

Namespace MoraGames.MapGen

Classes

[AdvancedSettings](#)

[Area](#)

[AreaDefinitionSO](#)

[AreaDefinitionsSOEditor](#)

[AreaOptionsSO](#)

[BakedArea](#)

[BakedAreaGraphExtensions](#)

[BakedGridSpace](#)

[BakedMap](#)

[BakedMapGraphExtensions](#)

[BakedMapPathExtensions](#)

[BasePropCategoriesEditor<T>](#)

[BoundsOverlapCalculator](#)

[Chunk](#)

[Chunks](#)

[ChunksStrategy](#)

[CommonMapGen](#)

[ComplexFloor](#)

[DebugOutputWorldGenInput](#)

[DivideIntoChunks](#)

[GameMap](#)

[GameMapManager](#)

[GameMapManagerEditor](#)

[GameMapSettingsEditor](#)

[GameMapSettingsSO](#)

[Grid2D<T>](#)

[HybridStrategy](#)

[MapGenConstants](#)

[MapGenInspectorLibrary](#)

[MapGenStrategies](#)

[MapSizeUtility](#)

[MapTileBlueprint](#)

[MapTileFactory](#)

[MapTileManager](#)

[Module](#)

[PrefabSelectionBase](#)

[PropCategoriesEditorUI](#)

[RandomStrategy](#)

[ReflectionProbePlacement](#)

[RoomShapeManager](#)

[RoomShapeSO](#)

[RoomShapeSOEditor](#)

[SearchTile](#)

[SearchTileExtensions](#)

[SerializableDictionaryEntryDrawer](#)

[SerializableDictionaryEntry< TKey, TValue >](#)

[SerializableDictionary< TKey, TValue >](#)

[SmartPrefabUtility](#)

[TestOverlap](#)

[TileGridSpace](#)

[TileModuleDefinitionsSO](#)

[TileModuleDefinitionsSOEditor](#)

Structs

[ChunkNeighborPair](#)

[ConnectingTiles](#)

[MapGenInspectorLibrary.GraphAxisData](#)

[PotentialConnection](#)

[TileEdgePair](#)

Interfaces

[IMapGenPostProcess](#)

[IMapGenStrategy](#)

Enums

[BakedAreaType](#)

[BlueprintRotation](#)

[ChunkRoomConnectionMode](#)

[Corner](#)

[DoorPlacementMode](#)

[Edge](#)

[EnvType](#)

[ModuleType](#)

[RoomCreationMode](#)

Class AdvancedSettings

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class AdvancedSettings
```

Inheritance

[object](#) ← AdvancedSettings

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

AdvancedSettings()

```
public AdvancedSettings()
```

Fields

corridorPixelColor

```
public Color corridorPixelColor
```

Field Value

Color

corridorWeight

```
public float corridorWeight
```

Field Value

[float](#) ↗

deadEndWeight

```
public float deadEndWeight
```

Field Value

[float](#) ↗

roomPixelColor

```
public Color roomPixelColor
```

Field Value

Color

roomWeight

```
public float roomWeight
```

Field Value

[float](#) ↗

spriteTemplatesPath

```
public string spriteTemplatesPath
```

Field Value

[string](#) ↗

wallPixelColor

public Color wallPixelColor

Field Value

Color

Class Area

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class Area
```

Inheritance

[object](#) ← Area

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

Area(int, List<TileGridSpace>)

```
public Area(int id, List<TileGridSpace> tiles)
```

Parameters

id [int](#)

tiles [List](#)<[TileGridSpace](#)>

Fields

PropSpots

```
public List<PropSpot> PropSpots
```

Field Value

[List](#)<[PropSpot](#)>

ValidDoorEdges

```
public List<TileEdgePair> ValidDoorEdges
```

Field Value

[List ↗ <TileEdgePair>](#)

Properties

AreaDefinition

```
public AreaDefinitionSO AreaDefinition { get; }
```

Property Value

[AreaDefinitionSO](#)

ID

```
public int ID { get; }
```

Property Value

[int ↗](#)

Tiles

```
public List<TileGridSpace> Tiles { get; }
```

Property Value

[List ↗ <TileGridSpace>](#)

