------------	----------------------

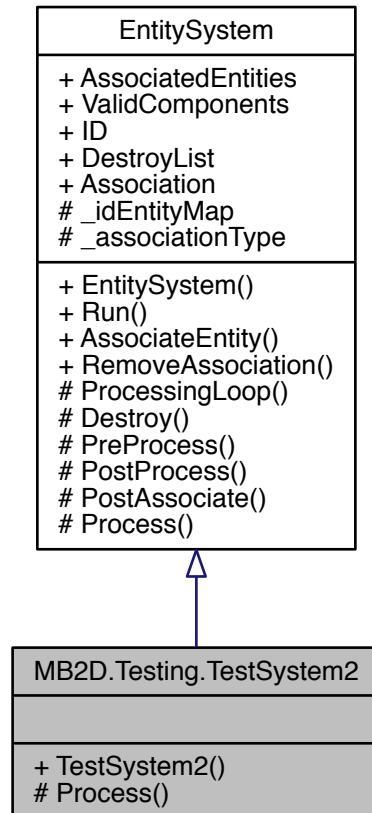
Implements [MB2D.EntityComponent.EntitySystem](#).

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

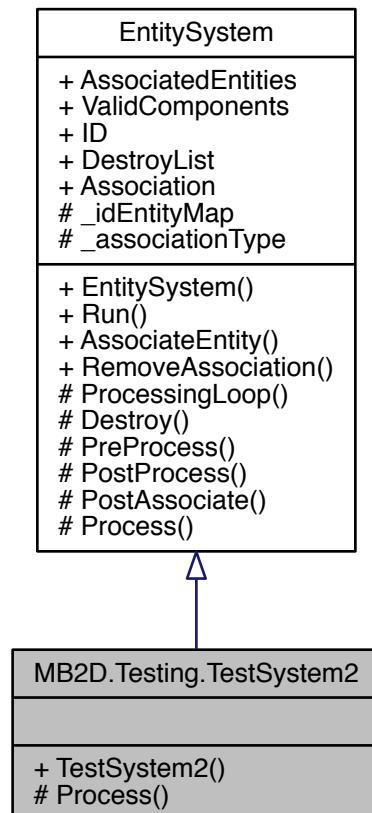
- MB2D/src/Test/TestSystem.cs

5.85 MB2D.Testing.TestSystem2 Class Reference

Inheritance diagram for MB2D.Testing.TestSystem2:



Collaboration diagram for MB2D.Testing.TestSystem2:



Protected Member Functions

- `override void Process (Entity entity)`
Executes this systems logic on a single entity

Additional Inherited Members

5.85.1 Member Function Documentation

5.85.1.1 `Process()`

```
override void MB2D.Testing.TestSystem2.Process (
    Entity entity ) [inline], [protected], [virtual]
```

Executes this systems logic on a single entity

Parameters

<i>entity</i>	Entity to operate on
---------------	----------------------

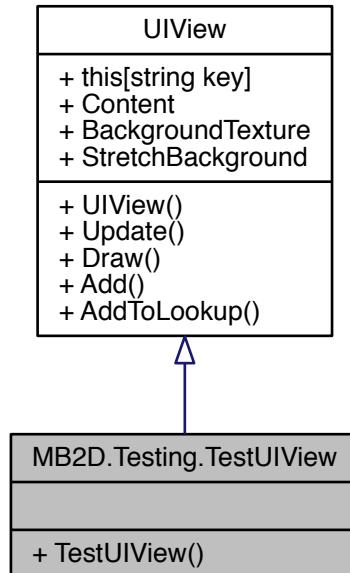
Implements [MB2D.EntityComponent.EntitySystem](#).

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

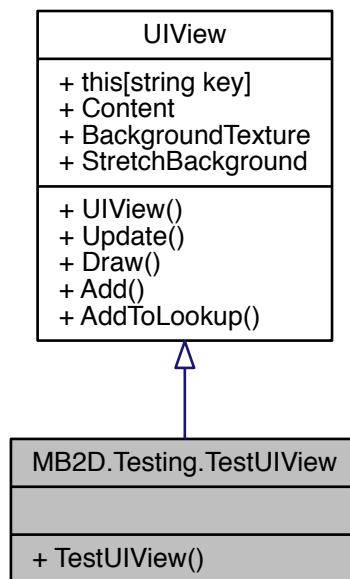
- MB2D/src/Test/TestSystem.cs

5.86 MB2D.Testing.TestUIView Class Reference

Inheritance diagram for MB2D.Testing.TestUIView:



Collaboration diagram for MB2D.Testing.TestUIView:



Public Member Functions

- **TestUIView** (ContentManager content)

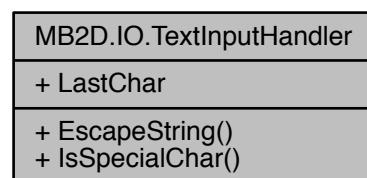
Additional Inherited Members

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

- MB2D/src/Test/TestUIView.cs

5.87 MB2D.IO.TextInputHandler Class Reference

Collaboration diagram for MB2D.IO.TextInputHandler:



Public Member Functions

- string **EscapeString** (char c)
- bool **IsSpecialChar** (char c)

Properties

- char [LastChar](#) [get]

Returns the last character entered by the user. Translates the keycode to an ASCII character taking into account space, backspace, and special character modifiers.

5.87.1 Property Documentation

5.87.1.1 [LastChar](#)

```
char MB2D.IO.TextInputHandler.LastChar [get]
```

Returns the last character entered by the user. Translates the keycode to an ASCII character taking into account space, backspace, and special character modifiers.

Returns

The char.

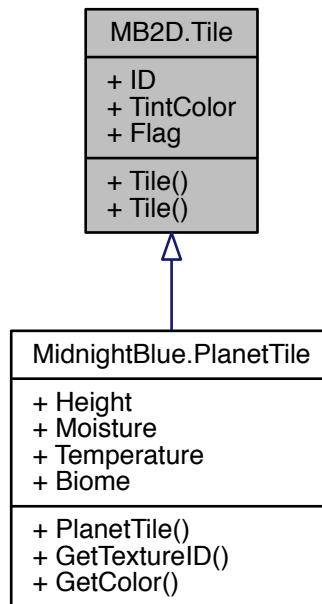
The documentation for this class was generated from the following file:

- MB2D/src/Input/TextInputHandler.cs

5.88 MB2D.Tile Class Reference

Represents a single tile in a tile map.

Inheritance diagram for MB2D.Tile:



Collaboration diagram for MB2D.Tile:



Public Member Functions

- `Tile` (int textureID, Color color)
Initializes a new instance of the T:MidnightBlue.Tile class
- `Tile` ()
Initializes a new instance of the T:MidnightBlue.Tile class.

Properties

- int **ID** [get, protected set]
Gets or sets the tile map region ID to use for this tile.
- Color **TintColor** [get, protected set]
Gets or sets the color of the tint.
- **TileFlag Flag** [get, set]
Gets or sets the tile flag for collision detection.

5.88.1 Detailed Description

Represents a single tile in a tile map.

5.88.2 Constructor & Destructor Documentation

5.88.2.1 Tile() [1/2]

```
MB2D.Tile.Tile (
    int textureID,
    Color color ) [inline]
```

Initializes a new instance of the T:MidnightBlue.Tile class

Parameters

<i>textureID</i>	ID of the texture region in the tile map to use for this tile, i.e. Grass or water.
<i>color</i>	Color tint to apply to the tile.

5.88.2.2 Tile() [2/2]

```
MB2D.Tile.Tile ( ) [inline]
```

Initializes a new instance of the T:MidnightBlue.Tile class.

5.88.3 Property Documentation

5.88.3.1 Flag

TileFlag MB2D.Tile.Flag [get], [set]

Gets or sets the tile flag for collision detection.

The flag.

5.88.3.2 ID

```
int MB2D.Tile.ID [get], [protected set]
```

Gets or sets the tile map region ID to use for this tile.

The tile map region ID.

