

Scripting API Reference - Dungeon Architect (Unity)

1.0.1

Generated by Doxygen 1.8.10

Wed Feb 17 2016 21:12:33

Contents

1	Namespace Index	1
1.1	Packages	1
2	Hierarchical Index	3
2.1	Class Hierarchy	3
3	Class Index	7
3.1	Class List	7
4	Namespace Documentation	13
4.1	DungeonArchitect Namespace Reference	13
4.1.1	Enumeration Type Documentation	16
4.1.1.1	CellType	16
4.1.1.2	DungeonModelBuildState	16
4.1.1.3	LandscapeTextureType	16
4.2	DungeonArchitect.Editors Namespace Reference	16
4.2.1	Enumeration Type Documentation	18
4.2.1.1	GraphMenuAction	18
4.3	DungeonArchitect.Graphs Namespace Reference	18
4.3.1	Enumeration Type Documentation	19
4.3.1.1	GraphPinMouseState	19
4.3.1.2	GraphPinType	19
4.4	DungeonArchitect.Navigation Namespace Reference	19
4.5	DungeonArchitect.Triangulator Namespace Reference	19
4.6	DungeonArchitect.Triangulator.Geometry Namespace Reference	20
4.7	DungeonArchitect.Utills Namespace Reference	20
5	Class Documentation	21
5.1	DungeonArchitect.Editors.AssetThumbnailCache Class Reference	21
5.1.1	Detailed Description	21
5.1.2	Member Function Documentation	21
5.1.2.1	GetThumb(Object asset)	21
5.1.2.2	Reset()	22

5.1.3	Property Documentation	22
5.1.3.1	Instance	22
5.2	DungeonArchitect.Utills.Blackboard Class Reference	22
5.2.1	Detailed Description	22
5.3	DungeonArchitect.Utills.BlackboardDatabase< T > Class Template Reference	22
5.4	DungeonArchitect.BlurFilter Class Reference	22
5.4.1	Detailed Description	23
5.5	DungeonArchitect.Cell Class Reference	23
5.5.1	Detailed Description	23
5.6	DungeonArchitect.CellDoor Class Reference	24
5.6.1	Detailed Description	24
5.6.2	Property Documentation	24
5.6.2.1	AdjacentTiles	24
5.7	DungeonArchitect.CellHeightFrameInfo Class Reference	24
5.7.1	Detailed Description	24
5.8	DungeonArchitect.CellHeightNode Class Reference	25
5.8.1	Detailed Description	25
5.9	DungeonArchitect.Navigation.Collision2DTriangleProvider Class Reference	25
5.10	DungeonArchitect.Navigation.CollisionTriangleProvider Class Reference	25
5.11	DungeonArchitect.Editors.CursorDragLink Class Reference	26
5.11.1	Detailed Description	26
5.12	DungeonArchitect.DebugDrawer Class Reference	26
5.12.1	Detailed Description	27
5.13	DungeonArchitect.Triangulator.Delauney Class Reference	27
5.13.1	Detailed Description	27
5.13.2	Member Function Documentation	27
5.13.2.1	Triangulate(List< Triangulator.Geometry.Point > Vertex)	27
5.14	DungeonArchitect.DoorManager Class Reference	28
5.14.1	Detailed Description	28
5.14.2	Member Function Documentation	28
5.14.2.1	ContainsDoorBetweenCells(int cellA, int cellB)	28
5.14.2.2	CreateDoor(IntVector p1, IntVector p2, int cellId1, int cellId2)	29
5.14.3	Property Documentation	29
5.14.3.1	Doors	29
5.15	DungeonArchitect.Dungeon Class Reference	29
5.15.1	Detailed Description	30
5.15.2	Member Function Documentation	30
5.15.2.1	AddPaintCell(IntVector location, bool automaticRebuild)	30
5.15.2.2	Build()	30
5.15.2.3	ClearToolOverlayData(bool automaticRebuild)	30

5.15.2.4	DestroyDungeon()	31
5.15.2.5	ReapplyTheme()	31
5.15.2.6	RemovePaintCell(IntVector location, bool automaticRebuild)	31
5.15.2.7	RequestRebuild()	31
5.15.3	Member Data Documentation	31
5.15.3.1	dungeonThemes	31
5.15.4	Property Documentation	31
5.15.4.1	ActiveModel	31
5.15.4.2	IsLayoutBuilt	31
5.16	DungeonArchitect.Editors.DungeonArchitectGraphEditor Class Reference	32
5.16.1	Detailed Description	32
5.17	DungeonArchitect.Editors.DungeonAssetPostprocessor Class Reference	32
5.17.1	Detailed Description	32
5.18	DungeonArchitect.DungeonBuilder Class Reference	32
5.18.1	Detailed Description	34
5.18.2	Member Function Documentation	34
5.18.2.1	BuildDungeon(DungeonConfig config, DungeonModel model)	34
5.18.2.2	EmitCustomMarkers()	34
5.18.2.3	EmitMarkers()	34
5.19	DungeonArchitect.DungeonConfig Class Reference	34
5.19.1	Detailed Description	35
5.20	DungeonArchitect.Utls.DungeonConstants Class Reference	35
5.20.1	Detailed Description	35
5.21	DungeonArchitect.Editors.DungeonEditorHelper Class Reference	35
5.21.1	Detailed Description	36
5.21.2	Member Function Documentation	36
5.21.2.1	AddToAsset(Graph graph, GraphNode node)	36
5.21.2.2	AddToAsset(Graph graph, GraphLink link)	36
5.21.2.3	CreateEditorTag(string tag)	36
5.21.2.4	MakeFilenameUnique(string dir, string filename)	36
5.21.2.5	MarkAsDirty(Graph graph)	37
5.21.2.6	OnOpenAsset(int instanceID, int line)	37
5.21.2.7	ShowEditor(Graph graph)	37
5.22	DungeonArchitect.Editors.DungeonEditorResources Class Reference	37
5.22.1	Detailed Description	38
5.22.2	Member Function Documentation	38
5.22.2.1	GetResource< T >(string path)	38
5.23	DungeonArchitect.DungeonEventListener Class Reference	38
5.23.1	Detailed Description	39
5.23.2	Member Function Documentation	39

5.23.2.1	OnDungeonDestroyed(Dungeon dungeon)	39
5.23.2.2	OnPostDungeonBuild(Dungeon dungeon, DungeonModel model)	39
5.23.2.3	OnPostDungeonLayoutBuild(Dungeon dungeon, DungeonModel model)	39
5.24	DungeonArchitect.DungeonMarkerEmitter Class Reference	39
5.24.1	Detailed Description	40
5.24.2	Member Function Documentation	40
5.24.2.1	EmitMarkers(DungeonBuilder builder)	40
5.25	DungeonArchitect.DungeonModel Class Reference	40
5.25.1	Detailed Description	41
5.26	DungeonArchitect.Navigation.DungeonNavAgent Class Reference	41
5.27	DungeonArchitect.Navigation.DungeonNavAgent2D Class Reference	41
5.28	DungeonArchitect.Navigation.DungeonNavAgent3D Class Reference	42
5.29	DungeonArchitect.Navigation.DungeonNavMesh Class Reference	43
5.30	DungeonNavMeshEditor Class Reference	44
5.31	DungeonArchitect.Editors.DungeonObjectTraker Class Reference	44
5.31.1	Detailed Description	45
5.31.2	Member Function Documentation	45
5.31.2.1	RequestRebuild()	45
5.31.3	Property Documentation	45
5.31.3.1	ActiveGraph	45
5.31.3.2	Dungeons	45
5.32	DungeonArchitect.DungeonPaintMode Class Reference	45
5.32.1	Detailed Description	46
5.32.2	Member Function Documentation	46
5.32.2.1	GetDungeon()	46
5.32.2.2	GetDungeonConfig()	46
5.32.2.3	GetDungeonModel()	46
5.33	DungeonArchitect.Editors.DungeonPaintModeEditor Class Reference	46
5.33.1	Detailed Description	47
5.34	DungeonArchitect.DungeonPaintModeGrid Class Reference	47
5.34.1	Detailed Description	47
5.34.2	Member Data Documentation	48
5.34.2.1	brushSize	48
5.34.2.2	cursorLogicalHeight	48
5.34.2.3	mode2D	48
5.34.2.4	overlayOpacity	48
5.35	DungeonArchitect.Editors.DungeonPaintModeGridEditor Class Reference	48
5.35.1	Detailed Description	48
5.36	DungeonArchitect.DungeonPropDataAsset Class Reference	49
5.36.1	Detailed Description	49

5.37	DungeonArchitect.Editors.DungeonPropertyEditor Class Reference	49
5.37.1	Detailed Description	49
5.38	DungeonArchitect.DungeonSceneProvider Class Reference	49
5.38.1	Detailed Description	50
5.38.2	Member Function Documentation	51
5.38.2.1	AddGameObject(GameObjectPropTypeData gameObjectProp, Matrix4x4 transform)	51
5.38.2.2	AddSprite(SpritePropTypeData spriteProp, Matrix4x4 transform)	52
5.38.2.3	OnDungeonBuildStart()	52
5.38.2.4	OnDungeonBuildStop()	52
5.38.3	Member Data Documentation	52
5.38.3.1	config	52
5.38.3.2	dungeon	52
5.38.3.3	itemParent	52
5.39	DungeonArchitect.DungeonSceneProviderData Class Reference	52
5.39.1	Detailed Description	53
5.39.2	Member Data Documentation	53
5.39.2.1	affectsNavigation	53
5.39.2.2	dungeon	53
5.39.2.3	NodeId	53
5.40	DungeonArchitect.DungeonToolData Class Reference	53
5.40.1	Detailed Description	54
5.41	DungeonArchitect.Triangulator.Geometry.Edge Class Reference	54
5.41.1	Detailed Description	54
5.41.2	Constructor & Destructor Documentation	54
5.41.2.1	Edge(int point1, int point2)	54
5.41.2.2	Edge()	55
5.41.3	Member Function Documentation	55
5.41.3.1	Equals(Edge other)	55
5.41.4	Member Data Documentation	55
5.41.4.1	p1	55
5.41.4.2	p2	55
5.42	DungeonArchitect.Filter Interface Reference	55
5.42.1	Detailed Description	56
5.43	DungeonArchitect.Graphs.GameObjectNode Class Reference	56
5.43.1	Member Function Documentation	56
5.43.1.1	CopyFrom(GraphNode node)	56
5.44	DungeonArchitect.GameObjectPropTypeData Class Reference	57
5.44.1	Detailed Description	57
5.45	DungeonArchitect.Graphs.Graph Class Reference	57

5.45.1	Detailed Description	58
5.45.2	Member Function Documentation	58
5.45.2.1	GetNode(string id)	58
5.45.2.2	GetNodes< T >()	58
5.45.2.3	MarkAsDirty()	58
5.45.2.4	NotifyStateChanged()	59
5.45.3	Property Documentation	59
5.45.3.1	Links	59
5.45.3.2	Nodes	59
5.45.3.3	TopZIndex	59
5.46	DungeonArchitect.Graphs.GraphCamera Class Reference	59
5.46.1	Detailed Description	60
5.46.2	Member Function Documentation	60
5.46.2.1	FocusOnBestFit(Graph graph, Rect editorBounds)	60
5.46.2.2	FocusOnMarker(Graph graph, Rect editorBounds, string markerName)	60
5.46.2.3	HandleInput(Event e)	60
5.46.2.4	Pan(int x, int y)	60
5.46.2.5	Pan(Vector2 delta)	60
5.46.2.6	ScreenToWorld(Vector2 screenCoord)	60
5.46.2.7	WorldToScreen(Vector2 worldCoord)	61
5.46.3	Property Documentation	61
5.46.3.1	Position	61
5.46.3.2	Scale	61
5.47	DungeonArchitect.Editors.GraphContextMenu Class Reference	61
5.47.1	Detailed Description	62
5.47.2	Member Function Documentation	62
5.47.2.1	HandleInput(Event e)	62
5.47.2.2	Show(Graph graph, GraphPin sourcePin, Vector2 mouseWorld)	62
5.47.2.3	Show(Graph graph)	62
5.48	DungeonArchitect.Editors.GraphContextMenuEvent Class Reference	62
5.48.1	Detailed Description	62
5.49	DungeonArchitect.Editors.GraphEditor Class Reference	63
5.49.1	Detailed Description	63
5.49.2	Member Function Documentation	63
5.49.2.1	CreateNode< T >(Vector2 screenCoord)	63
5.49.2.2	Draw(Rect bounds)	64
5.49.2.3	FocusCameraOnBestFit(Rect editorBounds)	64
5.49.2.4	FocusCameraOnMarker(string markerName, Rect editorBounds)	64
5.49.2.5	GetPinUnderPosition(Vector2 worldPosition)	64
5.49.2.6	HandleInput(Event e)	64

5.49.2.7	Init(Graph graph, Rect editorBounds)	65
5.49.2.8	SelectNode(GraphNode nodeToSelect)	65
5.49.3	Property Documentation	65
5.49.3.1	Graph	65
5.49.3.2	RealtimeUpdate	65
5.50	DungeonArchitect.Editors.GraphEditorConstants Class Reference	65
5.50.1	Detailed Description	66
5.51	DungeonArchitect.Editors.GraphInputHandler Class Reference	66
5.51.1	Member Function Documentation	66
5.51.1.1	HandleNodeInput(GraphNode node, Event e, GraphCamera camera)	66
5.51.1.2	HandlePinInput(GraphPin pin, Event e, GraphCamera camera)	66
5.52	DungeonArchitect.Editors.GraphInspector Class Reference	66
5.52.1	Detailed Description	67
5.52.2	Member Function Documentation	67
5.52.2.1	OnInspectorGUI()	67
5.53	DungeonArchitect.Graphs.GraphLink Class Reference	67
5.53.1	Detailed Description	67
5.53.2	Member Function Documentation	68
5.53.2.1	GetTangentStrength()	68
5.53.3	Property Documentation	68
5.53.3.1	Graph	68
5.53.3.2	Id	68
5.53.3.3	Input	68
5.53.3.4	Output	68
5.54	DungeonArchitect.Editors.GraphLinkRenderer Class Reference	68
5.54.1	Detailed Description	68
5.55	DungeonArchitect.Graphs.GraphNode Class Reference	69
5.55.1	Detailed Description	70
5.55.2	Member Function Documentation	70
5.55.2.1	CopyFrom(GraphNode node)	70
5.55.2.2	CreatePin(GraphPinType pinType, Vector2 position, Rect boundsOffset, Vector2 tangent)	70
5.55.2.3	DragNode(Vector2 delta)	71
5.55.2.4	GetChildNodes()	71
5.55.2.5	GetParentNodes()	71
5.55.3	Property Documentation	71
5.55.3.1	Bounds	71
5.55.3.2	Caption	71
5.55.3.3	Graph	71
5.55.3.4	Id	71

5.55.3.5	InputPin	71
5.55.3.6	InputPins	71
5.55.3.7	OutputPin	72
5.55.3.8	OutputPins	72
5.55.3.9	Position	72
5.55.3.10	Selected	72
5.55.3.11	Size	72
5.55.3.12	ZIndex	72
5.56	DungeonArchitectEditors.GraphNodeRenderer Class Reference	72
5.56.1	Detailed Description	73
5.57	DungeonArchitectEditors.GraphNodeRendererFactory Class Reference	73
5.58	DungeonArchitectEditors.GraphOperations Class Reference	73
5.58.1	Member Function Documentation	73
5.58.1.1	BreakInputLinks(GraphNode node)	73
5.58.1.2	BreakOutputLinks(GraphNode node)	73
5.58.1.3	CreateLink< T >(Graph graph, GraphPin output, GraphPin input)	74
5.58.1.4	CreateNode(Graph graph, Type t)	75
5.58.1.5	CreateNode< T >(Graph graph)	75
5.58.1.6	DestroyLink(GraphLink link)	75
5.58.1.7	DestroyNode(GraphNode node)	75
5.58.1.8	DuplicateNode< T >(Graph graph, T originalNode)	76
5.59	DungeonArchitectGraphs.GraphPin Class Reference	76
5.59.1	Detailed Description	77
5.59.2	Member Function Documentation	77
5.59.2.1	ContainsPoint(Vector2 worldPoint)	77
5.59.2.2	GetBounds()	77
5.59.2.3	GetConnctectedLinks()	78
5.59.2.4	GetWorldBounds()	78
5.59.3	Property Documentation	78
5.59.3.1	BoundsOffset	78
5.59.3.2	ClickState	78
5.59.3.3	Node	78
5.59.3.4	PinType	78
5.59.3.5	Position	78
5.59.3.6	Tangent	78
5.59.3.7	TangentStrength	78
5.59.3.8	WorldPosition	78
5.59.4	Event Documentation	79
5.59.4.1	PinLinksDestroyed	79
5.60	DungeonArchitectEditors.GraphPinHierarchyComparer Class Reference	79

5.60.1 Detailed Description	79
5.61 DungeonArchitect.Editors.GraphPinRenderer Class Reference	79
5.61.1 Detailed Description	79
5.62 DungeonArchitect.Editors.GraphRendererContext Class Reference	80
5.62.1 Detailed Description	80
5.63 DungeonArchitect.Graphs.GraphSchema Class Reference	80
5.63.1 Detailed Description	80
5.63.2 Member Function Documentation	80
5.63.2.1 CanCreateLink(GraphPin output, GraphPin input)	80
5.63.2.2 CanCreateLink(GraphPin output, GraphPin input, out string errorMessage)	80
5.64 DungeonArchitect.Editors.GraphSelectionBox Class Reference	81
5.64.1 Detailed Description	81
5.64.2 Member Function Documentation	81
5.64.2.1 HandleInput(Event e)	81
5.65 DungeonArchitect.Editors.GraphTooltip Class Reference	82
5.65.1 Detailed Description	82
5.65.2 Member Data Documentation	82
5.65.2.1 message	82
5.66 DungeonArchitect.Editors.GraphTooltipRenderer Class Reference	82
5.66.1 Detailed Description	82
5.67 DungeonArchitect.Utills.GraphUtils Class Reference	82
5.67.1 Detailed Description	83
5.68 DungeonArchitect.GridCellInfo Class Reference	83
5.68.1 Detailed Description	83
5.69 DungeonArchitect.GridDungeonBuilder Class Reference	83
5.69.1 Detailed Description	84
5.69.2 Member Function Documentation	84
5.69.2.1 BuildCells()	84
5.69.2.2 BuildDungeon(DungeonConfig config, DungeonModel model)	84
5.69.2.3 EmitMarkers()	84
5.69.2.4 Seperate(GridDungeonModel gridModel)	84
5.69.2.5 TriangulateRooms(GridDungeonModel gridModel)	85
5.70 DungeonArchitect.GridDungeonConfig Class Reference	85
5.70.1 Detailed Description	86
5.70.2 Member Data Documentation	86
5.70.2.1 CorridorPadding	86
5.70.2.2 CorridorPaddingDoubleSided	86
5.70.2.3 FloorHeight	86
5.70.2.4 HeightVariationProbability	87
5.70.2.5 InitialRoomRadius	87

5.70.2.6	MaxAllowedStairHeight	87
5.70.2.7	MaxCellSize	87
5.70.2.8	MinCellSize	87
5.70.2.9	NormalMean	87
5.70.2.10	NormalStd	87
5.70.2.11	NumCells	87
5.70.2.12	RoomAreaThreshold	87
5.70.2.13	RoomAspectDelta	88
5.70.2.14	SpanningTreeLoopProbability	88
5.70.2.15	StairConnectionTolerance	88
5.71	DungeonArchitect.Editors.GridDungeonConfigPropertyEditor Class Reference	88
5.71.1	Detailed Description	88
5.72	DungeonArchitect.GridDungeonModel Class Reference	88
5.72.1	Detailed Description	89
5.72.2	Member Function Documentation	90
5.72.2.1	BuildCellLookup()	90
5.72.2.2	BuildSpatialCellLookup()	90
5.72.2.3	ContainsStair(int cellA, int cellB)	90
5.72.2.4	FindCellByPosition(IntVector position)	90
5.72.2.5	GetCell(int Id)	90
5.72.2.6	GetGridCellLookup(int x, int z)	90
5.72.2.7	ResetModel()	91
5.72.3	Member Data Documentation	91
5.72.3.1	CellLookup	91
5.72.4	Property Documentation	91
5.72.4.1	Doors	91
5.73	DungeonArchitect.GridDungeonModelUtils Class Reference	91
5.74	DungeonArchitect.Graphs.IndexCounter Class Reference	91
5.74.1	Detailed Description	92
5.75	DungeonArchitect.Editors.InspectorUtils Class Reference	92
5.75.1	Detailed Description	92
5.75.2	Member Function Documentation	92
5.75.2.1	DrawMatrixProperty(string caption, ref Matrix4x4 matrix)	92
5.75.2.2	DrawVectorProperty(string caption, ref Vector3 vector)	92
5.75.2.3	Round(float f, int precision)	92
5.75.2.4	RoundVector(ref Vector3 vector, int precision)	93
5.76	DungeonArchitect.Utils.InstanceCache Class Reference	93
5.76.1	Detailed Description	93
5.76.2	Member Function Documentation	93
5.76.2.1	GetInstance(string typeName)	93

5.77	DungeonArchitect.IntVector Struct Reference	94
5.77.1	Detailed Description	94
5.78	DungeonArchitect.IntVector2Key Struct Reference	94
5.78.1	Detailed Description	95
5.79	DungeonArchitect.Editors.KeyboardState Class Reference	95
5.79.1	Detailed Description	95
5.80	DungeonArchitect.LandscapeDataRasterizer Class Reference	95
5.80.1	Detailed Description	96
5.80.2	Constructor & Destructor Documentation	96
5.80.2.1	LandscapeDataRasterizer(Terrain terrain, float elevation)	96
5.80.3	Member Function Documentation	96
5.80.3.1	DrawCell(float x, float y, float w, float h, float elevation)	96
5.80.3.2	GetHeight(Terrain terrain, float worldX, float worldZ)	97
5.80.3.3	LoadData()	97
5.80.3.4	SaveData()	97
5.80.3.5	SmoothCell(float x, float y, float w, float h, float elevation, int smoothingDistance, AnimationCurve smoothingCurve)	97
5.80.3.6	WorldToTerrainCoord(Terrain terrain, float x, float y, out int gx, out int gy)	97
5.80.3.7	WorldToTerrainTextureCoord(Terrain terrain, float x, float y, out int tx, out int ty)	98
5.81	DungeonArchitect.LandscapeTexture Class Reference	98
5.81.1	Detailed Description	98
5.82	DungeonArchitect.Editors.LandscapeTextureEditor Class Reference	98
5.82.1	Detailed Description	99
5.83	DungeonArchitect.LandscapeTransformerGrid Class Reference	99
5.83.1	Detailed Description	99
5.83.2	Member Function Documentation	100
5.83.2.1	OnPostDungeonLayoutBuild(Dungeon dungeon, DungeonModel model)	100
5.83.3	Member Data Documentation	101
5.83.3.1	groundLevelHeight	101
5.84	DungeonArchitect.Navigation.LayoutFloorTriangleProvider Class Reference	101
5.85	DungeonArchitect.LeftToRightNodeComparer Class Reference	101
5.85.1	Detailed Description	102
5.86	DungeonArchitect.MarkerEmitterCornerBeautifier Class Reference	102
5.86.1	Detailed Description	102
5.86.2	Member Function Documentation	102
5.86.2.1	EmitMarkers(DungeonBuilder builder)	102
5.87	DungeonArchitect.MarkerEmitterEmptySpace Class Reference	103
5.87.1	Detailed Description	103
5.87.2	Member Function Documentation	103
5.87.2.1	EmitMarkers(DungeonBuilder builder)	103

5.88	DungeonArchitect.MarkerEmitterFindLowestPoint Class Reference	103
5.88.1	Detailed Description	104
5.88.2	Member Function Documentation	104
5.88.2.1	EmitMarkers(DungeonBuilder builder)	104
5.89	DungeonArchitect.MarkerEmitterFreeSpaceDecorator Class Reference	104
5.89.1	Detailed Description	105
5.89.2	Member Function Documentation	105
5.89.2.1	EmitMarkers(DungeonBuilder builder)	105
5.90	DungeonArchitect.Graphs.MarkerEmitterNode Class Reference	105
5.91	DungeonArchitect.Editors.MarkerEmitterNodeEditor Class Reference	106
5.91.1	Detailed Description	106
5.92	DungeonArchitect.Editors.MarkerEmitterNodeRenderer Class Reference	106
5.92.1	Detailed Description	107
5.93	DungeonArchitect.Graphs.MarkerNode Class Reference	107
5.94	DungeonArchitect.Editors.MarkerNodeEditor Class Reference	107
5.94.1	Detailed Description	108
5.95	DungeonArchitect.Editors.MarkerNodeRenderer Class Reference	108
5.95.1	Detailed Description	108
5.96	DungeonArchitect.Utills.MathUtils Class Reference	108
5.96.1	Detailed Description	109
5.96.2	Member Function Documentation	109
5.96.2.1	Copy(Rectangle other)	109
5.96.2.2	Divide(Vector3 a, Vector3 b)	110
5.96.2.3	FlipYZ(ref Bounds bounds)	110
5.96.2.4	FlipYZ(Vector3 bounds)	110
5.96.2.5	FlipYZ(IntVector bounds)	110
5.96.2.6	GridToWorld(Vector3 GridCellSize, IntVector v)	111
5.96.2.7	GridToWorld(Vector3 GridCellSize, Vector3 v)	111
5.96.2.8	Intersects(Rect outer, Rect inner)	111
5.96.2.9	Intersects(Rect outer, GraphLink link)	111
5.96.2.10	RoundToIntVector(Vector3 v)	112
5.96.2.11	SnapToGrid(Vector3 position, Vector3 gridCellSize)	113
5.96.2.12	SnapToGrid(Vector3 position, Vector3 gridCellSize, bool useRounding)	113
5.96.2.13	ToIntVector(Vector3 v)	113
5.96.2.14	ToVector3(IntVector v)	113
5.96.2.15	WorldToGrid(Vector3 WorldCoord, Vector3 GridCellSize)	114
5.96.3	Member Data Documentation	114
5.96.3.1	INVALID_LOCATION	114
5.97	DungeonArchitect.Utills.Matrix Class Reference	114
5.97.1	Detailed Description	115

5.97.2	Member Function Documentation	115
5.97.2.1	Copy(Matrix4x4 In)	115
5.97.2.2	DecomposeMatrix(ref Matrix4x4 matrix, out Vector3 localPosition, out Quaternion localRotation, out Vector3 localScale)	115
5.97.2.3	FromGameTransform(Transform t)	115
5.97.2.4	GetRotation(ref Matrix4x4 matrix)	116
5.97.2.5	GetScale(ref Matrix4x4 matrix)	116
5.97.2.6	GetTranslation(ref Matrix4x4 matrix)	116
5.97.2.7	Identity()	116
5.97.2.8	SetTransform(out Matrix4x4 transform, Vector3 position, Quaternion rotation, Vector3 scale)	116
5.97.2.9	SetTransformFromMatrix(Transform transform, ref Matrix4x4 matrix)	117
5.97.2.10	SetTranslation(ref Matrix4x4 matrix, Vector3 translate)	117
5.97.2.11	TranslationMatrix(Vector3 offset)	117
5.97.3	Member Data Documentation	117
5.97.3.1	IdentityQuaternion	117
5.98	DungeonArchitect.Editors.MeshNodeEditor Class Reference	117
5.98.1	Detailed Description	118
5.99	DungeonArchitect.Editors.MeshNodeRenderer Class Reference	118
5.99.1	Detailed Description	118
5.100	DungeonArchitect.Navigation.NavigationBuildInvoker Class Reference	119
5.100.1	Detailed Description	119
5.100.2	Member Function Documentation	119
5.100.2.1	OnPostDungeonBuild(Dungeon dungeon, DungeonModel model)	119
5.101	DungeonArchitect.Navigation.NavigationTriangleProvider Class Reference	120
5.102	DungeonArchitect.NegationVolume Class Reference	120
5.102.1	Detailed Description	120
5.103	DungeonArchitect.Editors.NegationVolumeEditor Class Reference	120
5.103.1	Detailed Description	121
5.104	DungeonArchitect.Editors.NodeDeletionOrderComparer Class Reference	121
5.104.1	Detailed Description	121
5.105	DungeonArchitect.Editors.NodeReversedZIndexComparer Class Reference	121
5.105.1	Detailed Description	122
5.106	DungeonArchitect.Editors.NodeZIndexComparer Class Reference	122
5.106.1	Detailed Description	122
5.107	DungeonArchitect.Graphs.PlaceableNode Class Reference	122
5.107.1	Member Function Documentation	123
5.107.1.1	CopyFrom(GraphNode node)	123
5.108	DungeonArchitect.Editors.PlaceableNodeEditor Class Reference	123
5.108.1	Detailed Description	124

5.109DungeonArchitect.PlatformVolume Class Reference	124
5.109.1 Detailed Description	124
5.110DungeonArchitect.Editors.PlatformVolumeEditor Class Reference	124
5.110.1 Detailed Description	125
5.111DungeonArchitect.Utls.PMRandom Class Reference	125
5.111.1 Detailed Description	125
5.111.2 Constructor & Destructor Documentation	125
5.111.2.1 PMRandom()	125
5.111.2.2 PMRandom(uint seed)	125
5.111.3 Member Function Documentation	126
5.111.3.1 GetNextUniformFloat()	126
5.111.3.2 Initialize(uint seed)	126
5.111.3.3 NextGaussianFloat()	126
5.111.3.4 NextGaussianFloat(float mean, float stdDev)	126
5.112DungeonArchitect.Triangulator.Geometry.Point Class Reference	126
5.112.1 Detailed Description	127
5.113DungeonArchitect.Triangulator.Geometry.Point Class Reference	127
5.113.1 Detailed Description	127
5.114DungeonArchitect.PooledDungeonSceneProvider Class Reference	127
5.114.1 Detailed Description	128
5.114.2 Member Function Documentation	128
5.114.2.1 AddGameObject(GameObjectPropTypeData gameObjectProp, Matrix4x4 transform)	128
5.114.2.2 AddSprite(SpritePropTypeData spriteProp, Matrix4x4 transform)	128
5.114.2.3 OnDungeonBuildStart()	128
5.114.2.4 OnDungeonBuildStop()	128
5.115DungeonArchitect.PropChildSocketData Class Reference	129
5.115.1 Detailed Description	129
5.116DungeonArchitect.PropSocket Class Reference	129
5.116.1 Detailed Description	129
5.117DungeonArchitect.PropTypeData Class Reference	129
5.117.1 Detailed Description	130
5.117.2 Member Data Documentation	130
5.117.2.1 affectsNavigation	130
5.117.2.2 Affinity	130
5.117.2.3 AttachToSocket	130
5.117.2.4 ChildSockets	131
5.117.2.5 ConsumeOnAttach	131
5.117.2.6 IsStaticObject	131
5.117.2.7 NodeId	131

5.117.2.8 Offset	131
5.117.2.9 SelectorRuleClassName	131
5.117.2.10 TransformRuleClassName	131
5.117.2.11 UseSelectionRule	131
5.117.2.12 UseTransformRule	131
5.118 DungeonArchitect.Rectangle Struct Reference	131
5.118.1 Detailed Description	132
5.119 DungeonArchitect.SelectorRule Class Reference	132
5.119.1 Detailed Description	133
5.119.2 Member Function Documentation	133
5.119.2.1 CanSelect(PropSocket socket, Matrix4x4 propTransform, DungeonModel model, System.Random random)	133
5.120 DungeonArchitect.Graphs.SpriteNode Class Reference	133
5.120.1 Member Function Documentation	134
5.120.1.1 CopyFrom(GraphNode node)	134
5.121 DungeonArchitect.Editors.SpriteNodeEditor Class Reference	135
5.121.1 Detailed Description	135
5.122 DungeonArchitect.Editors.SpriteNodeRenderer Class Reference	135
5.122.1 Detailed Description	136
5.123 DungeonArchitect.SpritePropTypeData Class Reference	136
5.123.1 Detailed Description	136
5.124 DungeonArchitect.StairAdjacencyQueueNode Class Reference	136
5.124.1 Detailed Description	137
5.125 DungeonArchitect.StairEdgeInfo Struct Reference	137
5.125.1 Detailed Description	137
5.126 DungeonArchitect.StairInfo Class Reference	137
5.126.1 Detailed Description	137
5.127 DungeonArchitect.Navigation.StaticMeshTriangleProvider Class Reference	138
5.128 DungeonArchitect.Navigation.TerrainTriangleProvider Class Reference	138
5.129 DungeonArchitect.ThemeOverrideVolume Class Reference	139
5.129.1 Detailed Description	139
5.130 DungeonArchitect.Editors.ThemeOverrideVolumeEditor Class Reference	139
5.130.1 Detailed Description	139
5.131 DungeonArchitect.Editors.Timer Class Reference	140
5.131.1 Detailed Description	140
5.131.2 Member Function Documentation	140
5.131.2.1 Update(float deltaSeconds)	140
5.131.3 Property Documentation	140
5.131.3.1 Hertz	140
5.132 DungeonArchitect.TransformationRule Class Reference	140

5.132.1 Detailed Description	141
5.132.2 Member Function Documentation	141
5.132.2.1 GetTransform(PropSocket socket, DungeonModel model, Matrix4x4 prop↔ Transform, System.Random random, out Vector3 outPosition, out Quaternion outRotation, out Vector3 outScale)	141
5.133DungeonArchitect.Triangulator.Geometry.Triangle Struct Reference	141
5.133.1 Detailed Description	142
5.133.2 Constructor & Destructor Documentation	142
5.133.2.1 Triangle(int point1, int point2, int point3)	142
5.133.3 Member Data Documentation	142
5.133.3.1 p1	142
5.133.3.2 p2	142
5.133.3.3 p3	142
5.134DungeonArchitect.Graphs.VisualNode Class Reference	142
5.134.1 Member Function Documentation	143
5.134.1.1 CopyFrom(GraphNode node)	143
5.134.2 Member Data Documentation	144
5.134.2.1 affectsNavigation	144
5.134.2.2 IsStatic	144
5.134.2.3 selectionRuleClassName	144
5.134.2.4 selectionRuleEnabled	144
5.134.2.5 transformRuleClassName	144
5.134.2.6 transformRuleEnabled	144
5.135DungeonArchitect.Editors.VisualNodeEditor Class Reference	144
5.135.1 Detailed Description	145
5.136DungeonArchitect.Editors.VisualNodeRenderer Class Reference	145
5.136.1 Detailed Description	145
5.137DungeonArchitect.Volume Class Reference	145
5.137.1 Detailed Description	146
5.137.2 Member Function Documentation	146
5.137.2.1 GetBounds()	146
5.137.2.2 GetGridTransform(out IntVector positionGrid, out IntVector scaleGrid)	146
5.138DungeonArchitect.Editors.VolumeEditor Class Reference	147
5.138.1 Detailed Description	147

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

DungeonArchitect	13
DungeonArchitect.Editors	16
DungeonArchitect.Graphs	18
DungeonArchitect.Navigation	19
DungeonArchitect.Triangulator	19
DungeonArchitect.Triangulator.Geometry	20
DungeonArchitect.Utils	20

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

AssetPostprocessor	
DungeonArchitect.Editors.DungeonAssetPostprocessor	32
DungeonArchitect.Editors.AssetThumbnailCache	21
DungeonArchitect.Utils.Blackboard	22
DungeonArchitect.Utils.BlackboardDatabase< T >	22
DungeonArchitect.Utils.BlackboardDatabase< DungeonArchitect.IntVector >	22
DungeonArchitect.Utils.BlackboardDatabase< float >	22
DungeonArchitect.Utils.BlackboardDatabase< int >	22
DungeonArchitect.Utils.BlackboardDatabase< string >	22
DungeonArchitect.Utils.BlackboardDatabase< Vector3 >	22
DungeonArchitect.Cell	23
DungeonArchitect.CellDoor	24
DungeonArchitect.CellHeightFrameInfo	24
DungeonArchitect.CellHeightNode	25
DungeonArchitect.Editors.CursorDragLink	26
DungeonArchitect.DebugDrawer	26
DungeonArchitect.Triangulator.Delauney	27
DungeonArchitect.DoorManager	28
DungeonArchitect.Utils.DungeonConstants	35
DungeonArchitect.Editors.DungeonEditorHelper	35
DungeonArchitect.Editors.DungeonEditorResources	37
DungeonArchitect.Editors.DungeonObjectTraker	44
DungeonArchitect.DungeonPropDataAsset	49
DungeonArchitect.DungeonToolData	53
Editor	
DungeonArchitect.Editors.DungeonPaintModeEditor	46
DungeonArchitect.Editors.DungeonPaintModeGridEditor	48
DungeonArchitect.Editors.DungeonPropertyEditor	49
DungeonArchitect.Editors.GraphInspector	66
DungeonArchitect.Editors.GridDungeonConfigPropertyEditor	88
DungeonArchitect.Editors.LandscapeTextureEditor	98
DungeonArchitect.Editors.MarkerNodeEditor	107
DungeonArchitect.Editors.PlaceableNodeEditor	123
DungeonArchitect.Editors.MarkerEmitterNodeEditor	106
DungeonArchitect.Editors.VisualNodeEditor	144
DungeonArchitect.Editors.MeshNodeEditor	117
DungeonArchitect.Editors.SpriteNodeEditor	135
DungeonArchitect.Editors.VolumeEditor	147

DungeonArchitect.Editors.NegationVolumeEditor	120
DungeonArchitect.Editors.PlatformVolumeEditor	124
DungeonArchitect.Editors.ThemeOverrideVolumeEditor	139
DungeonNavMeshEditor	44
EditorWindow	
DungeonArchitect.Editors.DungeonArchitectGraphEditor	32
DungeonArchitect.Filter	55
DungeonArchitect.BlurFilter	22
DungeonArchitect.Graphs.GraphCamera	59
DungeonArchitect.Editors.GraphContextMenu	61
DungeonArchitect.Editors.GraphContextMenuEvent	62
DungeonArchitect.Editors.GraphEditorConstants	65
DungeonArchitect.Editors.GraphInputHandler	66
DungeonArchitect.Editors.GraphLinkRenderer	68
DungeonArchitect.Editors.GraphNodeRenderer	72
DungeonArchitect.Editors.MarkerEmitterNodeRenderer	106
DungeonArchitect.Editors.MarkerNodeRenderer	108
DungeonArchitect.Editors.VisualNodeRenderer	145
DungeonArchitect.Editors.MeshNodeRenderer	118
DungeonArchitect.Editors.SpriteNodeRenderer	135
DungeonArchitect.Editors.GraphNodeRendererFactory	73
DungeonArchitect.Editors.GraphOperations	73
DungeonArchitect.Editors.GraphPinRenderer	79
DungeonArchitect.Editors.GraphRendererContext	80
DungeonArchitect.Graphs.GraphSchema	80
DungeonArchitect.Editors.GraphSelectionBox	81
DungeonArchitect.Editors.GraphTooltip	82
DungeonArchitect.Editors.GraphTooltipRenderer	82
DungeonArchitect.Utills.GraphUtils	82
DungeonArchitect.GridCellInfo	83
DungeonArchitect.GridDungeonModelUtils	91
IComparer	
DungeonArchitect.Editors.GraphPinHierarchyComparer	79
DungeonArchitect.Editors.NodeDeletionOrderComparer	121
DungeonArchitect.Editors.NodeReversedZIndexComparer	121
DungeonArchitect.Editors.NodeZIndexComparer	122
DungeonArchitect.LeftToRightNodeComparer	101
IEquatable	
DungeonArchitect.Triangulator.Geometry.Edge	54
DungeonArchitect.Graphs.IndexCounter	91
DungeonArchitect.Editors.InspectorUtils	92
DungeonArchitect.Utills.InstanceCache	93
DungeonArchitect.IntVector	94
DungeonArchitect.IntVector2Key	94
DungeonArchitect.Editors.KeyboardState	95
DungeonArchitect.LandscapeDataRasterizer	95
DungeonArchitect.LandscapeTexture	98
DungeonArchitect.Utills.MathUtils	108
DungeonArchitect.Utills.Matrix	114
MonoBehaviour	
DungeonArchitect.Dungeon	29
DungeonArchitect.DungeonBuilder	32
DungeonArchitect.GridDungeonBuilder	83
DungeonArchitect.DungeonConfig	34
DungeonArchitect.GridDungeonConfig	85
DungeonArchitect.DungeonEventListener	38
DungeonArchitect.LandscapeTransformerGrid	99