Methods

GetSpecificDoorEdge(TileGridSpace, Edge)

```
public TileEdgePair GetSpecificDoorEdge(TileGridSpace tile, Edge edge)
```

Parameters

tile [TileGridSpace](#)

edge [Edge](#)

Returns

[TileEdgePair](#)

SetAreaDefinition(AreaDefinitionSO)

```
public void SetAreaDefinition(AreaDefinitionSO areaDefinition)
```

Parameters

areaDefinition [AreaDefinitionSO](#)

Class AreaDefinitionSO

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class AreaDefinitionSO : ScriptableObject
```

Inheritance

[object](#) ← AreaDefinitionSO

Constructors

AreaDefinitionSO()

```
public AreaDefinitionSO()
```

Properties

PropCategories

```
public PropCategories PropCategories { get; }
```

Property Value

[PropCategories](#)

PropDensity

```
public float PropDensity { get; }
```

Property Value

[float](#)

TileModuleDefinitions

```
public TileModuleDefinitionsSO TileModuleDefinitions { get; }
```

Property Value

[TileModuleDefinitionsSO](#)

Methods

GetPropChannelDensity(PropChannel)

```
public float GetPropChannelDensity(PropChannel channel)
```

Parameters

channel [PropChannel](#)

Returns

[float](#) ↗

Class AreaDefinitionsSOEditor

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class AreaDefinitionsSOEditor : BasePropCategoriesEditor<AreaDefinitionSO>
```

Inheritance

[object](#) ← [BasePropCategoriesEditor<AreaDefinitionSO>](#) ← [AreaDefinitionsSOEditor](#)

Inherited Members

[BasePropCategoriesEditor<AreaDefinitionSO>.propCategoriesUI](#) ,
[BasePropCategoriesEditor<AreaDefinitionSO>.OnEnable\(\)](#)

Constructors

[AreaDefinitionsSOEditor\(\)](#)

```
public AreaDefinitionsSOEditor()
```

Methods

[OnInspectorGUI\(\)](#)

```
public override void OnInspectorGUI()
```

Class AreaOptionsSO

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class AreaOptionsSO : ScriptableObject
```

Inheritance

[object](#) ← AreaOptionsSO

Constructors

AreaOptionsSO()

```
public AreaOptionsSO()
```

Fields

DefaultCorridor

```
public AreaDefinitionSO DefaultCorridor
```

Field Value

[AreaDefinitionSO](#)

DefaultRoom

```
public AreaDefinitionSO DefaultRoom
```

Field Value

[AreaDefinitionSO](#)

OptionalCommonCorridors

```
public AreaDefinitionSO[] OptionalCommonCorridors
```

Field Value

[AreaDefinitionSO\[\]](#)

OptionalCommonRooms

```
public AreaDefinitionSO[] OptionalCommonRooms
```

Field Value

[AreaDefinitionSO\[\]](#)

OptionalUniqueCorridors

```
public AreaDefinitionSO[] OptionalUniqueCorridors
```

Field Value

[AreaDefinitionSO\[\]](#)

OptionalUniqueRooms

```
public AreaDefinitionSO[] OptionalUniqueRooms
```

Field Value

[AreaDefinitionSO\[\]](#)

RequiredUniqueCorridors

```
public AreaDefinitionSO[] RequiredUniqueCorridors
```

Field Value

[AreaDefinitionSO\[\]](#)

RequiredUniqueRooms

```
public AreaDefinitionSO[] RequiredUniqueRooms
```

Field Value

[AreaDefinitionSO\[\]](#)

TileModuleEdgeSize

```
public int TileModuleEdgeSize
```

Field Value

[int↗](#)

Class BakedArea

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class BakedArea
```

Inheritance

[object](#) ← BakedArea

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Extension Methods

[BakedAreaGraphExtensions.GetFarthestAreas\(BakedArea, BakedMap, out List<BakedArea>, BakedAreaType?\)](#) ,
[BakedAreaGraphExtensions.GetImmeditateGraphConnections\(BakedArea, BakedMap, List<BakedArea>\)](#) ,
[BakedAreaGraphExtensions.TryGetGraphToArea\(BakedArea, BakedMap, BakedArea, out List<BakedArea>\)](#) ,
[BakedAreaGraphExtensions.TryGetGraphToArea\(BakedArea, BakedMap, BakedArea, out List<BakedArea>, List<BakedArea>\)](#).