5.88.3.3 TintColor

```
Color MB2D.Tile.TintColor [get], [protected set]
```

Gets or sets the color of the tint.

The color of the tint.

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

- MB2D/src/Tiles/Tile.cs

5.89 MB2D.Tiles.TileMap Class Reference

A grid of tiles with collision. Wraps coordinates when they fall out of bounds. Allows accessing tiles by index.

Collaboration diagram for MB2D.Tiles.TileMap:

MB2D.Tiles.TileMap
+ this[int x, int y] + Texture + TileSize + MapSize + Fill() + TileMap() + GetTile() + Draw() + HandleWrapping()

Public Member Functions

- void [Fill](#) (`Tile[,] tiles`)
Uses a 2D Array of previously defined tile information to fill a tile map with collision data and other information. Must be called in order for the [TileMap](#) to function.
- [TileMap](#) (`Texture2D texture, int cellSize, int margin=0, int spacing=0, int offset=0, float scale=1.0f`)
Initializes a new instance of the T:MB2D.Tiles.TileMap class. Separates the texture into a series of regions.
- Rectangle [GetTile](#) (`int id`)
Retrieves the bounding rectangle of a tile texture from the tilemap
- void [Draw](#) (`SpriteBatch spriteBatch`)
Draws the tile map to the specified SpriteBatch, wrapping the rendering when the camera reaches the bounds of the map.
- void [HandleWrapping](#) ([Movement](#) movement)
Handles wrapping an entity around the map if their movement falls out of bounds - gives the illusion of an infinitely looping map.

Properties

- [Tile](#) `this[int x, int y] [get]`
Gets the T:MB2D.Tiles.Tile at the specified x y.
- Texture2D [Texture](#) [get]
Gets the texture atlases undivided texture.
- Point [TileSize](#) [get]
Gets the size of each tile in the world.
- Point [MapSize](#) [get]
Gets the size of the map.

5.89.1 Detailed Description

A grid of tiles with collision. Wraps coordinates when they fall out of bounds. Allows accessing tiles by index.

5.89.2 Constructor & Destructor Documentation

5.89.2.1 [TileMap\(\)](#)

```
MB2D.Tiles.TileMap.TileMap (
    Texture2D texture,
    int cellSize,
    int margin = 0,
    int spacing = 0,
    int offset = 0,
    float scale = 1.0f ) [inline]
```

Initializes a new instance of the T:MB2D.Tiles.TileMap class. Separates the texture into a series of regions.

Parameters

<code>texture</code>	Texture to use in the texture atlas.
<code>cellSize</code>	The size of each cell in the texture atlas.
<code>margin</code>	Margin to apply to each rendered tile.
<code>spacing</code>	Spacing to apply to each rendered tile.
<code>offset</code>	Offset to apply to the x and y coordinates of each tile when rendering.
<code>scale</code>	Scale vector to apply to each cell when rendering.

5.89.3 Member Function Documentation

5.89.3.1 Draw()

```
void MB2D.Tiles.TileMap.Draw (
    SpriteBatch spriteBatch ) [inline]
```

Draws the tile map to the specified SpriteBatch, wrapping the rendering when the camera reaches the bounds of the map.

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
--------------------	--------------------------

Here is the caller graph for this function:



5.89.3.2 Fill()

```
void MB2D.Tiles.TileMap.Fill (
    Tile tiles[,] ) [inline]
```

Uses a 2D Array of previously defined tile information to fill a tile map with collision data and other information. Must be called in order for the [TileMap](#) to function.

Parameters

<i>tiles</i>	Tiles.
--------------	--------

Here is the caller graph for this function:



5.89.3.3 GetTile()

```
Rectangle MB2D.Tiles.TileMap.GetTile (
    int id ) [inline]
```

Retrieves the bounding rectangle of a tile texture from the tilemap

Returns

The tile ID's bounding rectangle.

Parameters

<i>id</i>	The ID to get.
-----------	----------------

5.89.3.4 HandleWrapping()

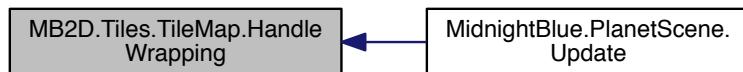
```
void MB2D.Tiles.TileMap.HandleWrapping (
    Movement movement ) [inline]
```

Handles wrapping an entity around the map if their movement falls out of bounds - gives the illusion of an infinitely looping map.

Parameters

<i>movement</i>	Movement component to operate on.
-----------------	-----------------------------------

Here is the caller graph for this function:



5.89.4 Property Documentation

5.89.4.1 MapSize

```
Point MB2D.Tiles.TileMap.MapSize [get]
```

Gets the size of the map.

The size of the map.

5.89.4.2 Texture

```
Texture2D MB2D.Tiles.TileMap.Texture [get]
```

Gets the texture atlases undivided texture.

The texture.

5.89.4.3 this[int x, int y]

```
Tile MB2D.Tiles.TileMap.this[int x, int y] [get]
```

Gets the T:MB2D.Tiles.Tile at the specified x y.

Parameters

x	The x coordinate.
y	The y coordinate.

5.89.4.4 TileSize

```
Point MB2D.Tiles.TileMap.TileSize [get]
```

Gets the size of each tile in the world.

The size of the tile.

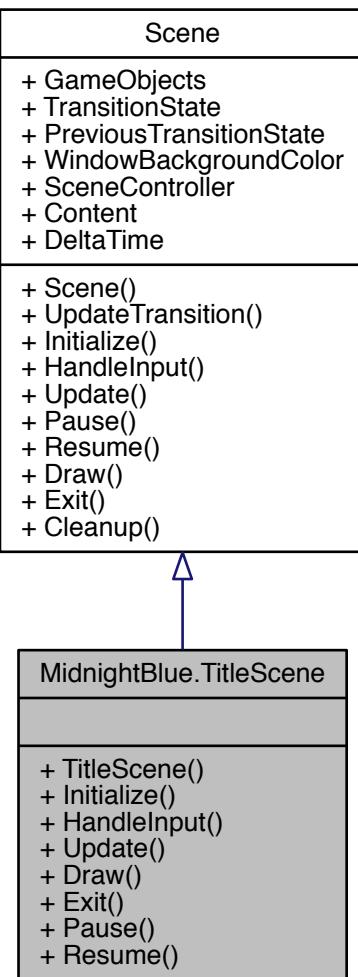
The documentation for this class was generated from the following file:

- MB2D/src/Tiles/TileMap.cs

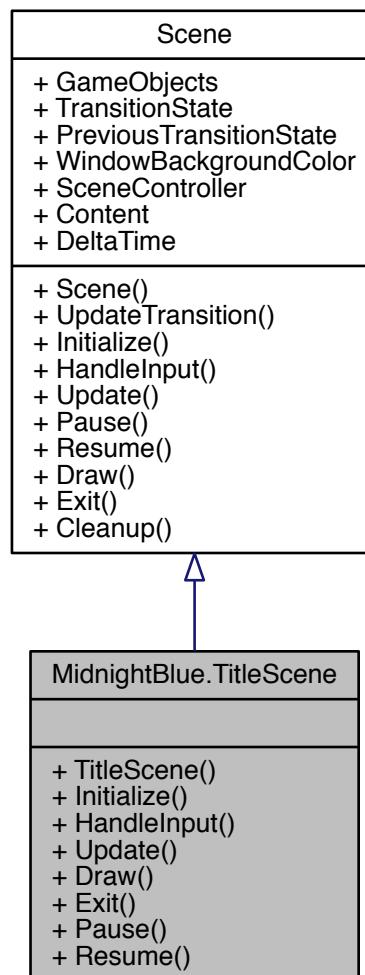
5.90 MidnightBlue.TitleScene Class Reference

The scene shown at the title screen.