DungeonArchitect.Navigation.NavigationBuildInvoker	119
DungeonArchitect.DungeonMarkerEmitter	39
DungeonArchitect.MarkerEmitterCornerBeautifier	102
DungeonArchitect.MarkerEmitterEmptySpace	103
DungeonArchitect.MarkerEmitterFindLowestPoint	103
DungeonArchitect.MarkerEmitterFreeSpaceDecorator	104
DungeonArchitect.DungeonModel	40
DungeonArchitect.GridDungeonModel	88
DungeonArchitect.DungeonPaintMode	45
DungeonArchitect.DungeonPaintModeGrid	47
DungeonArchitect.DungeonSceneProvider	49
DungeonArchitect.PooledDungeonSceneProvider	127
DungeonArchitect.DungeonSceneProviderData	52
DungeonArchitect.Navigation.DungeonNavAgent	41
DungeonArchitect.Navigation.DungeonNavAgent2D	41
DungeonArchitect.Navigation.DungeonNavAgent3D	42
DungeonArchitect.Navigation.DungeonNavMesh	43
DungeonArchitect.Navigation.NavigationTriangleProvider	120
DungeonArchitect.Navigation.Collision2DTriangleProvider	25
DungeonArchitect.Navigation.CollisionTriangleProvider	25
DungeonArchitect.Navigation.LayoutFloorTriangleProvider	101
DungeonArchitect.Navigation.StaticMeshTriangleProvider	138
DungeonArchitect.Navigation.TerrainTriangleProvider	138
DungeonArchitect.Volume	145
DungeonArchitect.NegationVolume	120
DungeonArchitect.PlatformVolume	124
DungeonArchitect.ThemeOverrideVolume	139
DungeonArchitect.Utills.PMRandom	125
DungeonArchitect.Triangulator.Geometry.Point	127
DungeonArchitect.Triangulator.Geometry.Point < T >	127
DungeonArchitect.PropChildSocketData	129
DungeonArchitect.PropSocket	129
DungeonArchitect.PropTypeData	129
DungeonArchitect.GameObjectPropTypeData	57
DungeonArchitect.SpritePropTypeData	136
DungeonArchitect.Rectangle	131
ScriptableObject	
DungeonArchitect.Editors.GraphEditor	63
DungeonArchitect.Graphs.Graph	57
DungeonArchitect.Graphs.GraphLink	67
DungeonArchitect.Graphs.GraphNode	69
DungeonArchitect.Graphs.MarkerNode	107
DungeonArchitect.Graphs.PlaceableNode	122
DungeonArchitect.Graphs.MarkerEmitterNode	105
DungeonArchitect.Graphs.VisualNode	142
DungeonArchitect.Graphs.GameObjectNode	56
DungeonArchitect.Graphs.SpriteNode	133
DungeonArchitect.Graphs.GraphPin	76
DungeonArchitect.SelectorRule	132
DungeonArchitect.TransformationRule	140
DungeonArchitect.StairAdjacencyQueueNode	136
DungeonArchitect.StairEdgeInfo	137
DungeonArchitect.StairInfo	137
DungeonArchitect.Editors.Timer	140
DungeonArchitect.Triangulator.Geometry.Triangle	141

Chapter 3

Class Index

3.1 Class List

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

DungeonArchitect.Editors.AssetThumbnailCache	
Manages the asset thumbnails to display in the visual nodes	21
DungeonArchitect.Utills.Blackboard	
A blackboard holds global data that can be shared across multiple scripts	22
DungeonArchitect.Utills.BlackboardDatabase< T >	22
DungeonArchitect.BlurFilter	
A fast Gaussian blurring filter applied over a 2D data array	22
DungeonArchitect.Cell	
Data-structure to hold the Cell information. A cell is a piece of the dungeon layout and can be either a room or a corridor	23
DungeonArchitect.CellDoor	
Data-structure to hold the door information	24
DungeonArchitect.CellHeightFrameInfo	
Temporary data-structure used while assigning heights on the dungeon.	24
DungeonArchitect.CellHeightNode	
Temporary data-structure to hold the height data of the cell node A graph is build of the dungeon layout while the heights are assigned and this node contains the cell's height information . . .	25
DungeonArchitect.Navigation.Collision2DTriangleProvider	25
DungeonArchitect.Navigation.CollisionTriangleProvider	25
DungeonArchitect.Editors.CursorDragLink	
Manages a link dragged out of a node with the other end following the mouse cursor	26
DungeonArchitect.DebugDrawer	
Helper functions to draw debug information of the dungeon layout in the scene view	26
DungeonArchitect.Triangulator.Delauney	
Performs the Delauney triangulation on a set of vertices.	27
DungeonArchitect.DoorManager	
Manages the doors in the grid based builder	28
DungeonArchitect.Dungeon	
The main dungeon behavior that manages the creation and destruction of dungeons	29
DungeonArchitect.Editors.DungeonArchitectGraphEditor	
The main editor window for the Theme graph editor. This hosts the graph editor for managing the theme graph	32
DungeonArchitect.Editors.DungeonAssetPostprocessor	
An asset processor to create dungeon tags when DungeonArchitect is imported into a new project	32
DungeonArchitect.DungeonBuilder	
Builds the layout of the dungeon and emits markers around the layout Implement this class to create your own builder	32

DungeonArchitect.DungeonConfig	
Base dungeon configuration. Create your own implementation of this configuration based on your dungeon builder's needs	34
DungeonArchitect.Utills.DungeonConstants	
Contains various project specific constants	35
DungeonArchitect.Editors.DungeonEditorHelper	
Utility functions for various editor based features of Dungeon Architect	35
DungeonArchitect.Editors.DungeonEditorResources	
The resource filename constants used by dungeon architect editor	37
DungeonArchitect.DungeonEventListener	
Listen to various dungeon events during the build and destroy phase	38
DungeonArchitect.DungeonMarkerEmitter	
Marker Emitters let you emit your own markers anywhere in the map. Implement this class and add it to the Dungeon object to add your own markers right after the dungeon layout is created	39
DungeonArchitect.DungeonModel	
Abstract dungeon model. Create your own implementation of the model depending on your builder's needs	40
DungeonArchitect.Navigation.DungeonNavAgent	41
DungeonArchitect.Navigation.DungeonNavAgent2D	41
DungeonArchitect.Navigation.DungeonNavAgent3D	42
DungeonArchitect.Navigation.DungeonNavMesh	43
DungeonNavMeshEditor	44
DungeonArchitect.Editors.DungeonObjectTraker	
Tracks active dungeon objects in the scene and finds ones that have the active graph being edited This is used for real-time updates on the dungeon object as the graph is modified from the editor	44
DungeonArchitect.DungeonPaintMode	
Manage the editor paint mode so you can paint the layout of you dungeon. You should implement your own paint mode depending on your dungeon builder's data structures and requirements	45
DungeonArchitect.Editors.DungeonPaintModeEditor	
Custom property editor for the paint mode object	46
DungeonArchitect.DungeonPaintModeGrid	
Editor tooling for the grid based dungeon builder. Lets you paint with a grid based brush	47
DungeonArchitect.Editors.DungeonPaintModeGridEditor	
Custom property editor for the Paint model game object	48
DungeonArchitect.DungeonPropDataAsset	
The data-structure for serializing the theme graph to disk	49
DungeonArchitect.Editors.DungeonPropertyEditor	
Custom property editor for the dungeon game object	49
DungeonArchitect.DungeonSceneProvider	
A scene provider instantiates game objects into the scene. Implementations can customize the instantiation process if needed (e.g. object pooling etc)	49
DungeonArchitect.DungeonSceneProviderData	
Meta-data added to each spawned game object in the scene. This is used to identify objects that belong to the dungeons, for later destruction and rebuilding	52
DungeonArchitect.DungeonToolData	
Tool Data represented by the grid based builder	53
DungeonArchitect.Triangulator.Geometry.Edge	
Edge made from two point indexes	54
DungeonArchitect.Filter	
A data filter applied over a 2D data array	55
DungeonArchitect.Graphs.GameObjectNode	56
DungeonArchitect.GameObjectPropTypeData	
Game Object node data asset attributes	57
DungeonArchitect.Graphs.Graph	
Theme Graph data structure holds all the theme nodes and their connections	57
DungeonArchitect.Graphs.GraphCamera	
A camera that manages the graph editor's viewport	59

DungeonArchitect.Editors.GraphContextMenu	
The context menu shown when the user right clicks on the theme graph editor	61
DungeonArchitect.Editors.GraphContextMenuEvent	
The graph context menu event data	62
DungeonArchitect.Editors.GraphEditor	
The graph editor script for managing a graph. This contains the bulk of the logic for graph editing	63
DungeonArchitect.Editors.GraphEditorConstants	
Graph editor constants	65
DungeonArchitect.Editors.GraphInputHandler	
	66
DungeonArchitect.Editors.GraphInspector	
Custom property editor for graph objects Shows the graph editor when a theme graph asset is selected	66
DungeonArchitect.Graphs.GraphLink	
A graph link is a directional connection between two graph nodes	67
DungeonArchitect.Editors.GraphLinkRenderer	
Renders the graph link in the graph editor	68
DungeonArchitect.Graphs.GraphNode	
Represents a graph node in the theme graph. This is the base class for all graph nodes	69
DungeonArchitect.Editors.GraphNodeRenderer	
Renders the graph node in the graph editor	72
DungeonArchitect.Editors.GraphNodeRendererFactory	
	73
DungeonArchitect.Editors.GraphOperations	
	73
DungeonArchitect.Graphs.GraphPin	
A pin is used to connect a link to a node	76
DungeonArchitect.Editors.GraphPinHierarchyComparer	
Sorts the pins based on their owning node's type	79
DungeonArchitect.Editors.GraphPinRenderer	
Renders a graph pin hosted inside a node	79
DungeonArchitect.Editors.GraphRendererContext	
The rendering context for drawing the theme editor	80
DungeonArchitect.Graphs.GraphSchema	
The graph schema defines the rules of the theme graph	80
DungeonArchitect.Editors.GraphSelectionBox	
Manages the selection box for selecting multiple objects in the graph editor	81
DungeonArchitect.Editors.GraphTooltip	
Graph tooltip singleton	82
DungeonArchitect.Editors.GraphTooltipRenderer	
Renders a tooltip in the graph editor. The tooltip message is defined in GraphTooltip.message .	82
DungeonArchitect.Utills.GraphUtils	
Theme graph utility functions	82
DungeonArchitect.GridCellInfo	
Contains meta data about the cells. This structure is used for caching cell information for faster lookup during and after generation of the dungeon	83
DungeonArchitect.GridDungeonBuilder	
A Dungeon Builder implementation that builds a grid based dungeon	83
DungeonArchitect.GridDungeonConfig	
The dungeon configuration for the Grid builder	85
DungeonArchitect.Editors.GridDungeonConfigPropertyEditor	
Custom property editor for the grid based dungeon configuration	88
DungeonArchitect.GridDungeonModel	
Data model for the grid based dungeon builder	88
DungeonArchitect.GridDungeonModelUtils	
	91
DungeonArchitect.Graphs.IndexCounter	
An ID provider for graph objects	91
DungeonArchitect.Editors.InspectorUtils	
Utility functions for drawing UI in the Inspector window	92

DungeonArchitect.Utills.InstanceCache	
Caches instances by their name so they can be reused when needed again instead of recreating it	93
DungeonArchitect.IntVector	
Represent an integer vector	94
DungeonArchitect.IntVector2Key	
Data-structure for IntVector pair. Used for caching	94
DungeonArchitect.Editors.KeyboardState	
Caches the keyboard state	95
DungeonArchitect.LandscapeDataRasterizer	
Manages the landscape data and performs various rasterization algorithms (draw cells, lines etc)	95
DungeonArchitect.LandscapeTexture	
Data-structure to hold the texture settings. This contains enough information to paint the texture on to the terrain	98
DungeonArchitect.Editors.LandscapeTextureEditor	
Custom property editor for the Landscape texture data-structure	98
DungeonArchitect.LandscapeTransformerGrid	
The terrain modifier that works with the grid based dungeon builder (DungeonBuilderGrid) It modifies the terrain by adjusting the height around the layout of the dungeon and painting it based on the specified texture settings	99
DungeonArchitect.Navigation.LayoutFloorTriangleProvider	101
DungeonArchitect.LeftToRightNodeComparer	
Sorts the nodes from left to right based on the X-axis. This is used for sorting the visual nodes for execution, since they are executed from left to right	101
DungeonArchitect.MarkerEmitterCornerBeautifier	
Emits markers to beautify the level around corners based on the surrounding tiles	102
DungeonArchitect.MarkerEmitterEmptySpace	
Emits markers in the nearby empty space of the dungeon layout	103
DungeonArchitect.MarkerEmitterFindLowestPoint	
Finds the lowest dungeon point and emits a marker at that position. Also sets the scale of the marker to match the width / height	103
DungeonArchitect.MarkerEmitterFreeSpaceDecorator	
A more specialized version of the EmptySpace emitter. Emits decorative markers in empty space near the layout	104
DungeonArchitect.Graphs.MarkerEmitterNode	105
DungeonArchitect.Editors.MarkerEmitterNodeEditor	
Custom property editors for MarkerEmitterNode	106
DungeonArchitect.Editors.MarkerEmitterNodeRenderer	
Renders a MarkerEmitterNode	106
DungeonArchitect.Graphs.MarkerNode	107
DungeonArchitect.Editors.MarkerNodeEditor	
Custom property editors for MarkerNode	107
DungeonArchitect.Editors.MarkerNodeRenderer	
Renders a marker node	108
DungeonArchitect.Utills.MathUtils	
Various math utility functions	108
DungeonArchitect.Utills.Matrix	
Utility function to extract and put data into a Matrix4x4 object	114
DungeonArchitect.Editors.MeshNodeEditor	
Custom property editors for GameObjectNode	117
DungeonArchitect.Editors.MeshNodeRenderer	
Renders a mesh node	118
DungeonArchitect.Navigation.NavigationBuildInvoker	
Drop this script into your dungeon object and assign the nav mesh prefab to automatically rebuild the nav mesh whenever the dungeon is rebuild (works both with runtime and design time)	119
DungeonArchitect.Navigation.NavigationTriangleProvider	120
DungeonArchitect.NegationVolume	
Negation volumes remove procedural geometries from the scene that lie with it's bounds	120

DungeonArchitect.Editors.NegationVolumeEditor	
Custom property editor for Negation volumes	120
DungeonArchitect.Editors.NodeDeletionOrderComparer	
Sorts based on the node's Z-index in decending order	121
DungeonArchitect.Editors.NodeReversedZIndexComparer	
Sorts based on the node's Z-index in decending order	121
DungeonArchitect.Editors.NodeZIndexComparer	
Sorts based on the node's Z-index	122
DungeonArchitect.Graphs.PlaceableNode	122
DungeonArchitect.Editors.PlaceableNodeEditor	
Custom property editor for placeable node	123
DungeonArchitect.PlatformVolume	
Platform volumes add a platform in the scene encompassing the volume	124
DungeonArchitect.Editors.PlatformVolumeEditor	
Custom property editor for Platform volumes	124
DungeonArchitect.Utils.PMRandom	
A random stream based on normal distribution. Also support uniform distrsribution	125
DungeonArchitect.Triangulator.Geometry.Point	
2D <i>Point</i> with double precision	127
DungeonArchitect.Triangulator.Geometry.Point< T >	
A point with an attribute value of type 'T'	127
DungeonArchitect.PooledDungeonSceneProvider	
Implementation of the Scene provider that adds object pooling over the existing functionality. This is useful for quick rebuilding and better performance, as object in the scene are reused while rebuilding, instead of destroying everything and rebuilding	127
DungeonArchitect.PropChildSocketData	
Props can emit new sockets when they are inserted, to add more child props relative to them	129
DungeonArchitect.PropSocket	
The data structure for a marker	129
DungeonArchitect.PropTypeData	
The data structure to hold information about a single node in the asset file	129
DungeonArchitect.Rectangle	
Represents an integer rectangle	131
DungeonArchitect.SelectorRule	
Selector rule allow you to attach selection behavior to decide if a visual node should be inserted into the scene	132
DungeonArchitect.Graphs.SpriteNode	133
DungeonArchitect.Editors.SpriteNodeEditor	
Custom property editor for a sprite node	135
DungeonArchitect.Editors.SpriteNodeRenderer	
Renders a sprite node	135
DungeonArchitect.SpritePropTypeData	
Sprite node data asset attributes	136
DungeonArchitect.StairAdjacencyQueueNode	
Temporary data-structure used while assigning stairs on the dungeon.	136
DungeonArchitect.StairEdgeInfo	
Data structure to hold the adjacent cells connected to the stairs (entry / exit)	137
DungeonArchitect.StairInfo	
Data-structure to hold the stair information in the grid based builder	137
DungeonArchitect.Navigation.StaticMeshTriangleProvider	138
DungeonArchitect.Navigation.TerrainTriangleProvider	138
DungeonArchitect.ThemeOverrideVolume	
Dungeon layout that lies within this volumes bounds picks up the theme set in this volume	139
DungeonArchitect.Editors.ThemeOverrideVolumeEditor	
Custom property editor for Theme override volumes	139
DungeonArchitect.Editors.Timer	
Ticks every few milli-seconds	140

DungeonArchitect.TransformationRule	
Selector rule allow you to attach selection behavior to decide if a visual node should be inserted into the scene	140
DungeonArchitect.Triangulator.Geometry.Triangle	
Triangle made from three point indexes	141
DungeonArchitect.Graphs.VisualNode	142
DungeonArchitect.Editors.VisualNodeEditor	
Custom property editor for visual nodes	144
DungeonArchitect.Editors.VisualNodeRenderer	
Renders a visual node	145
DungeonArchitect.Volume	
A volume is an abstract representation of space in the world. A volume can be scaled and moved around like any other game object and custom functionality can be added to volumes to influence the dungeon with it's spatial volume	145
DungeonArchitect.Editors.VolumeEditor	
Custom property editor for volumes game objects	147

Chapter 4

Namespace Documentation

4.1 DungeonArchitect Namespace Reference

Namespaces

- namespace [Editors](#)
- namespace [Graphs](#)
- namespace [Navigation](#)
- namespace [Triangulator](#)
- namespace [Utils](#)

Classes

- class [BlurFilter](#)
A fast Gaussian blurring filter applied over a 2D data array
- class [Cell](#)
Data-structure to hold the [Cell](#) information. A cell is a piece of the dungeon layout and can be either a room or a corridor
- class [CellDoor](#)
Data-structure to hold the door information
- class [CellHeightFrameInfo](#)
Temporary data-structure used while assigning heights on the dungeon.
- class [CellHeightNode](#)
Temporary data-structure to hold the height data of the cell node A graph is build of the dungeon layout while the heights are assigned and this node contains the cell's height information
- class [DebugDrawer](#)
Helper functions to draw debug information of the dungeon layout in the scene view
- class [DoorManager](#)
Manages the doors in the grid based builder
- class [Dungeon](#)
The main dungeon behavior that manages the creation and destruction of dungeons
- class [DungeonBuilder](#)
Builds the layout of the dungeon and emits markers around the layout Implement this class to create your own builder
- class [DungeonConfig](#)
Base dungeon configuration. Create your own implementation of this configuration based on your dungeon builder's needs
- class [DungeonEventListener](#)
Listen to various dungeon events during the build and destroy phase

- class [DungeonMarkerEmitter](#)
Marker Emitters let you emit your own markers anywhere in the map. Implement this class and add it to the [Dungeon](#) object to add your own markers right after the dungeon layout is created
- class [DungeonModel](#)
Abstract dungeon model. Create your own implementation of the model depending on your builder's needs
- class [DungeonPaintMode](#)
Manage the editor paint mode so you can paint the layout of you dungeon. You should implement your own paint mode depending on your dungeon builder's data structures and requirements
- class [DungeonPaintModeGrid](#)
Editor tooling for the grid based dungeon builder. Lets you paint with a grid based brush
- class [DungeonPropDataAsset](#)
The data-structure for serializing the theme graph to disk
- class [DungeonSceneProvider](#)
A scene provider instantiates game objects into the scene. Implementations can customize the instantiation process if needed (e.g. object pooling etc)
- class [DungeonSceneProviderData](#)
Meta-data added to each spawned game object in the scene. This is used to identify objects that belong to the dungeons, for later destruction and rebuilding
- class [DungeonToolData](#)
Tool Data represented by the grid based builder
- interface [Filter](#)
A data filter applied over a 2D data array
- class [GameObjectPropTypeData](#)
Game Object node data asset attributes
- class [GridCellInfo](#)
Contains meta data about the cells. This structure is used for caching cell information for faster lookup during and after generation of the dungeon
- class [GridDungeonBuilder](#)
A [Dungeon](#) Builder implementation that builds a grid based dungeon.
- class [GridDungeonConfig](#)
The dungeon configuration for the Grid builder
- class [GridDungeonModel](#)
Data model for the grid based dungeon builder
- class [GridDungeonModelUtils](#)
- struct [IntVector](#)
Represent an integer vector
- struct [IntVector2Key](#)
Data-structure for [IntVector](#) pair. Used for caching
- class [LandscapeDataRasterizer](#)
Manages the landscape data and performs various rasterization algorithms (draw cells, lines etc)
- class [LandscapeTexture](#)
Data-structure to hold the texture settings. This contains enough information to paint the texture on to the terrain
- class [LandscapeTransformerGrid](#)
The terrain modifier that works with the grid based dungeon builder ([DungeonBuilderGrid](#)) It modifies the terrain by adjusting the height around the layout of the dungeon and painting it based on the specified texture settings
- class [LeftToRightNodeComparer](#)
Sorts the nodes from left to right based on the X-axis. This is used for sorting the visual nodes for execution, since they are executed from left to right
- class [MarkerEmitterCornerBeautifier](#)
Emits markers to beautify the level around corners based on the surrounding tiles
- class [MarkerEmitterEmptySpace](#)
Emits markers in the nearby empty space of the dungeon layout

- class [MarkerEmitterFindLowestPoint](#)
Finds the lowest dungeon point and emits a marker at that position. Also sets the scale of the marker to match the width / height
- class [MarkerEmitterFreeSpaceDecorator](#)
A more specialized version of the EmptySpace emitter. Emits decorative markers in empty space near the layout
- class [NegationVolume](#)
Negation volumes remove procedural geometries from the scene that lie with it's bounds
- class [PlatformVolume](#)
Platform volumes add a platform in the scene encompassing the volume
- class [PooledDungeonSceneProvider](#)
Implementation of the Scene provider that adds object pooling over the existing functionality. This is useful for quick rebuilding and better performance, as object in the scene are reused while rebuilding, instead of destroying everything and rebuilding
- class [PropChildSocketData](#)
Props can emit new sockets when they are inserted, to add more child props relative to them
- class [PropSocket](#)
The data structure for a marker
- class [PropTypeData](#)
The data structure to hold information about a single node in the asset file
- struct [Rectangle](#)
Represents an integer rectangle
- class [SelectorRule](#)
Selector rule allow you to attach selection behavior to decide if a visual node should be inserted into the scene
- class [SpritePropTypeData](#)
Sprite node data asset attributes
- class [StairAdjacencyQueueNode](#)
Temporary data-structure used while assigning stairs on the dungeon.
- struct [StairEdgeInfo](#)
Data structure to hold the adjacent cells connected to the stairs (entry / exit)
- class [StairInfo](#)
Data-structure to hold the stair information in the grid based builder
- class [ThemeOverrideVolume](#)
[Dungeon](#) layout that lies within this volumes bounds picks up the theme set in this volume
- class [TransformationRule](#)
Selector rule allow you to attach selection behavior to decide if a visual node should be inserted into the scene
- class [Volume](#)
A volume is an abstract representation of space in the world. A volume can be scaled and moved around like any other game object and custom functionality can be added to volumes to influence the dungeon with it's spatial volume

Typedefs

- using **PropBySocketType_t** = Dictionary< string, List< [PropTypeData](#) >>
- using **PropBySocketTypeByTheme_t** = Dictionary< [DungeonPropDataAsset](#), Dictionary< string, List< [PropTypeData](#) >>>

Enumerations

- enum [LandscapeTextureType](#) { **Fill**, **Room**, **Corridor**, **Cliff** }
The type of the texture defined in the landscape paint settings. This determines how the specified texture would be painted in the modified terrain

- enum [DungeonModelBuildState](#) { **Initial, Separation, Triangulation, SpanningTree, Corridors, Complete** }
The build state used to track the progress
- enum [CellType](#) { **Room, Corridor, CorridorPadding, Unknown** }
The type of cell used in the grid builder

4.1.1 Enumeration Type Documentation

4.1.1.1 enum [DungeonArchitect.CellType](#) [strong]

The type of cell used in the grid builder

4.1.1.2 enum [DungeonArchitect.DungeonModelBuildState](#) [strong]

The build state used to track the progress

4.1.1.3 enum [DungeonArchitect.LandscapeTextureType](#) [strong]

The type of the texture defined in the landscape paint settings. This determines how the specified texture would be painted in the modified terrain

4.2 DungeonArchitect.Editors Namespace Reference

Classes

- class [AssetThumbnailCache](#)
Manages the asset thumbnails to display in the visual nodes
- class [CursorDragLink](#)
Manages a link dragged out of a node with the other end following the mouse cursor
- class [DungeonArchitectGraphEditor](#)
The main editor window for the Theme graph editor. This hosts the graph editor for managing the theme graph
- class [DungeonAssetPostprocessor](#)
An asset processor to create dungeon tags when [DungeonArchitect](#) is imported into a new project
- class [DungeonEditorHelper](#)
Utility functions for various editor based features of [Dungeon Architect](#)
- class [DungeonEditorResources](#)
The resource filename constants used by dungeon architect editor
- class [DungeonObjectTraker](#)
Tracks active dungeon objects in the scene and finds ones that have the active graph being edited This is used for real-time updates on the dungeon object as the graph is modified from the editor
- class [DungeonPaintModeEditor](#)
Custom property editor for the paint mode object
- class [DungeonPaintModeGridEditor](#)
Custom property editor for the Paint model game object
- class [DungeonPropertyEditor](#)
Custom property editor for the dungeon game object
- class [GraphContextMenu](#)
The context menu shown when the user right clicks on the theme graph editor
- class [GraphContextMenuEvent](#)

- The graph context menu event data*

 - class [GraphEditor](#)

The graph editor script for managing a graph. This contains the bulk of the logic for graph editing
 - class [GraphEditorConstants](#)

Graph editor constants
 - class [GraphInputHandler](#)
 - class [GraphInspector](#)

Custom property editor for graph objects Shows the graph editor when a theme graph asset is selected
 - class [GraphLinkRenderer](#)

Renders the graph link in the graph editor
 - class [GraphNodeRenderer](#)

Renders the graph node in the graph editor
 - class [GraphNodeRendererFactory](#)
 - class [GraphOperations](#)
 - class [GraphPinHierarchyComparer](#)

Sorts the pins based on their owning node's type
 - class [GraphPinRenderer](#)

Renders a graph pin hosted inside a node
 - class [GraphRendererContext](#)

The rendering context for drawing the theme editor
 - class [GraphSelectionBox](#)

Manages the selection box for selecting multiple objects in the graph editor
 - class [GraphTooltip](#)

Graph tooltip singleton
 - class [GraphTooltipRenderer](#)

Renders a tooltip in the graph editor. The tooltip message is defined in [GraphTooltip.message](#)
 - class [GridDungeonConfigPropertyEditor](#)

Custom property editor for the grid based dungeon configuration
 - class [InspectorUtils](#)

Utility functions for drawing UI in the Inspector window
 - class [KeyboardState](#)

Caches the keyboard state
 - class [LandscapeTextureEditor](#)

Custom property editor for the Landscape texture data-structure
 - class [MarkerEmitterNodeEditor](#)

Custom property editors for MarkerEmitterNode
 - class [MarkerEmitterNodeRenderer](#)

Renders a MarkerEmitterNode
 - class [MarkerNodeEditor](#)

Custom property editors for MarkerNode
 - class [MarkerNodeRenderer](#)

Renders a marker node
 - class [MeshNodeEditor](#)

Custom property editors for GameObjectNode
 - class [MeshNodeRenderer](#)

Renders a mesh node
 - class [NegationVolumeEditor](#)

Custom property editor for Negation volumes
 - class [NodeDeletionOrderComparer](#)

Sorts based on the node's Z-index in decending order
 - class [NodeReversedZIndexComparer](#)

- Sorts based on the node's Z-index in decending order*
 - class [NodeZIndexComparer](#)
- Sorts based on the node's Z-index*
 - class [PlaceableNodeEditor](#)
- Custom property editor for placeable node*
 - class [PlatformVolumeEditor](#)
- Custom property editor for Platform volumes*
 - class [SpriteNodeEditor](#)
- Custom property editor for a sprite node*
 - class [SpriteNodeRenderer](#)
- Renders a sprite node*
 - class [ThemeOverrideVolumeEditor](#)
- Custom property editor for Theme override volumes*
 - class [Timer](#)
- Ticks every few milli-seconds*
 - class [VisualNodeEditor](#)
- Custom property editor for visual nodes*
 - class [VisualNodeRenderer](#)
- Renders a visual node*
 - class [VolumeEditor](#)
- Custom property editor for volumes game objects*

Enumerations

- enum [GraphMenuAction](#) { **AddGameObjectNode**, **AddSpriteNode**, **AddMarkerNode**, **AddMarkerEmitterNode** }
- The type of menu action to perform*

4.2.1 Enumeration Type Documentation

4.2.1.1 enum **DungeonArchitect.Editors.GraphMenuAction** [strong]

The type of menu action to perform

4.3 DungeonArchitect.Graphs Namespace Reference

Classes

- class [GameObjectNode](#)
- class [Graph](#)
 - Theme [Graph](#) data structure holds all the theme nodes and their connections*
- class [GraphCamera](#)
 - A camera that manages the graph editor's viewport*
- class [GraphLink](#)
 - A graph link is a directional connection between two graph nodes*
- class [GraphNode](#)
 - Represents a graph node in the theme graph. This is the base class for all graph nodes*
- class [GraphPin](#)
 - A pin is used to connect a link to a node*
- class [GraphSchema](#)

The graph schema defines the rules of the theme graph

- class [IndexCounter](#)
- An ID provider for graph objects*
- class [MarkerEmitterNode](#)
- class [MarkerNode](#)
- class [PlaceableNode](#)
- class [SpriteNode](#)
- class [VisualNode](#)

Enumerations

- enum [GraphPinType](#) { **Input**, **Output**, **Unknown** }
- The graph pin type*
- enum [GraphPinMouseState](#) { **Hover**, **Clicked**, **None** }
- The state of the mouse input on a pin*
- enum [DungeonSpriteCollisionType](#) { **None**, **Box**, **Circle**, **Polygon** }

4.3.1 Enumeration Type Documentation

4.3.1.1 enum [DungeonArchitect.Graphs.GraphPinMouseState](#) [strong]

The state of the mouse input on a pin

4.3.1.2 enum [DungeonArchitect.Graphs.GraphPinType](#) [strong]

The graph pin type

4.4 DungeonArchitect.Navigation Namespace Reference

Classes

- class [Collision2DTriangleProvider](#)
- class [CollisionTriangleProvider](#)
- class [DungeonNavAgent](#)
- class [DungeonNavAgent2D](#)
- class [DungeonNavAgent3D](#)
- class [DungeonNavMesh](#)
- class [LayoutFloorTriangleProvider](#)
- class [NavigationBuildInvoker](#)
- Drop this script into your dungeon object and assign the nav mesh prefab to automatically rebuild the nav mesh whenever the dungeon is rebuild (works both with runtime and design time)*
- class [NavigationTriangleProvider](#)
- class [StaticMeshTriangleProvider](#)
- class [TerrainTriangleProvider](#)

4.5 DungeonArchitect.Triangulator Namespace Reference

Namespaces

- namespace [Geometry](#)

Classes

- class [Delauney](#)
Performs the [Delauney](#) triangulation on a set of vertices.

4.6 DungeonArchitect.Triangulator.Geometry Namespace Reference

Classes

- class [Edge](#)
[Edge](#) made from two point indexes
- class [Point](#)
2D [Point](#) with double precision
- struct [Triangle](#)
[Triangle](#) made from three point indexes

4.7 DungeonArchitect.Utils Namespace Reference

Classes

- class [Blackboard](#)
A blackboard holds global data that can be shared across multiple scripts
- class [BlackboardDatabase](#)
- class [DungeonConstants](#)
Contains various project specific constants
- class [GraphUtils](#)
Theme graph utility functions
- class [InstanceCache](#)
Caches instances by their name so they can be reused when needed again instead of recreating it
- class [MathUtils](#)
Various math utility functions
- class [Matrix](#)
Utility function to extract and put data into a Matrix4x4 object
- class [PMRandom](#)
A random stream based on normal distribution. Also support uniform distribution

Chapter 5

Class Documentation

5.1 DungeonArchitect.Editor.AssetThumbnailCache Class Reference

Manages the asset thumbnails to display in the visual nodes

Public Member Functions

- void [Reset](#) ()
Clears all the thumbnail from the cache
- Texture2D [GetThumb](#) (Object asset)
Gets the thumbnail of the specified asset. Tries to retrieve it from the cache, if it was accessed earlier
- void **Update** ()

Properties

- static [AssetThumbnailCache Instance](#) [get]
Singleton access

5.1.1 Detailed Description

Manages the asset thumbnails to display in the visual nodes

5.1.2 Member Function Documentation

5.1.2.1 Texture2D DungeonArchitect.Editor.AssetThumbnailCache.GetThumb (Object asset)

Gets the thumbnail of the specified asset. Tries to retrieve it from the cache, if it was accessed earlier

Parameters

<i>asset</i>	The asset to get the thumbnail for
--------------	------------------------------------

Returns

The thumbnail of the asset. If thumbnail cannot be created, returns the defaultTexture instead

5.1.2.2 void DungeonArchitect.Editors.AssetThumbnailCache.Reset ()

Clears all the thumbnail from the cache

5.1.3 Property Documentation

5.1.3.1 AssetThumbnailCache DungeonArchitect.Editors.AssetThumbnailCache.Instance [static], [get]

Singleton access

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

- [github/dungeon-architect-unity/Editor/Utils/AssetThumbnailCache.cs](#)

5.2 DungeonArchitect.Utils.Blackboard Class Reference

A blackboard holds global data that can be shared across multiple scripts

Properties

- [BlackboardDatabase](#)< int > **IntEntries** [get]
- [BlackboardDatabase](#)< float > **FloatEntries** [get]
- [BlackboardDatabase](#)< string > **StringEntries** [get]
- [BlackboardDatabase](#)< Vector3 > **VectorEntries** [get]
- [BlackboardDatabase](#)< IntVector > **IntVectorEntries** [get]

5.2.1 Detailed Description

A blackboard holds global data that can be shared across multiple scripts

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

- [github/dungeon-architect-unity/Scripts/Utils/Blackboard.cs](#)

5.3 DungeonArchitect.Utils.BlackboardDatabase< T > Class Template Reference

Public Member Functions

- **BlackboardDatabase** (T defaultValue)
- void **SetValue** (string key, T value)
- T **GetValue** (string key)

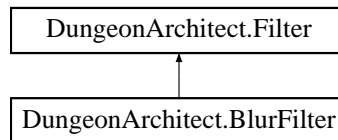
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Utils/Blackboard.cs](#)

5.4 DungeonArchitect.BlurFilter Class Reference

A fast Gaussian blurring filter applied over a 2D data array

Inheritance diagram for DungeonArchitect.BlurFilter:



Public Member Functions

- **BlurFilter** (int radius)
- float[,] **ApplyFilter** (float[,] data)

5.4.1 Detailed Description

A fast Gaussian blurring filter applied over a 2D data array

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Landscape/Filter/BlurFilter.cs](#)

5.5 DungeonArchitect.Cell Class Reference

Data-structure to hold the [Cell](#) information. A cell is a piece of the dungeon layout and can be either a room or a corridor

Public Member Functions

- **Cell** (int x, int z, int width, int length)
- override bool **Equals** (System.Object obj)
- override int **GetHashCode** ()

Properties

- int **Id** [get, set]
- [Rectangle](#) **Bounds** [get, set]
- [CellType](#) **CellType** [get, set]
- bool **UserDefined** [get, set]
- HashSet< int > **ConnectedRooms** [get, set]
- HashSet< int > **FixedRoomConnections** [get, set]
- HashSet< int > **AdjacentCells** [get, set]
- [IntVector](#) **Center** [get]
- Vector3 **CenterF** [get]

5.5.1 Detailed Description

Data-structure to hold the [Cell](#) information. A cell is a piece of the dungeon layout and can be either a room or a corridor

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Models/GridDungeonModel.cs](#)

5.6 DungeonArchitect.CellDoor Class Reference

Data-structure to hold the door information

Public Member Functions

- override string **ToString** ()

Properties

- [IntVector](#)[] **AdjacentTiles** [get]
The adjacent tile positions shared by this door (entry / exit tiles)
- int[] **AdjacentCells** [get, set]

5.6.1 Detailed Description

Data-structure to hold the door information

5.6.2 Property Documentation

5.6.2.1 [IntVector](#) [] **DungeonArchitect.CellDoor.AdjacentTiles** [get]

The adjacent tile positions shared by this door (entry / exit tiles)

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Models/GridDungeonModel.cs](#)

5.7 DungeonArchitect.CellHeightFrameInfo Class Reference

Temporary data-structure used while assigning heights on the dungeon.

Public Member Functions

- **CellHeightFrameInfo** (int pCellId, int pCurrentHeight)

Public Attributes

- int **CellId**
- int **CurrentHeight**

5.7.1 Detailed Description

Temporary data-structure used while assigning heights on the dungeon.

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Builders/GridDungeonBuilder.cs](#)

5.8 DungeonArchitect.CellHeightNode Class Reference

Temporary data-structure to hold the height data of the cell node A graph is build of the dungeon layout while the heights are assigned and this node contains the cell's height information

Public Attributes

- int **CellId**
- int **Height**
- bool **MarkForIncrease**
- bool **MarkForDecrease**

5.8.1 Detailed Description

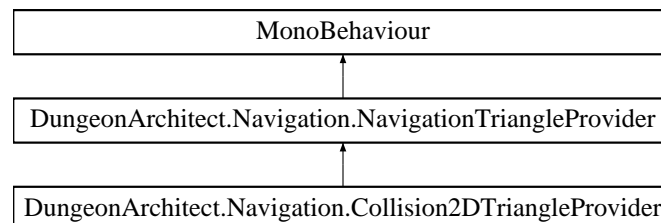
Temporary data-structure to hold the height data of the cell node A graph is build of the dungeon layout while the heights are assigned and this node contains the cell's height information

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Builders/GridDungeonBuilder.cs](#)

5.9 DungeonArchitect.Navigation.Collision2DTriangleProvider Class Reference

Inheritance diagram for DungeonArchitect.Navigation.Collision2DTriangleProvider:



Public Member Functions

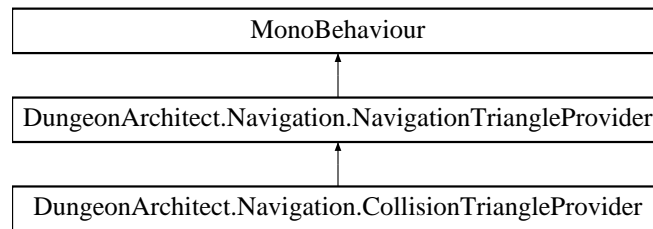
- override void **AddNavTriangles** (List< Triangle3 > triangles)

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

- [github/dungeon-architect-unity/Scripts/Navigation/TriangleProviders/Collision2DTriangleProvider.cs](#)

5.10 DungeonArchitect.Navigation.CollisionTriangleProvider Class Reference

Inheritance diagram for DungeonArchitect.Navigation.CollisionTriangleProvider:



Public Member Functions

- override void **AddNavTriangles** (List< Triangle3 > triangles)

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

- [github/dungeon-architect-unity/Scripts/Navigation/TriangleProviders/CollisionTriangleProvider.cs](#)

5.11 DungeonArchitect.Editors.CursorDragLink Class Reference

Manages a link dragged out of a node with the other end following the mouse cursor

Public Member Functions

- delegate void **OnDraggedLinkReleased** (Vector2 mousePositionScreen)
- **CursorDragLink** ([GraphEditor](#) graphEditor)
- void **Destroy** ()
- void **Activate** ([GraphPin](#) fromPin)
- void **Deactivate** ()
- void **Draw** ([GraphRendererContext](#) rendererContext, [GraphCamera](#) camera)
- void **HandleInput** (Event e)

Properties

- [GraphPin](#) **AttachedPin** [get]

Events

- OnDraggedLinkReleased **DraggedLinkReleased**

5.11.1 Detailed Description

Manages a link dragged out of a node with the other end following the mouse cursor

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

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphEditor.cs](#)

5.12 DungeonArchitect.DebugDrawer Class Reference

Helper functions to draw debug information of the dungeon layout in the scene view

Static Public Member Functions

- static void **DrawCell** ([Cell](#) cell, [Color](#) color, [Vector3](#) gridScale)
- static void **DrawBounds** ([Rectangle](#) bounds, [Color](#) color, [Vector3](#) gridScale)
- static void **DrawCellId** ([Cell](#) cell, [Vector3](#) gridScale)
- static void **DrawMarker** ([PropSocket](#) marker, [Color](#) color)
- static void **DrawAdjacentCells** ([Cell](#) cell, [GridDungeonModel](#) model, [Color](#) color)

5.12.1 Detailed Description

Helper functions to draw debug information of the dungeon layout in the scene view

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Dungeon.cs](#)

5.13 DungeonArchitect.Triangulator.Delauney Class Reference

Performs the [Delauney](#) triangulation on a set of vertices.