Constructors

BakedArea(BakedMap, Area, BakedAreaType)

```
public BakedArea(BakedMap bakedMap, Area area, BakedAreaType areaType)
```

Parameters

bakedMap [BakedMap](#)

area [Area](#)

areaType [BakedAreaType](#)

Properties

AreaDefinition

```
public AreaDefinitionSO AreaDefinition { get; }
```

Property Value

[AreaDefinitionSO](#)

AreaType

```
public BakedAreaType AreaType { get; }
```

Property Value

[BakedAreaType](#)

CentralGridSpace

```
public BakedGridSpace CentralGridSpace { get; }
```

Property Value

[BakedGridSpace](#)

ConnectedCorridors

```
public List<int> ConnectedCorridorIds { get; }
```

Property Value

[List<int>](#)

ConnectedRoomIds

```
public List<int> ConnectedRoomIds { get; }
```

Property Value

[List](#) <[int](#)>

DoorSpaces

```
public List<BakedGridSpace> DoorSpaces { get; }
```

Property Value

[List](#) <[BakedGridSpace](#)>

GridSpaces

```
public BakedGridSpace[] GridSpaces { get; }
```

Property Value

[BakedGridSpace](#)[]

ID

```
public int ID { get; }
```

Property Value

[int](#)

Name

```
public string Name { get; }
```

Property Value

[string](#)

Methods

GenerateGraphConnections(BakedMap)

```
public void GenerateGraphConnections(BakedMap context)
```

Parameters

context [BakedMap](#)

GetAllProceduralPropSpots(bool, bool)

```
public List<ProceduralPropSpot> GetAllProceduralPropSpots(bool randomOrder,  
bool includeUsed)
```

Parameters

randomOrder [bool](#)

includeUsed [bool](#)

Returns

[List](#) <[ProceduralPropSpot](#)>

GetAllPropSpots(bool, bool)

```
public List<PropSpot> GetAllPropSpots(bool randomOrder, bool includeUsed)
```

Parameters

randomOrder [bool](#)

includeUsed [bool](#)

Returns

[List](#)<[PropSpot](#)>

GetProceduralPropSpotByType(PropPlacementType, bool, bool)

```
public List<ProceduralPropSpot> GetProceduralPropSpotByType(PropPlacementType placementType,  
    bool randomOrder, bool includeUsed)
```

Parameters

placementType [PropPlacementType](#)

randomOrder [bool](#)

includeUsed [bool](#)

Returns

[List](#)<[ProceduralPropSpot](#)>

GetPropSpotsByChannel(PropChannel, bool, bool)

```
public List<PropSpot> GetPropSpotsByChannel(PropChannel channel, bool randomOrder,  
    bool includeUsed)
```

Parameters

channel [PropChannel](#)

randomOrder [bool](#)

includeUsed [bool](#)

Returns

[List ↗ <PropSpot>](#)

Class BakedAreaGraphExtensions

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class BakedAreaGraphExtensions
```

Inheritance

[object](#) ← BakedAreaGraphExtensions

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

GetFarthestAreas(BakedArea, BakedMap, out List<BakedArea>, BakedAreaType?)

```
public static int GetFarthestAreas(this BakedArea area, BakedMap context, out  
List<BakedArea> farthestAreas, BakedAreaType? typeFilter = null)
```

Parameters

area [BakedArea](#)

context [BakedMap](#)

farthestAreas [List](#)<[BakedArea](#)>

typeFilter [BakedAreaType](#)?

Returns

[int](#)

GetImmediateGraphConnections(BakedArea, BakedMap, List<BakedArea>)

```
public static List<BakedArea> GetImmediateGraphConnections(this BakedArea area, BakedMap context, List<BakedArea> exclude)
```

Parameters

area [BakedArea](#)

context [BakedMap](#)

exclude [List<BakedArea>](#)

Returns

[List<BakedArea>](#)

TryGetGraphToArea(BakedArea, BakedMap, BakedArea, out List<BakedArea>)