Inheritance diagram for `MidnightBlue.TitleScene`:



Collaboration diagram for MidnightBlue.TitleScene:



Public Member Functions

- **TitleScene (EntityMap map, ContentManager content)**
Initializes a new instance of the T:MidnightBlue.TitleScene class.
- **override void Initialize ()**
Creates the UIView and starts the background music.
- **override void HandleInput ()**
Handles the input for the menu.
- **override void Update ()**
Updates the UI
- **override void Draw (SpriteBatch spriteBatch, SpriteBatch uiSpriteBatch)**
Draws the UI to the uiSpriteBatch
- **override void Exit ()**
Stops the music and quits instantly

- `override void Pause ()`
Pauses the title screen, fading music while it does so.
- `override void Resume ()`
Resumes the title screen, fading music in while it does so.

Additional Inherited Members

5.90.1 Detailed Description

The scene shown at the title screen.

5.90.2 Constructor & Destructor Documentation

5.90.2.1 TitleScene()

```
MidnightBlue.TitleScene.TitleScene (
    EntityMap map,
    ContentManager content ) [inline]
```

Initializes a new instance of the T:MidnightBlue.TitleScene class.

Parameters

<code>map</code>	Game objects.
<code>content</code>	Content manager for loading textures and sounds.

5.90.3 Member Function Documentation

5.90.3.1 Draw()

```
override void MidnightBlue.TitleScene.Draw (
    SpriteBatch spriteBatch,
    SpriteBatch uiSpriteBatch ) [inline], [virtual]
```

Draws the UI to the uiSpriteBatch

Parameters

<code>spriteBatch</code>	Sprite batch for world-based entities.
<code>uiSpriteBatch</code>	User interface sprite batch.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.90.3.2 Exit()

```
override void MidnightBlue.TitleScene.Exit ( ) [inline], [virtual]
```

Stops the music and quits instantly

Implements [MB2D.Scenes.Scene](#).

5.90.3.3 HandleInput()

```
override void MidnightBlue.TitleScene.HandleInput ( ) [inline], [virtual]
```

Handles the input for the menu.

Implements [MB2D.Scenes.Scene](#).

5.90.3.4 Initialize()

```
override void MidnightBlue.TitleScene.Initialize ( ) [inline], [virtual]
```

Creates the UITableView and starts the background music.

Implements [MB2D.Scenes.Scene](#).

5.90.3.5 Pause()

```
override void MidnightBlue.TitleScene.Pause ( ) [inline], [virtual]
```

Pauses the title screen, fading music while it does so.

Implements [MB2D.Scenes.Scene](#).

5.90.3.6 Resume()

```
override void MidnightBlue.TitleScene.Resume ( ) [inline], [virtual]
```

Resumes the title screen, fading music in while it does so.

Implements [MB2D.Scenes.Scene](#).

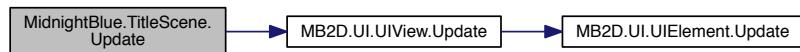
5.90.3.7 Update()

```
override void MidnightBlue.TitleScene.Update ( ) [inline], [virtual]
```

Updates the UI

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



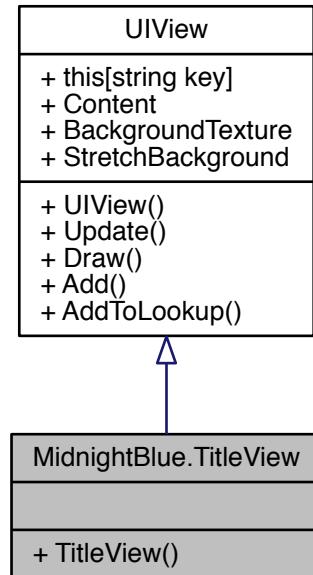
The documentation for this class was generated from the following file:

- Shared/src/Game/Scenes/TitleScene.cs

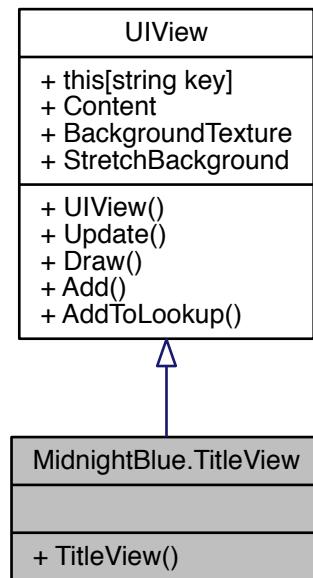
5.91 MidnightBlue.TitleView Class Reference

The title screens UI view

Inheritance diagram for MidnightBlue.TitleView:



Collaboration diagram for MidnightBlue.TitleView:



Public Member Functions

- `TitleView (ContentManager content, EntityMap gameObjects, SceneStack scenes)`
Initializes a new instance of the T:MidnightBlue.TitleView class.

Additional Inherited Members

5.91.1 Detailed Description

The title screens UI view

5.91.2 Constructor & Destructor Documentation

5.91.2.1 TitleView()

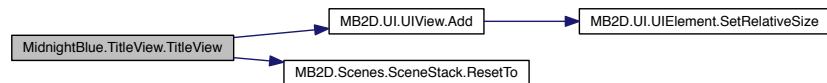
```
MidnightBlue.TitleView.TitleView (
    ContentManager content,
    EntityMap gameObjects,
    SceneStack scenes ) [inline]
```

Initializes a new instance of the T:MidnightBlue.TitleView class.

Parameters

<i>content</i>	Content to load sounds, fonts, and textures from.
<i>gameObjects</i>	Game objects to track.
<i>scenes</i>	Scenes to use in UI interactions.

Here is the call graph for this function:



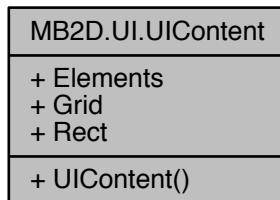
The documentation for this class was generated from the following file:

- Shared/src/Game/UIViews/TitleView.cs

5.92 MB2D.UI.UIContent Class Reference

Holds content in a grid structure for a UIContext or [Layout](#)

Collaboration diagram for MB2D.UI.UIContent:



Public Member Functions

- [UIContent](#) (int rows, int cols, Rectangle parent)

Initializes a new instance of the T:MB2D.UI.UIContent class.

Properties

- `UIElement [,] Elements [get]`
Gets the elements of the content.
- `Grid Grid [get]`
Gets a grid geometry representation of the content
- `Rectangle Rect [get, set]`
Gets or sets the rectangle encompassing the content.

5.92.1 Detailed Description

Holds content in a grid structure for a UIContext or [Layout](#)

5.92.2 Constructor & Destructor Documentation

5.92.2.1 UIContent()

```
MB2D.UI.UIContent.UIContent (
    int rows,
    int cols,
    Rectangle parent ) [inline]
```

Initializes a new instance of the T:MB2D.UI.UIContent class.

Parameters

<code>rows</code>	Rows.
<code>cols</code>	Cols.
<code>parent</code>	Parent.

5.92.3 Property Documentation

5.92.3.1 Elements

```
UIElement [,] MB2D.UI.UIContent.Elements [get]
```

Gets the elements of the content.

The [UI](#) elements.

5.92.3.2 Grid

```
Grid MB2D.UI.UIContent.Grid [get]
```

Gets a grid geometry representation of the content

The grid.

5.92.3.3 Rect

```
Rectangle MB2D.UI.UIContent.Rect [get], [set]
```

Gets or sets the rectangle encompassing the content.

The rectangle.