Static Public Member Functions

- static List< [Geometry.Triangle](#) > **Triangulate** (List< [Triangulator.Geometry.Point](#) > Vertex)
Performs [Delauney](#) triangulation on a set of points.

5.13.1 Detailed Description

Performs the [Delauney](#) triangulation on a set of vertices.

Based on Paul Bourke's "An Algorithm for Interpolating Irregularly-Spaced Data with Applications in Terrain Modelling" <http://astronomy.swin.edu.au/~pbourke/modelling/triangulate/>

5.13.2 Member Function Documentation

5.13.2.1 static List<[Geometry.Triangle](#)> **DungeonArchitect.Triangulator.Delauney.Triangulate** (List< [Triangulator.Geometry.Point](#) > Vertex) [static]

Performs [Delauney](#) triangulation on a set of points.

The triangulation doesn't support multiple points with the same planar location. Vertex-lists with duplicate points may result in strange triangulation with intersecting edges. To avoid adding multiple points to your vertex-list you can use the following anonymous predicate method:

```
if (!Vertices.Exists(delegate(Triangulator.Geometry.Point p) { return pNew.Equals2D(p); }))
    Vertices.Add(pNew);
```

The triangulation algorithm may be described in pseudo-code as follows:

```
subroutine Triangulate
input : vertex list
output : triangle list
    initialize the triangle list
    determine the supertriangle
    add supertriangle vertices to the end of the vertex list
    add the supertriangle to the triangle list
    for each sample point in the vertex list
        initialize the edge buffer
```

```

for each triangle currently in the triangle list
    calculate the triangle circumcircle center and radius
    if the point lies in the triangle circumcircle then
        add the three triangle edges to the edge buffer
        remove the triangle from the triangle list
    endif
endfor
delete all doubly specified edges from the edge buffer
this leaves the edges of the enclosing polygon only
add to the triangle list all triangles formed between the point
and the edges of the enclosing polygon
endfor
remove any triangles from the triangle list that use the supertriangle vertices
remove the supertriangle vertices from the vertex list
end

```

Parameters

<i>Vertex</i>	List of vertices to triangulate.
---------------	----------------------------------

Returns

Triangles referencing vertex indices arranged in clockwise order

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

- [github/dungeon-architect-unity/Scripts/Triangulator/Delauney.cs](#)

5.14 DungeonArchitect.DoorManager Class Reference

Manages the doors in the grid based builder

Public Member Functions

- void **Clear** ()
- [CellDoor](#) **CreateDoor** ([IntVector](#) p1, [IntVector](#) p2, int cellId1, int cellId2)
Creates a door between the two grid points
- bool [ContainsDoorBetweenCells](#) (int cellA, int cellB)
Check if a door exists between the two cells
- bool **ContainsDoor** (int x1, int z1, int x2, int z2)

Properties

- [CellDoor](#)[] [Doors](#) [get]
List of registered doors

5.14.1 Detailed Description

Manages the doors in the grid based builder

5.14.2 Member Function Documentation

5.14.2.1 bool DungeonArchitect.DoorManager.ContainsDoorBetweenCells (int cellA, int cellB)

Check if a door exists between the two cells

Parameters

<i>cellA</i>	Cell Id of the first cell
<i>cellB</i>	Cell Id of the second cell

Returns

5.14.2.2 **CellDoor** `DungeonArchitect.DoorManager.CreateDoor (IntVector p1, IntVector p2, int cellId1, int cellId2)`

Creates a door between the two grid points

Parameters

<i>p1</i>	The grid poition 1
<i>p2</i>	The grid poition 2
<i>cellId1</i>	Cell Id of the first adjacent cell
<i>cellId2</i>	Cell Id of the second adjacent cell

Returns

5.14.3 Property Documentation

5.14.3.1 **CellDoor** [] `DungeonArchitect.DoorManager.Doors` [get]

List of registered doors

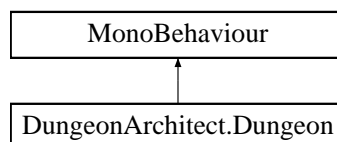
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/Models/GridDungeonModel.cs](#)

5.15 **DungeonArchitect.Dungeon Class Reference**

The main dungeon behavior that manages the creation and destruction of dungeons

Inheritance diagram for `DungeonArchitect.Dungeon`:



Public Member Functions

- void [Build](#) ()
Builds the complete dungeon (layout and visual phase)
- void [ReapplyTheme](#) ()
Runs the theming engine over the existing layout to rebuild the game objects from the theme file. The layout is not built in this stage
- void [DestroyDungeon](#) ()

- Destroys the dungeon*
- void [RequestRebuild](#) ()
- Requests the dungeon to be rebuilt in the next update phase*
- void [AddPaintCell](#) ([IntVector](#) location, bool automaticRebuild)
- Registers a painted cell*
- void [RemovePaintCell](#) ([IntVector](#) location, bool automaticRebuild)
- Remove a previous painted cell*
- void [ClearToolOverlayData](#) (bool automaticRebuild)
- Clears all overlay data*

Public Attributes

- bool **debugDraw** = false
- List< [Graph](#) > [dungeonThemes](#)
- List of themes assigned to this dungeon*

Properties

- [DungeonModel ActiveModel](#) [get]
- Active model used by the dungeon*
- bool [IsLayoutBuilt](#) [get]
- Flag to check if the layout has been built. This is used to quickly reapply the theme after the theme graph has been modified, without rebuilding the layout, if it has already been built*
- [DungeonConfig Config](#) [get]

5.15.1 Detailed Description

The main dungeon behavior that manages the creation and destruction of dungeons

5.15.2 Member Function Documentation

5.15.2.1 void [DungeonArchitect.Dungeon.AddPaintCell](#) ([IntVector](#) location, bool automaticRebuild)

Registers a painted cell

Parameters

<i>location</i>	the location of the painted cell, in grid coordinates
<i>automatic↔ Rebuild</i>	if true, the dungeon would be rebuilt, if the data model has changed due to this request

5.15.2.2 void [DungeonArchitect.Dungeon.Build](#) ()

Builds the complete dungeon (layout and visual phase)

5.15.2.3 void [DungeonArchitect.Dungeon.ClearToolOverlayData](#) (bool automaticRebuild)

Clears all overlay data

Parameters

<i>automatic↔ Rebuild</i>	
-------------------------------	--

5.15.2.4 void DungeonArchitect.Dungeon.DestroyDungeon ()

Destroys the dungeon

5.15.2.5 void DungeonArchitect.Dungeon.ReapplyTheme ()

Runs the theming engine over the existing layout to rebuild the game objects from the theme file. The layout is not built in this stage

5.15.2.6 void DungeonArchitect.Dungeon.RemovePaintCell (IntVector *location*, bool *automaticRebuild*)

Remove a previous painted cell

Parameters

<i>location</i>	the location of the painted cell to remove, in grid coordinates
<i>automatic↔ Rebuild</i>	if true, the dungeon would be rebuilt, if the data model has changed due to this request

5.15.2.7 void DungeonArchitect.Dungeon.RequestRebuild ()

Requests the dungeon to be rebuilt in the next update phase

5.15.3 Member Data Documentation

5.15.3.1 List<Graph> DungeonArchitect.Dungeon.dungeonThemes

List of themes assigned to this dungeon

5.15.4 Property Documentation

5.15.4.1 DungeonModel DungeonArchitect.Dungeon.ActiveModel [get]

Active model used by the dungeon

5.15.4.2 bool DungeonArchitect.Dungeon.IsLayoutBuilt [get]

Flag to check if the layout has been built. This is used to quickly reapply the theme after the theme graph has been modified, without rebuilding the layout, if it has already been built

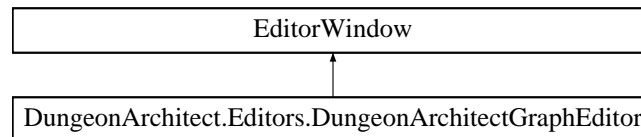
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/Dungeon.cs](#)

5.16 DungeonArchitect.Editor.DungeonArchitectGraphEditor Class Reference

The main editor window for the Theme graph editor. This hosts the graph editor for managing the theme graph

Inheritance diagram for DungeonArchitect.Editor.DungeonArchitectGraphEditor:



Public Member Functions

- void **Init** ([Graph](#) graph)

Properties

- [GraphEditor](#) **GraphEditor** [get]

5.16.1 Detailed Description

The main editor window for the Theme graph editor. This hosts the graph editor for managing the theme graph

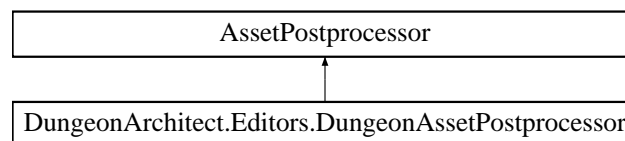
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/Dungeon/DungeonArchitectGraphEditor.cs](#)

5.17 DungeonArchitect.Editor.DungeonAssetPostprocessor Class Reference

An asset processor to create dungeon tags when [DungeonArchitect](#) is imported into a new project

Inheritance diagram for DungeonArchitect.Editor.DungeonAssetPostprocessor:



5.17.1 Detailed Description

An asset processor to create dungeon tags when [DungeonArchitect](#) is imported into a new project

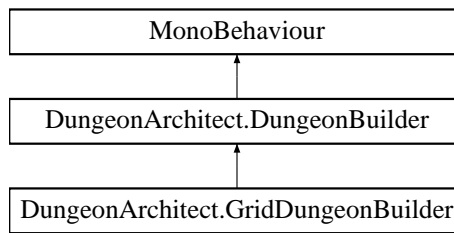
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/Dungeon/DungeonAssetPostprocessor.cs](#)

5.18 DungeonArchitect.DungeonBuilder Class Reference

Builds the layout of the dungeon and emits markers around the layout Implement this class to create your own builder

Inheritance diagram for DungeonArchitect.DungeonBuilder:



Public Member Functions

- virtual void [BuildDungeon](#) ([DungeonConfig](#) config, [DungeonModel](#) model)
Builds the dungeon layout
- virtual void **OnDestroyed** ()
- virtual void [EmitMarkers](#) ()
Emit markers defined by this builder
- void [EmitCustomMarkers](#) ()
Emit markers defined by the user (implementation of [DungeonMarkerEmitter](#))
- void **EmitMarker** (string SocketType, Matrix4x4 transform, [IntVector](#) gridPosition, int cellId)
- virtual void **ApplyTheme** (List< [DungeonPropDataAsset](#) > Themes, [DungeonSceneProvider](#) SceneProvider)

Protected Member Functions

- void **ClearSockets** ()
- void **EmitMarker** (string SocketType, Matrix4x4 _transform, int count, Vector3 InterOffset, [IntVector](#) gridPosition, int cellId)
- void **EmitMarker** (List< [PropSocket](#) > pPropSockets, string SocketType, Matrix4x4 transform, [IntVector](#) gridPosition, int cellId)
- void **CreatePropLookup** ([DungeonPropDataAsset](#) PropAsset, PropBySocketTypeByTheme_t PropBySocketTypeByTheme)
- [DungeonPropDataAsset](#) **GetBestMatchedTheme** (List< [DungeonPropDataAsset](#) > Themes, [PropSocket](#) socket, PropBySocketTypeByTheme_t PropBySocketTypeByTheme)

Protected Attributes

- [DungeonConfig](#) config
- [PMRandom](#) nrandom
- [PMRandom](#) random
- [DungeonModel](#) model
- Vector3 **GridToMeshScale**
- List< [PropSocket](#) > **PropSockets** = new List<[PropSocket](#)>()
- int **_SocketIdCounter** = 0
- [Blackboard](#) **blackboard** = new [Blackboard](#)()

Properties

- bool **IsLayoutBuilt** [get]
- [DungeonArchitect.DungeonModel](#) **Model** [get]
- [Blackboard](#) **Blackboard** [get]

5.18.1 Detailed Description

Builds the layout of the dungeon and emits markers around the layout Implement this class to create your own builder

5.18.2 Member Function Documentation

5.18.2.1 `virtual void DungeonArchitect.DungeonBuilder.BuildDungeon (DungeonConfig config, DungeonModel model)`
[virtual]

Builds the dungeon layout

Parameters

<i>config</i>	The builder configuration
<i>model</i>	The dungeon model that the builder will populate

Reimplemented in [DungeonArchitect.GridDungeonBuilder](#).

5.18.2.2 `void DungeonArchitect.DungeonBuilder.EmitCustomMarkers ()`

Emit markers defined by the user (implementation of [DungeonMarkerEmitter](#))

5.18.2.3 `virtual void DungeonArchitect.DungeonBuilder.EmitMarkers ()` [virtual]

Emit markers defined by this builder

Reimplemented in [DungeonArchitect.GridDungeonBuilder](#).

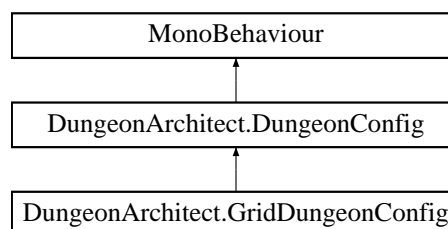
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/DungeonBuilder.cs](#)

5.19 DungeonArchitect.DungeonConfig Class Reference

Base dungeon configuration. Create your own implementation of this configuration based on your dungeon builder's needs

Inheritance diagram for [DungeonArchitect.DungeonConfig](#):



Public Attributes

- `uint Seed = 0`
- `Vector3 GridCellSize = new Vector3(1, 1, 1)`

5.19.1 Detailed Description

Base dungeon configuration. Create your own implementation of this configuration based on your dungeon builder's needs

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

- [github/dungeon-architect-unity/Scripts/Dungeon/DungeonConfig.cs](#)

5.20 DungeonArchitect.Utills.DungeonConstants Class Reference

Contains various project specific constants

Static Public Attributes

- static readonly string **ST_FENCE** = "Fence"
- static readonly string **ST_FENCESEPARATOR** = "FenceSeparator"
- static readonly string **ST_DOOR** = "Door"
- static readonly string **ST_WALL** = "Wall"
- static readonly string **ST_WALLSEPARATOR** = "WallSeparator"
- static readonly string **ST_GROUND** = "Ground"
- static readonly string **ST_STAIR** = "Stair"
- static readonly string **ST_STAIR2X** = "Stair2X"
- static readonly string **ST_WALLHALF** = "WallHalf"
- static readonly string **ST_WALLHALFSEPARATOR** = "WallHalfSeparator"
- static readonly string **ST_NONE** = "None"
- static readonly string **ST_ROOMWALL** = "RoomWall"
- static readonly string **ST_ROOMWALLSEPARATOR** = "RoomWallSeparator"
- static readonly string **ST_ROOMOPENSOURCE** = "RoomOpenSpace"
- static readonly string **ST_LIGHT** = "Light"

5.20.1 Detailed Description

Contains various project specific constants

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

- [github/dungeon-architect-unity/Scripts/Utills/DungeonConstants.cs](#)

5.21 DungeonArchitect.Editors.DungeonEditorHelper Class Reference

Utility functions for various editor based features of [Dungeon Architect](#)

Static Public Member Functions

- static bool [OnOpenAsset](#) (int instanceID, int line)
Handle opening of theme graphs. When the user right clicks on the theme graph and selects open, the graph is shown in the theme editor
- static void [ShowEditor](#) ([Graph](#) graph)
Shows the dungeon theme editor window and loads the specified graph into it
- static string [MakeFilenameUnique](#) (string dir, string filename)

- Creates a unique filename in the specified asset directory*
- static void [AddToAsset](#) ([Graph](#) graph, [GraphNode](#) node)
Adds the node to the graph asset so it can be serialized to disk
- static void [AddToAsset](#) ([Graph](#) graph, [GraphLink](#) link)
Adds the node to the graph asset so it can be serialized to disk
- static void [MarkAsDirty](#) ([Graph](#) graph)
Marks the graph as dirty so that it is serialized to disk again when saved
- static void [CreateEditorTag](#) (string tag)
Creates an editor tag
- static void [_Advanced_RecreateGraphNodeIds](#) ()

5.21.1 Detailed Description

Utility functions for various editor based features of [Dungeon Architect](#)

5.21.2 Member Function Documentation

5.21.2.1 `static void DungeonArchitect.Editors.DungeonEditorHelper.AddToAsset (Graph graph, GraphNode node)`
[static]

Adds the node to the graph asset so it can be serialized to disk

Parameters

<i>graph</i>	The owning graph
<i>node</i>	The node to add to the graph

5.21.2.2 `static void DungeonArchitect.Editors.DungeonEditorHelper.AddToAsset (Graph graph, GraphLink link)`
[static]

Adds the node to the graph asset so it can be serialized to disk

Parameters

<i>graph</i>	The owning graph
<i>link</i>	The link to add to the graph

5.21.2.3 `static void DungeonArchitect.Editors.DungeonEditorHelper.CreateEditorTag (string tag)` [static]

Creates an editor tag

Parameters

<i>tag</i>	
------------	--

5.21.2.4 `static string DungeonArchitect.Editors.DungeonEditorHelper.MakeFilenameUnique (string dir, string filename)`
[static]

Creates a unique filename in the specified asset directory

Parameters

<i>dir</i>	The target directory this file will be placed in. Used for finding non-colliding filenames
<i>filename</i>	The preferred filename. Will add incremental numbers to it till it finds a free filename

Returns

A filename not currently used in the specified directory

5.21.2.5 static void DungeonArchitect.Editors.DungeonEditorHelper.MarkAsDirty (Graph *graph*) [static]

Marks the graph as dirty so that it is serialized to disk again when saved

Parameters

<i>graph</i>	
--------------	--

5.21.2.6 static bool DungeonArchitect.Editors.DungeonEditorHelper.OnOpenAsset (int *instanceID*, int *line*) [static]

Handle opening of theme graphs. When the user right clicks on the theme graph and selects open, the graph is shown in the theme editor

Parameters

<i>instanceID</i>	
<i>line</i>	

Returns

true if trying to open a dungeon theme, indicating that it has been handled. false otherwise

5.21.2.7 static void DungeonArchitect.Editors.DungeonEditorHelper.ShowEditor (Graph *graph*) [static]

Shows the dungeon theme editor window and loads the specified graph into it

Parameters

<i>graph</i>	The graph to load in the dungeon theme editor window
--------------	--

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

- [github/dungeon-architect-unity/Editor/Utils/DungeonEditorHelper.cs](#)

5.22 DungeonArchitect.Editors.DungeonEditorResources Class Reference

The resource filename constants used by dungeon architect editor

Public Member Functions

- T [GetResource< T >](#) (string path)

Loads and retrieves the resource of the specified type

Static Public Attributes

- static readonly string **TEXTURE_GO_NODE_SELECTION** = "graph_node_go_selection"
- static readonly string **TEXTURE_GO_NODE_FRAME** = "graph_node_go_frame"
- static readonly string **TEXTURE_GO_NODE_BG** = "graph_node_go_bg"
- static readonly string **TEXTURE_PIN_GLOW** = "graph_pin_glow"
- static readonly string **TEXTURE_MARKER_NODE_SELECTION** = "graph_node_marker_selection"
- static readonly string **TEXTURE_MARKER_NODE_FRAME** = "graph_node_marker_frame"
- static readonly string **TEXTURE_MARKER_EMITTER_NODE_FRAME** = "graph_node_marker_emitter_↔frame"
- static readonly string **TEXTURE_MARKER_NODE_BG** = "graph_node_marker_bg"
- static readonly string **GUI_STYLE_BANNER** = "DABannerStyle"

5.22.1 Detailed Description

The resource filename constants used by dungeon architect editor

5.22.2 Member Function Documentation

5.22.2.1 T DungeonArchitect.Editors.DungeonEditorResources.GetResource< T > (string path)

Loads and retrieves the resource of the specified type

Template Parameters

<i>T</i>	The type of the resource (e.g. Texture2D)
----------	---

Parameters

<i>path</i>	The path to load the resource from. Usually specified from the constants defined in this class
-------------	--

Returns

The loaded resource

Type Constraints

T : Object

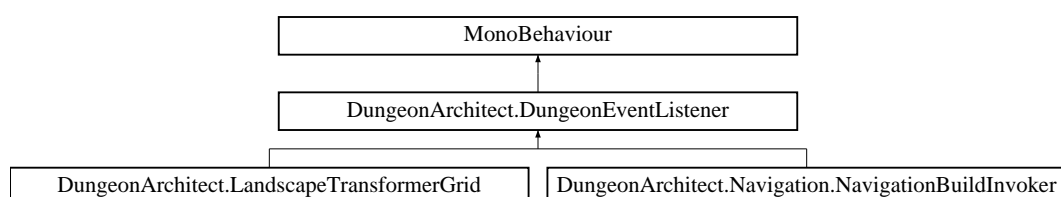
The documentation for this class was generated from the following file:

- github/dungeon-architect-unity/Editor/Utils/DungeonEditorResources.cs

5.23 DungeonArchitect.DungeonEventListener Class Reference

Listen to various dungeon events during the build and destroy phase

Inheritance diagram for DungeonArchitect.DungeonEventListener:



Public Member Functions

- virtual void [OnPostDungeonLayoutBuild](#) ([Dungeon](#) dungeon, [DungeonModel](#) model)
Called after the layout is built in memory, but before the markers are emitted
- virtual void [OnPostDungeonBuild](#) ([Dungeon](#) dungeon, [DungeonModel](#) model)
Called after the dungeon is completely built
- virtual void [OnDungeonDestroyed](#) ([Dungeon](#) dungeon)
Called after the dungeon is destroyed

5.23.1 Detailed Description

Listen to various dungeon events during the build and destroy phase

5.23.2 Member Function Documentation

5.23.2.1 virtual void [DungeonArchitect.DungeonEventListener.OnDungeonDestroyed](#) ([Dungeon](#) *dungeon*) [virtual]

Called after the dungeon is destroyed

Parameters

<i>model</i>	The dungeon model
--------------	-------------------

5.23.2.2 virtual void [DungeonArchitect.DungeonEventListener.OnPostDungeonBuild](#) ([Dungeon](#) *dungeon*, [DungeonModel](#) *model*) [virtual]

Called after the dungeon is completely built

Parameters

<i>model</i>	The dungeon model
--------------	-------------------

Reimplemented in [DungeonArchitect.Navigation.NavigationBuildInvoker](#).

5.23.2.3 virtual void [DungeonArchitect.DungeonEventListener.OnPostDungeonLayoutBuild](#) ([Dungeon](#) *dungeon*, [DungeonModel](#) *model*) [virtual]

Called after the layout is built in memory, but before the markers are emitted

Parameters

<i>model</i>	The dungeon model
--------------	-------------------

Reimplemented in [DungeonArchitect.LandscapeTransformerGrid](#).

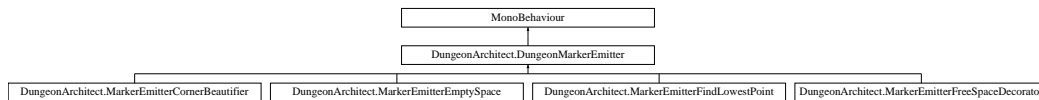
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/DungeonEventListener.cs](#)

5.24 DungeonArchitect.DungeonMarkerEmitter Class Reference

Marker Emitters let you emit your own markers anywhere in the map. Implement this class and add it to the [Dungeon](#) object to add your own markers right after the dungeon layout is created

Inheritance diagram for [DungeonArchitect.DungeonMarkerEmitter](#):



Public Member Functions

- virtual void [EmitMarkers](#) ([DungeonBuilder](#) builder)
Called by the dungeon object right after the dungeon is created

5.24.1 Detailed Description

Marker Emitters let you emit your own markers anywhere in the map. Implement this class and add it to the [Dungeon](#) object to add your own markers right after the dungeon layout is created

5.24.2 Member Function Documentation

5.24.2.1 virtual void DungeonArchitect.DungeonMarkerEmitter.EmitMarkers ([DungeonBuilder](#) *builder*) [virtual]

Called by the dungeon object right after the dungeon is created

Parameters

<i>builder</i>	reference to the builder object used to build the dungeon
----------------	---

Reimplemented in [DungeonArchitect.MarkerEmitterEmptySpace](#), [DungeonArchitect.MarkerEmitterFreeSpaceDecorator](#), [DungeonArchitect.MarkerEmitterFindLowestPoint](#), and [DungeonArchitect.MarkerEmitterCornerBeautifier](#).

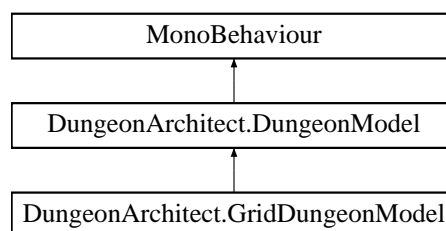
The documentation for this class was generated from the following file:

- github/dungeon-architect-unity/Scripts/Dungeon/DungeonMarkerEmitter.cs

5.25 DungeonArchitect.DungeonModel Class Reference

Abstract dungeon model. Create your own implementation of the model depending on your builder's needs

Inheritance diagram for DungeonArchitect.DungeonModel:



Public Member Functions

- virtual void **ResetModel** ()

Public Attributes

- [DungeonToolData](#) **ToolData** = new [DungeonToolData](#)()

5.25.1 Detailed Description

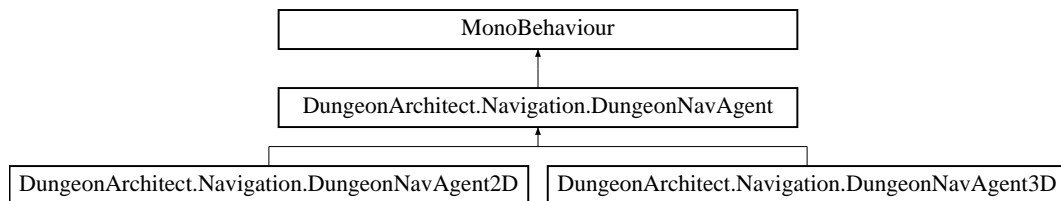
Abstract dungeon model. Create your own implementation of the model depending on your builder's needs

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

- [github/dungeon-architect-unity/Scripts/Dungeon/DungeonModel.cs](#)

5.26 DungeonArchitect.Navigation.DungeonNavAgent Class Reference

Inheritance diagram for DungeonArchitect.Navigation.DungeonNavAgent:



Public Member Functions

- abstract void **Resume** ()
- abstract void **Stop** ()
- abstract float **GetRemainingDistance** ()

Properties

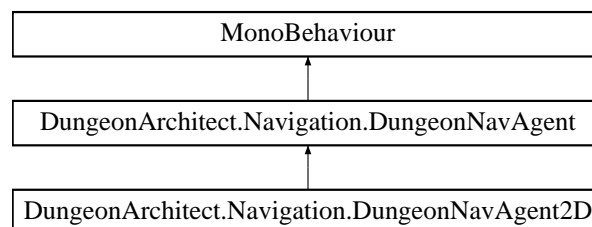
- abstract Vector3 **Destination** [get, set]
- abstract Vector3 **Velocity** [get, set]
- abstract Vector3 **Direction** [get]

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

- [github/dungeon-architect-unity/Scripts/Navigation/DungeonNavAgent.cs](#)

5.27 DungeonArchitect.Navigation.DungeonNavAgent2D Class Reference

Inheritance diagram for DungeonArchitect.Navigation.DungeonNavAgent2D:



Public Member Functions

- override void **Stop** ()
- override void **Resume** ()
- override float **GetRemainingDistance** ()

Static Public Member Functions

- static SharpNav.Geometry.Vector3 **ToSV3** (Vector3 v)
- static Vector3 **ToV3** (SharpNav.Geometry.Vector3 v)

Public Attributes

- float **radius** = 0.5f
- float **height** = 1f
- float **maxAcceleration** = 8
- float **maxSpeed** = 3f
- float **collisionQueryRange** = 4
- float **pathOptimizationRange** = 15
- float **separationWeight** = 3
- Vector2 **navAgentCollisionOffset**
- float **updateFrequency** = 2

Properties

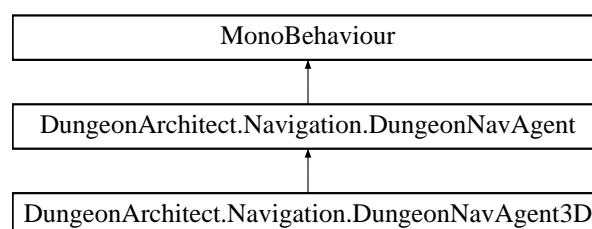
- [DungeonNavMesh](#) **NavMesh** [get]
- override Vector3 **Destination** [get, set]
- override Vector3 **Velocity** [get, set]
- override Vector3 **Direction** [get]
- float **DesiredSpeed** [get]

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

- github/dungeon-architect-unity/Scripts/Navigation/DungeonNavAgent2D.cs

5.28 DungeonArchitect.Navigation.DungeonNavAgent3D Class Reference

Inheritance diagram for DungeonArchitect.Navigation.DungeonNavAgent3D:



Public Member Functions

- override void **Stop** ()
- override void **Resume** ()
- override float **GetRemainingDistance** ()

Static Public Member Functions

- static SharpNav.Geometry.Vector3 **ToSV3** (Vector3 v)
- static Vector3 **ToV3** (SharpNav.Geometry.Vector3 v)

Public Attributes

- float **radius** = 0.5f
- float **height** = 1f
- float **maxAcceleration** = 8
- float **maxSpeed** = 3f
- float **collisionQueryRange** = 4
- float **pathOptimizationRange** = 15
- float **separationWeight** = 3
- float **gravity** = -10
- float **updateFrequency** = 2

Properties

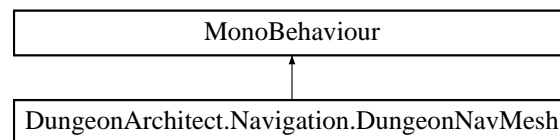
- [DungeonNavMesh NavMesh](#) [get]
- override Vector3 **Destination** [get, set]
- override Vector3 **Velocity** [get, set]
- override Vector3 **Direction** [get]
- float **DesiredSpeed** [get]

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

- [github/dungeon-architect-unity/Scripts/Navigation/DungeonNavAgent3D.cs](#)

5.29 DungeonArchitect.Navigation.DungeonNavMesh Class Reference

Inheritance diagram for DungeonArchitect.Navigation.DungeonNavMesh:



Public Member Functions

- void **SetNavMeshVisible** (bool show)
- void **Build** ()

Static Public Member Functions

- static Vector3 **ToV3** (SVector3 v)

Public Attributes

- float **agentHeight** = 2
- float **agentRadius** = 0.5f
- float **agentClimbHeight** = 0.5f
- float **cellSize** = 0.2f
- int **maxCrowdAgents** = 50
- Mesh **visualization**
- Color **visualizationColor** = new Color(0, 0.5f, 1, 0.25f)
- bool **visualize2D** = false
- Crowd **crowd**

Properties

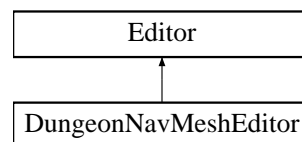
- SharpNav.NavMesh **NavMesh** [get]
- SharpNav.NavMeshQuery **NavMeshQuery** [get]
- Crowd **Crowd** [get]
- SharpNav.PolyMesh **PolyMesh** [get]
- SharpNav.PolyMeshDetail **PolyMeshDetail** [get]

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

- [github/dungeon-architect-unity/Scripts/Navigation/DungeonNavMesh.cs](#)

5.30 DungeonNavMeshEditor Class Reference

Inheritance diagram for DungeonNavMeshEditor:



Public Member Functions

- void **OnEnable** ()
- void **OnDisable** ()
- override void **OnInspectorGUI** ()

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

- [github/dungeon-architect-unity/Editor/Navigation/DungeonNavMeshEditor.cs](#)

5.31 DungeonArchitect.Editors.DungeonObjectTraker Class Reference

Tracks active dungeon objects in the scene and finds ones that have the active graph being edited This is used for real-time updates on the dungeon object as the graph is modified from the editor

Public Member Functions

- void **Update** ()
- void **RequestRebuild** ()
Rebuilds the dungeons that reference the theme graphs tracked by this object

Properties

- **Graph ActiveGraph** [get, set]
The active graph being edited by the theme graph editor
- **Dungeon[] Dungeons** [get]
The dungeon objects in the scene that uses the graph tracked by this object

5.31.1 Detailed Description

Tracks active dungeon objects in the scene and finds ones that have the active graph being edited This is used for real-time updates on the dungeon object as the graph is modified from the editor

5.31.2 Member Function Documentation

5.31.2.1 void DungeonArchitect.Editors.DungeonObjectTraker.RequestRebuild ()

Rebuilds the dungeons that reference the theme graphs tracked by this object

5.31.3 Property Documentation

5.31.3.1 Graph DungeonArchitect.Editors.DungeonObjectTraker.ActiveGraph [get], [set]

The active graph being edited by the theme graph editor

5.31.3.2 Dungeon [] DungeonArchitect.Editors.DungeonObjectTraker.Dungeons [get]

The dungeon objects in the scene that uses the graph tracked by this object

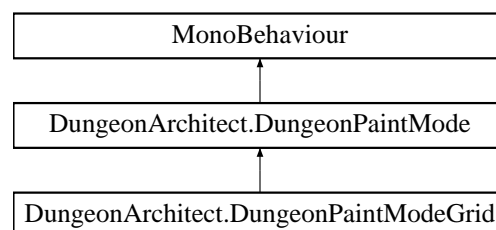
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphEditor.cs](#)

5.32 DungeonArchitect.DungeonPaintMode Class Reference

Manage the editor paint mode so you can paint the layout of you dungeon. You should implement your own paint mode depending on your dungeon builder's data structures and requirements

Inheritance diagram for DungeonArchitect.DungeonPaintMode:



Public Member Functions

- [DungeonConfig GetDungeonConfig \(\)](#)
Gets the configuration of the dungeon
- [DungeonModel GetDungeonModel \(\)](#)
Gets the model used by the owning dungeon
- [Dungeon GetDungeon \(\)](#)
Gets the owning dungeon
- [T GetSiblingComponent< T > \(\)](#)

Protected Attributes

- [Dungeon](#) **dungeon**
- [DungeonModel](#) **dungeonModel**
- [DungeonConfig](#) **dungeonConfig**

5.32.1 Detailed Description

Manage the editor paint mode so you can paint the layout of you dungeon. You should implement your own paint mode depending on your dungeon builder's data structures and requirements

5.32.2 Member Function Documentation

5.32.2.1 **Dungeon** `DungeonArchitect.DungeonPaintMode.GetDungeon ()`

Gets the owning dungeon

Returns

The owning dungeon

5.32.2.2 **DungeonConfig** `DungeonArchitect.DungeonPaintMode.GetDungeonConfig ()`

Gets the configuration of the dungeon

Returns

5.32.2.3 **DungeonModel** `DungeonArchitect.DungeonPaintMode.GetDungeonModel ()`

Gets the model used by the owning dungeon

Returns

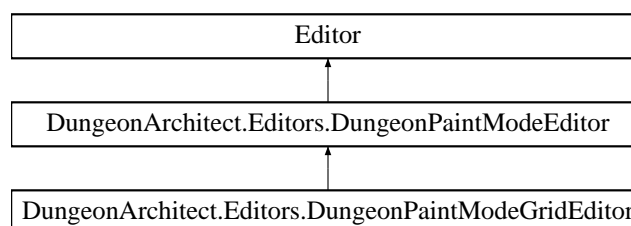
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/DungeonPaintMode.cs](#)

5.33 `DungeonArchitect.Editors.DungeonPaintModeEditor` Class Reference

Custom property editor for the paint mode object

Inheritance diagram for `DungeonArchitect.Editors.DungeonPaintModeEditor`:



Protected Member Functions

- virtual void **SceneGUI** (SceneView sceneview)

5.33.1 Detailed Description

Custom property editor for the paint mode object

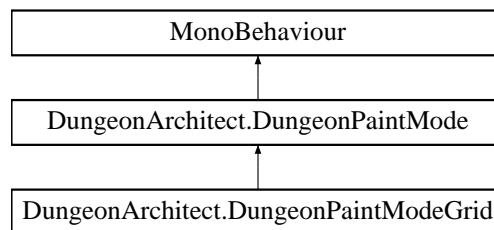
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/Dungeon/DungeonPaintModeEditor.cs](#)

5.34 DungeonArchitect.DungeonPaintModeGrid Class Reference

Editor tooling for the grid based dungeon builder. Lets you paint with a grid based brush

Inheritance diagram for DungeonArchitect.DungeonPaintModeGrid:



Public Member Functions

- float **GetCursorHeight** ()
- void **SetElevationDelta** (int delta)
- [GridDungeonModel](#) **GetDungeonModelGrid** ()

Public Attributes

- int [cursorLogicalHeight](#) = 0
The height of the cursor in grid coordinates. Can also be changed with the mouse wheel in the editor when activated
- float [overlayOpacity](#) = 0.1f
The opacity of the overlay colored tiles
- bool [mode2D](#) = false
Indicates if the painting is to be done in 2D mode (for 2D dungeons) This flag is used for the editor tooling. The model still stores it in 3D
- int [brushSize](#) = 1
The size of the brush. This would create a brush of size NxN

Additional Inherited Members

5.34.1 Detailed Description

Editor tooling for the grid based dungeon builder. Lets you paint with a grid based brush

5.34.2 Member Data Documentation

5.34.2.1 `int DungeonArchitect.DungeonPaintModeGrid.brushSize = 1`

The size of the brush. This would create a brush of size NxN

5.34.2.2 `int DungeonArchitect.DungeonPaintModeGrid.cursorLogicalHeight = 0`

The height of the cursor in grid coordinates. Can also be changed with the mouse wheel in the editor when activated

5.34.2.3 `bool DungeonArchitect.DungeonPaintModeGrid.mode2D = false`

Indicates if the painting is to be done in 2D mode (for 2D dungeons) This flag is used for the editor tooling. The model still stores it in 3D

5.34.2.4 `float DungeonArchitect.DungeonPaintModeGrid.overlayOpacity = 0.1f`

The opacity of the overlay colored tiles

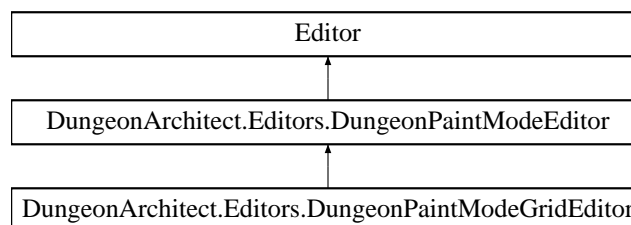
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/PaintModes/DungeonPaintModeGrid.cs](https://github.com/dungeon-architect-unity/Scripts/Dungeon/PaintModes/DungeonPaintModeGrid.cs)

5.35 `DungeonArchitect.Editors.DungeonPaintModeGridEditor` Class Reference

Custom property editor for the Paint model game object

Inheritance diagram for `DungeonArchitect.Editors.DungeonPaintModeGridEditor`:



Public Member Functions

- override void **OnInspectorGUI** ()

Protected Member Functions

- override void **SceneGUI** (SceneView sceneview)

5.35.1 Detailed Description

Custom property editor for the Paint model game object

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

- [github/dungeon-architect-unity/Editor/Dungeon/PaintModes/DungeonPaintModeGridEditor.cs](https://github.com/dungeon-architect-unity/Editor/Dungeon/PaintModes/DungeonPaintModeGridEditor.cs)

5.36 DungeonArchitect.DungeonPropDataAsset Class Reference

The data-structure for serializing the theme graph to disk

Public Member Functions

- void **BuildFromGraph** ([Graph](#) graph)

Public Attributes

- List< [PropTypeData](#) > **Props** = new List<[PropTypeData](#)>()

5.36.1 Detailed Description

The data-structure for serializing the theme graph to disk

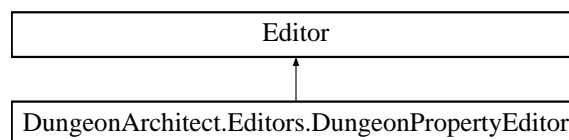
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/DungeonPropDataAsset.cs](#)

5.37 DungeonArchitect.Editors.DungeonPropertyEditor Class Reference

Custom property editor for the dungeon game object

Inheritance diagram for DungeonArchitect.Editors.DungeonPropertyEditor:



Public Member Functions

- override void **OnInspectorGUI** ()

5.37.1 Detailed Description

Custom property editor for the dungeon game object

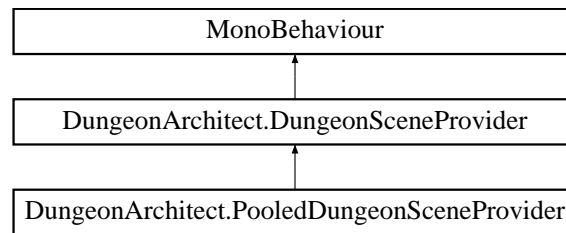
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/Dungeon/DungeonPropertyEditor.cs](#)

5.38 DungeonArchitect.DungeonSceneProvider Class Reference

A scene provider instantiates game objects into the scene. Implementations can customize the instantiation process if needed (e.g. object pooling etc)

Inheritance diagram for DungeonArchitect.DungeonSceneProvider:



Public Member Functions

- virtual void [OnDungeonBuildStart](#) ()
Called when build is started
- virtual void [OnDungeonBuildStop](#) ()
Called after build has ended
- virtual void [AddGameObject](#) ([GameObjectPropTypeData](#) gameObjectProp, Matrix4x4 transform)
Request the creation of a game object
- virtual void [AddSprite](#) ([SpritePropTypeData](#) spriteProp, Matrix4x4 transform)
Request the creation of a sprite object

Public Attributes

- GameObject [itemParent](#)
The parent for all spawned game objects. Assign this to create all spawned objects underneath it to avoid cluttering up the hierarchy

Protected Member Functions

- void **Initialize** ()
- GameObject **BuildGameObject** ([GameObjectPropTypeData](#) gameObjectProp, Matrix4x4 transform)
- void **FlipSpriteTransform** (ref Matrix4x4 transform, Sprite sprite)
- void **FlipSpritePosition** (ref Matrix4x4 transform)
- void **FlipSpritePosition** (ref Vector3 position)
- GameObject **BuildSpriteObject** ([SpritePropTypeData](#) spriteData, Matrix4x4 transform, string NodeId)
- void **SetTransform** (Transform transform, Matrix4x4 matrix)

Protected Attributes

- [DungeonConfig](#) [config](#)
Dungeon config used by the builder
- [Dungeon](#) [dungeon](#)
The owning dungeon actor reference
- Vector3 **_position** = new Vector3()
- Quaternion **_rotation** = new Quaternion()
- Vector3 **_scale** = new Vector3()

5.38.1 Detailed Description

A scene provider instantiates game objects into the scene. Implementations can customize the instantiation process if needed (e.g. object pooling etc)

5.38.2 Member Function Documentation

5.38.2.1 virtual void DungeonArchitect.DungeonSceneProvider.AddGameObject (GameObjectPropTypeData *gameObjectProp*, Matrix4x4 *transform*) [virtual]

Request the creation of a game object

Parameters

<i>gameObjectProp</i>	The template to use for instantiation
<i>transform</i>	The transform of the instantiated game object

Reimplemented in [DungeonArchitect.PooledDungeonSceneProvider](#).