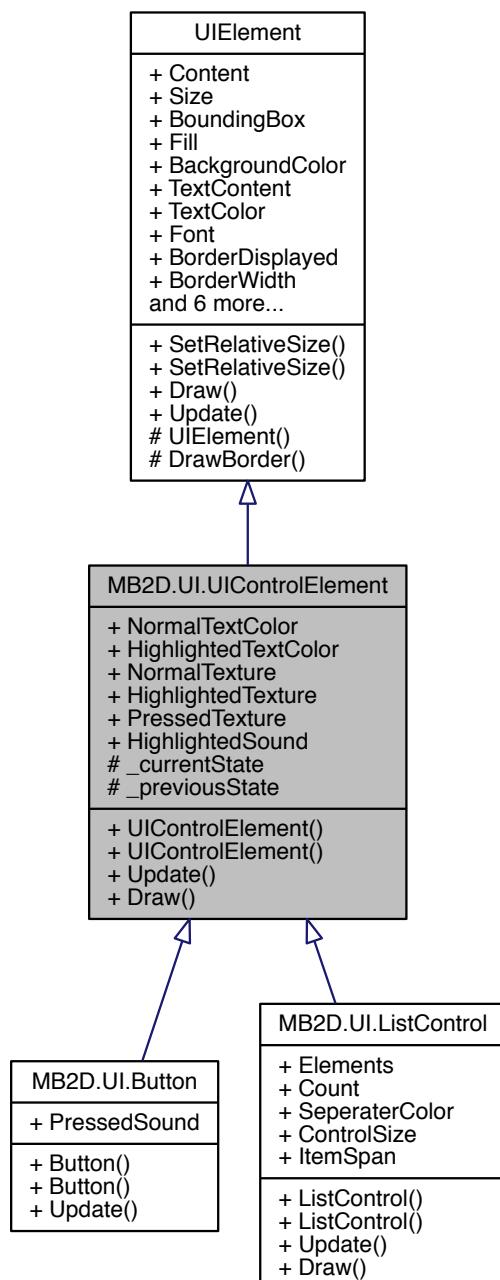
The documentation for this class was generated from the following file:

- MB2D/src/UI/UIData.cs

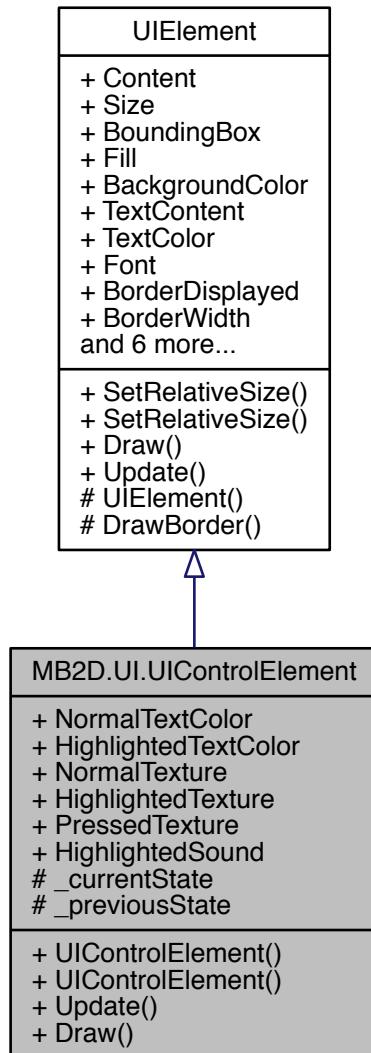
5.93 MB2D.UI.UIControlElement Class Reference

An interactive and controllable [UIElement](#)

Inheritance diagram for MB2D.UI.UIControlElement:



Collaboration diagram for MB2D.UI.UIControlElement:



Public Member Functions

- [UIControlElement](#) (Texture2D normal, Texture2D selected, Texture2D pressed)

Initializes a new instance of the T:MB2D.UI.UIControlElement class.
- [UIControlElement \(\)](#)

Initializes a new instance of the T:MB2D.UI.UIControlElement class with no textures associated
- [override void Update \(\)](#)

Update the UIState of the control element based on mouse position
- [override void Draw \(SpriteBatch spriteBatch\)](#)

Draws the texture associated with the elements current UIState and then its TextContent on top of the texture

Protected Attributes

- **UIState _currentState**
The current UIState of the element
- **UIState _previousState**
The last state of the element

Properties

- Color **NormalTextColor** [get, set]
Gets or sets the TextContent color associated with the Normal UIState of the element.
- Color **HighlightedTextColor** [get, set]
Gets or sets the TextContent color associated with the Selected UIState of the element.
- Texture2D **NormalTexture** [get, set]
Gets or sets the normal UIState texture.
- Texture2D **HighlightedTexture** [get, set]
Gets or sets the selected UIState texture.
- Texture2D **PressedTexture** [get, set]
Gets or sets the pressed UIState texture.
- SoundEffect **HighlightedSound** [get, set]
Gets or sets the sound played when an element switches to the selected state.

Additional Inherited Members

5.93.1 Detailed Description

An interactive and controllable [UIElement](#)

5.93.2 Constructor & Destructor Documentation

5.93.2.1 UIControlElement() [1/2]

```
MB2D.UI.UIControlElement.UIControlElement (
    Texture2D normal,
    Texture2D selected,
    Texture2D pressed ) [inline]
```

Initializes a new instance of the T:MB2D.UI.UIControlElement class.

Parameters

<i>normal</i>	Normal state texture
<i>selected</i>	Selected state texture
<i>pressed</i>	Pressed state texture

5.93.2.2 UIControlElement() [2/2]

```
MB2D.UI.UIControlElement.UIControlElement ( ) [inline]
```

Initializes a new instance of the T:MB2D.UI.UIControlElement class with no textures associated

5.93.3 Member Function Documentation

5.93.3.1 Draw()

```
override void MB2D.UI.UIControlElement.Draw (
    SpriteBatch spriteBatch) [inline], [virtual]
```

Draws the texture associated with the elements current UIState and then its TextContent on top of the texture

Parameters

<i>spriteBatch</i>	Sprite batch to draw to
--------------------	-------------------------

Reimplemented from [MB2D.UI.UIElement](#).

5.93.3.2 Update()

```
override void MB2D.UI.UIControlElement.Update () [inline], [virtual]
```

Update the UIState of the control element based on mouse position

Implements [MB2D.UI.UIElement](#).

5.93.4 Member Data Documentation

5.93.4.1 _currentState

```
UIState MB2D.UI.UIControlElement._currentState [protected]
```

The current UIState of the element

5.93.4.2 _previousState

```
UIState MB2D.UI.UIControlElement._previousState [protected]
```

The last state of the element

5.93.5 Property Documentation

5.93.5.1 HighlightedSound

```
SoundEffect MB2D.UI.UIControlElement.HighlightedSound [get], [set]
```

Gets or sets the sound played when an element switches to the selected state.

The highlighted state sound effect.

5.93.5.2 HighlightedTextColor

```
Color MB2D.UI.UIControlElement.HighlightedTextColor [get], [set]
```

Gets or sets the TextContent color associated with the Selected UIState of the element.

The TextContents selected color

5.93.5.3 HighlightedTexture

```
Texture2D MB2D.UI.UIControlElement.HighlightedTexture [get], [set]
```

Gets or sets the selected UIState texture.

The selected texture.

5.93.5.4 NormalTextColor

```
Color MB2D.UI.UIControlElement.NormalTextColor [get], [set]
```

Gets or sets the TextContent color associated with the Normal UIState of the element.

The TextContents normal color

5.93.5.5 NormalTexture

```
Texture2D MB2D.UI.UIControlElement.NormalTexture [get], [set]
```

Gets or sets the normal UIState texture.

The normal texture.

5.93.5.6 PressedTexture

```
Texture2D MB2D.UI.UIControlElement.PressedTexture [get], [set]
```

Gets or sets the pressed UIState texture.

The pressed texture.