5.38.2.2 `virtual void DungeonArchitect.DungeonSceneProvider.AddSprite (SpritePropTypeData spriteProp, Matrix4x4 transform) [virtual]`

Request the creation of a sprite object

Parameters

<i>spriteProp</i>	The sprite game object template reference
<i>transform</i>	The transform of the prop

Reimplemented in [DungeonArchitect.PooledDungeonSceneProvider](#).

5.38.2.3 `virtual void DungeonArchitect.DungeonSceneProvider.OnDungeonBuildStart () [virtual]`

Called when build is started

Reimplemented in [DungeonArchitect.PooledDungeonSceneProvider](#).

5.38.2.4 `virtual void DungeonArchitect.DungeonSceneProvider.OnDungeonBuildStop () [virtual]`

Called after build has ended

Reimplemented in [DungeonArchitect.PooledDungeonSceneProvider](#).

5.38.3 Member Data Documentation

5.38.3.1 `DungeonConfig DungeonArchitect.DungeonSceneProvider.config [protected]`

[Dungeon](#) config used by the builder

5.38.3.2 `Dungeon DungeonArchitect.DungeonSceneProvider.dungeon [protected]`

The owning dungeon actor reference

5.38.3.3 `GameObject DungeonArchitect.DungeonSceneProvider.itemParent`

The parent for all spawned game objects. Assign this to create all spawned objects underneath it to avoid cluttering up the hierarchy

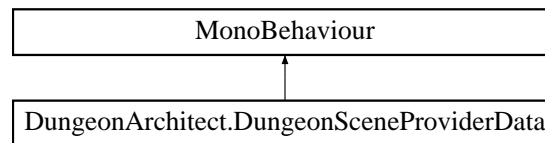
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/DungeonSceneProvider.cs](#)

5.39 DungeonArchitect.DungeonSceneProviderData Class Reference

Meta-data added to each spawned game object in the scene. This is used to identify objects that belong to the dungeons, for later destruction and rebuilding

Inheritance diagram for [DungeonArchitect.DungeonSceneProviderData](#):



Public Attributes

- string [Nodeld](#)
The graph node id this game object was spawned from in the theme graph
- [Dungeon](#) [dungeon](#)
The dungeon this game object belongs to
- bool [affectsNavigation](#) = false
Indicates if the geometry in this node contributes to navigation mesh generation This flag reflects the state set in the theme graph's visual node affectsNavigation flag

5.39.1 Detailed Description

Meta-data added to each spawned game object in the scene. This is used to identify objects that belong to the dungeons, for later destruction and rebuilding

5.39.2 Member Data Documentation

5.39.2.1 bool DungeonArchitect.DungeonSceneProviderData.affectsNavigation = false

Indicates if the geometry in this node contributes to navigation mesh generation This flag reflects the state set in the theme graph's visual node affectsNavigation flag

5.39.2.2 Dungeon DungeonArchitect.DungeonSceneProviderData.dungeon

The dungeon this game object belongs to

5.39.2.3 string DungeonArchitect.DungeonSceneProviderData.Nodeld

The graph node id this game object was spawned from in the theme graph

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

- [github/dungeon-architect-unity/Scripts/Dungeon/DungeonSceneProviderData.cs](#)

5.40 DungeonArchitect.DungeonToolData Class Reference

Tool Data represented by the grid based builder

Properties

- List< [IntVector](#) > **PaintedCells** [get]

5.40.1 Detailed Description

Tool Data represented by the grid based builder

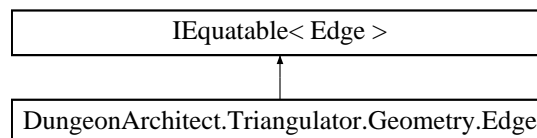
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/Models/GridDungeonModel.cs](#)

5.41 DungeonArchitect.Triangulator.Geometry.Edge Class Reference

[Edge](#) made from two point indexes

Inheritance diagram for DungeonArchitect.Triangulator.Geometry.Edge:



Public Member Functions

- [Edge](#) (int point1, int point2)
Initializes a new edge instance
- [Edge](#) ()
Initializes a new edge instance with start/end indexes of '0'
- bool [Equals](#) ([Edge](#) other)
Checks whether two edges are equal disregarding the direction of the edges

Public Attributes

- int [p1](#)
Start of edge index
- int [p2](#)
End of edge index

5.41.1 Detailed Description

[Edge](#) made from two point indexes

5.41.2 Constructor & Destructor Documentation

5.41.2.1 DungeonArchitect.Triangulator.Geometry.Edge.Edge (int point1, int point2)

Initializes a new edge instance

Parameters

<i>point1</i>	Start edge vertex index
<i>point2</i>	End edge vertex index

5.41.2.2 DungeonArchitect.Triangulator.Geometry.Edge.Edge ()

Initializes a new edge instance with start/end indexes of '0'

5.41.3 Member Function Documentation

5.41.3.1 bool DungeonArchitect.Triangulator.Geometry.Edge.Equals (Edge *other*)

Checks whether two edges are equal disregarding the direction of the edges

Parameters

<i>other</i>	
--------------	--

Returns

5.41.4 Member Data Documentation

5.41.4.1 int DungeonArchitect.Triangulator.Geometry.Edge.p1

Start of edge index

5.41.4.2 int DungeonArchitect.Triangulator.Geometry.Edge.p2

End of edge index

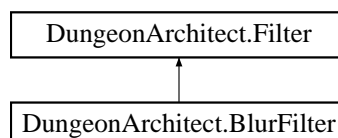
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Triangulator/Geometry/Edge.cs](#)

5.42 DungeonArchitect.Filter Interface Reference

A data filter applied over a 2D data array

Inheritance diagram for DungeonArchitect.Filter:



Public Member Functions

- float[,] **ApplyFilter** (float[,] data)

5.42.1 Detailed Description

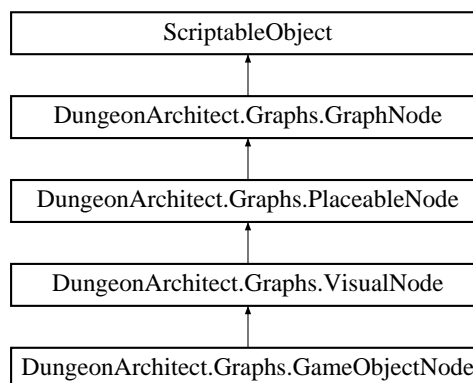
A data filter applied over a 2D data array

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Landscape/Filter/Filter.cs](#)

5.43 DungeonArchitect.Graphs.GameObjectNode Class Reference

Inheritance diagram for DungeonArchitect.Graphs.GameObjectNode:



Public Member Functions

- override void **Initialize** (string id, [Graph](#) graph)
- override void **CopyFrom** ([GraphNode](#) node)

Called when the node is copied. The implementations should implement copy here (e.g. deep / shallow copy depending on implementation)

Public Attributes

- GameObject **Template**

Additional Inherited Members

5.43.1 Member Function Documentation

5.43.1.1 override void **DungeonArchitect.Graphs.GameObjectNode.CopyFrom** ([GraphNode](#) node) [virtual]

Called when the node is copied. The implementations should implement copy here (e.g. deep / shallow copy depending on implementation)

Parameters

<i>node</i>	
-------------	--

Reimplemented from [DungeonArchitect.Graphs.GraphNode](#).

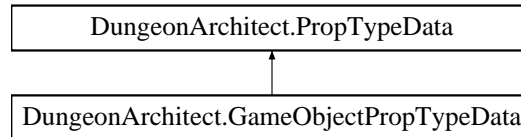
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Graph/ThemeEditor/GameObjectNode.cs](#)

5.44 DungeonArchitect.GameObjectPropTypeData Class Reference

Game Object node data asset attributes

Inheritance diagram for DungeonArchitect.GameObjectPropTypeData:



Public Attributes

- GameObject **Template**

5.44.1 Detailed Description

Game Object node data asset attributes

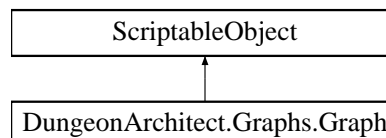
The documentation for this class was generated from the following file:

- `github/dungeon-architect-unity/Scripts/Dungeon/DungeonProp.cs`

5.45 DungeonArchitect.Graphs.Graph Class Reference

Theme [Graph](#) data structure holds all the theme nodes and their connections

Inheritance diagram for DungeonArchitect.Graphs.Graph:



Public Member Functions

- delegate void **OnMarkedDirty** ([Graph](#) graph)
- delegate void **OnGraphStateChanged** ([Graph](#) graph)
- void **OnEnable** ()
- [GraphNode](#) **GetNode** (string id)
Gets the node by it's id
- T[] **GetNodes< T >** ()
Get all nodes of the specified type
- void **MarkAsDirty** ()
Marks the model as dirty
- void **NotifyStateChanged** ()
Call to notify all listeners that the graph state has changed

Properties

- [DungeonArchitect.Graphs.IndexCounter](#) **IndexCounter** [get]
- List< [GraphNode](#) > **Nodes** [get]
List of graph nodes
- List< [GraphLink](#) > **Links** [get]
List of graph links connecting the nodes
- [IndexCounter](#) **TopZIndex** [get]
The z index of the top most node

Events

- OnMarkedDirty **MarkedDirty**
- OnGraphStateChanged **GraphStateChanged**

5.45.1 Detailed Description

Theme [Graph](#) data structure holds all the theme nodes and their connections

5.45.2 Member Function Documentation

5.45.2.1 [GraphNode](#) DungeonArchitect.Graphs.Graph.GetNode (string *id*)

Gets the node by it's id

Parameters

<i>id</i>	The ID of the node
-----------	--------------------

Returns

The retrieved node. null if node with this id doesn't exist

5.45.2.2 `T []` DungeonArchitect.Graphs.Graph.GetNodes< T > ()

Get all nodes of the specified type

Template Parameters

<i>T</i>	The type of nodes to retrieve. Should be a subclass of GraphNode
----------	--

Returns

List of all the nodes of the specified type

Type Constraints

***T* : [GraphNode](#)**

5.45.2.3 void DungeonArchitect.Graphs.Graph.MarkAsDirty ()

Marks the model as dirty

5.45.2.4 void DungeonArchitect.Graphs.Graph.NotifyStateChanged ()

Call to notify all listeners that the graph state has changed

5.45.3 Property Documentation

5.45.3.1 List<GraphLink> DungeonArchitect.Graphs.Graph.Links [get]

List of graph links connecting the nodes

5.45.3.2 List<GraphNode> DungeonArchitect.Graphs.Graph.Nodes [get]

List of graph nodes

5.45.3.3 IndexCounter DungeonArchitect.Graphs.Graph.TopZIndex [get]

The z index of the top most node

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

- [github/dungeon-architect-unity/Scripts/Graph/Graph.cs](#)

5.46 DungeonArchitect.Graphs.GraphCamera Class Reference

A camera that manages the graph editor's viewport

Public Member Functions

- void [Pan](#) (int x, int y)
Pan the camera along the specified delta value
- void [Pan](#) (Vector2 delta)
Pan the camera along the specified delta value
- void [HandleInput](#) (Event e)
Handles the user mouse and keyboard input
- Vector2 [WorldToScreen](#) (Vector2 worldCoord)
Converts world coordinates (in the graph view) into Screen coordinates (relative to the editor window)
- Vector2 [ScreenToWorld](#) (Vector2 screenCoord)
Converts the Screen coordinates (of the editor window) into the graph's world coordinate
- void [FocusOnBestFit](#) ([Graph](#) graph, Rect editorBounds)
Moves the camera so most of the nodes are visible
- void [FocusOnMarker](#) ([Graph](#) graph, Rect editorBounds, string markerName)
Moves the camera to the marker node

Properties

- Vector2 [Position](#) [get, set]
Position of the camera
- Vector2 [Scale](#) [get, set]
Zoom scale of the graph camera

5.46.1 Detailed Description

A camera that manages the graph editor's viewport

5.46.2 Member Function Documentation

5.46.2.1 void DungeonArchitect.Graphs.GraphCamera.FocusOnBestFit (Graph *graph*, Rect *editorBounds*)

Moves the camera so most of the nodes are visible

Parameters

<i>graph</i>	The graph to query
<i>editorBounds</i>	The bounds of the editor window

5.46.2.2 void DungeonArchitect.Graphs.GraphCamera.FocusOnMarker (Graph *graph*, Rect *editorBounds*, string *markerName*)

Moves the camera to the marker node

Parameters

<i>graph</i>	The graph to work on
<i>editorBounds</i>	The bounds of the editor window
<i>markerName</i>	The marker name to focus on

5.46.2.3 void DungeonArchitect.Graphs.GraphCamera.HandleInput (Event *e*)

Handles the user mouse and keyboard input

Parameters

<i>e</i>	
----------	--

5.46.2.4 void DungeonArchitect.Graphs.GraphCamera.Pan (int *x*, int *y*)

Pan the camera along the specified delta value

Parameters

<i>x</i>	Delta value to move along the X value
<i>y</i>	Delta value to move along the Y value

5.46.2.5 void DungeonArchitect.Graphs.GraphCamera.Pan (Vector2 *delta*)

Pan the camera along the specified delta value

Parameters

<i>delta</i>	The delta offset to move the camera to
--------------	--

5.46.2.6 Vector2 DungeonArchitect.Graphs.GraphCamera.ScreenToWorld (Vector2 *screenCoord*)

Converts the Screen coordinates (of the editor window) into the graph's world coordinate

Parameters

<i>screenCoord</i>	
--------------------	--

Returns

The world coordinates in the graph view

5.46.2.7 Vector2 DungeonArchitect.Graphs.GraphCamera.WorldToScreen (Vector2 worldCoord)

Converts world coordinates (in the graph view) into Screen coordinates (relative to the editor window)

Parameters

<i>worldCoord</i>	The world coordnates of the graph view
-------------------	--

Returns

The screen coordnates relative to the editor window

5.46.3 Property Documentation

5.46.3.1 Vector2 DungeonArchitect.Graphs.GraphCamera.Position [get], [set]

Position of the camera

5.46.3.2 Vector2 DungeonArchitect.Graphs.GraphCamera.Scale [get], [set]

Zoom scale of the graph camera

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

- [github/dungeon-architect-unity/Scripts/Graph/GraphCamera.cs](#)

5.47 DungeonArchitect.Editors.GraphContextMenu Class Reference

The context menu shown when the user right clicks on the theme graph editor

Public Member Functions

- delegate void **OnRequestContextMenuCreation** (Event e)
- delegate void **OnMenuItemClicked** ([GraphMenuAction](#) action, [GraphContextMenuEvent](#) e)
- void [HandleInput](#) (Event e)
Handles mouse input
- void [Show](#) ([Graph](#) graph, [GraphPin](#) sourcePin, Vector2 mouseWorld)
Shows the context menu in the theme graph editor
- void [Show](#) ([Graph](#) graph)
Show the context menu

Events

- OnRequestContextMenuCreation **RequestContextMenuCreation**
- OnMenuItemClicked **MenuItemClicked**

5.47.1 Detailed Description

The context menu shown when the user right clicks on the theme graph editor

5.47.2 Member Function Documentation

5.47.2.1 void DungeonArchitect.Editors.GraphContextMenu.HandleInput (Event *e*)

Handles mouse input

Parameters

<i>e</i>	Input event data
----------	------------------

5.47.2.2 void DungeonArchitect.Editors.GraphContextMenu.Show (Graph *graph*, GraphPin *sourcePin*, Vector2 *mouseWorld*)

Shows the context menu in the theme graph editor

Parameters

<i>graph</i>	The graph shown in the graph editor
<i>sourcePin</i>	The source pin, if the user dragged a link out of a pin. null otherwise
<i>mouseWorld</i>	The position of the mouse. The context menu would be shown from here

5.47.2.3 void DungeonArchitect.Editors.GraphContextMenu.Show (Graph *graph*)

Show the context menu

Parameters

<i>graph</i>	The owning graph
--------------	------------------

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

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphContextMenu.cs](#)

5.48 DungeonArchitect.Editors.GraphContextMenuEvent Class Reference

The graph context menu event data

Public Attributes

- [GraphPin](#) **sourcePin**
- Vector2 **mouseWorldPosition**
- object **userdata**

5.48.1 Detailed Description

The graph context menu event data

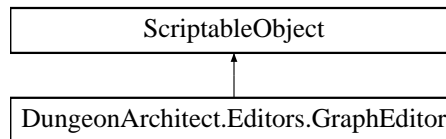
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphContextMenu.cs](#)

5.49 DungeonArchitect.Editors.GraphEditor Class Reference

The graph editor script for managing a graph. This contains the bulk of the logic for graph editing

Inheritance diagram for DungeonArchitect.Editors.GraphEditor:



Public Member Functions

- void **Init** ([Graph](#) graph, Rect editorBounds)
Initializes the graph editor with the specified graph
- void **FocusCameraOnMarker** (string markerName, Rect editorBounds)
Moves the graph editor viewport to show the marker on the screen
- void **FocusCameraOnBestFit** (Rect editorBounds)
Moves the graph editor viewport to show as many markers as possible. Called when a new graph is loaded
- void **OnEnable** ()
- void **OnDisable** ()
- void **OnDestroy** ()
- void **Update** ()
- void **HandleInput** (Event e)
Handles user input (mouse and keyboard)
- void **OnNodeSelectionChanged** ()
- void **Draw** (Rect bounds)
Renders the graph editor in the editor window
- T **CreateNode**< T > (Vector2 screenCoord)
Creates a new node in the specified screen coordinate
- void **SelectNode** ([GraphNode](#) nodeToSelect)
Selects and highlights a node
- [GraphPin](#) **GetPinUnderPosition** (Vector2 worldPosition)
Gets the node pin under the mouse position. Takes the owning node's Z-order into consideration

Properties

- [Graph](#) **Graph** [get]
The owning graph
- bool **RealtimeUpdate** [get, set]
If set, updates the dungeon in the viewport whenever the state of the graph is modified

5.49.1 Detailed Description

The graph editor script for managing a graph. This contains the bulk of the logic for graph editing

5.49.2 Member Function Documentation

5.49.2.1 T DungeonArchitect.Editors.GraphEditor.CreateNode< T > (Vector2 screenCoord)

Creates a new node in the specified screen coordinate

Template Parameters

<i>T</i>	The type of node to created. Should be a subclass of <code>GraphNode</code>
----------	---

Parameters

<i>screenCoord</i>	The screen coordinate to place the node at
--------------------	--

Returns

The created graph node

Type Constraints

***T* : `GraphNode`**

***T* : `new()`**

5.49.2.2 void `DungeonArchitect.Editors.GraphEditor.Draw (Rect bounds)`

Renders the graph editor in the editor window

Parameters

<i>bounds</i>	The bounds of the editor window
---------------	---------------------------------

5.49.2.3 void `DungeonArchitect.Editors.GraphEditor.FocusCameraOnBestFit (Rect editorBounds)`

Moves the graph editor viewport to show as many markers as possible. Called when a new graph is loaded

Parameters

<i>editorBounds</i>	The bounds of the editor window
---------------------	---------------------------------

5.49.2.4 void `DungeonArchitect.Editors.GraphEditor.FocusCameraOnMarker (string markerName, Rect editorBounds)`

Moves the graph editor viewport to show the marker on the screen

Parameters

<i>markerName</i>	The name of the marker to focus on
<i>editorBounds</i>	The bounds of the editor

5.49.2.5 **GraphPin** `DungeonArchitect.Editors.GraphEditor.GetPinUnderPosition (Vector2 worldPosition)`

Gets the node pin under the mouse position. Takes the owning node's Z-order into consideration

Parameters

<i>worldPosition</i>	The world position in graph coordinates
----------------------	---

Returns

The pin under the specified position. null otherwise

5.49.2.6 void `DungeonArchitect.Editors.GraphEditor.HandleInput (Event e)`

Handles user input (mouse and keyboard)

Parameters

<i>e</i>	
----------	--

5.49.2.7 void DungeonArchitect.Editors.GraphEditor.Init (Graph *graph*, Rect *editorBounds*)

Initializes the graph editor with the specified graph

Parameters

<i>graph</i>	The owning graph
<i>editorBounds</i>	The bounds of the editor window

5.49.2.8 void DungeonArchitect.Editors.GraphEditor.SelectNode (GraphNode *nodeToSelect*)

Selects and highlights a node

Parameters

<i>nodeToSelect</i>	
---------------------	--

5.49.3 Property Documentation

5.49.3.1 Graph DungeonArchitect.Editors.GraphEditor.Graph [get]

The owning graph

5.49.3.2 bool DungeonArchitect.Editors.GraphEditor.RealtimeUpdate [get], [set]

If set, updates the dungeon in the viewport whenever the state of the graph is modified

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

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphEditor.cs](#)

5.50 DungeonArchitect.Editors.GraphEditorConstants Class Reference

Graph editor constants

Static Public Attributes

- static readonly Color **PIN_COLOR** = new Color(0.4f, 0.4f, 0.4f)
- static readonly Color **PIN_COLOR_HOVER** = new Color(1, 0.6f, 0.0f)
- static readonly Color **PIN_COLOR_CLICK** = new Color(1, 0.9f, 0.0f)
- static readonly Color **NODE_COLOR** = new Color(0.2824f, 0.2824f, 0.2824f)
- static readonly Color **NODE_COLOR_SELECTED** = new Color(.9f, 0.5f, 0.0f)
- static readonly Color **TEXT_COLOR** = new Color(0.9f, 0.9f, 0.9f)
- static readonly Color **TEXT_COLOR_SELECTED** = Color.white

5.50.1 Detailed Description

Graph editor constants

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

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphEditorConstants.cs](#)

5.51 DungeonArchitect.Editors.GraphInputHandler Class Reference

Static Public Member Functions

- static bool [HandleNodeInput](#) ([GraphNode](#) node, Event e, [GraphCamera](#) camera)
Handles user input (keyboard and mouse)
- static bool [HandlePinInput](#) ([GraphPin](#) pin, Event e, [GraphCamera](#) camera)
Handles the mouse input and returns true if handled

5.51.1 Member Function Documentation

5.51.1.1 static bool [DungeonArchitect.Editors.GraphInputHandler.HandleNodeInput](#) ([GraphNode](#) node, Event e, [GraphCamera](#) camera) [static]

Handles user input (keyboard and mouse)

Parameters

<i>e</i>	Input event
<i>camera</i>	Graph camera to convert to / from screen to world coordinates

Returns

true if the input was processed, false otherwise.

5.51.1.2 static bool [DungeonArchitect.Editors.GraphInputHandler.HandlePinInput](#) ([GraphPin](#) pin, Event e, [GraphCamera](#) camera) [static]

Handles the mouse input and returns true if handled

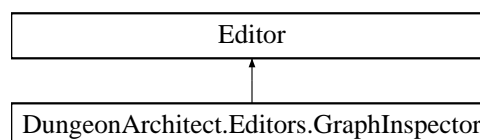
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphOperations.cs](#)

5.52 DungeonArchitect.Editors.GraphInspector Class Reference

Custom property editor for graph objects Shows the graph editor when a theme graph asset is selected

Inheritance diagram for [DungeonArchitect.Editors.GraphInspector](#):



Public Member Functions

- void **OnEnable** ()
- override void [OnInspectorGUI](#) ()

5.52.1 Detailed Description

Custom property editor for graph objects Shows the graph editor when a theme graph asset is selected

5.52.2 Member Function Documentation

5.52.2.1 override void DungeonArchitect.Editors.GraphInspector.OnInspectorGUI ()

ShowEditor();

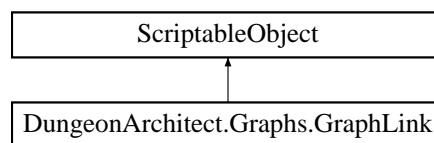
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/Utils/GraphInspector.cs](#)

5.53 DungeonArchitect.Graphs.GraphLink Class Reference

A graph link is a directional connection between two graph nodes

Inheritance diagram for DungeonArchitect.Graphs.GraphLink:



Public Member Functions

- void **OnEnable** ()
- float [GetTangentStrength](#) ()

Determines the spring strength of the link. It reduces as it gets smaller to draw good looking link at any distance

Properties

- int [Id](#) [get, set]
The ID of the link
- [GraphPin Input](#) [get, set]
The input pin this link originates from
- [GraphPin Output](#) [get, set]
The output pin this link points to
- [Graph Graph](#) [get, set]
The graph this link belongs to

5.53.1 Detailed Description

A graph link is a directional connection between two graph nodes

5.53.2 Member Function Documentation

5.53.2.1 float `DungeonArchitect.Graphs.GraphLink.GetTangentStrength ()`

Determines the spring strength of the link. It reduces as it gets smaller to draw good looking link at any distance

Returns

5.53.3 Property Documentation

5.53.3.1 `Graph` `DungeonArchitect.Graphs.GraphLink.Graph` `[get]`, `[set]`

The graph this link belongs to

5.53.3.2 `int` `DungeonArchitect.Graphs.GraphLink.Id` `[get]`, `[set]`

The ID of the link

5.53.3.3 `GraphPin` `DungeonArchitect.Graphs.GraphLink.Input` `[get]`, `[set]`

The input pin this link originates from

5.53.3.4 `GraphPin` `DungeonArchitect.Graphs.GraphLink.Output` `[get]`, `[set]`

The output pin this link points to

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

- [github/dungeon-architect-unity/Scripts/Graph/GraphLink.cs](#)

5.54 `DungeonArchitect.Editors.GraphLinkRenderer` Class Reference

Renders the graph link in the graph editor

Static Public Member Functions

- static void **DrawGraphLink** ([GraphRendererContext](#) rendererContext, [GraphLink](#) link, [GraphCamera](#) camera)

5.54.1 Detailed Description

Renders the graph link in the graph editor

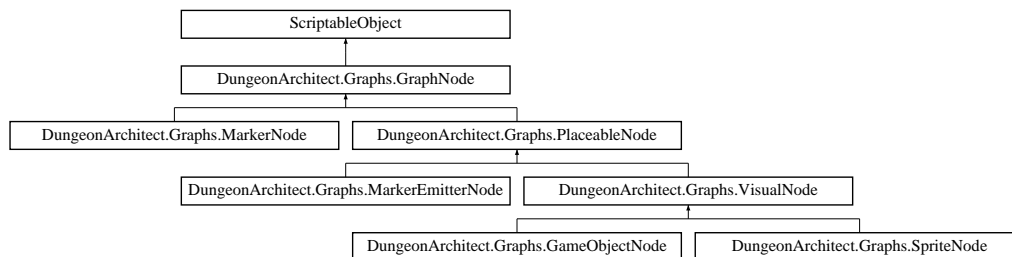
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/GraphEditor/Renderers/GraphLinkRenderer.cs](#)

5.55 DungeonArchitect.Graphs.GraphNode Class Reference

Represents a graph node in the theme graph. This is the base class for all graph nodes

Inheritance diagram for DungeonArchitect.Graphs.GraphNode:



Public Member Functions

- virtual void **OnEnable** ()
- virtual void **Initialize** (string id, [Graph](#) graph)
- virtual void **CopyFrom** ([GraphNode](#) node)
Called when the node is copied. The implementations should implement copy here (e.g. deep / shallow copy depending on implementation)
- [GraphNode](#)[] **GetParentNodes** ()
Gets the list of parent graph nodes
- [GraphNode](#)[] **GetChildNodes** ()
Gets the list of child nodes
- void **DragNode** ([Vector2](#) delta)
Moves the node by the specified delta

Protected Member Functions

- void **UpdateName** (string prefix)
- void **CreatePin** ([GraphPinType](#) pinType, [Vector2](#) position, [Rect](#) boundsOffset, [Vector2](#) tangent)
Creates a pin with the specified configuration

Protected Attributes

- string **id**
- string **caption**
- [Rect](#) **bounds** = new [Rect](#)(10, 10, 120, 120)
- bool **selected** = false
- int **zIndex**
- List< [GraphPin](#) > **inputPins**
- List< [GraphPin](#) > **outputPins**
- [Graph](#) **graph**

Properties

- string **Id** [get, set]
The ID of the graph node
- string **Caption** [get, set]
The caption label of the node. It is up to the implementation to draw this label, if needed

- Rect **Bounds** [get, set]
The bounds of the node
- bool **Selected** [get, set]
Flag to indicate if the node has been selected
- Vector2 **Size** [get, set]
The size of the node
- Vector2 **Position** [get, set]
The position of the node
- int **ZIndex** [get, set]
The Z-index of the node. It determines if the node is on top of other nodes
- List< **GraphPin** > **InputPins** [get]
List of input pins owned by this node
- List< **GraphPin** > **OutputPins** [get]
List of output pins owned by this node
- **GraphPin** **OutputPin** [get]
Gets the first output pin. Returns null if no output pins are defined
- **GraphPin** **InputPin** [get]
Gets the first input pin. Returns null if no input pins are defined
- **Graph** **Graph** [get]
The graph that owns this node
- bool **Dragging** [get, set]

5.55.1 Detailed Description

Represents a graph node in the theme graph. This is the base class for all graph nodes

5.55.2 Member Function Documentation

5.55.2.1 virtual void DungeonArchitect.Graphs.GraphNode.CopyFrom (**GraphNode** *node*) [virtual]

Called when the node is copied. The implementations should implement copy here (e.g. deep / shallow copy depending on implementation)

Parameters

<i>node</i>	
-------------	--

Reimplemented in [DungeonArchitect.Graphs.VisualNode](#), [DungeonArchitect.Graphs.SpriteNode](#), [DungeonArchitect.Graphs.GameObjectNode](#), and [DungeonArchitect.Graphs.PlaceableNode](#).

5.55.2.2 void DungeonArchitect.Graphs.GraphNode.CreatePin (**GraphPinType** *pinType*, Vector2 *position*, Rect *boundsOffset*, Vector2 *tangent*) [protected]

Creates a pin with the specified configuration

Parameters

<i>pinType</i>	The type of pin (input / output)
<i>position</i>	The position of the pin, relative to the node bounds
<i>boundsOffset</i>	The bounds of the pin, relative to the position

<i>tangent</i>	The tangent of the pin. Links connected to the pin would come out from this direction
----------------	---

5.55.2.3 void DungeonArchitect.Graphs.GraphNode.DragNode (Vector2 *delta*)

Moves the node by the specified delta

Parameters

<i>delta</i>	The delta offset to move the node by
--------------	--------------------------------------

5.55.2.4 GraphNode [] DungeonArchitect.Graphs.GraphNode.GetChildNodes ()

Gets the list of child nodes

Returns

List of child nodes

5.55.2.5 GraphNode [] DungeonArchitect.Graphs.GraphNode.GetParentNodes ()

Gets the list of parent graph nodes

Returns

List of parent graph nodes

5.55.3 Property Documentation

5.55.3.1 Rect DungeonArchitect.Graphs.GraphNode.Bounds [get], [set]

The bounds of the node

5.55.3.2 string DungeonArchitect.Graphs.GraphNode.Caption [get], [set]

The caption label of the node. It is up to the implementation to draw this label, if needed

5.55.3.3 Graph DungeonArchitect.Graphs.GraphNode.Graph [get]

The graph that owns this node

5.55.3.4 string DungeonArchitect.Graphs.GraphNode.Id [get], [set]

The ID of the graph node

5.55.3.5 GraphPin DungeonArchitect.Graphs.GraphNode.InputPin [get]

Gets the first input pin. Returns null if no input pins are defined

5.55.3.6 List<GraphPin> DungeonArchitect.Graphs.GraphNode.InputPins [get]

List of input pins owned by this node

5.55.3.7 **GraphPin** `DungeonArchitect.Graphs.GraphNode.OutputPin` [get]

Gets the first output pin. Returns null if no output pins are defined

5.55.3.8 **List<GraphPin>** `DungeonArchitect.Graphs.GraphNode.OutputPins` [get]

List of output pins owned by this node

5.55.3.9 **Vector2** `DungeonArchitect.Graphs.GraphNode.Position` [get], [set]

The position of the node

5.55.3.10 **bool** `DungeonArchitect.Graphs.GraphNode.Selected` [get], [set]

Flag to indicate if the node has been selected

5.55.3.11 **Vector2** `DungeonArchitect.Graphs.GraphNode.Size` [get], [set]

The size of the node

5.55.3.12 **int** `DungeonArchitect.Graphs.GraphNode.ZIndex` [get], [set]

The Z-index of the node. It determines if the node is on top of other nodes

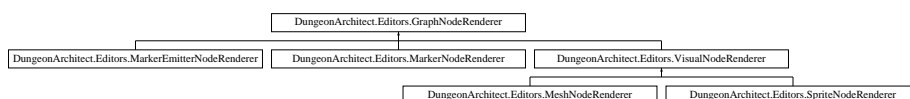
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Graph/GraphNode.cs](#)

5.56 **DungeonArchitectEditors.GraphNodeRenderer** Class Reference

Renders the graph node in the graph editor

Inheritance diagram for `DungeonArchitectEditors.GraphNodeRenderer`:



Public Member Functions

- virtual void **Draw** ([GraphRendererContext](#) rendererContext, [GraphNode](#) node, [GraphCamera](#) camera)

Protected Member Functions

- virtual Color **getBackgroundColor** ([GraphNode](#) node)
- void **DrawBackgroundBox** ([GraphRendererContext](#) rendererContext, [GraphNode](#) node, [GraphCamera](#) camera)
- void **DrawNodeTexture** ([GraphRendererContext](#) rendererContext, [GraphNode](#) node, [GraphCamera](#) camera, string textureName)

5.56.1 Detailed Description

Renders the graph node in the graph editor

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

- [github/dungeon-architect-unity/Editor/GraphEditor/Renderers/GraphNodeRenderer.cs](#)

5.57 DungeonArchitect.Editors.GraphNodeRendererFactory Class Reference

Public Member Functions

- void **RegisterNodeRenderer** (Type nodeType, [GraphNodeRenderer](#) renderer)
- [GraphNodeRenderer](#) **GetRenderer** (Type nodeType)

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

- [github/dungeon-architect-unity/Editor/GraphEditor/Renderers/GraphNodeRenderer.cs](#)

5.58 DungeonArchitect.Editors.GraphOperations Class Reference

Static Public Member Functions

- static T [CreateNode](#)< T > ([Graph](#) graph)
Creates a new graph node of the specified type
- static [GraphNode](#) [CreateNode](#) ([Graph](#) graph, Type t)
Creates a graph node of the specified type
- static T [DuplicateNode](#)< T > ([Graph](#) graph, T originalNode)
Makes a deep copy of a node. Called when a node is copy pasted
- static void [DestroyNode](#) ([GraphNode](#) node)
Destroys a node and removes all references of it from the graph model. Called when the node is deleted from the editor
- static void [DestroyLink](#) ([GraphLink](#) link)
Destroys a node and removes all references of it from the graph model. Called when the node is deleted from the editor
- static void [BreakInputLinks](#) ([GraphNode](#) node)
Breaks all links connected to the input pins
- static void [BreakOutputLinks](#) ([GraphNode](#) node)
Breaks all links connected to the output pins
- static T [CreateLink](#)< T > ([Graph](#) graph, [GraphPin](#) output, [GraphPin](#) input)
Creates a graph link between the two specified pins

5.58.1 Member Function Documentation

5.58.1.1 static void [DungeonArchitect.Editors.GraphOperations.BreakInputLinks](#) ([GraphNode](#) node) [static]

Breaks all links connected to the input pins

5.58.1.2 static void [DungeonArchitect.Editors.GraphOperations.BreakOutputLinks](#) ([GraphNode](#) node) [static]

Breaks all links connected to the output pins

5.58.1.3 **static** T **DungeonArchitect**.Editors.**GraphOperations**.**CreateLink**< T > (**Graph** *graph*, **GraphPin** *output*, **GraphPin** *input*) [static]

Creates a graph link between the two specified pins

Template Parameters

<i>T</i>	The type of the link. Should be GraphLink or one of its subclass
----------	--

Parameters

<i>output</i>	The output pin from where the link originates
<i>input</i>	The input pin, where the link points to

Returns

Type Constraints

***T* : GraphLink**

5.58.1.4 static GraphNode DungeonArchitect.Editors.GraphOperations.CreateNode (Graph *graph*, Type *t*) [static]

Creates a graph node of the specified type

Parameters

<i>t</i>	The type of node to create. Should be a subclass of GraphNode
----------	---

Returns

The created graph node

5.58.1.5 static T DungeonArchitect.Editors.GraphOperations.CreateNode< T > (Graph *graph*) [static]

Creates a new graph node of the specified type

Template Parameters

<i>T</i>	The type of node to create. Should be a subclass of GraphNode
----------	---

Returns

The created graph node

Type Constraints

***T* : GraphNode**

5.58.1.6 static void DungeonArchitect.Editors.GraphOperations.DestroyLink (GraphLink *link*) [static]

Destroys a node and removes all references of it from the graph model. Called when the node is deleted from the editor

Parameters

<i>node</i>	
-------------	--

5.58.1.7 static void DungeonArchitect.Editors.GraphOperations.DestroyNode (GraphNode *node*) [static]

Destroys a node and removes all references of it from the graph model. Called when the node is deleted from the editor

Parameters

<i>node</i>	
-------------	--

5.58.1.8 `static T DungeonArchitect.Editors.GraphOperations.DuplicateNode< T > (Graph graph, T originalNode)`
`[static]`

Makes a deep copy of a node. Called when a node is copy pasted

Template Parameters

<i>T</i>	
----------	--

Parameters

<i>originalNode</i>	
---------------------	--

Returns

Type Constraints

T* : *GraphNode

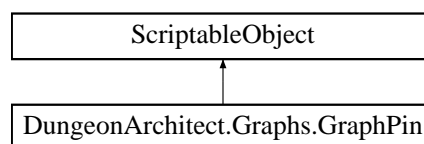
The documentation for this class was generated from the following file:

- `github/dungeon-architect-unity/Editor/GraphEditor/GraphOperations.cs`

5.59 DungeonArchitect.Graphs.GraphPin Class Reference

A pin is used to connect a link to a node

Inheritance diagram for `DungeonArchitect.Graphs.GraphPin`:



Public Member Functions

- delegate void **OnPinLinksDestroyed** ([GraphPin](#) pin)
- void **OnEnable** ()
- [GraphLink](#)[] **GetConnctectedLinks** ()
Gets all the links connected to this pin
- bool **ContainsPoint** ([Vector2](#) worldPoint)
Checks if a point is inside the pin
- Rect **GetWorldBounds** ()
Gets the bounds of the pin, in world coordinates
- Rect **GetBounds** ()
Gets the bounds of the pin, relative to the node position
- void **NotifyPinLinksDestroyed** ()

Properties

- [GraphPinMouseState ClickState](#) [get, set]
The state of the mouse input on this pin
- [GraphPinType PinType](#) [get, set]
The type of this pin
- [GraphNode Node](#) [get, set]
The owning graph node
- [Vector2 Position](#) [get, set]
The position of the graph pin, relative to the owning node's position
- [Vector2 WorldPosition](#) [get]
The world position of the pin
- [Rect BoundsOffset](#) [get, set]
The bounds of the pin, relative to the node's position
- [Vector2 Tangent](#) [get, set]
The tangent of the pin. Links connected to this pin would come in or out from this direction
- [float TangentStrength](#) [get, set]
The spring strength of the link connected to this pin
- [bool RequestLinkDeletionInitiated](#) [get, set]

Events

- [OnPinLinksDestroyed PinLinksDestroyed](#)
Notifies whenever the pin is destroyed

5.59.1 Detailed Description

A pin is used to connect a link to a node

5.59.2 Member Function Documentation

5.59.2.1 bool DungeonArchitect.Graphs.GraphPin.ContainsPoint ([Vector2 worldPoint](#))

Checks if a point is inside the pin

Parameters

worldPoint	The point to test in world coordinates
----------------------------	--

Returns

true, if inside the bounds of this pin, false otherwise

5.59.2.2 Rect DungeonArchitect.Graphs.GraphPin.GetBounds ()

Gets the bounds of the pin, relative to the node position

Returns

The bounds of the pin, relative to the node position

5.59.2.3 **GraphLink []** `DungeonArchitect.Graphs.GraphPin.GetConntectedLinks ()`

Gets all the links connected to this pin

Returns

The connected links.