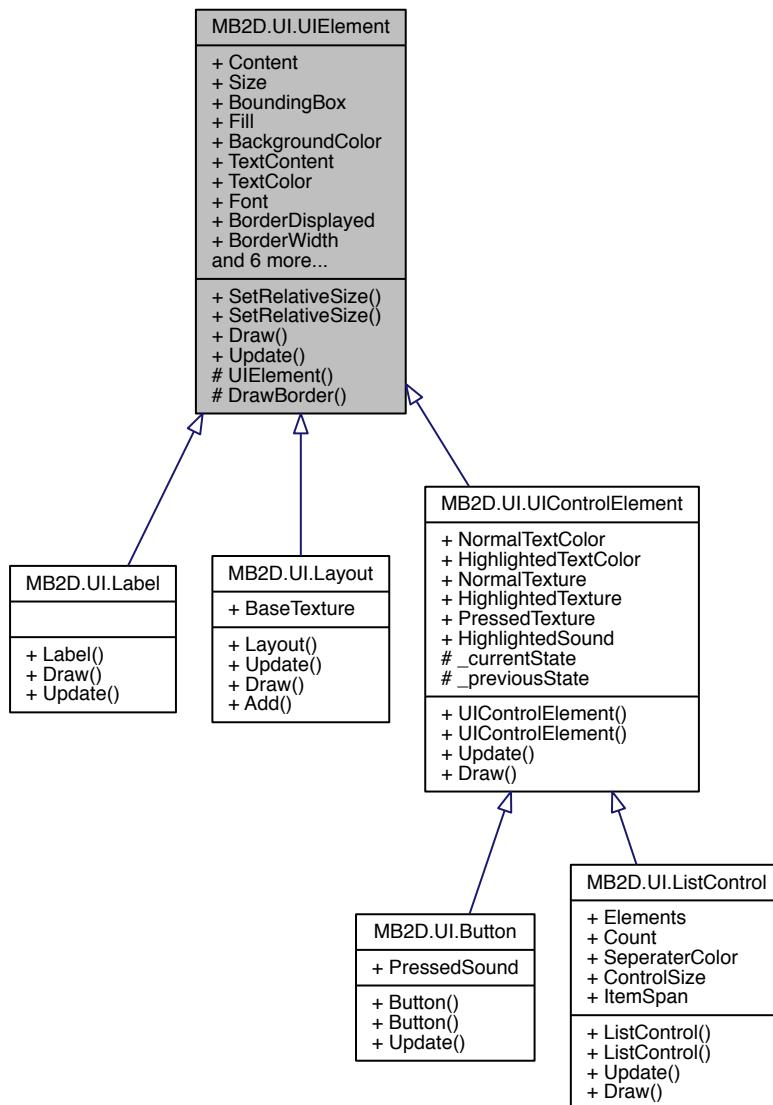
The documentation for this class was generated from the following file:

- MB2D/src/UI/UIControlElement.cs

5.94 MB2D.UI.UIElement Class Reference

Defines a [UI](#) object that can be contained within Views and Layouts, drawn, updated, and moved about

Inheritance diagram for MB2D.UI.UIElement:



Collaboration diagram for MB2D.UI.UIElement:



Public Member Functions

- void [SetRelativeSize](#) ([UIContent](#) parent, [Point](#) at, [Point](#) span)
Sets the size of the element relative to its parent
- void [SetRelativeSize](#) ([UIContent](#) parent, int atRow, int atCol, int rowSpan, int colSpan)
Sets the size of the element relative to its parent
- virtual void [Draw](#) ([SpriteBatch](#) spriteBatch)
Draws the element to the window. Overridden in derived classes
- abstract void [Update](#) ()
Update the elements state and handles input. Overridden in derived classes

Protected Member Functions

- [UIElement](#) (int rows, int cols)
Initializes a new instance of the T:MB2D.UI.UIElement class. Sets default property values
- void [DrawBorder](#) ([SpriteBatch](#) spriteBatch)
Draws the elements border to the window. Skips sides that have color set to Color.Transparent

Properties

- [UIContent](#) [Content](#) [get]
Gets the [UIContent](#) of the element, only available in container elements
- [Vector2](#) [Size](#) [get]
Gets the column and row count of the element

- Rectangle **BoundingBox** [get]

Gets the bounding box of this element
- bool **Fill** [get, set]

Gets or sets a value that indicates the element should be stretched or shrunk to fill its parents bounds exactly
- Color **BackgroundColor** [get, set]

Gets or sets the color of the elements background.
- string **TextContent** [get, set]

Gets or sets the string rendered by the element.
- Color **TextColor** [get, set]

Gets or sets the current color of the text.
- SpriteFont **Font** [get, set]

Gets or sets the font used in rendering the elements TextContent
- bool **BorderDisplayed** [get, set]

Gets or sets a value indicating whether this T:MB2D.UI.UIElement has its border displayed.
- int **BorderWidth** [get, set]

Gets or sets the width of the border.
- Color **BorderColor** [get, set]

Gets or sets the color of the border.
- Color **BorderTopColor** [get, set]

Gets or sets the color of the border top.
- Color **BorderRightColor** [get, set]

Gets or sets the color of the border right.
- Color **BorderBottomColor** [get, set]

Gets or sets the color of the border bottom.
- Color **BorderLeftColor** [get, set]

Gets or sets the color of the border left.
- string **Tag** [get, set]

Gets or sets the tag used to quickly access this element and uniquely identify it.

5.94.1 Detailed Description

Defines a [UI](#) object that can be contained within Views and Layouts, drawn, updated, and moved about

5.94.2 Constructor & Destructor Documentation

5.94.2.1 UIElement()

```
MB2D.UI.UIElement.UIElement (
    int rows,
    int cols ) [inline], [protected]
```

Initializes a new instance of the T:MB2D.UI.UIElement class. Sets default property values

Parameters

<i>rows</i>	Number of rows this element should span - used only be used for container elements
<i>cols</i>	Number of columns this element should span - used only be used for container elements.

5.94.3 Member Function Documentation

5.94.3.1 Draw()

```
virtual void MB2D.UI.UIElement.Draw (
    SpriteBatch spriteBatch ) [inline], [virtual]
```

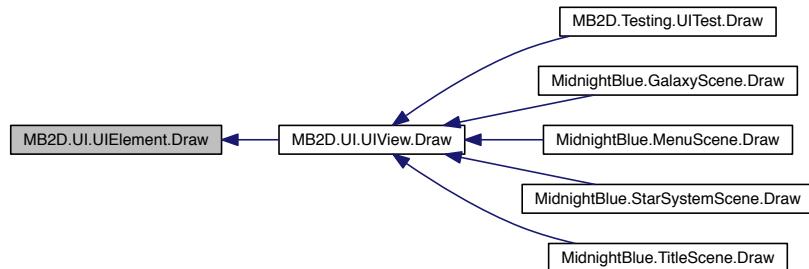
Draws the element to the window. Overridden in derived classes

Parameters

<i>spriteBatch</i>	Sprite batch to draw to.
--------------------	--------------------------

Reimplemented in [MB2D.UI.ListControl](#), [MB2D.UI.UIControlElement](#), [MB2D.UI.Layout](#), and [MB2D.UI.Label](#).

Here is the caller graph for this function:



5.94.3.2 DrawBorder()

```
void MB2D.UI.UIElement.DrawBorder (
    SpriteBatch spriteBatch ) [inline], [protected]
```

Draws the elements border to the window. Skips sides that have color set to Color.Transparent

Parameters

<i>spriteBatch</i>	Sprite batch to draw the border to.
--------------------	-------------------------------------

5.94.3.3 SetRelativeSize() [1/2]

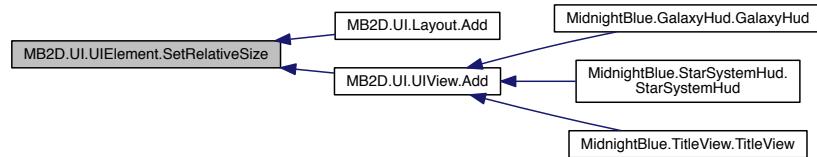
```
void MB2D.UI.UIElement.SetRelativeSize (
    UIContent parent,
    Point at,
    Point span ) [inline]
```

Sets the size of the element relative to its parent

Parameters

<i>parent</i>	Parent content to align to
<i>at</i>	Position element should be set to
<i>span</i>	Number of columns/rows the element should span.

Here is the caller graph for this function:



5.94.3.4 SetRelativeSize() [2/2]

```
void MB2D.UI.UIElement.SetRelativeSize (
    UIContent parent,
    int atRow,
    int atCol,
    int rowSpan,
    int colSpan ) [inline]
```

Sets the size of the element relative to its parent

Parameters

<i>parent</i>	Parent content to align to
<i>atRow</i>	Row position the element should align to.
<i>atCol</i>	Column position the element should align to.
<i>rowSpan</i>	Number of rows in the parent the element should span.
<i>colSpan</i>	Number of columns in the parent the element should span.

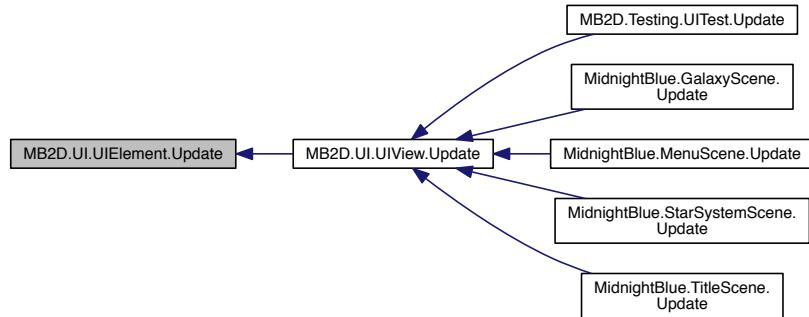
5.94.3.5 Update()

```
abstract void MB2D.UI.UIElement.Update ( ) [pure virtual]
```

Update the elements state and handles input. Overriden in derived classes

Implemented in [MB2D.UI.ListControl](#), [MB2D.UI.UIControlElement](#), [MB2D.UI.Label](#), [MB2D.UI.Button](#), and [MB2D.UI.Layout](#).

Here is the caller graph for this function:



5.94.4 Property Documentation

5.94.4.1 BackgroundColor

```
Color MB2D.UI.UIElement.BackgroundColor [get], [set]
```

Gets or sets the color of the elements background.

The color of the background.

5.94.4.2 BorderBottomColor

```
Color MB2D.UI.UIElement.BorderBottomColor [get], [set]
```

Gets or sets the color of the border bottom.

The color of the border bottom.

5.94.4.3 BorderColor

```
Color MB2D.UI.UIElement.BorderColor [get], [set]
```

Gets or sets the color of the border.

The color of the border. Resets all border sides' colors to this color.

5.94.4.4 BorderDisplayed

```
bool MB2D.UI.UIElement.BorderDisplayed [get], [set]
```

Gets or sets a value indicating whether this T:MB2D.UI.UIElement has its border displayed.

`true` if border should be displayed; otherwise, `false`.

5.94.4.5 BorderLeftColor

```
Color MB2D.UI.UIElement.BorderLeftColor [get], [set]
```

Gets or sets the color of the border left.

The color of the border left.

5.94.4.6 BorderRightColor

```
Color MB2D.UI.UIElement.BorderRightColor [get], [set]
```

Gets or sets the color of the border right.

The color of the border right.

5.94.4.7 BorderTopColor

```
Color MB2D.UI.UIElement.BorderTopColor [get], [set]
```

Gets or sets the color of the border top.

The color of the border top.

5.94.4.8 BorderWidth

```
int MB2D.UI.UIElement.BorderWidth [get], [set]
```

Gets or sets the width of the border.

The width of the border.

5.94.4.9 BoundingBox

```
Rectangle MB2D.UI.UIElement.BoundingBox [get]
```

Gets the bounding box of this element

The bounding box.

5.94.4.10 Content

```
UIContent MB2D.UI.UIElement.Content [get]
```

Gets the [UIContent](#) of the element, only available in container elements

The content.

5.94.4.11 Fill

```
bool MB2D.UI.UIElement.Fill [get], [set]
```

Gets or sets a value that indicates the element should be stretched or shrunk to fill its parents bounds exactly
true if set to fill parent; otherwise, false.

5.94.4.12 Font

```
SpriteFont MB2D.UI.UIElement.Font [get], [set]
```

Gets or sets the font used in rendering the elements TextContent

The font.

5.94.4.13 Size

```
Vector2 MB2D.UI.UIElement.Size [get]
```

Gets the column and row count of the element

The elements grid size.

5.94.4.14 Tag

```
string MB2D.UI.UIElement.Tag [get], [set]
```

Gets or sets the tag used to quickly access this element and uniquely identify it.

The tag.

5.94.4.15 TextColor

```
Color MB2D.UI.UIElement.TextColor [get], [set]
```

Gets or sets the current color of the text.

The text contents current color value.

5.94.4.16 TextContent

```
string MB2D.UI.UIElement.TextContent [get], [set]
```

Gets or sets the string rendered by the element.

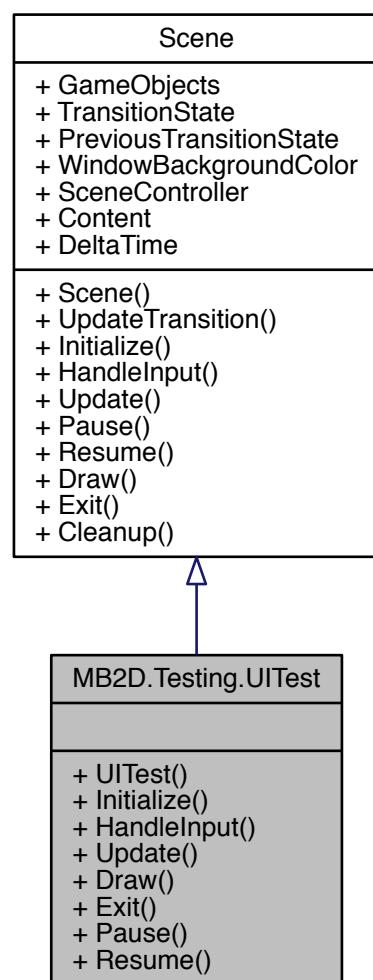
The content of the text.

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

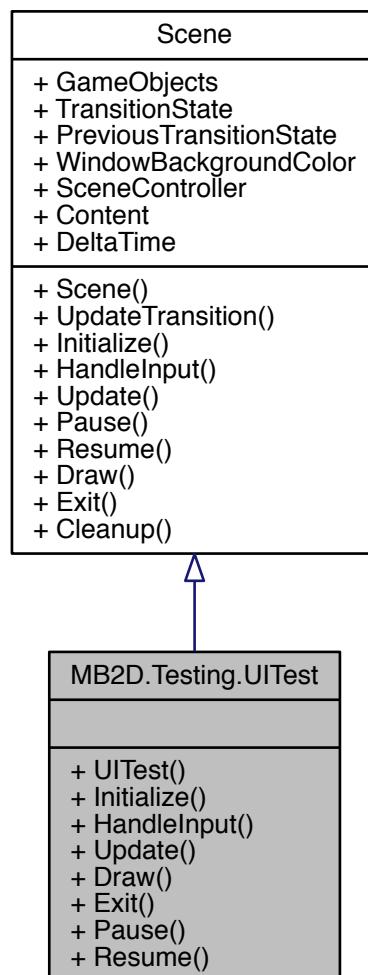
- MB2D/src/UI/UIElement.cs

5.95 MB2D.Testing.UITest Class Reference

Inheritance diagram for MB2D.Testing.UITest:



Collaboration diagram for MB2D.Testing.UITest:



Public Member Functions

- **UITest** ([EntityMap](#) map, [ContentManager](#) content)
- override void [Initialize](#) ()

Initialize this scene and loads all resources. Runs logic to execute during the Initializing state. Set state to None to end.

- override void [HandleInput](#) ()

Handles all input for the scene

- override void [Update](#) ()

Updates game logic and changes scene state.

- override void [Draw](#) ([SpriteBatch](#) spriteBatch, [SpriteBatch](#) uiSpriteBatch)

Draws entities and UI elements to the specified SpriteBatches

- override void [Exit](#) ()

Runs logic to execute while the scene is in the Exiting state. Set state to Null to end.

- `override void Pause ()`
Runs logic to execute while the scene is in the Pausing state. Set state to None to end.
- `override void Resume ()`
Runs logic to execute while the scene is in the Resuming state. Set state to None to end.