5.59.2.4 **Rect** `DungeonArchitect.Graphs.GraphPin.GetWorldBounds ()`

Gets the bounds of the pin, in world coordinates

Returns

The bounds of the pin, in world coordinates

5.59.3 Property Documentation

5.59.3.1 **Rect** `DungeonArchitect.Graphs.GraphPin.BoundsOffset` [get], [set]

The bounds of the pin, relative to the node's position

5.59.3.2 **GraphPinMouseState** `DungeonArchitect.Graphs.GraphPin.ClickState` [get], [set]

The state of the mouse input on this pin

5.59.3.3 **GraphNode** `DungeonArchitect.Graphs.GraphPin.Node` [get], [set]

The owning graph node

5.59.3.4 **GraphPinType** `DungeonArchitect.Graphs.GraphPin.PinType` [get], [set]

The type of this pin

5.59.3.5 **Vector2** `DungeonArchitect.Graphs.GraphPin.Position` [get], [set]

The position of the graph pin, relative to the owning node's position

5.59.3.6 **Vector2** `DungeonArchitect.Graphs.GraphPin.Tangent` [get], [set]

The tangent of the pin. Links connected to this pin would come in or out from this direction

5.59.3.7 **float** `DungeonArchitect.Graphs.GraphPin.TangentStrength` [get], [set]

The spring strength of the link connected to this pin

5.59.3.8 **Vector2** `DungeonArchitect.Graphs.GraphPin.WorldPosition` [get]

The world position of the pin

5.59.4 Event Documentation

5.59.4.1 OnPinLinksDestroyed DungeonArchitect.Graphs.GraphPin.PinLinksDestroyed

Notifies whenever the pin is destroyed

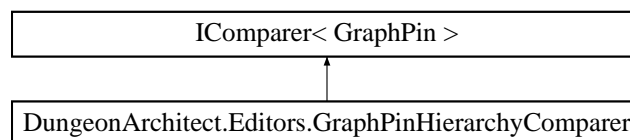
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Graph/GraphPin.cs](#)

5.60 DungeonArchitect.Editor.GraphPinHierarchyComparer Class Reference

Sorts the pins based on their owning node's type

Inheritance diagram for DungeonArchitect.Editor.GraphPinHierarchyComparer:



Public Member Functions

- int **Compare** ([GraphPin](#) x, [GraphPin](#) y)

5.60.1 Detailed Description

Sorts the pins based on their owning node's type

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

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphEditor.cs](#)

5.61 DungeonArchitect.Editor.GraphPinRenderer Class Reference

Renders a graph pin hosted inside a node

Static Public Member Functions

- static void **Draw** ([GraphRendererContext](#) rendererContext, [GraphPin](#) pin, [GraphCamera](#) camera)

5.61.1 Detailed Description

Renders a graph pin hosted inside a node

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

- [github/dungeon-architect-unity/Editor/GraphEditor/Renderers/GraphPinRenderer.cs](#)

5.62 DungeonArchitect.Editors.GraphRendererContext Class Reference

The rendering context for drawing the theme editor

Properties

- [DungeonEditorResources](#) **Resources** [get]

5.62.1 Detailed Description

The rendering context for drawing the theme editor

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

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphEditor.cs](#)

5.63 DungeonArchitect.Graphs.GraphSchema Class Reference

The graph schema defines the rules of the theme graph

Static Public Member Functions

- static bool [CanCreateLink](#) ([GraphPin](#) output, [GraphPin](#) input)
Checks if a link between the two nodes can be created
- static bool [CanCreateLink](#) ([GraphPin](#) output, [GraphPin](#) input, out string errorMessage)

5.63.1 Detailed Description

The graph schema defines the rules of the theme graph

5.63.2 Member Function Documentation

5.63.2.1 static bool [DungeonArchitect.Graphs.GraphSchema.CanCreateLink](#) ([GraphPin](#) output, [GraphPin](#) input)
[static]

Checks if a link between the two nodes can be created

Parameters

<i>output</i>	The pin from which the link originates and goes out
<i>input</i>	The pin where the link points to

Returns

true, if the link is allowed, false otherwise

5.63.2.2 static bool [DungeonArchitect.Graphs.GraphSchema.CanCreateLink](#) ([GraphPin](#) output, [GraphPin](#) input, out string errorMessage) [static]

Parameters

<i>output</i>	The pin from which the link originates and goes out
<i>input</i>	The pin where the link points to
<i>errorMessage</i>	

Returns

true, if the link is allowed, false otherwise

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

- [github/dungeon-architect-unity/Scripts/Graph/GraphSchema.cs](#)

5.64 DungeonArchitect.Editors.GraphSelectionBox Class Reference

Manages the selection box for selecting multiple objects in the graph editor

Public Member Functions

- delegate void **OnSelectionPerformed** (Rect boundsScreenSpace)
- void [HandleInput](#) (Event e)
Handles user input (mouse)
- bool **IsSelectionValid** ()
- void **Draw** ()

Properties

- Rect **Bounds** [get, set]
- bool **Dragging** [get]

Events

- OnSelectionPerformed **SelectionPerformed**

5.64.1 Detailed Description

Manages the selection box for selecting multiple objects in the graph editor

5.64.2 Member Function Documentation

5.64.2.1 void DungeonArchitect.Editors.GraphSelectionBox.HandleInput (Event e)

Handles user input (mouse)

Parameters

<i>e</i>	
----------	--

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

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphEditor.cs](#)

5.65 DungeonArchitect.Editors.GraphTooltip Class Reference

Graph tooltip singleton

Static Public Member Functions

- static void **Clear** ()

Static Public Attributes

- static string **message** = ""
Set this to display a tooltip in the graph editor

5.65.1 Detailed Description

Graph tooltip singleton

5.65.2 Member Data Documentation

5.65.2.1 string **DungeonArchitect.Editors.GraphTooltip.message** = "" [static]

Set this to display a tooltip in the graph editor

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

- github/dungeon-architect-unity/Editor/GraphEditor/Renderers/TooltipRenderer.cs

5.66 DungeonArchitect.Editors.GraphTooltipRenderer Class Reference

Renders a tooltip in the graph editor. The tooltip message is defined in [GraphTooltip.message](#)

Static Public Member Functions

- static void **Draw** ([GraphRendererContext](#) rendererContext, Vector2 mousePosition)

5.66.1 Detailed Description

Renders a tooltip in the graph editor. The tooltip message is defined in [GraphTooltip.message](#)

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

- github/dungeon-architect-unity/Editor/GraphEditor/Renderers/TooltipRenderer.cs

5.67 DungeonArchitect.Utils.GraphUtils Class Reference

Theme graph utility functions

5.67.1 Detailed Description

Theme graph utility functions

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

- [github/dungeon-architect-unity/Scripts/Utils/GraphUtils.cs](#)

5.68 DungeonArchitect.GridCellInfo Class Reference

Contains meta data about the cells. This structure is used for caching cell information for faster lookup during and after generation of the dungeon

Public Member Functions

- **GridCellInfo** (int pCellId, [CellType](#) pCellType)

Public Attributes

- int **CellId**
- [CellType](#) **CellType**
- bool **ContainsDoor**

5.68.1 Detailed Description

Contains meta data about the cells. This structure is used for caching cell information for faster lookup during and after generation of the dungeon

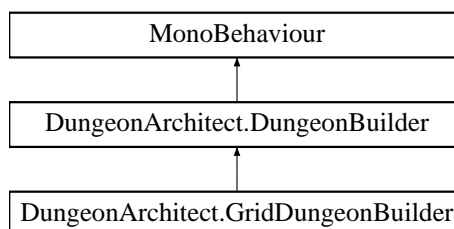
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/Builders/GridDungeonBuilder.cs](#)

5.69 DungeonArchitect.GridDungeonBuilder Class Reference

A [Dungeon](#) Builder implementation that builds a grid based dungeon.

Inheritance diagram for DungeonArchitect.GridDungeonBuilder:



Public Member Functions

- override void **BuildDungeon** ([DungeonConfig](#) config, [DungeonModel](#) model)
Builds the dungeon
- void **BuildCells** ()

- builds the cells in the dungeon*
- override void [EmitMarkers](#) ()
Emit markers defined by this builder
- bool **V3Equal** (Vector3 a, Vector3 b)

Static Public Member Functions

- static void [Seperate](#) ([GridDungeonModel](#) gridModel)
Separates the cells built in the previous phase
- static void [TriangulateRooms](#) ([GridDungeonModel](#) gridModel)
Triangulates the rooms identified in the previous phase This is required to connect the corridors. Delauney triangulation is used to find nice evenly spaced triangles for good connections

Additional Inherited Members

5.69.1 Detailed Description

A [Dungeon](#) Builder implementation that builds a grid based dungeon.

It is based on the awesome algorithm described here by the TinyKeep game's author https://www.reddit.com/r/gamedev/comments/1dlwc4/procedural_dungeon_generation_algorithm_explained/

5.69.2 Member Function Documentation

5.69.2.1 void DungeonArchitect.GridDungeonBuilder.BuildCells ()

builds the cells in the dungeon

5.69.2.2 override void DungeonArchitect.GridDungeonBuilder.BuildDungeon ([DungeonConfig](#) config, [DungeonModel](#) model) [virtual]

Builds the dungeon

Parameters

<i>config</i>	The dungeon configuration
<i>model</i>	The dungeon model

Reimplemented from [DungeonArchitect.DungeonBuilder](#).

5.69.2.3 override void DungeonArchitect.GridDungeonBuilder.EmitMarkers () [virtual]

Emit markers defined by this builder

Reimplemented from [DungeonArchitect.DungeonBuilder](#).

5.69.2.4 static void DungeonArchitect.GridDungeonBuilder.Seperate ([GridDungeonModel](#) gridModel) [static]

Separates the cells built in the previous phase

Parameters

<i>gridModel</i>	
------------------	--

5.69.2.5 static void DungeonArchitect.GridDungeonBuilder.TriangulateRooms (GridDungeonModel *gridModel*) [static]

Triangulates the rooms identified in the previous phase This is required to connect the corridors. Delauney triangulation is used to find nice evenly spaced triangles for good connections

Parameters

<i>gridModel</i>	
------------------	--

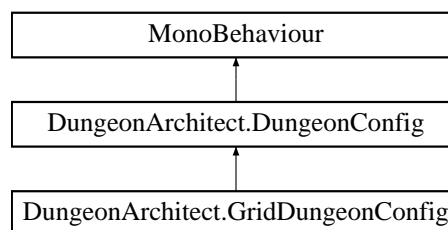
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/Builders/GridDungeonBuilder.cs](#)

5.70 DungeonArchitect.GridDungeonConfig Class Reference

The dungeon configuration for the Grid builder

Inheritance diagram for DungeonArchitect.GridDungeonConfig:



Public Attributes

- int [NumCells](#) = 150
Changing this number would completely change the layout of the dungeon. This is the base random number seed that is used to build the dungeon. There is a convenience function to randomize this value (button labeled R)
- int [MinCellSize](#) = 2
This is how small a cell size can be. While generation, a cell is either converted to a room, corridor or is discarded completely. The [Cell](#) width / height is randomly chosen within this range
- int [MaxCellSize](#) = 5
This is how big a cell size can be. While generation, a cell is either converted to a room, corridor or is discarded completely. The [Cell](#) width / height is randomly chosen within this range
- int [RoomAreaThreshold](#) = 15
If a cell size exceeds past this limit, it is converted into a room. After cells are promoted to rooms, all rooms are connected to each other through corridors (either directly or indirectly. See spanning tree later)
- float [RoomAspectDelta](#) = 0.4f
The aspect ratio of the cells (width to height ratio). Keeping this value near 0 would create square rooms. Bringing this close to 1 would create elongated / stretched rooms with a high width to height ratio
- int [CorridorPadding](#) = 1
The extra width to apply to one side of a corridor
- bool [CorridorPaddingDoubleSided](#) = false
Flag to apply the padding on both sides of the corridor

- float `HeightVariationProbability` = 0.2f

Tweak this value to increase / reduce the height variations (and stairs) in your dungeon. A value close to 0 reduces the height variation and increases as you approach 1. Increasing this value to a higher level might create dungeons with no place for proper stair placement since there is too much height variation. A value of 0.2 to 0.4 seems good

- int `MaxAllowedStairHeight` = 2

The number of logical floor units the dungeon height can vary. This determines how high the dungeon's height can vary (e.g. max 2 floors high). Set this value depending on the stair meshes you designer has created. In the sample demo, there are two stair meshes, one 200 units high (1 floor) and another 400 units high (2 floors). So the default is set to 2

- float `SpanningTreeLoopProbability` = 0.15f

Determines how many loops you would like to have in your dungeon. A value near 0 will create fewer loops creating linear dungeons. A value near 1 would create lots of loops, which would look unoriginal. Its good to allow a few loops so a value close to zero (like 0.2 should be good)

- float `StairConnectionTolerance` = 3

The generator would add stairs to make different areas of the dungeon accessible. However, we do not want too many stairs. For e.g., before adding a stair in a particular elevated area, the generator would check if this area is already accessible from a nearby stair. If so, it would not add it. This tolerance parameter determines how far to look for an existing path before we can add a stair. Play with this parameter if you see too many stairs close to each other, or too few

- float `NormalMean` = 0

The random number generator used in the dungeon generator does not use a uniform distribution. Instead it uses a normal distribution to get higher frequency of lower values and fewer higher values (and hence fewer room cells and a lot more corridor cells). Play with these parameters for different results

- float `NormalStd` = 0.3f

The random number generator used in the dungeon generator does not use a uniform distribution. Instead it uses a normal distribution to get higher frequency of lower values and fewer higher values (and hence fewer room cells and a lot more corridor cells). Play with these parameters for different results

- bool `Mode2D` = false

- float `InitialRoomRadius` = 15

The radius within which to spawn the initial cells before they are separated. Keep to a low value like 10-15

- int `FloorHeight` = 0

`__Internal`

5.70.1 Detailed Description

The dungeon configuration for the Grid builder

5.70.2 Member Data Documentation

5.70.2.1 int `DungeonArchitect.GridDungeonConfig.CorridorPadding` = 1

The extra width to apply to one side of a corridor

5.70.2.2 bool `DungeonArchitect.GridDungeonConfig.CorridorPaddingDoubleSided` = false

Flag to apply the padding on both sides of the corridor

5.70.2.3 int `DungeonArchitect.GridDungeonConfig.FloorHeight` = 0

`__Internal`

5.70.2.4 float DungeonArchitect.GridDungeonConfig.HeightVariationProbability = 0.2f

Tweak this value to increase / reduce the height variations (and stairs) in your dungeon. A value close to 0 reduces the height variation and increases as you approach 1. Increasing this value to a higher level might create dungeons with no place for proper stair placement since there is too much height variation. A value of 0.2 to 0.4 seems good

5.70.2.5 float DungeonArchitect.GridDungeonConfig.InitialRoomRadius = 15

The radius within which to spawn the initial cells before they are separated. Keep to a low value like 10-15

5.70.2.6 int DungeonArchitect.GridDungeonConfig.MaxAllowedStairHeight = 2

The number of logical floor units the dungeon height can vary. This determines how high the dungeon's height can vary (e.g. max 2 floors high). Set this value depending on the stair meshes you designer has created. In the sample demo, there are two stair meshes, one 200 units high (1 floor) and another 400 units high (2 floors). So the default is set to 2

5.70.2.7 int DungeonArchitect.GridDungeonConfig.MaxCellSize = 5

This is how big a cell size can be. While generation, a cell is either converted to a room, corridor or is discarded completely. The [Cell](#) width / height is randomly chosen within this range

5.70.2.8 int DungeonArchitect.GridDungeonConfig.MinCellSize = 2

This is how small a cell size can be. While generation, a cell is either converted to a room, corridor or is discarded completely. The [Cell](#) width / height is randomly chosen within this range

5.70.2.9 float DungeonArchitect.GridDungeonConfig.NormalMean = 0

The random number generator used in the dungeon generator does not use a uniform distribution. Instead it uses a normal distribution to get higher frequency of lower values and fewer higher values (and hence fewer room cells and a lot more corridor cells). Play with these parameters for different results

5.70.2.10 float DungeonArchitect.GridDungeonConfig.NormalStd = 0.3f

The random number generator used in the dungeon generator does not use a uniform distribution. Instead it uses a normal distribution to get higher frequency of lower values and fewer higher values (and hence fewer room cells and a lot more corridor cells). Play with these parameters for different results

5.70.2.11 int DungeonArchitect.GridDungeonConfig.NumCells = 150

Changing this number would completely change the layout of the dungeon. This is the base random number seed that is used to build the dungeon. There is a convenience function to randomize this value (button labeled R)

5.70.2.12 int DungeonArchitect.GridDungeonConfig.RoomAreaThreshold = 15

If a cell size exceeds past this limit, it is converted into a room. After cells are promoted to rooms, all rooms are connected to each other through corridors (either directly or indirectly. See spanning tree later)

5.70.2.13 float `DungeonArchitect.GridDungeonConfig.RoomAspectDelta` = 0.4f

The aspect ratio of the cells (width to height ratio). Keeping this value near 0 would create square rooms. Bringing this close to 1 would create elongated / stretched rooms with a high width to height ratio

5.70.2.14 float `DungeonArchitect.GridDungeonConfig.SpanningTreeLoopProbability` = 0.15f

Determines how many loops you would like to have in your dungeon. A value near 0 will create fewer loops creating linear dungeons. A value near 1 would create lots of loops, which would look unoriginal. Its good to allow a few loops so a value close to zero (like 0.2 should be good)

5.70.2.15 float `DungeonArchitect.GridDungeonConfig.StairConnectionTolerance` = 3

The generator would add stairs to make different areas of the dungeon accessible. However, we do not want too many stairs. For e.g., before adding a stair in a particular elevated area, the generator would check if this area is already accessible from a nearby stair. If so, it would not add it. This tolerance parameter determines how far to look for an existing path before we can add a stair. Play with this parameter if you see too many stairs close to each other, or too few

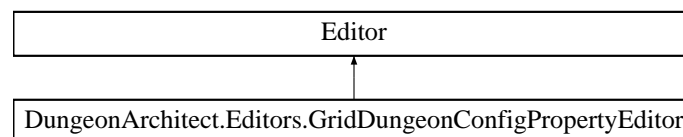
The documentation for this class was generated from the following file:

- `github/dungeon-architect-unity/Scripts/Dungeon/Configs/GridDungeonConfig.cs`

5.71 `DungeonArchitect.Editors.GridDungeonConfigPropertyEditor` Class Reference

Custom property editor for the grid based dungeon configuration

Inheritance diagram for `DungeonArchitect.Editors.GridDungeonConfigPropertyEditor`:



Public Member Functions

- void **OnEnable** ()
- override void **OnInspectorGUI** ()

5.71.1 Detailed Description

Custom property editor for the grid based dungeon configuration

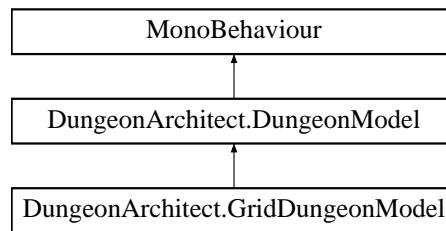
The documentation for this class was generated from the following file:

- `github/dungeon-architect-unity/Editor/Dungeon/DungeonConfigPropertyEditor.cs`

5.72 `DungeonArchitect.GridDungeonModel` Class Reference

Data model for the grid based dungeon builder

Inheritance diagram for `DungeonArchitect.GridDungeonModel`:



Public Member Functions

- [GridCellInfo GetGridCellLookup](#) (int x, int z)
Get meta-data about the grid in x, z grid coordinate
- void [BuildSpatialCellLookup](#) ()
Builds a lookup for fast data retrieval
- void [BuildCellLookup](#) ()
Builds the cell lookup for faster cell retrieval
- [Cell GetCell](#) (int Id)
Gets the cell information
- [Cell FindCellByPosition](#) ([IntVector](#) position)
Finds the cell based on the position in grid coordinates
- override void [ResetModel](#) ()
Clears the dungeon data model
- bool **ContainsStairAtLocation** (int x, int z)
- [StairInfo GetStairAtLocation](#) (int x, int z)
- bool [ContainsStair](#) (int cellA, int cellB)
Check if a stair exists between the two cells

Public Attributes

- [DoorManager DoorManager](#) = new [DoorManager](#)()
- [DungeonModelBuildState State](#) = DungeonModelBuildState.Initial
- [DungeonConfig Config](#)
- List< [Cell](#) > **Cells** = new List<[Cell](#)>()
- Dictionary< int, List< [StairInfo](#) > > **CellStairs** = new Dictionary<int, List<[StairInfo](#)>>()
- Dictionary< int, Dictionary< int, [GridCellInfo](#) > > **GridCellInfoLookup** = new Dictionary<int, Dictionary<int, [GridCellInfo](#)>>()
- Dictionary< int, [Cell](#) > **CellLookup** = new Dictionary<int, [Cell](#)>()
[Cell](#) lookup based on the [Cell](#) Id

Properties

- [CellDoor\[\] Doors](#) [get]
The list of registered doors

5.72.1 Detailed Description

Data model for the grid based dungeon builder

5.72.2 Member Function Documentation

5.72.2.1 void DungeonArchitect.GridDungeonModel.BuildCellLookup ()

Builds the cell lookup for faster cell retrieval

5.72.2.2 void DungeonArchitect.GridDungeonModel.BuildSpatialCellLookup ()

Builds a lookup for fast data retrieval

5.72.2.3 bool DungeonArchitect.GridDungeonModel.ContainsStair (int *cellA*, int *cellB*)

Check if a stair exists between the two cells

Parameters

<i>cellA</i>	
<i>cellB</i>	

Returns

5.72.2.4 Cell DungeonArchitect.GridDungeonModel.FindCellByPosition (IntVector *position*)

Finds the cell based on the position in grid coordinates

Parameters

<i>position</i>	Position to lookup in grid coordinates
-----------------	--

Returns

[Cell](#) information at that location. Returns null if none found

5.72.2.5 Cell DungeonArchitect.GridDungeonModel.GetCell (int *Id*)

Gets the cell information

Parameters

<i>Id</i>	Id of the cell to lookup
-----------	--------------------------

Returns

5.72.2.6 GridCellInfo DungeonArchitect.GridDungeonModel.GetGridCellLookup (int *x*, int *z*)

Get meta-data about the grid in x, z grid coordinate

Parameters

x	X value in grid coordinate
z	Z value in grid coordinate

Returns

5.72.2.7 `override void DungeonArchitect.GridDungeonModel.ResetModel () [virtual]`

Clears the dungeon data model

Reimplemented from [DungeonArchitect.DungeonModel](#).

5.72.3 Member Data Documentation

5.72.3.1 `Dictionary<int, Cell> DungeonArchitect.GridDungeonModel.CellLookup = new Dictionary<int, Cell>()`

[Cell](#) lookup based on the [Cell](#) Id

5.72.4 Property Documentation

5.72.4.1 `CellIDoor [] DungeonArchitect.GridDungeonModel.Doors [get]`

The list of registered doors

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Models/GridDungeonModel.cs](#)

5.73 DungeonArchitect.GridDungeonModelUtils Class Reference

Static Public Member Functions

- static [Cell\[\]](#) **FindFurthestRooms** ([GridDungeonModel](#) model)

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Models/GridDungeonModel.cs](#)

5.74 DungeonArchitect.Graphs.IndexCounter Class Reference

An ID provider for graph objects

Public Member Functions

- int **GetNext** ()

5.74.1 Detailed Description

An ID provider for graph objects

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

- [github/dungeon-architect-unity/Scripts/Graph/Graph.cs](#)

5.75 DungeonArchitect.Editors.InspectorUtils Class Reference

Utility functions for drawing UI in the Inspector window

Static Public Member Functions

- static void [DrawMatrixProperty](#) (string caption, ref Matrix4x4 matrix)
Draws the translation / rotation / scale widgets for a Matrix4x4
- static void [RoundVector](#) (ref Vector3 vector, int precision)
Rounds the Vector to the nearest precision
- static float [Round](#) (float f, int precision)
rounds a float to the nearest precision
- static void [DrawVectorProperty](#) (string caption, ref Vector3 vector)
Draws XYZ components of a Vector3 in the inspector window within the same line for better usability and aesthetics

5.75.1 Detailed Description

Utility functions for drawing UI in the Inspector window

5.75.2 Member Function Documentation

5.75.2.1 static void [DungeonArchitect.Editors.InspectorUtils.DrawMatrixProperty](#) (string *caption*, ref Matrix4x4 *matrix*)
[static]

Draws the translation / rotation / scale widgets for a Matrix4x4

Parameters

<i>caption</i>	The caption to display above the widget
<i>matrix</i>	The transform matrix to modify

5.75.2.2 static void [DungeonArchitect.Editors.InspectorUtils.DrawVectorProperty](#) (string *caption*, ref Vector3 *vector*)
[static]

Draws XYZ components of a Vector3 in the inspector window within the same line for better usability and aesthetics

Parameters

<i>caption</i>	The caption to use on the property
<i>vector</i>	The vector to modify

5.75.2.3 static float [DungeonArchitect.Editors.InspectorUtils.Round](#) (float *f*, int *precision*) [static]

rounds a float to the nearest precision

Parameters

<i>f</i>	The value to round
<i>precision</i>	The precision in digits

Returns

5.75.2.4 static void DungeonArchitect.Editors.InspectorUtils.RoundVector (ref Vector3 *vector*, int *precision*) [static]

Rounds the Vector to the nearest precision

Parameters

<i>vector</i>	The vector to round
<i>precision</i>	The precision in digits

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

- github/dungeon-architect-unity/Editor/Utls/InspectorUtils.cs

5.76 DungeonArchitect.Utls.InstanceCache Class Reference

Caches instances by their name so they can be reused when needed again instead of recreating it

Public Member Functions

- object [GetInstance](#) (string typeName)
Retrieves the instance of the specified ScriptableObject type name. If none exists, a new one is created and stored

5.76.1 Detailed Description

Caches instances by their name so they can be reused when needed again instead of recreating it

5.76.2 Member Function Documentation

5.76.2.1 object DungeonArchitect.Utls.InstanceCache.GetInstance (string typeName)

Retrieves the instance of the specified ScriptableObject type name. If none exists, a new one is created and stored

Parameters

<i>typeName</i>	The typename of the ScriptableObject
-----------------	--------------------------------------

Returns

The cached instance of the specified ScriptableObject typename

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

- github/dungeon-architect-unity/Scripts/Utls/InstanceCache.cs

5.77 DungeonArchitect.IntVector Struct Reference

Represent an integer vector

Public Member Functions

- **IntVector** (int x, int y, int z)
- void **Set** (int x, int y, int z)
- float **DistanceSq** ()
- float **Distance** ()
- override bool **Equals** (System.Object obj)
- override int **GetHashCode** ()

Static Public Member Functions

- static **IntVector operator+** (**IntVector** a, **IntVector** b)
- static **IntVector operator-** (**IntVector** a, **IntVector** b)
- static **IntVector operator*** (**IntVector** a, **IntVector** b)
- static **Vector3 operator*** (**IntVector** a, **Vector3** b)
- static **IntVector operator/** (**IntVector** a, **IntVector** b)
- static **IntVector operator+** (**IntVector** a, int b)
- static **IntVector operator-** (**IntVector** a, int b)
- static **IntVector operator*** (**IntVector** a, int b)
- static **IntVector operator/** (**IntVector** a, int b)
- static **Vector3 ToV3** (**IntVector** iv)

Public Attributes

- int **x**
- int **y**
- int **z**

Static Public Attributes

- static readonly **IntVector Zero** = new **IntVector**(0, 0, 0)

5.77.1 Detailed Description

Represent an integer vector

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

- github/dungeon-architect-unity/Scripts/Math/IntVector.cs

5.78 DungeonArchitect.IntVector2Key Struct Reference

Data-structure for **IntVector** pair. Used for caching

Public Member Functions

- **IntVector2Key** ([IntVector](#) a, [IntVector](#) b)
- override bool **Equals** (System.Object obj)
- override int **GetHashCode** ()

Public Attributes

- [IntVector](#) **a**
- [IntVector](#) **b**

5.78.1 Detailed Description

Data-structure for [IntVector](#) pair. Used for caching

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Models/GridDungeonModel.cs](#)

5.79 DungeonArchitect.Editor.KeyboardState Class Reference

Caches the keyboard state

Public Member Functions

- void **SetState** (KeyCode keyCode, bool pressed)
- void **HandleInput** (Event e)
- bool **GetSate** (KeyCode keyCode)

Properties

- bool **ControlPressed** [get]
- bool **ShiftPressed** [get]
- bool **AltPressed** [get]

5.79.1 Detailed Description

Caches the keyboard state

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

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphEditor.cs](#)

5.80 DungeonArchitect.LandscapeDataRasterizer Class Reference

Manages the landscape data and performs various rasterization algorithms (draw cells, lines etc)

Public Member Functions

- [LandscapeDataRasterizer](#) (Terrain terrain, float elevation)
Creates a new instance
- void [LoadData](#) ()
Loads the data from the terrain into memory for modification
- void [DrawCell](#) (float x, float y, float w, float h, float elevation)
Rasterizes the terrain height along the specified world coordinate with the specified elevation height
- void [SmoothCell](#) (float x, float y, float w, float h, float elevation, int smoothingDistance, AnimationCurve smoothingCurve)
Applies a smoothing blur filter based on the user-defined smoothing curve
- void [SaveData](#) ()
Saves the data in memory back into the terrain. This modifies the terrain object

Static Public Member Functions

- static float [GetHeight](#) (Terrain terrain, float worldX, float worldZ)
Gets the height of the terrain at the specified world space
- static void [WorldToTerrainCoord](#) (Terrain terrain, float x, float y, out int gx, out int gy)
Converts the world coordinate to internal terrain coordinate where the data is loaded
- static void [WorldToTerrainTextureCoord](#) (Terrain terrain, float x, float y, out int tx, out int ty)
Converts the world coordinate to terrain texture coordinate

5.80.1 Detailed Description

Manages the landscape data and performs various rasterization algorithms (draw cells, lines etc)

5.80.2 Constructor & Destructor Documentation

5.80.2.1 DungeonArchitect.LandscapeDataRasterizer.LandscapeDataRasterizer (Terrain terrain, float elevation)

Creates a new instance

Parameters

<i>terrain</i>	The terrain object to modify
<i>elevation</i>	The preferred ground level elevation

5.80.3 Member Function Documentation

5.80.3.1 void DungeonArchitect.LandscapeDataRasterizer.DrawCell (float x, float y, float w, float h, float elevation)

Rasterizes the terrain height along the specified world coordinate with the specified elevation height

Parameters

<i>x</i>	x coordinate in world space
<i>y</i>	z coordinate in world space
<i>w</i>	width in world space

<i>h</i>	height in world space
<i>elevation</i>	The elevation to set in the specified bounds

5.80.3.2 static float DungeonArchitect.LandscapeDataRasterizer.GetHeight (Terrain *terrain*, float *worldX*, float *worldZ*)
[static]

Gets the height of the terrain at the specified world space

Parameters

<i>terrain</i>	The terrain object
<i>worldX</i>	X coordinate in world space
<i>worldZ</i>	Z coordinate in world space

Returns

The Y height of the terrain at the specified location

5.80.3.3 void DungeonArchitect.LandscapeDataRasterizer.LoadData ()

Loads the data from the terrain into memory for modification

5.80.3.4 void DungeonArchitect.LandscapeDataRasterizer.SaveData ()

Saves the data in memory back into the terrain. This modifies the terrain object

5.80.3.5 void DungeonArchitect.LandscapeDataRasterizer.SmoothCell (float *x*, float *y*, float *w*, float *h*, float *elevation*, int *smoothingDistance*, AnimationCurve *smoothingCurve*)

Applies a smoothing blur filter based on the user-defined smoothing curve

Parameters

<i>x</i>	x coordinate in world space
<i>y</i>	z coordinate in world space
<i>w</i>	width in world space
<i>h</i>	height in world space
<i>elevation</i>	The elevation to set in the specified bounds
<i>smoothingDistance</i>	The distance to apply the smoothing transition on. For e.g. if the distance is 5, the smoothing would occur over 5 units
<i>smoothingCurve</i>	The user defined curve to control the steepness of cliffs

5.80.3.6 static void DungeonArchitect.LandscapeDataRasterizer.WorldToTerrainCoord (Terrain *terrain*, float *x*, float *y*, out int *gx*, out int *gy*) [static]

Converts the world coordinate to internal terrain coordinate where the data is loaded

Parameters

<i>terrain</i>	The terrain to query
----------------	----------------------

<i>x</i>	x coordinate in world coordinate
<i>y</i>	z coordinate in world coordinate
<i>gx</i>	x coordiante in the 2D terrain height data coordinate
<i>gy</i>	y coordiante in the 2D terrain height data coordinate

5.80.3.7 static void `DungeonArchitect.LandscapeDataRasterizer.WorldToTerrainTextureCoord` (Terrain *terrain*, float *x*, float *y*, out int *tx*, out int *ty*) [static]

Converts the world coordinate to terrain texture coordinate

Parameters

<i>terrain</i>	The terrain to query
<i>x</i>	x coordinate in world coordinate
<i>y</i>	z coordinate in world coordinate
<i>tx</i>	x coordiante in the 2D terrain texture data coordinate
<i>ty</i>	y coordiante in the 2D terrain texture data coordinate

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

- `github/dungeon-architect-unity/Scripts/Dungeon/Landscape/LandscapeDataRasterizer.cs`

5.81 DungeonArchitect.LandscapeTexture Class Reference

Data-structure to hold the texture settings. This contains enough information to paint the texture on to the terrain

Public Attributes

- [LandscapeTextureType](#) **textureType**
- Texture2D **diffuse**
- Texture2D **normal**
- float **metallic** = 0
- Vector2 **size** = new Vector2(15, 15)
- Vector2 **offset** = Vector2.zero

5.81.1 Detailed Description

Data-structure to hold the texture settings. This contains enough information to paint the texture on to the terrain

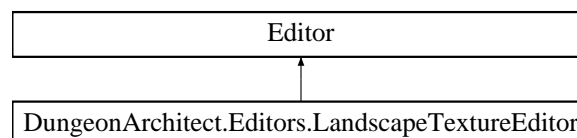
The documentation for this class was generated from the following file:

- `github/dungeon-architect-unity/Scripts/Dungeon/Landscape/LandscapeTransformerGrid.cs`

5.82 DungeonArchitectEditors.LandscapeTextureEditor Class Reference

Custom property editor for the Landscape texture data-structure

Inheritance diagram for `DungeonArchitectEditors.LandscapeTextureEditor`:



Public Member Functions

- void **OnEnable** ()
- override void **OnInspectorGUI** ()

5.82.1 Detailed Description

Custom property editor for the Landscape texture data-structure

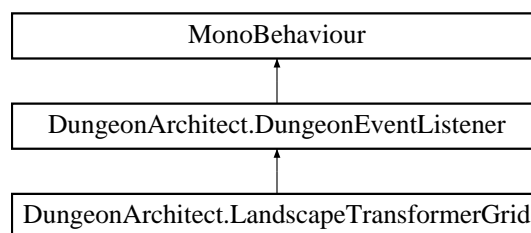
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/Dungeon/Landscape/LandscapeTextureEditor.cs](https://github.com/dungeon-architect-unity/Editor/Dungeon/Landscape/LandscapeTextureEditor.cs)

5.83 DungeonArchitect.LandscapeTransformerGrid Class Reference

The terrain modifier that works with the grid based dungeon builder (DungeonBuilderGrid) It modifies the terrain by adjusting the height around the layout of the dungeon and painting it based on the specified texture settings

Inheritance diagram for DungeonArchitect.LandscapeTransformerGrid:



Public Member Functions

- override void **OnPostDungeonLayoutBuild** ([Dungeon](#) dungeon, [DungeonModel](#) model)
Called after the layout is built in memory, but before the markers are emitted

Public Attributes

- Terrain **terrain**
- [LandscapeTexture](#)[] **textures**
- float [groundLevelHeight](#) = 0
The height of the default ground level
- float **layoutLevelOffset** = 0
- int **smoothingDistance** = 5
- AnimationCurve **roomElevationCurve**
- AnimationCurve **corridorElevationCurve**
- int **roadBlurDistance** = 6
- float **corridorBlurThreshold** = 0.5f

5.83.1 Detailed Description

The terrain modifier that works with the grid based dungeon builder (DungeonBuilderGrid) It modifies the terrain by adjusting the height around the layout of the dungeon and painting it based on the specified texture settings

5.83.2 Member Function Documentation

5.83.2.1 `override void DungeonArchitect.LandscapeTransformerGrid.OnPostDungeonLayoutBuild (Dungeon dungeon,
DungeonModel model) [virtual]`

Called after the layout is built in memory, but before the markers are emitted

Parameters

<i>model</i>	The dungeon model
--------------	-------------------

Reimplemented from [DungeonArchitect.DungeonEventListener](#).

5.83.3 Member Data Documentation

5.83.3.1 float DungeonArchitect.LandscapeTransformerGrid.groundLevelHeight = 0

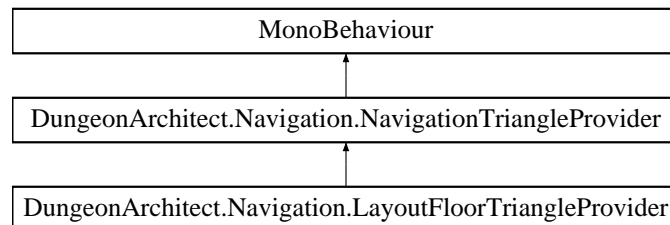
The height of the default ground level

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Landscape/LandscapeTransformerGrid.cs](#)

5.84 DungeonArchitect.Navigation.LayoutFloorTriangleProvider Class Reference

Inheritance diagram for `DungeonArchitect.Navigation.LayoutFloorTriangleProvider`:



Public Member Functions

- override void **AddNavTriangles** (List< Triangle3 > triangles)

Public Attributes

- [Dungeon](#) **dungeon**

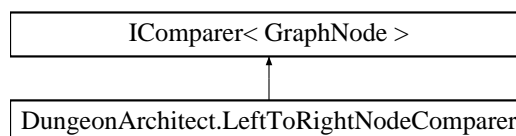
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Navigation/TriangleProviders/LayoutFloorTriangleProvider.cs](#)

5.85 DungeonArchitect.LeftToRightNodeComparer Class Reference

Sorts the nodes from left to right based on the X-axis. This is used for sorting the visual nodes for execution, since they are executed from left to right

Inheritance diagram for `DungeonArchitect.LeftToRightNodeComparer`:



Public Member Functions

- int **Compare** ([GraphNode](#) a, [GraphNode](#) b)

5.85.1 Detailed Description

Sorts the nodes from left to right based on the X-axis. This is used for sorting the visual nodes for execution, since they are executed from left to right

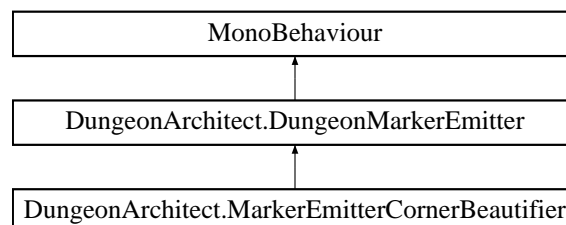
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/DungeonPropDataAsset.cs](#)

5.86 DungeonArchitect.MarkerEmitterCornerBeautifier Class Reference

Emits markers to beautify the level around corners based on the surrounding tiles

Inheritance diagram for `DungeonArchitect.MarkerEmitterCornerBeautifier`:



Public Member Functions

- override void **EmitMarkers** ([DungeonBuilder](#) builder)
Called by the dungeon object right after the dungeon is created

5.86.1 Detailed Description

Emits markers to beautify the level around corners based on the surrounding tiles

5.86.2 Member Function Documentation

- 5.86.2.1 **override void `DungeonArchitect.MarkerEmitterCornerBeautifier.EmitMarkers` (`DungeonBuilder` *builder*)**
[virtual]

Called by the dungeon object right after the dungeon is created

Parameters

<i>builder</i>	reference to the builder object used to build the dungeon
----------------	---

Reimplemented from [DungeonArchitect.DungeonMarkerEmitter](#).

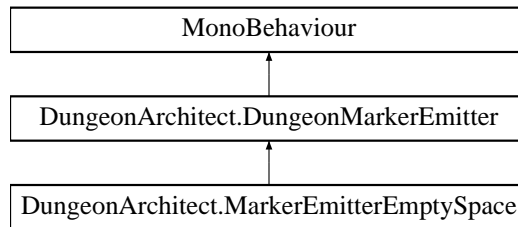
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/MarkerEmitters/Grid/MarkerEmitterCornerBeautifier.cs](#)

5.87 DungeonArchitect.MarkerEmitterEmptySpace Class Reference

Emits markers in the nearby empty space of the dungeon layout

Inheritance diagram for DungeonArchitect.MarkerEmitterEmptySpace:



Public Member Functions

- override void [EmitMarkers](#) ([DungeonBuilder](#) builder)
Called by the dungeon object right after the dungeon is created

Public Attributes

- int **distanceToCover** = 3
- string **markerName** = "EmptySpace"
- string **indexedMarkerNamePrefix** = "EmptySpace_"
- bool **overrideY** = false
- string **overrideYBlackboardKey** = "DungeonLowestY"

5.87.1 Detailed Description

Emits markers in the nearby empty space of the dungeon layout

5.87.2 Member Function Documentation

5.87.2.1 override void [DungeonArchitect.MarkerEmitterEmptySpace.EmitMarkers](#) ([DungeonBuilder](#) builder)
[virtual]

Called by the dungeon object right after the dungeon is created

Parameters

<i>builder</i>	reference to the builder object used to build the dungeon
----------------	---

Reimplemented from [DungeonArchitect.DungeonMarkerEmitter](#).

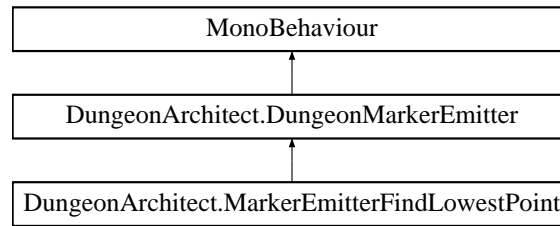
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/MarkerEmitters/Grid/MarkerEmitterEmptySpace.cs](#)

5.88 DungeonArchitect.MarkerEmitterFindLowestPoint Class Reference

Finds the lowest dungeon point and emits a marker at that position. Also sets the scale of the marker to match the width / height

Inheritance diagram for DungeonArchitect.MarkerEmitterFindLowestPoint:



Public Member Functions

- override void [EmitMarkers](#) ([DungeonBuilder](#) builder)
Called by the dungeon object right after the dungeon is created

Public Attributes

- string **MarkerName** = "LowestPoint"
- string **BlackboardKeyLowestY** = "DungeonLowestY"

5.88.1 Detailed Description

Finds the lowest dungeon point and emits a marker at that position. Also sets the scale of the marker to match the width / height

5.88.2 Member Function Documentation

5.88.2.1 override void [DungeonArchitect.MarkerEmitterFindLowestPoint.EmitMarkers](#) ([DungeonBuilder](#) *builder*)
 [virtual]

Called by the dungeon object right after the dungeon is created

Parameters

<i>builder</i>	reference to the builder object used to build the dungeon
----------------	---

Reimplemented from [DungeonArchitect.DungeonMarkerEmitter](#).

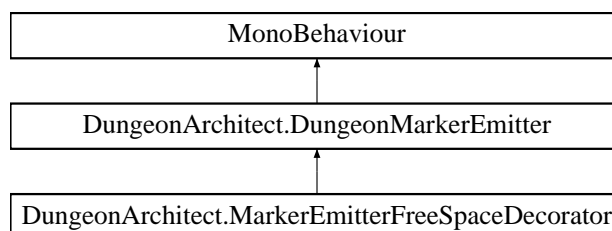
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/MarkerEmitters/Grid/MarkerEmitterFindLowestPoint.cs](#)

5.89 DungeonArchitect.MarkerEmitterFreeSpaceDecorator Class Reference

A more specialized version of the EmptySpace emitter. Emits decorative markers in empty space near the layout

Inheritance diagram for [DungeonArchitect.MarkerEmitterFreeSpaceDecorator](#):



Public Member Functions

- override void [EmitMarkers](#) ([DungeonBuilder](#) builder)
Called by the dungeon object right after the dungeon is created

Public Attributes

- int **distanceFromEdge** = 2
- string **markerName** = "EmptySpaceDecoration"
- float **pushDownAmount** = 6
- Vector3[] **pushDownTestAxis** = new Vector3[0]

5.89.1 Detailed Description

A more specialized version of the EmptySpace emitter. Emits decorative markers in empty space near the layout

5.89.2 Member Function Documentation

5.89.2.1 override void [DungeonArchitect.MarkerEmitterFreeSpaceDecorator.EmitMarkers](#) ([DungeonBuilder](#) *builder*)
[virtual]

Called by the dungeon object right after the dungeon is created

Parameters

<i>builder</i>	reference to the builder object used to build the dungeon
----------------	---

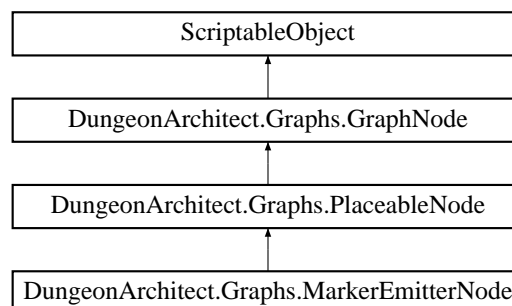
Reimplemented from [DungeonArchitect.DungeonMarkerEmitter](#).

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

- [github/dungeon-architect-unity/Scripts/Dungeon/MarkerEmitters/Grid/MarkerEmitterFreeSpaceDecorator.cs](#)

5.90 DungeonArchitect.Graphs.MarkerEmitterNode Class Reference

Inheritance diagram for [DungeonArchitect.Graphs.MarkerEmitterNode](#):



Public Member Functions

- override void **Initialize** (string id, [Graph](#) graph)

Properties

- [MarkerNode](#) **Marker** [get, set]

Additional Inherited Members

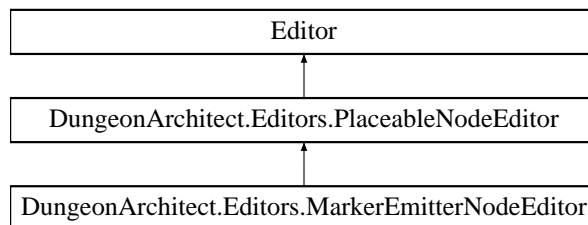
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Graph/ThemeEditor/MarkerEmitterNode.cs](#)

5.91 DungeonArchitect.Editors.MarkerEmitterNodeEditor Class Reference

Custom property editors for MarkerEmitterNode

Inheritance diagram for DungeonArchitect.Editors.MarkerEmitterNodeEditor:



Public Member Functions

- override void **OnEnable** ()

Protected Member Functions

- override void **DrawPreInspectorGUI** ()

Additional Inherited Members

5.91.1 Detailed Description

Custom property editors for MarkerEmitterNode

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

- [github/dungeon-architect-unity/Editor/ThemeEditor/MarkerEmitterNodeEditor.cs](#)

5.92 DungeonArchitect.Editors.MarkerEmitterNodeRenderer Class Reference

Renders a MarkerEmitterNode

Inheritance diagram for DungeonArchitect.Editors.MarkerEmitterNodeRenderer:



Public Member Functions

- override void **Draw** ([GraphNodeRendererContext](#) rendererContext, [GraphNode](#) node, [GraphCamera](#) camera)

Protected Member Functions

- override Color **getBackgroundColor** ([GraphNode](#) node)

5.92.1 Detailed Description

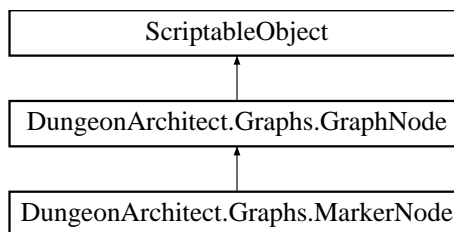
Renders a MarkerEmitterNode

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

- [github/dungeon-architect-unity/Editor/ThemeEditor/MarkerEmitterNodeEditor.cs](#)

5.93 DungeonArchitect.Graphs.MarkerNode Class Reference

Inheritance diagram for DungeonArchitect.Graphs.MarkerNode:



Public Member Functions

- override void **Initialize** (string id, [Graph](#) graph)

Additional Inherited Members

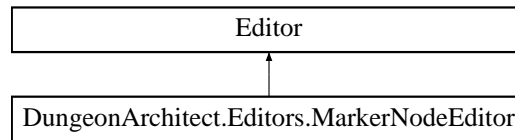
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Graph/ThemeEditor/MarkerNode.cs](#)

5.94 DungeonArchitect.Editors.MarkerNodeEditor Class Reference

Custom property editors for MarkerNode

Inheritance diagram for DungeonArchitect.Editors.MarkerNodeEditor:



Public Member Functions

- void **OnEnable** ()
- override void **OnInspectorGUI** ()

5.94.1 Detailed Description

Custom property editors for MarkerNode

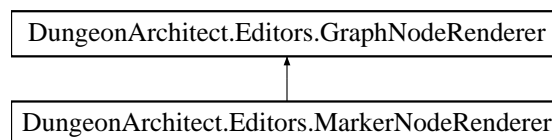
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/ThemeEditor/MarkerNodeEditor.cs](#)

5.95 DungeonArchitect.Editors.MarkerNodeRenderer Class Reference

Renders a marker node

Inheritance diagram for DungeonArchitect.Editors.MarkerNodeRenderer:



Public Member Functions

- override void **Draw** ([GraphNodeRendererContext](#) rendererContext, [GraphNode](#) node, [GraphCamera](#) camera)

Additional Inherited Members

5.95.1 Detailed Description

Renders a marker node

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

- [github/dungeon-architect-unity/Editor/ThemeEditor/MarkerNodeEditor.cs](#)

5.96 DungeonArchitect.Utils.MathUtils Class Reference

Various math utility functions

Static Public Member Functions

- static [Rectangle Copy](#) ([Rectangle](#) other)
Copies the rectangle object
- static [Vector3 Divide](#) ([Vector3](#) a, [Vector3](#) b)
Divides two vector3 objects
- static [Vector3 ToVector3](#) ([IntVector](#) v)
Converts an [IntVector](#) to a [Vector3](#)
- static [IntVector WorldToGrid](#) ([Vector3](#) WorldCoord, [Vector3](#) GridCellSize)
Converts the world coordinates to grid coordinates
- static [Vector3 GridToWorld](#) ([Vector3](#) GridCellSize, [IntVector](#) v)
Converts the grid coordinate to world coordinate
- static [Vector3 GridToWorld](#) ([Vector3](#) GridCellSize, [Vector3](#) v)
Converts the grid coordinate to world coordinate
- static [IntVector ToIntVector](#) ([Vector3](#) v)
Converts an [IntVector](#) to a [Vector3](#), with the XYZ components floored
- static [IntVector RoundToIntVector](#) ([Vector3](#) v)
Rounds to an [IntVector](#), with the XYZ components rounded to the nearest int
- static [Vector3 SnapToGrid](#) ([Vector3](#) position, [Vector3](#) gridCellSize)
Snaps the position to the nearest grid cell location
- static [Vector3 SnapToGrid](#) ([Vector3](#) position, [Vector3](#) gridCellSize, bool useRounding)
Snaps the position to the nearest grid cell location
- static bool [Intersects](#) ([Rect](#) outer, [Rect](#) inner)
Checks if the two rectangles intersect
- static bool [Intersects](#) ([Rect](#) outer, [GraphLink](#) link)
Test if the graph link lies within the rectangle
- static void [FlipYZ](#) (ref [Bounds](#) bounds)
Flips the coordinates for 2D mode
- static [Vector3 FlipYZ](#) ([Vector3](#) bounds)
Flips the coordinates for 2D mode
- static [IntVector FlipYZ](#) ([IntVector](#) bounds)
Flips the coordinates for 2D mode

Static Public Attributes

- static readonly int [INVALID_LOCATION](#) = -1000000
Flag to indicate an invalid location

5.96.1 Detailed Description

Various math utility functions

5.96.2 Member Function Documentation

5.96.2.1 static [Rectangle](#) [DungeonArchitect.Utils.MathUtils.Copy](#) ([Rectangle](#) other) [static]

Copies the rectangle object

Parameters

<i>other</i>	The object to copy
--------------	--------------------

Returns

The copied object

5.96.2.2 static Vector3 DungeonArchitect.Utils.MathUtils.Divide (Vector3 *a*, Vector3 *b*) [static]

Divides two vector3 objects

Parameters

<i>a</i>	
<i>b</i>	

Returns

The divided vector

5.96.2.3 static void DungeonArchitect.Utils.MathUtils.FlipYZ (ref Bounds *bounds*) [static]

Flips the coordinates for 2D mode

Parameters

<i>bounds</i>	Bounds.
---------------	---------

5.96.2.4 static Vector3 DungeonArchitect.Utils.MathUtils.FlipYZ (Vector3 *bounds*) [static]

Flips the coordinates for 2D mode

Returns

The Y.

Parameters

<i>bounds</i>	Bounds.
---------------	---------

5.96.2.5 static IntVector DungeonArchitect.Utils.MathUtils.FlipYZ (IntVector *bounds*) [static]

Flips the coordinates for 2D mode

Returns

The Y.

Parameters

<i>bounds</i>	Bounds.
---------------	---------

5.96.2.6 static Vector3 DungeonArchitect.Utils.MathUtils.GridToWorld (Vector3 *GridCellSize*, IntVector *v*) [static]

Converts the grid coordinate to world coordinate

Parameters

<i>GridCellSize</i>	The grid cell size
<i>v</i>	The input grid coordinate

Returns

The resulting world coordinate

5.96.2.7 static Vector3 DungeonArchitect.Utils.MathUtils.GridToWorld (Vector3 *GridCellSize*, Vector3 *v*) [static]

Converts the grid coordinate to world coordinate

Parameters

<i>GridCellSize</i>	The grid cell size
<i>v</i>	The input grid coordinate

Returns

The resulting world coordinate

5.96.2.8 static bool DungeonArchitect.Utils.MathUtils.Intersects (Rect *outer*, Rect *inner*) [static]

Checks if the two rectangles intersect

Parameters

<i>outer</i>	The outer rect
<i>inner</i>	The inner rect

Returns

True if they intersect, false otherwise

5.96.2.9 static bool DungeonArchitect.Utils.MathUtils.Intersects (Rect *outer*, GraphLink *link*) [static]

Test if the graph link lies within the rectangle

Parameters

<i>outer</i>	The rect to test against
<i>link</i>	The link to test the intersection

Returns

True if intersects, false otherwise

5.96.2.10 `static IntVector DungeonArchitect.Utils.MathUtils.RoundToIntVector (Vector3 v) [static]`

Rounds to an [IntVector](#), with the XYZ components rounded to the nearest int

Parameters

<i>v</i>	The input Vector3 to convert
----------	------------------------------

Returns

The rounded [IntVector](#)

5.96.2.11 `static Vector3 DungeonArchitect.Utils.MathUtils.SnapToGrid (Vector3 position, Vector3 gridCellSize)` `[static]`

Snaps the position to the nearest grid cell location

Parameters

<i>position</i>	The position to snap
<i>gridCellSize</i>	The size of the grid cell

Returns

The snapped position

5.96.2.12 `static Vector3 DungeonArchitect.Utils.MathUtils.SnapToGrid (Vector3 position, Vector3 gridCellSize, bool useRounding)` `[static]`

Snaps the position to the nearest grid cell location

Parameters

<i>position</i>	The position to snap
<i>gridCellSize</i>	The size of the grid cell
<i>useRounding</i>	Flag to indicate if rounding is to be used. Uses floor if false

Returns

The snapped position

5.96.2.13 `static IntVector DungeonArchitect.Utils.MathUtils.ToIntVector (Vector3 v)` `[static]`

Converts an [IntVector](#) to a Vector3, with the XYZ components floored

Parameters

<i>v</i>	The input Vector3 to convert
----------	------------------------------

Returns

The corresponding [IntVector](#), floored in each component

5.96.2.14 `static Vector3 DungeonArchitect.Utils.MathUtils.ToVector3 (IntVector v)` `[static]`

Converts an [IntVector](#) to a Vector3

Parameters

<i>v</i>	the input int vector
----------	----------------------

Returns

5.96.2.15 `static IntVector DungeonArchitect.Utils.MathUtils.WorldToGrid (Vector3 WorldCoord, Vector3 GridCellSize)`
`[static]`

Converts the world coordinates to grid coordinates

Parameters

<i>WorldCoord</i>	The world coordnate
<i>GridCellSize</i>	The grid cell size

Returns

The resulting grid coordinate

5.96.3 Member Data Documentation

5.96.3.1 `readonly int DungeonArchitect.Utils.MathUtils.INVALID_LOCATION = -1000000` `[static]`

Flag to indicate an invalid location

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

- `github/dungeon-architect-unity/Scripts/Utils/MathUtils.cs`

5.97 DungeonArchitect.Utils.Matrix Class Reference

Utility function to extract and put data into a Matrix4x4 object

Static Public Member Functions

- static Vector3 [GetTranslation](#) (ref Matrix4x4 matrix)
Extract translation from transform matrix.
- static void [SetTranslation](#) (ref Matrix4x4 matrix, Vector3 translate)
Sets the translation of the matrix object
- static void [SetTransform](#) (out Matrix4x4 transform, Vector3 position, Quaternion rotation, Vector3 scale)
Sets the transform of a matrix
- static Quaternion [GetRotation](#) (ref Matrix4x4 matrix)
Extract rotation quaternion from transform matrix.
- static Vector3 [GetScale](#) (ref Matrix4x4 matrix)
Extract scale from transform matrix.
- static void [DecomposeMatrix](#) (ref Matrix4x4 matrix, out Vector3 localPosition, out Quaternion localRotation, out Vector3 localScale)
Extract position, rotation and scale from TRS matrix.
- static void [SetTransformFromMatrix](#) (Transform transform, ref Matrix4x4 matrix)
Set transform component from TRS matrix.

- static Matrix4x4 [Identity](#) ()
Identity matrix.
- static Matrix4x4 [TranslationMatrix](#) (Vector3 offset)
Get translation matrix.
- static Matrix4x4 [Copy](#) (Matrix4x4 In)
Copies the matrix object
- static Matrix4x4 [FromGameTransform](#) (Transform t)
Creates a Matrix4x4 object from the game object's transform

Static Public Attributes

- static readonly Quaternion [IdentityQuaternion](#) = Quaternion.identity
Identity quaternion.

5.97.1 Detailed Description

Utility function to extract and put data into a Matrix4x4 object

5.97.2 Member Function Documentation

5.97.2.1 static Matrix4x4 DungeonArchitect.Utls.Matrix.Copy (Matrix4x4 In) [static]

Copies the matrix object

Parameters

<i>In</i>	The matrix object to copy
-----------	---------------------------

Returns

The copied object

5.97.2.2 static void DungeonArchitect.Utls.Matrix.DecomposeMatrix (ref Matrix4x4 matrix, out Vector3 localPosition, out Quaternion localRotation, out Vector3 localScale) [static]

Extract position, rotation and scale from TRS matrix.

Parameters

<i>matrix</i>	Transform matrix. This parameter is passed by reference to improve performance; no changes will be made to it.
<i>localPosition</i>	Output position.
<i>localRotation</i>	Output rotation.
<i>localScale</i>	Output scale.

5.97.2.3 static Matrix4x4 DungeonArchitect.Utls.Matrix.FromGameTransform (Transform t) [static]

Creates a Matrix4x4 object from the game object's transform

Parameters

<i>t</i>	The game object's transform
----------	-----------------------------

Returns

The resulting matrix

5.97.2.4 static Quaternion DungeonArchitect.Utils.Matrix.GetRotation (ref Matrix4x4 *matrix*) [static]

Extract rotation quaternion from transform matrix.

Parameters

<i>matrix</i>	Transform matrix. This parameter is passed by reference to improve performance; no changes will be made to it.
---------------	--

Returns

Quaternion representation of rotation transform.

5.97.2.5 static Vector3 DungeonArchitect.Utils.Matrix.GetScale (ref Matrix4x4 *matrix*) [static]

Extract scale from transform matrix.

Parameters

<i>matrix</i>	Transform matrix. This parameter is passed by reference to improve performance; no changes will be made to it.
---------------	--

Returns

Scale vector.

5.97.2.6 static Vector3 DungeonArchitect.Utils.Matrix.GetTranslation (ref Matrix4x4 *matrix*) [static]

Extract translation from transform matrix.

Parameters

<i>matrix</i>	Transform matrix. This parameter is passed by reference to improve performance; no changes will be made to it.
---------------	--

Returns

Translation offset.

5.97.2.7 static Matrix4x4 DungeonArchitect.Utils.Matrix.Identity () [static]

Identity matrix.

It is faster to access this variation than `Matrix4x4.identity`.

5.97.2.8 static void DungeonArchitect.Utils.Matrix.SetTransform (out Matrix4x4 *transform*, Vector3 *position*, Quaternion *rotation*, Vector3 *scale*) [static]

Sets the transform of a matrix

Parameters

<i>transform</i>	The matrix object to apply the transformation on
<i>position</i>	The position to set
<i>rotation</i>	The rotation to set
<i>scale</i>	The scale ot set

5.97.2.9 `static void DungeonArchitect.Utills.Matrix.SetTransformFromMatrix (Transform transform, ref Matrix4x4 matrix)` `[static]`

Set transform component from TRS matrix.

Parameters

<i>transform</i>	Transform component.
<i>matrix</i>	Transform matrix. This parameter is passed by reference to improve performance; no changes will be made to it.

5.97.2.10 `static void DungeonArchitect.Utills.Matrix.SetTranslation (ref Matrix4x4 matrix, Vector3 translate)` `[static]`

Sets the translation of the matrix object

Parameters

<i>matrix</i>	The matrix to set the translation on
<i>translate</i>	The translation to apply on the matrix

5.97.2.11 `static Matrix4x4 DungeonArchitect.Utills.Matrix.TranslationMatrix (Vector3 offset)` `[static]`

Get translation matrix.

Parameters

<i>offset</i>	Translation offset.
---------------	---------------------

Returns

The translation transform matrix.

5.97.3 Member Data Documentation

5.97.3.1 `readonly Quaternion DungeonArchitect.Utills.Matrix.IdentityQuaternion = Quaternion.identity` `[static]`

Identity quaternion.

It is faster to access this variation than `Quaternion.identity`.

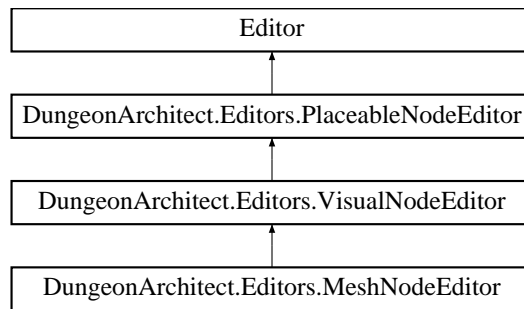
The documentation for this class was generated from the following file:

- `github/dungeon-architect-unity/Scripts/Utills/Matrix.cs`

5.98 DungeonArchitectEditors.MeshNodeEditor Class Reference

Custom property editors for GameObjectNode

Inheritance diagram for `DungeonArchitectEditors.MeshNodeEditor`:



Public Member Functions

- override void **OnEnable** ()

Protected Member Functions

- override void **DrawPreInspectorGUI** ()

Additional Inherited Members

5.98.1 Detailed Description

Custom property editors for GameObjectNode

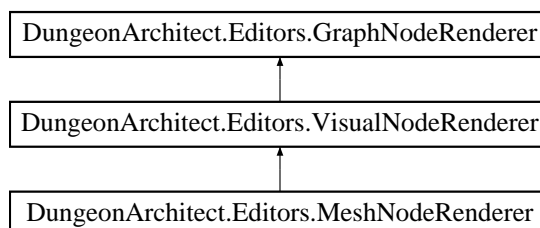
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/ThemeEditor/MeshNodeEditor.cs](#)

5.99 DungeonArchitect.Eeditors.MeshNodeRenderer Class Reference

Renders a mesh node

Inheritance diagram for DungeonArchitect.Eeditors.MeshNodeRenderer:



Protected Member Functions

- override Object **GetThumbObject** ([GraphNode](#) node)

Additional Inherited Members

5.99.1 Detailed Description

Renders a mesh node

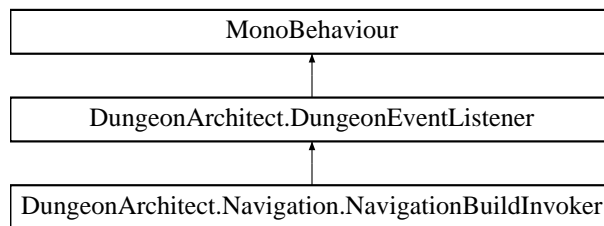
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/ThemeEditor/MeshNodeEditor.cs](#)

5.100 DungeonArchitect.Navigation.NavigationBuildInvoker Class Reference

Drop this script into your dungeon object and assign the nav mesh prefab to automatically rebuild the nav mesh whenever the dungeon is rebuild (works both with runtime and design time)

Inheritance diagram for DungeonArchitect.Navigation.NavigationBuildInvoker:



Public Member Functions

- override void [OnPostDungeonBuild](#) ([Dungeon](#) dungeon, [DungeonModel](#) model)

Called after the dungeon is completely built

Public Attributes

- [DungeonNavMesh](#) **navMesh**

5.100.1 Detailed Description

Drop this script into your dungeon object and assign the nav mesh prefab to automatically rebuild the nav mesh whenever the dungeon is rebuild (works both with runtime and design time)

5.100.2 Member Function Documentation

- 5.100.2.1 override void [DungeonArchitect.Navigation.NavigationBuildInvoker.OnPostDungeonBuild](#) ([Dungeon](#) *dungeon*, [DungeonModel](#) *model*) [virtual]

Called after the dungeon is completely built

Parameters

<i>model</i>	The dungeon model
--------------	-------------------

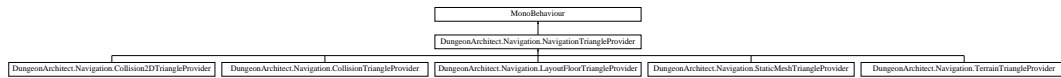
Reimplemented from [DungeonArchitect.DungeonEventListener](#).

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

- [github/dungeon-architect-unity/Scripts/Navigation/NavigationBuildInvoker.cs](#)

5.101 DungeonArchitect.Navigation.NavigationTriangleProvider Class Reference

Inheritance diagram for DungeonArchitect.Navigation.NavigationTriangleProvider:



Public Member Functions

- virtual void **AddNavTriangles** (List< Triangle3 > triangles)

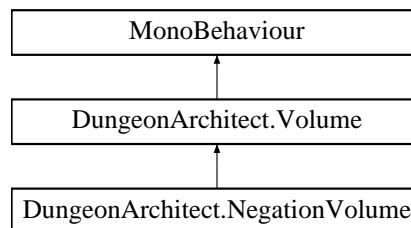
The documentation for this class was generated from the following file:

- github/dungeon-architect-unity/Scripts/Navigation/NavigationTriangleProvider.cs

5.102 DungeonArchitect.NegationVolume Class Reference

Negation volumes remove procedural geometries from the scene that lie with it's bounds

Inheritance diagram for DungeonArchitect.NegationVolume:



Additional Inherited Members

5.102.1 Detailed Description

Negation volumes remove procedural geometries from the scene that lie with it's bounds

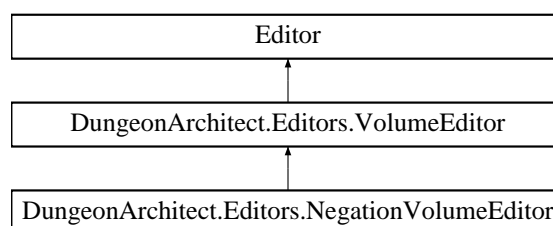
The documentation for this class was generated from the following file:

- github/dungeon-architect-unity/Scripts/Volumes/NegationVolume.cs

5.103 DungeonArchitect.Editors.NegationVolumeEditor Class Reference

Custom property editor for Negation volumes

Inheritance diagram for DungeonArchitect.Editors.NegationVolumeEditor:



Additional Inherited Members

5.103.1 Detailed Description

Custom property editor for Negation volumes

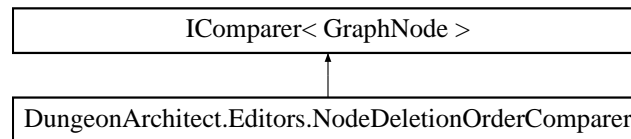
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/Volumes/NegationVolumeEditor.cs](#)

5.104 DungeonArchitect.Editors.NodeDeletionOrderComparer Class Reference

Sorts based on the node's Z-index in decending order

Inheritance diagram for DungeonArchitect.Editors.NodeDeletionOrderComparer:



Public Member Functions

- `int Compare (GraphNode x, GraphNode y)`

5.104.1 Detailed Description

Sorts based on the node's Z-index in decending order

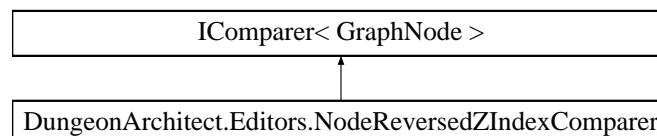
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphEditor.cs](#)

5.105 DungeonArchitect.Editors.NodeReversedZIndexComparer Class Reference

Sorts based on the node's Z-index in decending order

Inheritance diagram for DungeonArchitect.Editors.NodeReversedZIndexComparer:



Public Member Functions

- `int Compare (GraphNode x, GraphNode y)`

5.105.1 Detailed Description

Sorts based on the node's Z-index in decending order

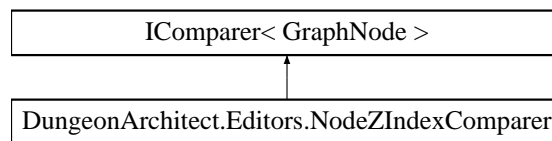
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphEditor.cs](#)

5.106 DungeonArchitect.Editors.NodeZIndexComparer Class Reference

Sorts based on the node's Z-index

Inheritance diagram for DungeonArchitect.Editors.NodeZIndexComparer:



Public Member Functions

- `int Compare (GraphNode x, GraphNode y)`

5.106.1 Detailed Description

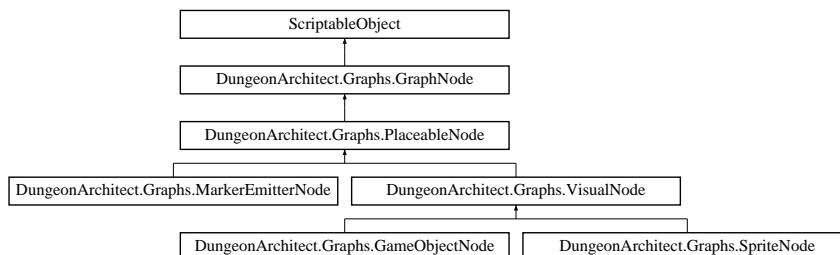
Sorts based on the node's Z-index

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

- [github/dungeon-architect-unity/Editor/GraphEditor/GraphEditor.cs](#)

5.107 DungeonArchitect.Graphs.PlaceableNode Class Reference

Inheritance diagram for DungeonArchitect.Graphs.PlaceableNode:



Public Member Functions

- `override void CopyFrom (GraphNode node)`

Called when the node is copied. The implementations should implement copy here (e.g. deep / shallow copy depending on implementation)

Public Attributes

- Matrix4x4 **offset** = Matrix4x4.identity
- bool **consumeOnAttach** = true
- float **attachmentProbability** = 1.0f

Additional Inherited Members

5.107.1 Member Function Documentation

5.107.1.1 override void DungeonArchitect.Graphs.PlaceableNode.CopyFrom (GraphNode node) [virtual]

Called when the node is copied. The implementations should implement copy here (e.g. deep / shallow copy depending on implementation)

Parameters

<i>node</i>	
-------------	--

Reimplemented from [DungeonArchitect.Graphs.GraphNode](#).

Reimplemented in [DungeonArchitect.Graphs.VisualNode](#), and [DungeonArchitect.Graphs.SpriteNode](#).