Additional Inherited Members

5.95.1 Member Function Documentation

5.95.1.1 Draw()

```
override void MB2D.Testing.UITest.Draw (
    SpriteBatch spriteBatch,
    SpriteBatch uiSpriteBatch ) [inline], [virtual]
```

Draws entities and [UI](#) elements to the specified SpriteBatches

Parameters

<i>spriteBatch</i>	World-coordinate based sprite batch.
<i>uiSpriteBatch</i>	Camera-based User Interface sprite batch.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



5.95.1.2 Exit()

```
override void MB2D.Testing.UITest.Exit ( ) [inline], [virtual]
```

Runs logic to execute while the scene is in the Exiting state. Set state to Null to end.

Implements [MB2D.Scenes.Scene](#).

5.95.1.3 HandleInput()

```
override void MB2D.Testing.UITest.HandleInput ( ) [inline], [virtual]
```

Handles all input for the scene

Implements [MB2D.Scenes.Scene](#).

5.95.1.4 Initialize()

```
override void MB2D.Testing.UITest.Initialize ( ) [inline], [virtual]
```

Initialize this scene and loads all resources. Runs logic to execute during the Initializing state. Set state to None to end.

Implements [MB2D.Scenes.Scene](#).

5.95.1.5 Pause()

```
override void MB2D.Testing.UITest.Pause ( ) [inline], [virtual]
```

Runs logic to execute while the scene is in the Pausing state. Set state to None to end.

Implements [MB2D.Scenes.Scene](#).

5.95.1.6 Resume()

```
override void MB2D.Testing.UITest.Resume ( ) [inline], [virtual]
```

Runs logic to execute while the scene is in the Resuming state. Set state to None to end.

Implements [MB2D.Scenes.Scene](#).

5.95.1.7 Update()

```
override void MB2D.Testing.UITest.Update ( ) [inline], [virtual]
```

Updates game logic and changes scene state.

Implements [MB2D.Scenes.Scene](#).

Here is the call graph for this function:



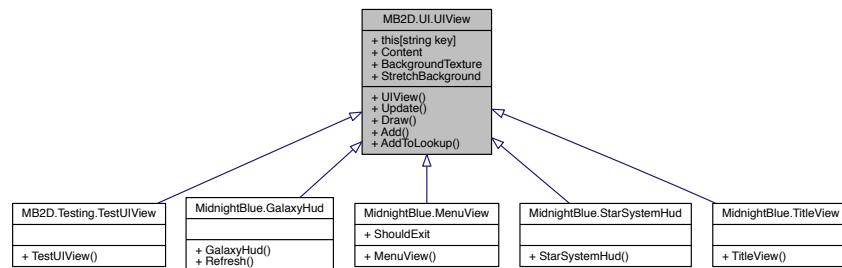
The documentation for this class was generated from the following file:

- MB2D/src/Test/UITest.cs

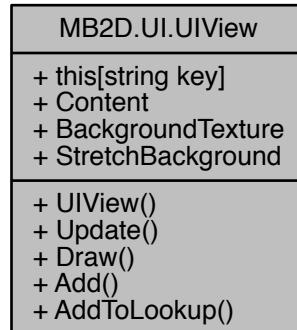
5.96 MB2D.UI.UIView Class Reference

A single context for all [UI](#) elements and layouts.

Inheritance diagram for MB2D.UI.UIView:



Collaboration diagram for MB2D.UI.UIView:



Public Member Functions

- [UIView](#) (int rows, int cols)

Initializes a new instance of the T:MB2D.UI.UIView class. Divides itself into the number of rows and columns evenly based on the size of the current viewport.
- void [Update](#) ()

Updates and handles input for all elements in the View.
- void [Draw](#) (SpriteBatch spriteBatch, SpriteBatch uiSpriteBatch)

Draws the View and its elements to the window
- void [Add](#) ([UIElement](#) element, int atRow, int atCol, int rowSpan, int colSpan)

Adds a new [UIElement](#) to the View
- void [AddToLookup](#) ([UIElement](#) element)

Adds an element to the lookup table used by the view

Properties

- **UIElement this[string key]** [get]
Gets the T:MB2D.UI.UIElement with the specified key.
- **UIElement[,] Content** [get]
Gets the elements this View contains in a 2D array
- **Texture2D BackgroundTexture** [get, set]
Gets or sets the background texture of the view.
- **bool StretchBackground** [get, set]
Gets or sets a value indicating whether the background image should stretch to fit the window.

5.96.1 Detailed Description

A single context for all **UI** elements and layouts.

5.96.2 Constructor & Destructor Documentation

5.96.2.1 UIView()

```
MB2D.UI.UIView.UIView (
    int rows,
    int cols ) [inline]
```

Initializes a new instance of the T:MB2D.UI.UIView class. Divides itself into the number of rows and columns evenly based on the size of the current viewport.

Parameters

<i>rows</i>	Number of rows in the view
<i>cols</i>	Number of columns in the view

5.96.3 Member Function Documentation

5.96.3.1 Add()

```
void MB2D.UI.UIView.Add (
    UIElement element,
    int atRow,
    int atCol,
    int rowSpan,
    int colSpan ) [inline]
```

Adds a new **UIElement** to the View

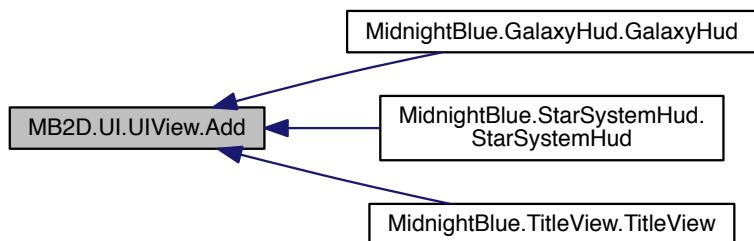
Parameters

<i>element</i>	Element to add.
<i>atRow</i>	Row position in the View.
<i>atCol</i>	Column position in the View.
<i>rowSpan</i>	Number of rows the element takes up. Generated By Doxygen
<i>colSpan</i>	Number of columns the element takes up.

Here is the call graph for this function:



Here is the caller graph for this function:



5.96.3.2 AddToLookup()

```
void MB2D.UI.UIVIEW.AddToLookup (
    UIELEMENT element ) [inline]
```

Adds an element to the lookup table used by the view

Parameters

<i>element</i>	Element to add.
----------------	-----------------

Here is the caller graph for this function:



5.96.3.3 Draw()

```
void MB2D.UI.UIView.Draw (
    SpriteBatch spriteBatch,
    SpriteBatch uiSpriteBatch ) [inline]
```

Draws the View and its elements to the window

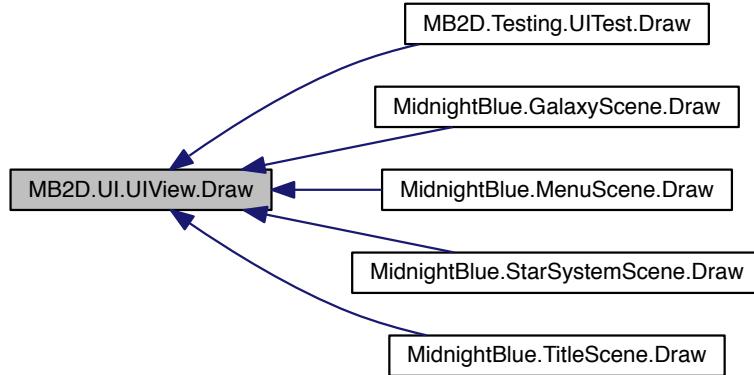
Parameters

<i>uiSpriteBatch</i>	Sprite batch to draw to.
----------------------	--------------------------

Here is the call graph for this function:



Here is the caller graph for this function:



5.96.3.4 Update()

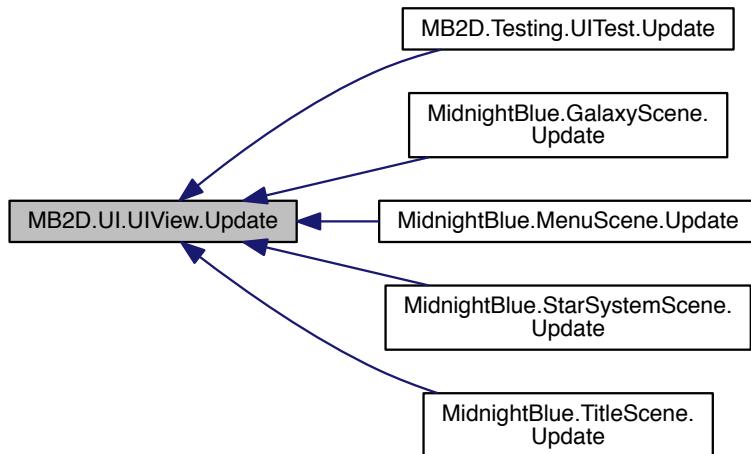
```
void MB2D.UI.UIView.Update ( ) [inline]
```

Updates and handles input for all elements in the View.