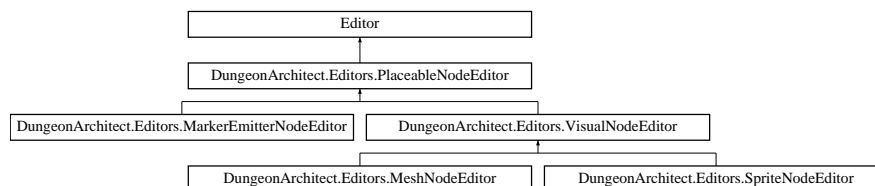
The documentation for this class was generated from the following file:

- github/dungeon-architect-unity/Scripts/Graph/ThemeEditor/PlaceableNode.cs

5.108 DungeonArchitect.Eeditors.PlaceableNodeEditor Class Reference

Custom property editor for placeable node

Inheritance diagram for DungeonArchitect.Eeditors.PlaceableNodeEditor:



Public Member Functions

- virtual void **OnEnable** ()
- override void **OnInspectorGUI** ()

Protected Member Functions

- virtual void **DrawPreInspectorGUI** ()
- virtual void **DrawPostInspectorGUI** ()

Protected Attributes

- SerializedObject **subject**
- bool **drawOffset** = false
- bool **drawAttachments** = false
- const int **CATEGORY_SPACING** = 10

5.108.1 Detailed Description

Custom property editor for placeable node

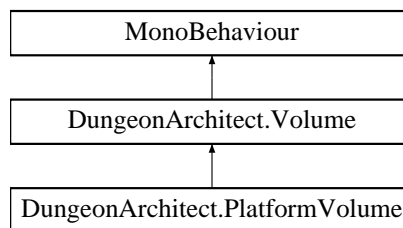
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/ThemeEditor/PlaceableNodeEditor.cs](#)

5.109 DungeonArchitect.PlatformVolume Class Reference

Platform volumes add a platform in the scene encompassing the volume

Inheritance diagram for `DungeonArchitect.PlatformVolume`:



Public Attributes

- [CellType](#) **cellType** = `CellType.Corridor`

Additional Inherited Members

5.109.1 Detailed Description

Platform volumes add a platform in the scene encompassing the volume

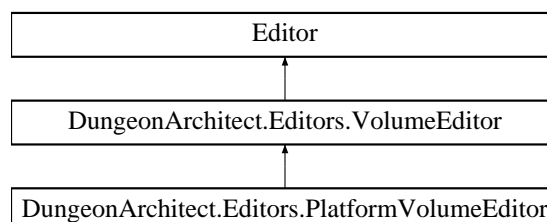
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Volumes/PlatformVolume.cs](#)

5.110 DungeonArchitect.Editors.PlatformVolumeEditor Class Reference

Custom property editor for Platform volumes

Inheritance diagram for `DungeonArchitect.Editors.PlatformVolumeEditor`:



Public Member Functions

- override void **OnUpdate** (`SceneView sceneView`)

Additional Inherited Members

5.110.1 Detailed Description

Custom property editor for Platform volumes

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

- [github/dungeon-architect-unity/Editor/Volumes/PlatformVolumeEditor.cs](#)

5.111 DungeonArchitect.Utls.PMRandom Class Reference

A random stream based on normal distribution. Also support uniform distribution

Public Member Functions

- [PMRandom](#) ()
Creates a new random stream with seed 0
- [PMRandom](#) (uint seed)
Creates a new random stream with the specified seed
- void [Initialize](#) (uint seed)
Initializes the stream with the given seed
- float [NextGaussianFloat](#) ()
Gets the next random number from a uniform distribution
- float [NextGaussianFloat](#) (float mean, float stdDev)
Gets the next random number from a uniform distribution
- float [GetNextUniformFloat](#) ()
Gets a random number from the uniformly distributed stream

Properties

- Random **UniformRandom** [get]

5.111.1 Detailed Description

A random stream based on normal distribution. Also support uniform distribution

5.111.2 Constructor & Destructor Documentation

5.111.2.1 DungeonArchitect.Utls.PMRandom.PMRandom ()

Creates a new random stream with seed 0

5.111.2.2 DungeonArchitect.Utls.PMRandom.PMRandom (uint seed)

Creates a new random stream with the specified seed

Parameters

<i>seed</i>	The seed to initialize the random stream
-------------	--

5.111.3 Member Function Documentation

5.111.3.1 float DungeonArchitect.Utls.PMRandom.GetNextUniformFloat ()

Gets a random number from the uniformly distributed stream

Returns

5.111.3.2 void DungeonArchitect.Utls.PMRandom.Initialize (uint *seed*)

Initializes the stream with the given seed

Parameters

<i>seed</i>	
-------------	--

5.111.3.3 float DungeonArchitect.Utls.PMRandom.NextGaussianFloat ()

Gets the next random number from a uniform distribution

Returns

Random number from a uniform stream

5.111.3.4 float DungeonArchitect.Utls.PMRandom.NextGaussianFloat (float *mean*, float *stdDev*)

Gets the next random number from a uniform distribution

Parameters

<i>mean</i>	The mean used for the normal distribution
<i>stdDev</i>	The standard deviation used for the normal distribution

Returns

The resulting random number from the normal distributed random stream

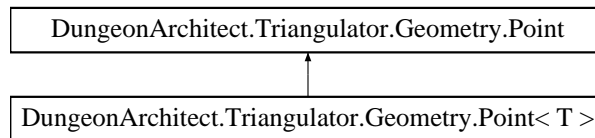
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Utls/PMRandom.cs](#)

5.112 DungeonArchitect.Triangulator.Geometry.Point Class Reference

2D [Point](#) with double precision

Inheritance diagram for DungeonArchitect.Triangulator.Geometry.Point:



5.112.1 Detailed Description

2D [Point](#) with double precision

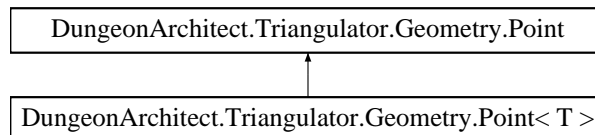
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Triangulator/Geometry/Point.cs](#)

5.113 DungeonArchitect.Triangulator.Geometry.Point Class Reference

2D [Point](#) with double precision

Inheritance diagram for `DungeonArchitect.Triangulator.Geometry.Point`:



5.113.1 Detailed Description

2D [Point](#) with double precision

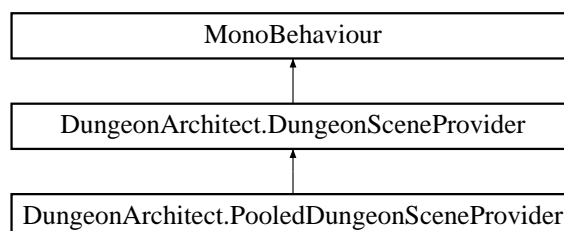
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Triangulator/Geometry/Point.cs](#)

5.114 DungeonArchitect.PooledDungeonSceneProvider Class Reference

Implementation of the Scene provider that adds object pooling over the existing functionality. This is useful for quick rebuilding and better performance, as object in the scene are reused while rebuilding, instead of destroying everything and rebuilding

Inheritance diagram for `DungeonArchitect.PooledDungeonSceneProvider`:



Public Member Functions

- override void [OnDungeonBuildStart](#) ()
Called when build is started
- override void [OnDungeonBuildStop](#) ()
Called after build has ended
- override void [AddSprite](#) ([SpritePropTypeData](#) spriteProp, Matrix4x4 transform)
Request the creation of a sprite object
- override void [AddGameObject](#) ([GameObjectPropTypeData](#) gameObjectProp, Matrix4x4 transform)
Request the creation of a game object

Additional Inherited Members

5.114.1 Detailed Description

Implementation of the Scene provider that adds object pooling over the existing functionality. This is useful for quick rebuilding and better performance, as object in the scene are reused while rebuilding, instead of destroying everything and rebuilding

5.114.2 Member Function Documentation

- 5.114.2.1 **override void [DungeonArchitect.PooledDungeonSceneProvider.AddGameObject](#) ([GameObjectPropTypeData](#) *gameObjectProp*, Matrix4x4 *transform*)** [virtual]

Request the creation of a game object

Parameters

<i>gameObjectProp</i>	The template to use for instantiation
<i>transform</i>	The transform of the instantiated game object

Reimplemented from [DungeonArchitect.DungeonSceneProvider](#).

- 5.114.2.2 **override void [DungeonArchitect.PooledDungeonSceneProvider.AddSprite](#) ([SpritePropTypeData](#) *spriteProp*, Matrix4x4 *transform*)** [virtual]

Request the creation of a sprite object

Parameters

<i>spriteProp</i>	The sprite game object template reference
<i>transform</i>	The transform of the prop

Reimplemented from [DungeonArchitect.DungeonSceneProvider](#).

- 5.114.2.3 **override void [DungeonArchitect.PooledDungeonSceneProvider.OnDungeonBuildStart](#) ()** [virtual]

Called when build is started

Reimplemented from [DungeonArchitect.DungeonSceneProvider](#).

- 5.114.2.4 **override void [DungeonArchitect.PooledDungeonSceneProvider.OnDungeonBuildStop](#) ()** [virtual]

Called after build has ended

Reimplemented from [DungeonArchitect.DungeonSceneProvider](#).

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

- [github/dungeon-architect-unity/Scripts/Dungeon/SceneProviders/PooledDungeonSceneProvider.cs](#)

5.115 DungeonArchitect.PropChildSocketData Class Reference

Props can emit new sockets when they are inserted, to add more child props relative to them

Public Attributes

- string **SocketType**
- Matrix4x4 **Offset**

5.115.1 Detailed Description

Props can emit new sockets when they are inserted, to add more child props relative to them

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

- [github/dungeon-architect-unity/Scripts/Dungeon/DungeonProp.cs](#)

5.116 DungeonArchitect.PropSocket Class Reference

The data structure for a marker

Public Attributes

- int **Id**
- string **SocketType**
- Matrix4x4 **Transform**
- bool **IsConsumed**
- [IntVector](#) **gridPosition**
- int **cellId**

5.116.1 Detailed Description

The data structure for a marker

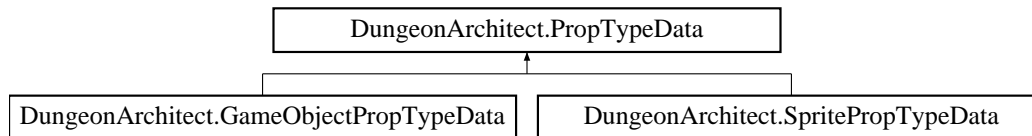
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Dungeon/DungeonProp.cs](#)

5.117 DungeonArchitect.PropTypeData Class Reference

The data structure to hold information about a single node in the asset file

Inheritance diagram for `DungeonArchitect.PropTypeData`:



Public Attributes

- string [Nodeld](#)
The unique guid of the node that generated this prop
- string [AttachToSocket](#)
The socket to attach to
- float [Affinity](#)
The probability of attachment
- bool [ConsumeOnAttach](#)
Should this prop consume the node (i.e. stop further processing of the sibling nodes)
- Matrix4x4 [Offset](#)
The offset to apply from the node's marker position
- List< [PropChildSocketData](#) > [ChildSockets](#) = new List<[PropChildSocketData](#)>()
The child socket markers emitted from this node
- bool [IsStaticObject](#)
Indicates if the object's static flag is to be set
- bool [affectsNavigation](#)
Flag to indicate if this node's geometry affects the navmesh
- bool [UseSelectionRule](#)
Flag to indicate if a selection rule script is used to determine if this prop is selected for insertion
- string [SelectorRuleClassName](#)
The script to to determine if this prop is selected for insertion. This holds the class of type [SelectorRule](#)
- bool [UseTransformRule](#)
Flag to indicate if a transformation rule script is used to determine the transform offset while inserting this mesh
- string [TransformRuleClassName](#)
The script that calculates the transform offset to be used when inserting this mesh. This holds a class of type [TransformationRule](#)

5.117.1 Detailed Description

The data structure to hold information about a single node in the asset file

5.117.2 Member Data Documentation

5.117.2.1 bool `DungeonArchitect.PropTypeData.affectsNavigation`

Flag to indicate if this node's geometry affects the navmesh

5.117.2.2 float `DungeonArchitect.PropTypeData.Affinity`

The probability of attachment

5.117.2.3 string `DungeonArchitect.PropTypeData.AttachToSocket`

The socket to attach to

5.117.2.4 `List<PropChildSocketData> DungeonArchitect.PropTypeData.ChildSockets = new List<PropChildSocketData>()`

The child socket markers emitted from this node

5.117.2.5 `bool DungeonArchitect.PropTypeData.ConsumeOnAttach`

Should this prop consume the node (i.e. stop further processing of the sibling nodes)

5.117.2.6 `bool DungeonArchitect.PropTypeData.IsStaticObject`

Indicates if the object's static flag is to be set

5.117.2.7 `string DungeonArchitect.PropTypeData.NodeId`

The unique guid of the node that generated this prop

5.117.2.8 `Matrix4x4 DungeonArchitect.PropTypeData.Offset`

The offset to apply from the node's marker position

5.117.2.9 `string DungeonArchitect.PropTypeData.SelectorRuleClassName`

The script to determine if this prop is selected for insertion. This holds the class of type [SelectorRule](#)

5.117.2.10 `string DungeonArchitect.PropTypeData.TransformRuleClassName`

The script that calculates the transform offset to be used when inserting this mesh. This holds a class of type [TransformationRule](#)

5.117.2.11 `bool DungeonArchitect.PropTypeData.UseSelectionRule`

Flag to indicate if a selection rule script is used to determine if this prop is selected for insertion

5.117.2.12 `bool DungeonArchitect.PropTypeData.UseTransformRule`

Flag to indicate if a transformation rule script is used to determine the transform offset while inserting this mesh

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

- `github/dungeon-architect-unity/Scripts/Dungeon/DungeonProp.cs`

5.118 DungeonArchitect.Rectangle Struct Reference

Represents an integer rectangle

Public Member Functions

- **Rectangle** (int x, int z, int width, int length)
- **Rectangle** ([IntVector](#) location, [IntVector](#) size)
- void **SetY** (int y)
- [IntVector](#) **Center** ()
- Vector3 **CenterF** ()
- bool **Contains** ([Rectangle](#) rect)
- bool **Contains** ([IntVector](#) Point)
- bool **Contains** (int x, int z)
- bool **IntersectsWith** ([Rectangle](#) rect)
- [IntVector](#)[] **GetBorderPoints** ()

Static Public Member Functions

- static Vector3 **ToVector3** ([IntVector](#) iv)
- static [Rectangle](#) **Intersect** ([Rectangle](#) a, [Rectangle](#) b)
- static [Rectangle](#) **ExpandBounds** ([Rectangle](#) rect, int distance)

Public Attributes

- [IntVector](#) **location**
- [IntVector](#) **size**

Properties

- [IntVector](#) **Location** [get, set]
- [IntVector](#) **Size** [get, set]
- int **X** [get]
- int **Z** [get]
- int **Width** [get]
- int **Length** [get]
- int **Left** [get]
- int **Right** [get]
- int **Back** [get]
- int **Front** [get]

5.118.1 Detailed Description

Represents an integer rectangle

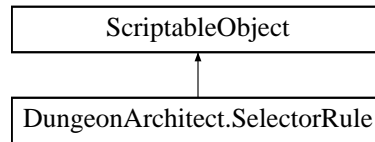
The documentation for this struct was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Math/Rectangle.cs](#)

5.119 DungeonArchitect.SelectorRule Class Reference

Selector rule allow you to attach selection behavior to decide if a visual node should be inserted into the scene

Inheritance diagram for [DungeonArchitect.SelectorRule](#):



Public Member Functions

- virtual bool [CanSelect](#) ([PropSocket](#) socket, Matrix4x4 propTransform, [DungeonModel](#) model, System.Random random)

Implementations should override this and determine if the node should be selected (inserted into the scene)

5.119.1 Detailed Description

Selector rule allow you to attach selection behavior to decide if a visual node should be inserted into the scene

5.119.2 Member Function Documentation

- 5.119.2.1 virtual bool `DungeonArchitect.SelectorRule.CanSelect (PropSocket socket, Matrix4x4 propTransform, DungeonModel model, System.Random random)` `[virtual]`

Implementations should override this and determine if the node should be selected (inserted into the scene)

Parameters

<i>socket</i>	The marker data-structure
<i>propTransform</i>	The combined transform of the visual node that invoked this rule
<i>model</i>	The dungeon model
<i>random</i>	The random stream used by the builder. User this random stream for any randomness for consistency

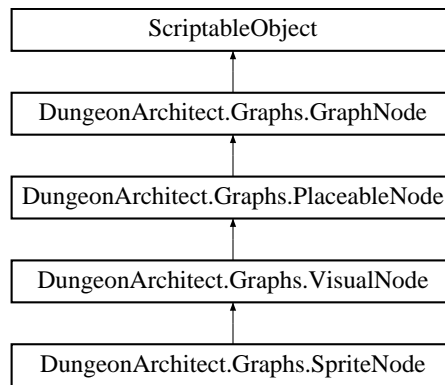
Returns

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

- github/dungeon-architect-unity/Scripts/Dungeon/Rules/SelectorRule.cs

5.120 DungeonArchitect.Graphs.SpriteNode Class Reference

Inheritance diagram for `DungeonArchitect.Graphs.SpriteNode`:



Public Member Functions

- override void **Initialize** (string id, [Graph](#) graph)
- override void **CopyFrom** ([GraphNode](#) node)

Called when the node is copied. The implementations should implement copy here (e.g. deep / shallow copy depending on implementation)

Public Attributes

- Sprite **sprite**
- Color **color** = new Color(1, 1, 1, 1)
- Material **materialOverride**
- string **sortingLayerName**
- int **orderInLayer**
- DungeonSpriteCollisionType **collisionType** = DungeonSpriteCollisionType.None
- PhysicsMaterial2D **physicsMaterial**
- Vector2 **physicsOffset** = Vector2.zero
- Vector2 **physicsSize** = Vector2.one
- float **physicsRadius** = 0.5f

Additional Inherited Members

5.120.1 Member Function Documentation

5.120.1.1 override void DungeonArchitect.Graphs.SpriteNode.CopyFrom ([GraphNode](#) *node*) [virtual]

Called when the node is copied. The implementations should implement copy here (e.g. deep / shallow copy depending on implementation)

Parameters

<i>node</i>	
-------------	--

Reimplemented from [DungeonArchitect.Graphs.PlaceableNode](#).

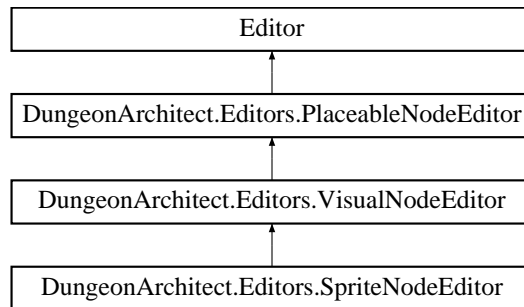
The documentation for this class was generated from the following file:

- github/dungeon-architect-unity/Scripts/Graph/ThemeEditor/SpriteNode.cs

5.121 DungeonArchitect.Editors.SpriteNodeEditor Class Reference

Custom property editor for a sprite node

Inheritance diagram for DungeonArchitect.Editors.SpriteNodeEditor:



Public Member Functions

- override void **OnEnable** ()

Protected Member Functions

- override void **DrawPreInspectorGUI** ()
- override void **DrawPostInspectorGUI** ()

Additional Inherited Members

5.121.1 Detailed Description

Custom property editor for a sprite node

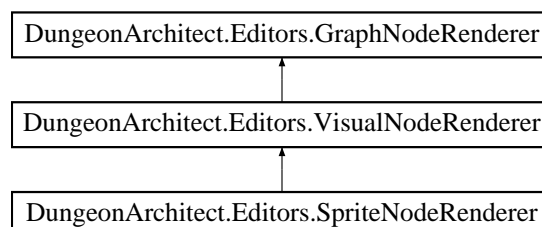
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/ThemeEditor/SpriteNodeEditor.cs](#)

5.122 DungeonArchitect.Editors.SpriteNodeRenderer Class Reference

Renders a sprite node

Inheritance diagram for DungeonArchitect.Editors.SpriteNodeRenderer:



Protected Member Functions

- override Object **GetThumbObject** ([GraphNode](#) node)

Additional Inherited Members

5.122.1 Detailed Description

Renders a sprite node

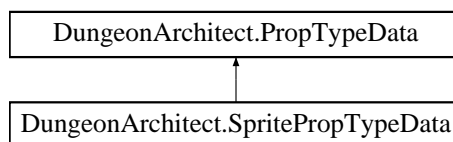
The documentation for this class was generated from the following file:

- `github/dungeon-architect-unity/Editor/ThemeEditor/SpriteNodeEditor.cs`

5.123 DungeonArchitect.SpritePropTypeData Class Reference

Sprite node data asset attributes

Inheritance diagram for `DungeonArchitect.SpritePropTypeData`:



Public Attributes

- Sprite **sprite**
- Color **color**
- Material **materialOverride**
- string **sortingLayerName**
- int **orderInLayer**
- `DungeonSpriteCollisionType` **collisionType**
- `PhysicsMaterial2D` **physicsMaterial**
- `Vector2` **physicsOffset**
- `Vector2` **physicsSize**
- float **physicsRadius**

5.123.1 Detailed Description

Sprite node data asset attributes

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

- `github/dungeon-architect-unity/Scripts/Dungeon/DungeonProp.cs`

5.124 DungeonArchitect.StairAdjacencyQueueNode Class Reference

Temporary data-structure used while assigning stairs on the dungeon.

Public Member Functions

- **StairAdjacencyQueueNode** (int pCellId, int pDepth)

Public Attributes

- int **cellId**
- int **depth**

5.124.1 Detailed Description

Temporary data-structure used while assigning stairs on the dungeon.

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Builders/GridDungeonBuilder.cs](#)

5.125 DungeonArchitect.StairEdgeInfo Struct Reference

Data structure to hold the adjacent cells connected to the stairs (entry / exit)

Public Member Functions

- **StairEdgeInfo** (int pCellIdA, int pCellIdB)

Public Attributes

- int **CellIdA**
- int **CellIdB**

5.125.1 Detailed Description

Data structure to hold the adjacent cells connected to the stairs (entry / exit)

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Builders/GridDungeonBuilder.cs](#)

5.126 DungeonArchitect.StairInfo Class Reference

Data-structure to hold the stair information in the grid based builder

Public Attributes

- int **OwnerCell**
- int **ConnectedToCell**
- Vector3 **Position**
- Quaternion **Rotation**
- [IntVector](#) **IPosition**

5.126.1 Detailed Description

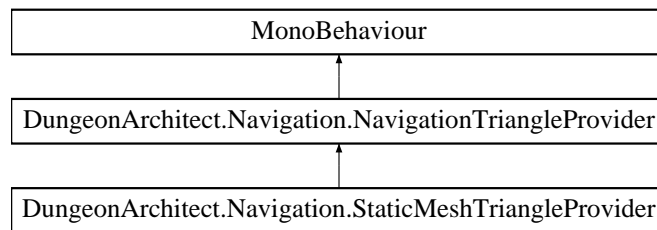
Data-structure to hold the stair information in the grid based builder

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

- [github/dungeon-architect-unity/Scripts/Dungeon/Models/GridDungeonModel.cs](#)

5.127 DungeonArchitect.Navigation.StaticMeshTriangleProvider Class Reference

Inheritance diagram for DungeonArchitect.Navigation.StaticMeshTriangleProvider:



Public Member Functions

- override void **AddNavTriangles** (List< Triangle3 > triangles)

Static Public Member Functions

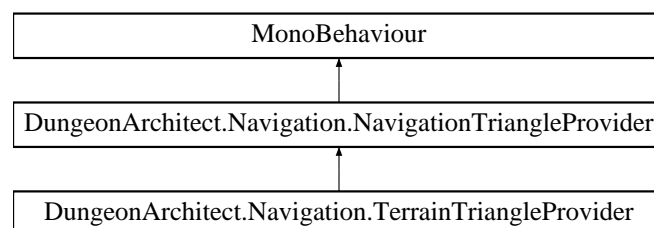
- static void **AddMeshTriangles** (List< Triangle3 > triangles, Mesh mesh, Matrix4x4 transform)
- static void **AddMeshTriangles** (List< Triangle3 > triangles, Vector3[] vertices, int[] indices, Matrix4x4 transform)

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

- [github/dungeon-architect-unity/Scripts/Navigation/TriangleProviders/StaticMeshTriangleProvider.cs](#)

5.128 DungeonArchitect.Navigation.TerrainTriangleProvider Class Reference

Inheritance diagram for DungeonArchitect.Navigation.TerrainTriangleProvider:



Public Member Functions

- override void **AddNavTriangles** (List< Triangle3 > triangles)

Public Attributes

- float **terrainCellSize** = 2

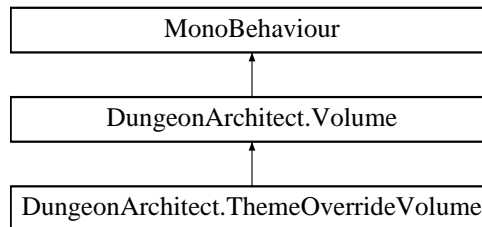
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Navigation/TriangleProviders/TerrainTriangleProvider.cs](#)

5.129 DungeonArchitect.ThemeOverrideVolume Class Reference

[Dungeon](#) layout that lies within this volumes bounds picks up the theme set in this volume

Inheritance diagram for DungeonArchitect.ThemeOverrideVolume:



Public Attributes

- [Graph](#) **overrideTheme**

Additional Inherited Members

5.129.1 Detailed Description

[Dungeon](#) layout that lies within this volumes bounds picks up the theme set in this volume

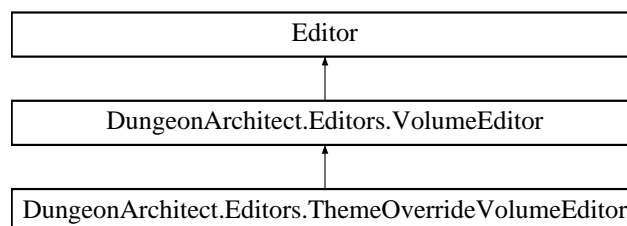
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Volumes/ThemeOverrideVolume.cs](#)

5.130 DungeonArchitect.Editors.ThemeOverrideVolumeEditor Class Reference

Custom property editor for Theme override volumes

Inheritance diagram for DungeonArchitect.Editors.ThemeOverrideVolumeEditor:



Additional Inherited Members

5.130.1 Detailed Description

Custom property editor for Theme override volumes

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

- [github/dungeon-architect-unity/Editor/Volumes/ThemeOverrideVolumeEditor.cs](#)

5.131 DungeonArchitect.Editors.Timer Class Reference

Ticks every few milli-seconds

Public Member Functions

- delegate void **OnTick** (float elapsedTime)
- void **Update** (float deltaSeconds)

Update should be called once per frame

Properties

- float **Hertz** [get, set]

Ticks per second

Events

- OnTick **Tick**

5.131.1 Detailed Description

Ticks every few milli-seconds

5.131.2 Member Function Documentation

5.131.2.1 void DungeonArchitect.Editors.Timer.Update (float *deltaSeconds*)

Update should be called once per frame

Parameters

<i>deltaSeconds</i>	The frame time between calls
---------------------	------------------------------

5.131.3 Property Documentation

5.131.3.1 float DungeonArchitect.Editors.Timer.Hertz [get], [set]

Ticks per second

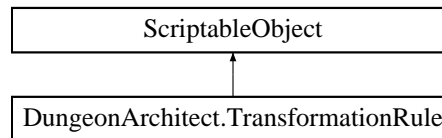
The documentation for this class was generated from the following file:

- github/dungeon-architect-unity/Editor/Utils/Timer.cs

5.132 DungeonArchitect.TransformationRule Class Reference

Selector rule allow you to attach selection behavior to decide if a visual node should be inserted into the scene

Inheritance diagram for DungeonArchitect.TransformationRule:



Public Member Functions

- virtual void [GetTransform](#) ([PropSocket](#) socket, [DungeonModel](#) model, Matrix4x4 propTransform, System.Random random, out Vector3 outPosition, out Quaternion outRotation, out Vector3 outScale)
Implement this method to provide a transform based on your logic.

5.132.1 Detailed Description

Selector rule allow you to attach selection behavior to decide if a visual node should be inserted into the scene

5.132.2 Member Function Documentation

- 5.132.2.1 virtual void DungeonArchitect.TransformationRule.GetTransform ([PropSocket](#) *socket*, [DungeonModel](#) *model*, Matrix4x4 *propTransform*, System.Random *random*, out Vector3 *outPosition*, out Quaternion *outRotation*, out Vector3 *outScale*) [virtual]

Implement this method to provide a transform based on your logic.

Parameters

<i>socket</i>	The marker data structure
<i>model</i>	The dungeon model
<i>propTransform</i>	The combined transform of the visual node that invoked this rule
<i>random</i>	The random stream used by the builder. Use this random stream for any randomness for consistency
<i>outPosition</i>	Set your position offset here
<i>outRotation</i>	Set your rotation offset here
<i>outScale</i>	Set your scale offset here

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

- github/dungeon-architect-unity/Scripts/Dungeon/Rules/TransformationRule.cs

5.133 DungeonArchitect.Triangulator.Geometry.Triangle Struct Reference

[Triangle](#) made from three point indexes

Public Member Functions

- [Triangle](#) (int point1, int point2, int point3)
Initializes a new instance of a triangle

Public Attributes

- int [p1](#)
First vertex index in triangle

- int [p2](#)

Second vertex index in triangle

- int [p3](#)

Third vertex index in triangle

5.133.1 Detailed Description

[Triangle](#) made from three point indexes

5.133.2 Constructor & Destructor Documentation

5.133.2.1 `DungeonArchitect.Triangulator.Geometry.Triangle.Triangle (int point1, int point2, int point3)`

Initializes a new instance of a triangle

Parameters

<i>point1</i>	Vertex 1
<i>point2</i>	Vertex 2
<i>point3</i>	Vertex 3

5.133.3 Member Data Documentation

5.133.3.1 `int DungeonArchitect.Triangulator.Geometry.Triangle.p1`

First vertex index in triangle

5.133.3.2 `int DungeonArchitect.Triangulator.Geometry.Triangle.p2`

Second vertex index in triangle

5.133.3.3 `int DungeonArchitect.Triangulator.Geometry.Triangle.p3`

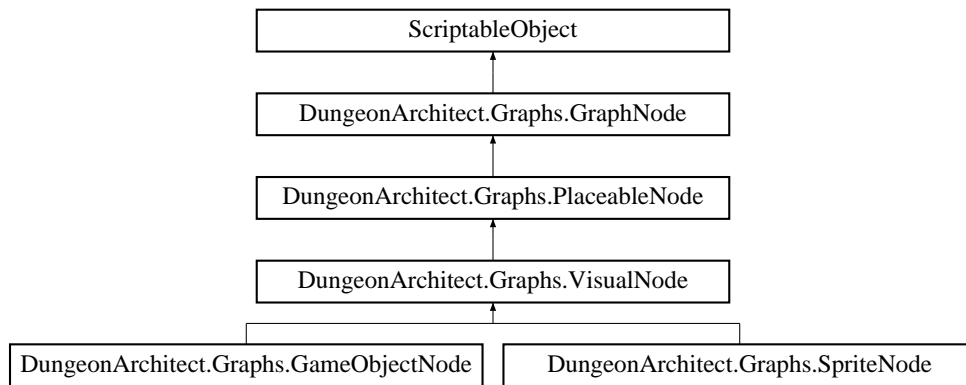
Third vertex index in triangle

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

- `github/dungeon-architect-unity/Scripts/Triangulator/Geometry/Triangle.cs`

5.134 DungeonArchitect.Graphs.VisualNode Class Reference

Inheritance diagram for `DungeonArchitect.Graphs.VisualNode`:



Public Member Functions

- override void **Initialize** (string id, [Graph](#) graph)
- override void **CopyFrom** ([GraphNode](#) node)

Called when the node is copied. The implementations should implement copy here (e.g. deep / shallow copy depending on implementation)

Public Attributes

- bool **IsStatic** = true
Indicates if the game object created from this visual node is set to static. If you are spawning NPCs or other dynamic objects, uncheck this.
- bool **affectsNavigation** = false
Indicates if the geometry in this node contributes to the navigation mesh. You should enable this only if necessary to improve navmesh generation performance.
- bool **selectionRuleEnabled** = false
Indicates if the selection rule is enabled. The selection rule will not run if this is disabled.
- string **selectionRuleClassName**
The class name of the selection rule. Selection rules let you specify behavior logic for selecting your nodes.
- bool **transformRuleEnabled** = false
Indicates if the transform rule is enabled. The transform rule will not run if this is disabled.
- string **transformRuleClassName**
The class name of the transformation rule. Transform rules let you specify behavior logic to apply the offset on the nodes.

Additional Inherited Members

5.134.1 Member Function Documentation

5.134.1.1 override void DungeonArchitect.Graphs.VisualNode.CopyFrom ([GraphNode](#) node) [virtual]

Called when the node is copied. The implementations should implement copy here (e.g. deep / shallow copy depending on implementation)

Parameters

<i>node</i>	
-------------	--

Reimplemented from [DungeonArchitect.Graphs.PlaceableNode](#).

5.134.2 Member Data Documentation

5.134.2.1 `bool DungeonArchitect.Graphs.VisualNode.affectsNavigation = false`

Indicates if the geometry in this node contributes to the navigation mesh. You should enable this only if necessary to improve navmesh generation performance.

5.134.2.2 `bool DungeonArchitect.Graphs.VisualNode.IsStatic = true`

Indicates if the game object created from this visual node is set to static. If you are spawning NPCs or other dynamic objects, uncheck this.

5.134.2.3 `string DungeonArchitect.Graphs.VisualNode.selectionRuleClassName`

The class name of the selection rule. Selection rules let you specify behavior logic for selecting your nodes.

5.134.2.4 `bool DungeonArchitect.Graphs.VisualNode.selectionRuleEnabled = false`

Indicates if the selection rule is enabled. The selection rule will not run if this is disabled.

5.134.2.5 `string DungeonArchitect.Graphs.VisualNode.transformRuleClassName`

The class name of the transformation rule. Transform rules let you specify behavior logic to apply the offset on the nodes.

5.134.2.6 `bool DungeonArchitect.Graphs.VisualNode.transformRuleEnabled = false`

Indicates if the transform rule is enabled. The transform rule will not run if this is disabled.

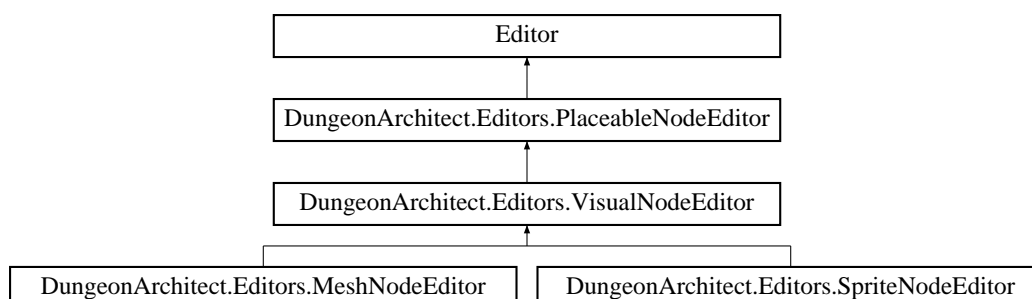
The documentation for this class was generated from the following file:

- `github/dungeon-architect-unity/Scripts/Graph/ThemeEditor/VisualNode.cs`

5.135 `DungeonArchitect.Editors.VisualNodeEditor` Class Reference

Custom property editor for visual nodes.

Inheritance diagram for `DungeonArchitect.Editors.VisualNodeEditor`:



Public Member Functions

- override void **OnEnable** ()

Protected Member Functions

- override void **DrawPreInspectorGUI** ()
- override void **DrawPostInspectorGUI** ()

Additional Inherited Members

5.135.1 Detailed Description

Custom property editor for visual nodes

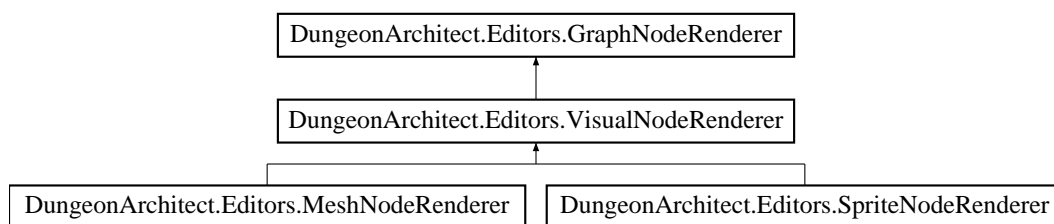
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/ThemeEditor/VisualNodeEditor.cs](#)

5.136 DungeonArchitect.Eeditors.VisualNodeRenderer Class Reference

Renders a visual node

Inheritance diagram for DungeonArchitect.Eeditors.VisualNodeRenderer:



Public Member Functions

- override void **Draw** ([GraphNodeRendererContext](#) rendererContext, [GraphNode](#) node, [GraphCamera](#) camera)

Protected Member Functions

- abstract Object **GetThumbObject** ([GraphNode](#) node)

5.136.1 Detailed Description

Renders a visual node

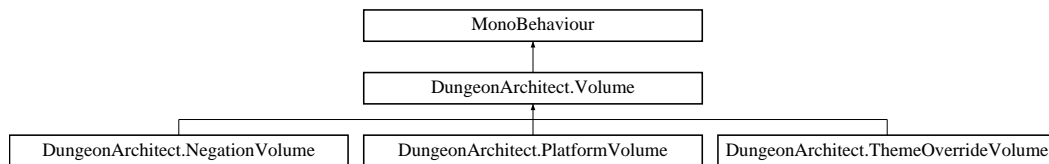
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Editor/ThemeEditor/VisualNodeEditor.cs](#)

5.137 DungeonArchitect.Volume Class Reference

A volume is an abstract representation of space in the world. A volume can be scaled and moved around like any other game object and custom functionality can be added to volumes to influence the dungeon with it's spatial volume

Inheritance diagram for DungeonArchitect.Volume:



Public Member Functions

- Bounds [GetBounds](#) ()
Gets the bounds of the volume
- void [GetGridTransform](#) (out [IntVector](#) positionGrid, out [IntVector](#) scaleGrid)
Gets the position and scale of the volume in grid space

Public Attributes

- [Dungeon](#) **dungeon**
- bool **mode2D** = false

Protected Attributes

- Color **COLOR_WIRE** = Color.yellow
- Color **COLOR_SOLID_DESELECTED** = new Color(1, 1, 0, 0.0f)
- Color **COLOR_SOLID** = new Color(1, 1, 0, 0.1f)

5.137.1 Detailed Description

A volume is an abstract representation of space in the world. A volume can be scaled and moved around like any other game object and custom functionality can be added to volumes to influence the dungeon with it's spatial volume

5.137.2 Member Function Documentation

5.137.2.1 Bounds [DungeonArchitect.Volume.GetBounds](#) ()

Gets the bounds of the volume

Returns

The bounds of the dungeon

5.137.2.2 void [DungeonArchitect.Volume.GetGridTransform](#) (out [IntVector](#) *positionGrid*, out [IntVector](#) *scaleGrid*)

Gets the position and scale of the volume in grid space

Parameters

<i>positionGrid</i>	The grid position (out)
---------------------	-------------------------

<i>scaleGrid</i>	The grid scale (out)
------------------	----------------------

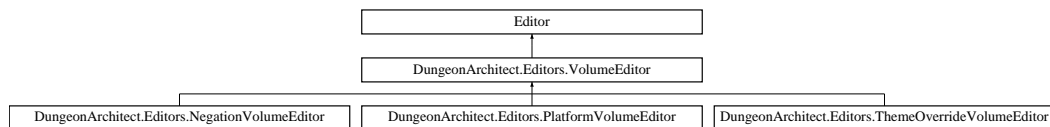
The documentation for this class was generated from the following file:

- [github/dungeon-architect-unity/Scripts/Volumes/Volume.cs](#)

5.138 DungeonArchitect.Editors.VolumeEditor Class Reference

Custom property editor for volumes game objects

Inheritance diagram for DungeonArchitect.Editors.VolumeEditor:



Public Member Functions

- override void **OnInspectorGUI** ()
- virtual void **OnUpdate** (SceneView sceneView)

Protected Member Functions

- void **OnTransformModified** ([Volume](#) volume)

5.138.1 Detailed Description

Custom property editor for volumes game objects

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

- [github/dungeon-architect-unity/Editor/Volumes/VolumeEditor.cs](#)

Index

- ActiveGraph
 - DungeonArchitect::Editors::DungeonObjectTraker, [45](#)
- ActiveModel
 - DungeonArchitect::Dungeon, [31](#)
- AddGameObject
 - DungeonArchitect::DungeonSceneProvider, [51](#)
 - DungeonArchitect::PooledDungeonSceneProvider, [128](#)
- AddPaintCell
 - DungeonArchitect::Dungeon, [30](#)
- AddSprite
 - DungeonArchitect::DungeonSceneProvider, [52](#)
 - DungeonArchitect::PooledDungeonSceneProvider, [128](#)
- AddToAsset
 - DungeonArchitect::Editors::DungeonEditorHelper, [36](#)
- AdjacentTiles
 - DungeonArchitect::CellDoor, [24](#)
- affectsNavigation
 - DungeonArchitect::DungeonSceneProviderData, [53](#)
 - DungeonArchitect::Graphs::VisualNode, [144](#)
 - DungeonArchitect::PropTypeData, [130](#)
- Affinity
 - DungeonArchitect::PropTypeData, [130](#)
- AttachToSocket
 - DungeonArchitect::PropTypeData, [130](#)
- Bounds
 - DungeonArchitect::Graphs::GraphNode, [71](#)
- BoundsOffset
 - DungeonArchitect::Graphs::GraphPin, [78](#)
- BreakInputLinks
 - DungeonArchitect::Editors::GraphOperations, [73](#)
- BreakOutputLinks
 - DungeonArchitect::Editors::GraphOperations, [73](#)
- brushSize
 - DungeonArchitect::DungeonPaintModeGrid, [48](#)
- Build
 - DungeonArchitect::Dungeon, [30](#)
- BuildCellLookup
 - DungeonArchitect::GridDungeonModel, [90](#)
- BuildCells
 - DungeonArchitect::GridDungeonBuilder, [84](#)
- BuildDungeon
 - DungeonArchitect::DungeonBuilder, [34](#)
 - DungeonArchitect::GridDungeonBuilder, [84](#)
- BuildSpatialCellLookup
 - DungeonArchitect::GridDungeonModel, [90](#)
- CanCreateLink
 - DungeonArchitect::Graphs::GraphSchema, [80](#)
- CanSelect
 - DungeonArchitect::SelectorRule, [133](#)
- Caption
 - DungeonArchitect::Graphs::GraphNode, [71](#)
- CellLookup
 - DungeonArchitect::GridDungeonModel, [91](#)
- CellType
 - DungeonArchitect, [16](#)
- ChildSockets
 - DungeonArchitect::PropTypeData, [130](#)
- ClearToolOverlayData
 - DungeonArchitect::Dungeon, [30](#)
- ClickState
 - DungeonArchitect::Graphs::GraphPin, [78](#)
- config
 - DungeonArchitect::DungeonSceneProvider, [52](#)
- ConsumeOnAttach
 - DungeonArchitect::PropTypeData, [131](#)
- ContainsDoorBetweenCells
 - DungeonArchitect::DoorManager, [28](#)
- ContainsPoint
 - DungeonArchitect::Graphs::GraphPin, [77](#)
- ContainsStair
 - DungeonArchitect::GridDungeonModel, [90](#)
- Copy
 - DungeonArchitect::Utils::MathUtils, [109](#)
 - DungeonArchitect::Utils::Matrix, [115](#)
- CopyFrom
 - DungeonArchitect::Graphs::GameObjectNode, [56](#)
 - DungeonArchitect::Graphs::GraphNode, [70](#)
 - DungeonArchitect::Graphs::PlaceableNode, [123](#)
 - DungeonArchitect::Graphs::SpriteNode, [134](#)
 - DungeonArchitect::Graphs::VisualNode, [143](#)
- CorridorPadding
 - DungeonArchitect::GridDungeonConfig, [86](#)
- CorridorPaddingDoubleSided
 - DungeonArchitect::GridDungeonConfig, [86](#)
- CreateDoor
 - DungeonArchitect::DoorManager, [29](#)
- CreateEditorTag
 - DungeonArchitect::Editors::DungeonEditorHelper, [36](#)
- CreateLink< T >
 - DungeonArchitect::Editors::GraphOperations, [73](#)
- CreateNode
 - DungeonArchitect::Editors::GraphOperations, [75](#)

- CreateNode< T >
 - DungeonArchitect::Editors::GraphEditor, [63](#)
 - DungeonArchitect::Editors::GraphOperations, [75](#)
- CreatePin
 - DungeonArchitect::Graphs::GraphNode, [70](#)
- cursorLogicalHeight
 - DungeonArchitect::DungeonPaintModeGrid, [48](#)
- DecomposeMatrix
 - DungeonArchitect::Utils::Matrix, [115](#)
- DestroyDungeon
 - DungeonArchitect::Dungeon, [31](#)
- DestroyLink
 - DungeonArchitect::Editors::GraphOperations, [75](#)
- DestroyNode
 - DungeonArchitect::Editors::GraphOperations, [75](#)
- Divide
 - DungeonArchitect::Utils::MathUtils, [110](#)
- Doors
 - DungeonArchitect::DoorManager, [29](#)
 - DungeonArchitect::GridDungeonModel, [91](#)
- DragNode
 - DungeonArchitect::Graphs::GraphNode, [71](#)
- Draw
 - DungeonArchitect::Editors::GraphEditor, [64](#)
- DrawCell
 - DungeonArchitect::LandscapeDataRasterizer, [96](#)
- DrawMatrixProperty
 - DungeonArchitect::Editors::InspectorUtils, [92](#)
- DrawVectorProperty
 - DungeonArchitect::Editors::InspectorUtils, [92](#)
- dungeon
 - DungeonArchitect::DungeonSceneProvider, [52](#)
 - DungeonArchitect::DungeonSceneProviderData, [53](#)
- DungeonArchitect, [13](#)
 - CellType, [16](#)
 - DungeonModelBuildState, [16](#)
 - LandscapeTextureType, [16](#)
- DungeonArchitect.BlurFilter, [22](#)
- DungeonArchitect.Cell, [23](#)
- DungeonArchitect.CellDoor, [24](#)
- DungeonArchitect.CellHeightFrameInfo, [24](#)
- DungeonArchitect.CellHeightNode, [25](#)
- DungeonArchitect.DebugDrawer, [26](#)
- DungeonArchitect.DoorManager, [28](#)
- DungeonArchitect.Dungeon, [29](#)
- DungeonArchitect.DungeonBuilder, [32](#)
- DungeonArchitect.DungeonConfig, [34](#)
- DungeonArchitect.DungeonEventListener, [38](#)
- DungeonArchitect.DungeonMarkerEmitter, [39](#)
- DungeonArchitect.DungeonModel, [40](#)
- DungeonArchitect.DungeonPaintMode, [45](#)
- DungeonArchitect.DungeonPaintModeGrid, [47](#)
- DungeonArchitect.DungeonPropDataAsset, [49](#)
- DungeonArchitect.DungeonSceneProvider, [49](#)
- DungeonArchitect.DungeonSceneProviderData, [52](#)
- DungeonArchitect.DungeonToolData, [53](#)
- DungeonArchitect.Editors, [16](#)
- DungeonArchitect.Editors.AssetThumbnailCache, [21](#)
- DungeonArchitect.Editors.CursorDragLink, [26](#)
- DungeonArchitect.Editors.DungeonArchitectGraph↔Editor, [32](#)
- DungeonArchitect.Editors.DungeonAssetPostprocessor, [32](#)
- DungeonArchitect.Editors.DungeonEditorHelper, [35](#)
- DungeonArchitect.Editors.DungeonEditorResources, [37](#)
- DungeonArchitect.Editors.DungeonObjectTraker, [44](#)
- DungeonArchitect.Editors.DungeonPaintModeEditor, [46](#)
- DungeonArchitect.Editors.DungeonPaintModeGrid↔Editor, [48](#)
- DungeonArchitect.Editors.DungeonPropertyEditor, [49](#)
- DungeonArchitect.Editors.GraphContextMenu, [61](#)
- DungeonArchitect.Editors.GraphContextMenuEvent, [62](#)
- DungeonArchitect.Editors.GraphEditor, [63](#)
- DungeonArchitect.Editors.GraphEditorConstants, [65](#)
- DungeonArchitect.Editors.GraphInputHandler, [66](#)
- DungeonArchitect.Editors.GraphInspector, [66](#)
- DungeonArchitect.Editors.GraphLinkRenderer, [68](#)
- DungeonArchitect.Editors.GraphNodeRenderer, [72](#)
- DungeonArchitect.Editors.GraphNodeRendererFactory, [73](#)
- DungeonArchitect.Editors.GraphOperations, [73](#)
- DungeonArchitect.Editors.GraphPinHierarchyComparer, [79](#)
- DungeonArchitect.Editors.GraphPinRenderer, [79](#)
- DungeonArchitect.Editors.GraphRendererContext, [80](#)
- DungeonArchitect.Editors.GraphSelectionBox, [81](#)
- DungeonArchitect.Editors.GraphTooltip, [82](#)
- DungeonArchitect.Editors.GraphTooltipRenderer, [82](#)
- DungeonArchitect.Editors.GridDungeonConfigProperty↔Editor, [88](#)
- DungeonArchitect.Editors.InspectorUtils, [92](#)
- DungeonArchitect.Editors.KeyboardState, [95](#)
- DungeonArchitect.Editors.LandscapeTextureEditor, [98](#)
- DungeonArchitect.Editors.MarkerEmitterNodeEditor, [106](#)
- DungeonArchitect.Editors.MarkerEmitterNodeRenderer, [106](#)
- DungeonArchitect.Editors.MarkerNodeEditor, [107](#)
- DungeonArchitect.Editors.MarkerNodeRenderer, [108](#)
- DungeonArchitect.Editors.MeshNodeEditor, [117](#)
- DungeonArchitect.Editors.MeshNodeRenderer, [118](#)
- DungeonArchitect.Editors.NegationVolumeEditor, [120](#)
- DungeonArchitect.Editors.NodeDeletionOrderComparer, [121](#)
- DungeonArchitect.Editors.NodeReversedZIndex↔Comparer, [121](#)
- DungeonArchitect.Editors.NodeZIndexComparer, [122](#)
- DungeonArchitect.Editors.PlaceableNodeEditor, [123](#)
- DungeonArchitect.Editors.PlatformVolumeEditor, [124](#)
- DungeonArchitect.Editors.SpriteNodeEditor, [135](#)
- DungeonArchitect.Editors.SpriteNodeRenderer, [135](#)
- DungeonArchitect.Editors.ThemeOverrideVolume↔Editor, [139](#)
- DungeonArchitect.Editors.Timer, [140](#)
- DungeonArchitect.Editors.VisualNodeEditor, [144](#)

- DungeonArchitect.Editors.VisualNodeRenderer, 145
- DungeonArchitect.Editors.VolumeEditor, 147
- DungeonArchitect.Filter, 55
- DungeonArchitect.GameObjectPropTypeData, 57
- DungeonArchitect.Graphs, 18
- DungeonArchitect.Graphs.GameObjectNode, 56
- DungeonArchitect.Graphs.Graph, 57
- DungeonArchitect.Graphs.GraphCamera, 59
- DungeonArchitect.Graphs.GraphLink, 67
- DungeonArchitect.Graphs.GraphNode, 69
- DungeonArchitect.Graphs.GraphPin, 76
- DungeonArchitect.Graphs.GraphSchema, 80
- DungeonArchitect.Graphs.IndexCounter, 91
- DungeonArchitect.Graphs.MarkerEmitterNode, 105
- DungeonArchitect.Graphs.MarkerNode, 107
- DungeonArchitect.Graphs.PlaceableNode, 122
- DungeonArchitect.Graphs.SpriteNode, 133
- DungeonArchitect.Graphs.VisualNode, 142
- DungeonArchitect.GridCellInfo, 83
- DungeonArchitect.GridDungeonBuilder, 83
- DungeonArchitect.GridDungeonConfig, 85
- DungeonArchitect.GridDungeonModel, 88
- DungeonArchitect.GridDungeonModelUtils, 91
- DungeonArchitect.IntVector, 94
- DungeonArchitect.IntVector2Key, 94
- DungeonArchitect.LandscapeDataRasterizer, 95
- DungeonArchitect.LandscapeTexture, 98
- DungeonArchitect.LandscapeTransformerGrid, 99
- DungeonArchitect.LeftToRightNodeComparer, 101
- DungeonArchitect.MarkerEmitterCornerBeautifier, 102
- DungeonArchitect.MarkerEmitterEmptySpace, 103
- DungeonArchitect.MarkerEmitterFindLowestPoint, 103
- DungeonArchitect.MarkerEmitterFreeSpaceDecorator, 104
- DungeonArchitect.Navigation, 19
- DungeonArchitect.Navigation.Collision2DTriangle↔Provider, 25
- DungeonArchitect.Navigation.CollisionTriangleProvider, 25
- DungeonArchitect.Navigation.DungeonNavAgent, 41
- DungeonArchitect.Navigation.DungeonNavAgent2D, 41
- DungeonArchitect.Navigation.DungeonNavAgent3D, 42
- DungeonArchitect.Navigation.DungeonNavMesh, 43
- DungeonArchitect.Navigation.LayoutFloorTriangle↔Provider, 101
- DungeonArchitect.Navigation.NavigationBuildInvoker, 119
- DungeonArchitect.Navigation.NavigationTriangle↔Provider, 120
- DungeonArchitect.Navigation.StaticMeshTriangle↔Provider, 138
- DungeonArchitect.Navigation.TerrainTriangleProvider, 138
- DungeonArchitect.NegationVolume, 120
- DungeonArchitect.PlatformVolume, 124
- DungeonArchitect.PooledDungeonSceneProvider, 127
- DungeonArchitect.PropChildSocketData, 129
- DungeonArchitect.PropSocket, 129
- DungeonArchitect.PropTypeData, 129
- DungeonArchitect.Rectangle, 131
- DungeonArchitect.SelectorRule, 132
- DungeonArchitect.SpritePropTypeData, 136
- DungeonArchitect.StairAdjacencyQueueNode, 136
- DungeonArchitect.StairEdgeInfo, 137
- DungeonArchitect.StairInfo, 137
- DungeonArchitect.ThemeOverrideVolume, 139
- DungeonArchitect.TransformationRule, 140
- DungeonArchitect.Triangulator, 19
- DungeonArchitect.Triangulator.Delauney, 27
- DungeonArchitect.Triangulator.Geometry, 20
- DungeonArchitect.Triangulator.Geometry.Edge, 54
- DungeonArchitect.Triangulator.Geometry.Point, 126, 127
- DungeonArchitect.Triangulator.Geometry.Triangle, 141
- DungeonArchitect.Utils, 20
- DungeonArchitect.Utils.Blackboard, 22
- DungeonArchitect.Utils.BlackboardDatabase< T >, 22
- DungeonArchitect.Utils.DungeonConstants, 35
- DungeonArchitect.Utils.GraphUtils, 82
- DungeonArchitect.Utils.InstanceCache, 93
- DungeonArchitect.Utils.MathUtils, 108
- DungeonArchitect.Utils.Matrix, 114
- DungeonArchitect.Utils.PMRandom, 125
- DungeonArchitect.Volume, 145
- DungeonArchitect::CellDoor
 - AdjacentTiles, 24
- DungeonArchitect::DoorManager
 - ContainsDoorBetweenCells, 28
 - CreateDoor, 29
 - Doors, 29
- DungeonArchitect::Dungeon
 - ActiveModel, 31
 - AddPaintCell, 30
 - Build, 30
 - ClearToolOverlayData, 30
 - DestroyDungeon, 31
 - dungeonThemes, 31
 - IsLayoutBuilt, 31
 - ReapplyTheme, 31
 - RemovePaintCell, 31
 - RequestRebuild, 31
- DungeonArchitect::DungeonBuilder
 - BuildDungeon, 34
 - EmitCustomMarkers, 34
 - EmitMarkers, 34
- DungeonArchitect::DungeonEventListener
 - OnDungeonDestroyed, 39
 - OnPostDungeonBuild, 39
 - OnPostDungeonLayoutBuild, 39
- DungeonArchitect::DungeonMarkerEmitter
 - EmitMarkers, 40
- DungeonArchitect::DungeonPaintMode
 - GetDungeon, 46
 - GetDungeonConfig, 46
 - GetDungeonModel, 46
- DungeonArchitect::DungeonPaintModeGrid

- brushSize, [48](#)
- cursorLogicalHeight, [48](#)
- mode2D, [48](#)
- overlayOpacity, [48](#)
- DungeonArchitect::DungeonSceneProvider
 - AddGameObject, [51](#)
 - AddSprite, [52](#)
 - config, [52](#)
 - dungeon, [52](#)
 - itemParent, [52](#)
 - OnDungeonBuildStart, [52](#)
 - OnDungeonBuildStop, [52](#)
- DungeonArchitect::DungeonSceneProviderData
 - affectsNavigation, [53](#)
 - dungeon, [53](#)
 - NodeId, [53](#)
- DungeonArchitect::Editors
 - GraphMenuAction, [18](#)
- DungeonArchitect::Editors::AssetThumbnailCache
 - GetThumb, [21](#)
 - Instance, [22](#)
 - Reset, [21](#)
- DungeonArchitect::Editors::DungeonEditorHelper
 - AddToAsset, [36](#)
 - CreateEditorTag, [36](#)
 - MakeFilenameUnique, [36](#)
 - MarkAsDirty, [37](#)
 - OnOpenAsset, [37](#)
 - ShowEditor, [37](#)
- DungeonArchitect::Editors::DungeonEditorResources
 - GetResource< T >, [38](#)
- DungeonArchitect::Editors::DungeonObjectTraker
 - ActiveGraph, [45](#)
 - Dungeons, [45](#)
 - RequestRebuild, [45](#)
- DungeonArchitect::Editors::GraphContextMenu
 - HandleInput, [62](#)
 - Show, [62](#)
- DungeonArchitect::Editors::GraphEditor
 - CreateNode< T >, [63](#)
 - Draw, [64](#)
 - FocusCameraOnBestFit, [64](#)
 - FocusCameraOnMarker, [64](#)
 - GetPinUnderPosition, [64](#)
 - Graph, [65](#)
 - HandleInput, [64](#)
 - Init, [65](#)
 - RealtimeUpdate, [65](#)
 - SelectNode, [65](#)
- DungeonArchitect::Editors::GraphInputHandler
 - HandleNodeInput, [66](#)
 - HandlePinInput, [66](#)
- DungeonArchitect::Editors::GraphInspector
 - OnInspectorGUI, [67](#)
- DungeonArchitect::Editors::GraphOperations
 - BreakInputLinks, [73](#)
 - BreakOutputLinks, [73](#)
 - CreateLink< T >, [73](#)
 - CreateNode, [75](#)
 - CreateNode< T >, [75](#)
 - DestroyLink, [75](#)
 - DestroyNode, [75](#)
 - DuplicateNode< T >, [76](#)
- DungeonArchitect::Editors::GraphSelectionBox
 - HandleInput, [81](#)
- DungeonArchitect::Editors::GraphTooltip
 - message, [82](#)
- DungeonArchitect::Editors::InspectorUtils
 - DrawMatrixProperty, [92](#)
 - DrawVectorProperty, [92](#)
 - Round, [92](#)
 - RoundVector, [93](#)
- DungeonArchitect::Editors::Timer
 - Hertz, [140](#)
 - Update, [140](#)
- DungeonArchitect::Graphs
 - GraphPinMouseState, [19](#)
 - GraphPinType, [19](#)
- DungeonArchitect::Graphs::GameObjectNode
 - CopyFrom, [56](#)
- DungeonArchitect::Graphs::Graph
 - GetNode, [58](#)
 - GetNodes< T >, [58](#)
 - Links, [59](#)
 - MarkAsDirty, [58](#)
 - Nodes, [59](#)
 - NotifyStateChanged, [58](#)
 - TopZIndex, [59](#)
- DungeonArchitect::Graphs::GraphCamera
 - FocusOnBestFit, [60](#)
 - FocusOnMarker, [60](#)
 - HandleInput, [60](#)
 - Pan, [60](#)
 - Position, [61](#)
 - Scale, [61](#)
 - ScreenToWorld, [60](#)
 - WorldToScreen, [61](#)
- DungeonArchitect::Graphs::GraphLink
 - GetTangentStrength, [68](#)
 - Graph, [68](#)
 - Id, [68](#)
 - Input, [68](#)
 - Output, [68](#)
- DungeonArchitect::Graphs::GraphNode
 - Bounds, [71](#)
 - Caption, [71](#)
 - CopyFrom, [70](#)
 - CreatePin, [70](#)
 - DragNode, [71](#)
 - GetChildNodes, [71](#)
 - GetParentNodes, [71](#)
 - Graph, [71](#)
 - Id, [71](#)
 - InputPin, [71](#)
 - InputPins, [71](#)
 - OutputPin, [71](#)

- OutputPins, [72](#)
- Position, [72](#)
- Selected, [72](#)
- Size, [72](#)
- ZIndex, [72](#)
- DungeonArchitect::Graphs::GraphPin
 - BoundsOffset, [78](#)
 - ClickState, [78](#)
 - ContainsPoint, [77](#)
 - GetBounds, [77](#)
 - GetConntectedLinks, [77](#)
 - GetWorldBounds, [78](#)
 - Node, [78](#)
 - PinLinksDestroyed, [79](#)
 - PinType, [78](#)
 - Position, [78](#)
 - Tangent, [78](#)
 - TangentStrength, [78](#)
 - WorldPosition, [78](#)
- DungeonArchitect::Graphs::GraphSchema
 - CanCreateLink, [80](#)
- DungeonArchitect::Graphs::PlaceableNode
 - CopyFrom, [123](#)
- DungeonArchitect::Graphs::SpriteNode
 - CopyFrom, [134](#)
- DungeonArchitect::Graphs::VisualNode
 - affectsNavigation, [144](#)
 - CopyFrom, [143](#)
 - IsStatic, [144](#)
 - selectionRuleClassName, [144](#)
 - selectionRuleEnabled, [144](#)
 - transformRuleClassName, [144](#)
 - transformRuleEnabled, [144](#)
- DungeonArchitect::GridDungeonBuilder
 - BuildCells, [84](#)
 - BuildDungeon, [84](#)
 - EmitMarkers, [84](#)
 - Seperate, [84](#)
 - TriangulateRooms, [85](#)
- DungeonArchitect::GridDungeonConfig
 - CorridorPadding, [86](#)
 - CorridorPaddingDoubleSided, [86](#)
 - FloorHeight, [86](#)
 - HeightVariationProbability, [86](#)
 - InitialRoomRadius, [87](#)
 - MaxAllowedStairHeight, [87](#)
 - MaxCellSize, [87](#)
 - MinCellSize, [87](#)
 - NormalMean, [87](#)
 - NormalStd, [87](#)
 - NumCells, [87](#)
 - RoomAreaThreshold, [87](#)
 - RoomAspectDelta, [87](#)
 - SpanningTreeLoopProbability, [88](#)
 - StairConnectionTolerance, [88](#)
- DungeonArchitect::GridDungeonModel
 - BuildCellLookup, [90](#)
 - BuildSpatialCellLookup, [90](#)
 - CellLookup, [91](#)
 - ContainsStair, [90](#)
 - Doors, [91](#)
 - FindCellByPosition, [90](#)
 - GetCell, [90](#)
 - GetGridCellLookup, [90](#)
 - ResetModel, [91](#)
- DungeonArchitect::LandscapeDataRasterizer
 - DrawCell, [96](#)
 - GetHeight, [97](#)
 - LandscapeDataRasterizer, [96](#)
 - LoadData, [97](#)
 - SaveData, [97](#)
 - SmoothCell, [97](#)
 - WorldToTerrainCoord, [97](#)
 - WorldToTerrainTextureCoord, [98](#)
- DungeonArchitect::LandscapeTransformerGrid
 - groundLevelHeight, [101](#)
 - OnPostDungeonLayoutBuild, [100](#)
- DungeonArchitect::MarkerEmitterCornerBeautifier
 - EmitMarkers, [102](#)
- DungeonArchitect::MarkerEmitterEmptySpace
 - EmitMarkers, [103](#)
- DungeonArchitect::MarkerEmitterFindLowestPoint
 - EmitMarkers, [104](#)
- DungeonArchitect::MarkerEmitterFreeSpaceDecorator
 - EmitMarkers, [105](#)
- DungeonArchitect::Navigation::NavigationBuildInvoker
 - OnPostDungeonBuild, [119](#)
- DungeonArchitect::PooledDungeonSceneProvider
 - AddGameObject, [128](#)
 - AddSprite, [128](#)
 - OnDungeonBuildStart, [128](#)
 - OnDungeonBuildStop, [128](#)
- DungeonArchitect::PropTypeData
 - affectsNavigation, [130](#)
 - Affinity, [130](#)
 - AttachToSocket, [130](#)
 - ChildSockets, [130](#)
 - ConsumeOnAttach, [131](#)
 - IsStaticObject, [131](#)
 - NodeId, [131](#)
 - Offset, [131](#)
 - SelectorRuleClassName, [131](#)
 - TransformRuleClassName, [131](#)
 - UseSelectionRule, [131](#)
 - UseTransformRule, [131](#)
- DungeonArchitect::SelectorRule
 - CanSelect, [133](#)
- DungeonArchitect::TransformationRule
 - GetTransform, [141](#)
- DungeonArchitect::Triangulator::Delauney
 - Triangulate, [27](#)
- DungeonArchitect::Triangulator::Geometry::Edge
 - Edge, [54](#), [55](#)
 - Equals, [55](#)
 - p1, [55](#)
 - p2, [55](#)

- DungeonArchitect::Triangulator::Geometry::Triangle
 - p1, [142](#)
 - p2, [142](#)
 - p3, [142](#)
 - Triangle, [142](#)
- DungeonArchitect::Utils::InstanceCache
 - GetInstance, [93](#)
- DungeonArchitect::Utils::MathUtils
 - Copy, [109](#)
 - Divide, [110](#)
 - FlipYZ, [110](#)
 - GridToWorld, [111](#)
 - INVALID_LOCATION, [114](#)
 - Intersects, [111](#)
 - RoundToIntVector, [111](#)
 - SnapToGrid, [113](#)
 - ToIntVector, [113](#)
 - ToVector3, [113](#)
 - WorldToGrid, [114](#)
- DungeonArchitect::Utils::Matrix
 - Copy, [115](#)
 - DecomposeMatrix, [115](#)
 - FromGameTransform, [115](#)
 - GetRotation, [116](#)
 - GetScale, [116](#)
 - GetTranslation, [116](#)
 - Identity, [116](#)
 - IdentityQuaternion, [117](#)
 - SetTransform, [116](#)
 - SetTransformFromMatrix, [117](#)
 - SetTranslation, [117](#)
 - TranslationMatrix, [117](#)
- DungeonArchitect::Utils::PMRandom
 - GetNextUniformFloat, [126](#)
 - Initialize, [126](#)
 - NextGaussianFloat, [126](#)
 - PMRandom, [125](#)
- DungeonArchitect::Volume
 - GetBounds, [146](#)
 - GetGridTransform, [146](#)
- DungeonModelBuildState
 - DungeonArchitect, [16](#)
- DungeonNavMeshEditor, [44](#)
- dungeonThemes
 - DungeonArchitect::Dungeon, [31](#)
- Dungeons
 - DungeonArchitect::Editors::DungeonObjectTraker, [45](#)
- DuplicateNode< T >
 - DungeonArchitect::Editors::GraphOperations, [76](#)
- Edge
 - DungeonArchitect::Triangulator::Geometry::Edge, [54](#), [55](#)
- EmitCustomMarkers
 - DungeonArchitect::DungeonBuilder, [34](#)
- EmitMarkers
 - DungeonArchitect::DungeonBuilder, [34](#)
 - DungeonArchitect::DungeonMarkerEmitter, [40](#)
 - DungeonArchitect::GridDungeonBuilder, [84](#)
 - DungeonArchitect::MarkerEmitterCornerBeautifier, [102](#)
 - DungeonArchitect::MarkerEmitterEmptySpace, [103](#)
 - DungeonArchitect::MarkerEmitterFindLowestPoint, [104](#)
 - DungeonArchitect::MarkerEmitterFreeSpace↔Decorator, [105](#)
- Equals
 - DungeonArchitect::Triangulator::Geometry::Edge, [55](#)
- FindCellByPosition
 - DungeonArchitect::GridDungeonModel, [90](#)
- FlipYZ
 - DungeonArchitect::Utils::MathUtils, [110](#)
- FloorHeight
 - DungeonArchitect::GridDungeonConfig, [86](#)
- FocusCameraOnBestFit
 - DungeonArchitect::Editors::GraphEditor, [64](#)
- FocusCameraOnMarker
 - DungeonArchitect::Editors::GraphEditor, [64](#)
- FocusOnBestFit
 - DungeonArchitect::Graphs::GraphCamera, [60](#)
- FocusOnMarker
 - DungeonArchitect::Graphs::GraphCamera, [60](#)
- FromGameTransform
 - DungeonArchitect::Utils::Matrix, [115](#)
- GetBounds
 - DungeonArchitect::Graphs::GraphPin, [77](#)
 - DungeonArchitect::Volume, [146](#)
- GetCell
 - DungeonArchitect::GridDungeonModel, [90](#)
- GetChildNodes
 - DungeonArchitect::Graphs::GraphNode, [71](#)
- GetConnectedLinks
 - DungeonArchitect::Graphs::GraphPin, [77](#)
- GetDungeon
 - DungeonArchitect::DungeonPaintMode, [46](#)
- GetDungeonConfig
 - DungeonArchitect::DungeonPaintMode, [46](#)
- GetDungeonModel
 - DungeonArchitect::DungeonPaintMode, [46](#)
- GetGridCellLookup
 - DungeonArchitect::GridDungeonModel, [90](#)
- GetGridTransform
 - DungeonArchitect::Volume, [146](#)
- GetHeight
 - DungeonArchitect::LandscapeDataRasterizer, [97](#)
- GetInstance
 - DungeonArchitect::Utils::InstanceCache, [93](#)
- GetNextUniformFloat
 - DungeonArchitect::Utils::PMRandom, [126](#)
- GetNode
 - DungeonArchitect::Graphs::Graph, [58](#)
- GetNodes< T >
 - DungeonArchitect::Graphs::Graph, [58](#)

- GetParentNodes
 - DungeonArchitect::Graphs::GraphNode, 71
- GetPinUnderPosition
 - DungeonArchitect::Editors::GraphEditor, 64
- GetResource< T >
 - DungeonArchitect::Editors::DungeonEditor↔Resources, 38
- GetRotation
 - DungeonArchitect::Utils::Matrix, 116
- GetScale
 - DungeonArchitect::Utils::Matrix, 116
- GetTangentStrength
 - DungeonArchitect::Graphs::GraphLink, 68
- GetThumb
 - DungeonArchitect::Editors::AssetThumbnailCache, 21
- GetTransform
 - DungeonArchitect::TransformationRule, 141
- GetTranslation
 - DungeonArchitect::Utils::Matrix, 116
- GetWorldBounds
 - DungeonArchitect::Graphs::GraphPin, 78
- Graph
 - DungeonArchitect::Editors::GraphEditor, 65
 - DungeonArchitect::Graphs::GraphLink, 68
 - DungeonArchitect::Graphs::GraphNode, 71
- GraphMenuAction
 - DungeonArchitect::Editors, 18
- GraphPinMouseState
 - DungeonArchitect::Graphs, 19
- GraphPinType
 - DungeonArchitect::Graphs, 19
- GridToWorld
 - DungeonArchitect::Utils::MathUtils, 111
- groundLevelHeight
 - DungeonArchitect::LandscapeTransformerGrid, 101
- HandleInput
 - DungeonArchitect::Editors::GraphContextMenu, 62
 - DungeonArchitect::Editors::GraphEditor, 64
 - DungeonArchitect::Editors::GraphSelectionBox, 81
 - DungeonArchitect::Graphs::GraphCamera, 60
- HandleNodeInput
 - DungeonArchitect::Editors::GraphInputHandler, 66
- HandlePinInput
 - DungeonArchitect::Editors::GraphInputHandler, 66
- HeightVariationProbability
 - DungeonArchitect::GridDungeonConfig, 86
- Hertz
 - DungeonArchitect::Editors::Timer, 140
- INVALID_LOCATION
 - DungeonArchitect::Utils::MathUtils, 114
- Id
 - DungeonArchitect::Graphs::GraphLink, 68
 - DungeonArchitect::Graphs::GraphNode, 71
- Identity
 - DungeonArchitect::Utils::Matrix, 116
- IdentityQuaternion
 - DungeonArchitect::Utils::Matrix, 117
- Init
 - DungeonArchitect::Editors::GraphEditor, 65
- InitialRoomRadius
 - DungeonArchitect::GridDungeonConfig, 87
- Initialize
 - DungeonArchitect::Utils::PMRandom, 126
- Input
 - DungeonArchitect::Graphs::GraphLink, 68
- InputPin
 - DungeonArchitect::Graphs::GraphNode, 71
- InputPins
 - DungeonArchitect::Graphs::GraphNode, 71
- Instance
 - DungeonArchitect::Editors::AssetThumbnailCache, 22
- Intersects
 - DungeonArchitect::Utils::MathUtils, 111
- IsLayoutBuilt
 - DungeonArchitect::Dungeon, 31
- IsStatic
 - DungeonArchitect::Graphs::VisualNode, 144
- IsStaticObject
 - DungeonArchitect::PropTypeData, 131
- itemParent
 - DungeonArchitect::DungeonSceneProvider, 52
- LandscapeDataRasterizer
 - DungeonArchitect::LandscapeDataRasterizer, 96
- LandscapeTextureType
 - DungeonArchitect, 16
- Links
 - DungeonArchitect::Graphs::Graph, 59
- LoadData
 - DungeonArchitect::LandscapeDataRasterizer, 97
- MakeFilenameUnique
 - DungeonArchitect::Editors::DungeonEditorHelper, 36
- MarkAsDirty
 - DungeonArchitect::Editors::DungeonEditorHelper, 37
 - DungeonArchitect::Graphs::Graph, 58
- MaxAllowedStairHeight
 - DungeonArchitect::GridDungeonConfig, 87
- MaxCellSize
 - DungeonArchitect::GridDungeonConfig, 87
- message
 - DungeonArchitect::Editors::GraphTooltip, 82
- MinCellSize
 - DungeonArchitect::GridDungeonConfig, 87
- mode2D
 - DungeonArchitect::DungeonPaintModeGrid, 48
- NextGaussianFloat
 - DungeonArchitect::Utils::PMRandom, 126
- Node
 - DungeonArchitect::Graphs::GraphPin, 78

- NodeId
 - DungeonArchitect::DungeonSceneProviderData, [53](#)
 - DungeonArchitect::PropTypeData, [131](#)
- Nodes
 - DungeonArchitect::Graphs::Graph, [59](#)
- NormalMean
 - DungeonArchitect::GridDungeonConfig, [87](#)
- NormalStd
 - DungeonArchitect::GridDungeonConfig, [87](#)
- NotifyStateChanged
 - DungeonArchitect::Graphs::Graph, [58](#)
- NumCells
 - DungeonArchitect::GridDungeonConfig, [87](#)
- Offset
 - DungeonArchitect::PropTypeData, [131](#)
- OnDungeonBuildStart
 - DungeonArchitect::DungeonSceneProvider, [52](#)
 - DungeonArchitect::PooledDungeonSceneProvider, [128](#)
- OnDungeonBuildStop
 - DungeonArchitect::DungeonSceneProvider, [52](#)
 - DungeonArchitect::PooledDungeonSceneProvider, [128](#)
- OnDungeonDestroyed
 - DungeonArchitect::DungeonEventListener, [39](#)
- OnInspectorGUI
 - DungeonArchitect::Editors::GraphInspector, [67](#)
- OnOpenAsset
 - DungeonArchitect::Editors::DungeonEditorHelper, [37](#)
- OnPostDungeonBuild
 - DungeonArchitect::DungeonEventListener, [39](#)
 - DungeonArchitect::Navigation::NavigationBuildInvoker, [119](#)
- OnPostDungeonLayoutBuild
 - DungeonArchitect::DungeonEventListener, [39](#)
 - DungeonArchitect::LandscapeTransformerGrid, [100](#)
- Output
 - DungeonArchitect::Graphs::GraphLink, [68](#)
- OutputPin
 - DungeonArchitect::Graphs::GraphNode, [71](#)
- OutputPins
 - DungeonArchitect::Graphs::GraphNode, [72](#)
- overlayOpacity
 - DungeonArchitect::DungeonPaintModeGrid, [48](#)
- p1
 - DungeonArchitect::Triangulator::Geometry::Edge, [55](#)
 - DungeonArchitect::Triangulator::Geometry::Triangle, [142](#)
- p2
 - DungeonArchitect::Triangulator::Geometry::Edge, [55](#)
 - DungeonArchitect::Triangulator::Geometry::Triangle, [142](#)
- p3
 - DungeonArchitect::Triangulator::Geometry::Triangle, [142](#)
- PMRandom
 - DungeonArchitect::Utils::PMRandom, [125](#)
- Pan
 - DungeonArchitect::Graphs::GraphCamera, [60](#)
- PinLinksDestroyed
 - DungeonArchitect::Graphs::GraphPin, [79](#)
- PinType
 - DungeonArchitect::Graphs::GraphPin, [78](#)
- Position
 - DungeonArchitect::Graphs::GraphCamera, [61](#)
 - DungeonArchitect::Graphs::GraphNode, [72](#)
 - DungeonArchitect::Graphs::GraphPin, [78](#)
- RealtimeUpdate
 - DungeonArchitect::Editors::GraphEditor, [65](#)
- ReapplyTheme
 - DungeonArchitect::Dungeon, [31](#)
- RemovePaintCell
 - DungeonArchitect::Dungeon, [31](#)
- RequestRebuild
 - DungeonArchitect::Dungeon, [31](#)
 - DungeonArchitect::Editors::DungeonObjectTraker, [45](#)
- Reset
 - DungeonArchitect::Editors::AssetThumbnailCache, [21](#)
- ResetModel
 - DungeonArchitect::GridDungeonModel, [91](#)
- RoomAreaThreshold
 - DungeonArchitect::GridDungeonConfig, [87](#)
- RoomAspectDelta
 - DungeonArchitect::GridDungeonConfig, [87](#)
- Round
 - DungeonArchitect::Editors::InspectorUtils, [92](#)
- RoundToIntVector
 - DungeonArchitect::Utils::MathUtils, [111](#)
- RoundVector
 - DungeonArchitect::Editors::InspectorUtils, [93](#)
- SaveData
 - DungeonArchitect::LandscapeDataRasterizer, [97](#)
- Scale
 - DungeonArchitect::Graphs::GraphCamera, [61](#)
- ScreenToWorld
 - DungeonArchitect::Graphs::GraphCamera, [60](#)
- SelectNode
 - DungeonArchitect::Editors::GraphEditor, [65](#)
- Selected
 - DungeonArchitect::Graphs::GraphNode, [72](#)
- selectionRuleClassName
 - DungeonArchitect::Graphs::VisualNode, [144](#)
- selectionRuleEnabled
 - DungeonArchitect::Graphs::VisualNode, [144](#)
- SelectorRuleClassName
 - DungeonArchitect::PropTypeData, [131](#)
- Seperate

- DungeonArchitect::GridDungeonBuilder, [84](#)
- SetTransform
 - DungeonArchitect::Utils::Matrix, [116](#)
- SetTransformFromMatrix
 - DungeonArchitect::Utils::Matrix, [117](#)
- SetTranslation
 - DungeonArchitect::Utils::Matrix, [117](#)
- Show
 - DungeonArchitect::Editors::GraphContextMenu, [62](#)
- ShowEditor
 - DungeonArchitect::Editors::DungeonEditorHelper, [37](#)
- Size
 - DungeonArchitect::Graphs::GraphNode, [72](#)
- SmoothCell
 - DungeonArchitect::LandscapeDataRasterizer, [97](#)
- SnapToGrid
 - DungeonArchitect::Utils::MathUtils, [113](#)
- SpanningTreeLoopProbability
 - DungeonArchitect::GridDungeonConfig, [88](#)
- StairConnectionTolerance
 - DungeonArchitect::GridDungeonConfig, [88](#)
- Tangent
 - DungeonArchitect::Graphs::GraphPin, [78](#)
- TangentStrength
 - DungeonArchitect::Graphs::GraphPin, [78](#)
- ToIntVector
 - DungeonArchitect::Utils::MathUtils, [113](#)
- ToVector3
 - DungeonArchitect::Utils::MathUtils, [113](#)
- TopZIndex
 - DungeonArchitect::Graphs::Graph, [59](#)
- TransformRuleClassName
 - DungeonArchitect::PropTypeData, [131](#)
- transformRuleClassName
 - DungeonArchitect::Graphs::VisualNode, [144](#)
- transformRuleEnabled
 - DungeonArchitect::Graphs::VisualNode, [144](#)
- TranslationMatrix
 - DungeonArchitect::Utils::Matrix, [117](#)
- Triangle
 - DungeonArchitect::Triangulator::Geometry::↔
Triangle, [142](#)
- Triangulate
 - DungeonArchitect::Triangulator::Delauney, [27](#)
- TriangulateRooms
 - DungeonArchitect::GridDungeonBuilder, [85](#)
- Update
 - DungeonArchitect::Editors::Timer, [140](#)
- UseSelectionRule
 - DungeonArchitect::PropTypeData, [131](#)
- UseTransformRule
 - DungeonArchitect::PropTypeData, [131](#)
- WorldPosition
 - DungeonArchitect::Graphs::GraphPin, [78](#)
- WorldToGrid
 - DungeonArchitect::Utils::MathUtils, [114](#)
- WorldToScreen
 - DungeonArchitect::Graphs::GraphCamera, [61](#)
- WorldToTerrainCoord
 - DungeonArchitect::LandscapeDataRasterizer, [97](#)
- WorldToTerrainTextureCoord
 - DungeonArchitect::LandscapeDataRasterizer, [98](#)
- ZIndex
 - DungeonArchitect::Graphs::GraphNode, [72](#)