Here is the call graph for this function:



Here is the caller graph for this function:



5.96.4 Property Documentation

5.96.4.1 BackgroundTexture

Texture2D MB2D.UI.UIView.BackgroundTexture [get], [set]

Gets or sets the background texture of the view.

The background texture.

5.96.4.2 Content

UIElement [,] MB2D.UI.UIView.Content [get]

Gets the elements this View contains in a 2D array

All UIElements.

5.96.4.3 StretchBackground

```
bool MB2D.UI.UIView.StretchBackground [get], [set]
```

Gets or sets a value indicating whether the background image should stretch to fit the window.

true if background should be stretched; otherwise, false.

5.96.4.4 this[string key]

```
UIElement MB2D.UI.UIView.this[string key] [get]
```

Gets the T:MB2D.UI.UIElement with the specified key.

Parameters

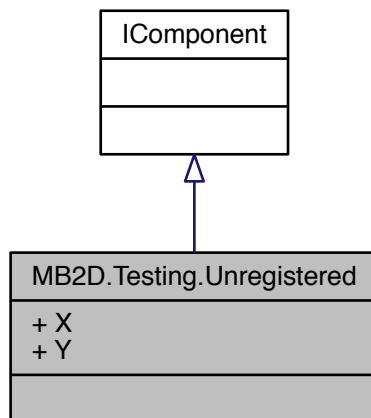
<i>key</i>	Tag of the element.
------------	---------------------

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

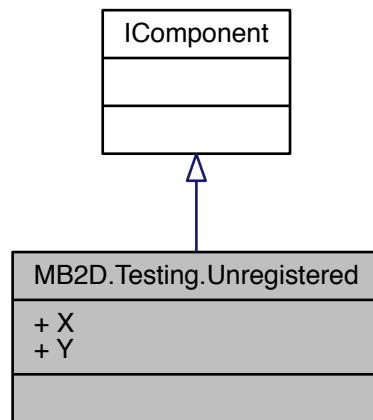
- MB2D/src/UI/UIView.cs

5.97 MB2D.Testing.Unregistered Class Reference

Inheritance diagram for MB2D.Testing.Unregistered:



Collaboration diagram for MB2D.Testing.Unregistered:



Properties

- int **X** [get, set]
- int **Y** [get, set]

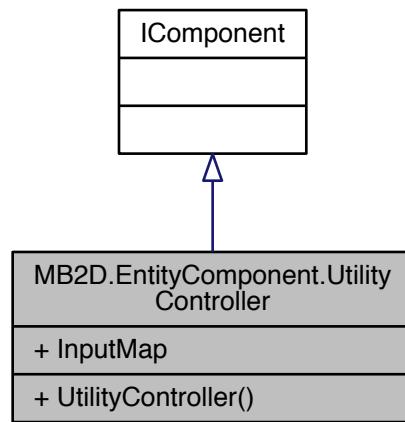
The documentation for this class was generated from the following file:

- MB2D/src/Test/TestSystem.cs

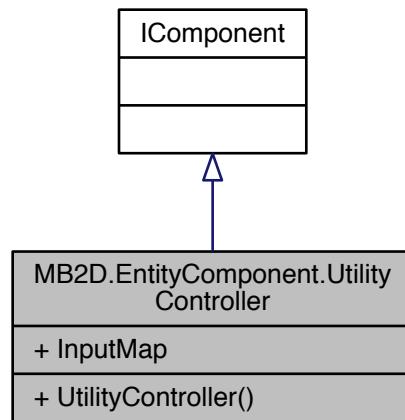
5.98 MB2D.EntityComponent.UtilityController Class Reference

Declares the attached entity as able to control utility commands such as opening the debug console

Inheritance diagram for MB2D.EntityComponent.UtilityController:



Collaboration diagram for MB2D.EntityComponent.UtilityController:



Public Member Functions

- [UtilityController \(\)](#)

Initializes a new instance of the T:MidnightBlue.UtilityController component with default input assignment

Properties

- [InputMap InputMap \[get\]](#)
Gets the input map.

5.98.1 Detailed Description

Declares the attached entity as able to control utility commands such as opening the debug console

5.98.2 Constructor & Destructor Documentation

5.98.2.1 UtilityController()

```
MB2D.EntityComponent.UtilityController.UtilityController () [inline]
```

Initializes a new instance of the T:MidnightBlue.UtilityController component with default input assignment

5.98.3 Property Documentation

5.98.3.1 InputMap

```
InputMap MB2D.EntityComponent.UtilityController.InputMap [get]
```

Gets the input map.

The input map.

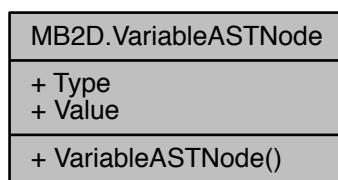
The documentation for this class was generated from the following file:

- MB2D/src/EntityComponent/Components/UtilityController.cs

5.99 MB2D.VariableASTNode Class Reference

Represents a variable with a type and a value

Collaboration diagram for MB2D.VariableASTNode:



Public Member Functions

- **VariableASTNode (Type type, object value)**
Initializes a new instance of the T:MB2D.VariableASTNode class.

Properties

- Type **Type** [get]
Gets the variables type info
- object **Value** [get]
Gets the value to assign to the variable

5.99.1 Detailed Description

Represents a variable with a type and a value

5.99.2 Constructor & Destructor Documentation

5.99.2.1 VariableASTNode()

```
MB2D.VariableASTNode.VariableASTNode (
    Type type,
    object value ) [inline]
```

Initializes a new instance of the T:MB2D.VariableASTNode class.

Parameters

<i>type</i>	Type of the variable.
<i>value</i>	Value to assign.

5.99.3 Property Documentation

5.99.3.1 Type

Type MB2D.VariableASTNode.Type [get]

Gets the variables type info

The type.

5.99.3.2 Value

object MB2D.VariableASTNode.Value [get]

Gets the value to assign to the variable

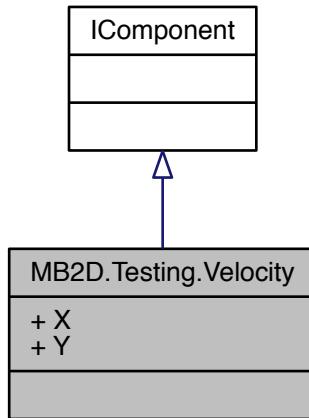
The value.

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

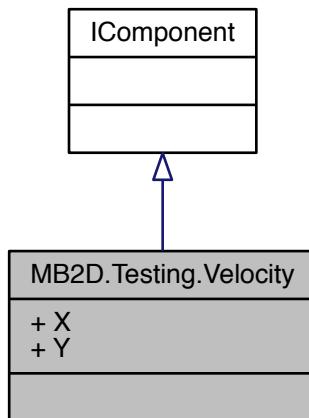
- MB2D/src/MBConsole/MBConsoleAST.cs

5.100 MB2D.Testing.Velocity Class Reference

Inheritance diagram for MB2D.Testing.Velocity:



Collaboration diagram for MB2D.Testing.Velocity:



Properties

- int **X** [get, set]
- int **Y** [get, set]

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

- MB2D/src/Test/TestSystem.cs