```
public static bool TryGetGraphToArea(this BakedArea thisArea, BakedMap context, BakedArea area, out List<BakedArea> graph)
```

Parameters

thisArea [BakedArea](#)

context [BakedMap](#)

area [BakedArea](#)

graph [List<BakedArea>](#)

Returns

[bool](#)

TryGetGraphToArea(BakedArea, BakedMap, BakedArea, out List<BakedArea>, List<BakedArea>)

```
public static bool TryGetGraphToArea(this BakedArea thisArea, BakedMap context, BakedArea area, out List<BakedArea> graph, List<BakedArea> areasToAvoid)
```

Parameters

thisArea [BakedArea](#)

context [BakedMap](#)

area [BakedArea](#)

graph [List](#)<[BakedArea](#)>

areasToAvoid [List](#)<[BakedArea](#)>

Returns

[bool](#)

Enum BakedAreaType

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public enum BakedAreaType
```

Fields

Corridor = 1

Room = 0

Class BakedGridSpace

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class BakedGridSpace
```

Inheritance

[object](#) ← BakedGridSpace

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

BakedGridSpace(TileGridSpace, Vector3, GameObject[])

```
public BakedGridSpace(TileGridSpace tile, Vector3 centerWorldPosition,  
GameObject[] doorObjects)
```

Parameters

tile [TileGridSpace](#)

centerWorldPosition Vector3

doorObjects GameObject[]

Fields

DoorObjects

```
public GameObject[] DoorObjects
```

Field Value

GameObject[]

name

`public string name`

Field Value

[string](#) ↗

Properties

CenterType

`public EnvType CenterType { get; }`

Property Value

[EnvType](#)

CenterWorldPosition

`public Vector3 CenterWorldPosition { get; }`

Property Value

Vector3

DoorEdges

`public Edge[] DoorEdges { get; }`

Property Value

[Edge](#)[]

GridPosition

`public Vector2Int GridPosition { get; }`

Property Value

Vector2Int

IsDoorSpace

`public bool IsDoorSpace { get; }`

Property Value

[bool](#) ↗

PixelEnvTypes

`public EnvType[] PixelEnvTypes { get; }`

Property Value

[EnvType](#)[]

RootObject

`public GameObject RootObject { get; }`

Property Value

GameObject

Methods

GetArea(BakedMap)

```
public BakedArea GetArea(BakedMap context)
```

Parameters

context [BakedMap](#)

Returns

[BakedArea](#)

GetEdgeType(Edge)

```
public EnvType GetEdgeType(Edge edge)
```

Parameters

edge [Edge](#)

Returns

[EnvType](#)

GetEdgesByCenterType(EnvType)

```
public Edge[] GetEdgesByCenterType(EnvType desiredType)
```

Parameters

desiredType [EnvType](#)

Returns

[Edge\[\]](#)

Class BakedMap

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class BakedMap
```

Inheritance

[object](#) ← BakedMap

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Extension Methods

[BakedMapGraphExtensions.Find2FarthestAreas\(BakedMap, List<BakedArea>, out BakedArea, out BakedArea, Random, BakedAreaType?\)](#) ,
[BakedMapGraphExtensions.Find2FarthestAreas\(BakedMap, List<BakedArea>, out BakedArea, out BakedArea, string, BakedAreaType?\)](#) ,
[BakedMapGraphExtensions.Find2FarthestGridSpaces\(BakedMap, out BakedGridSpace, out BakedGridSpace, Random, BakedAreaType?\)](#) ,
[BakedMapGraphExtensions.Find2FarthestGridSpaces\(BakedMap, out BakedGridSpace, out BakedGridSpace, string, BakedAreaType?\)](#) ,
[BakedMapGraphExtensions.GetFarthestGridSpaces\(BakedMap, BakedGridSpace, out List<BakedGridSpace>, BakedAreaType?\)](#) ,
[BakedMapPathExtensions.TryGetPath\(BakedMap, Vector2Int, Vector2Int, out SearchTile, params EnvType\[\]\)](#).

Constructors

BakedMap(GameMapManager)

```
public BakedMap(GameMapManager gameMapManager)
```

Parameters

gameMapManager [GameMapManager](#)

Properties

Corridors

```
public List<BakedArea> Corridors { get; }
```

Property Value

[List](#) <[BakedArea](#)>

GridSpaceCount

```
public int GridSpaceCount { get; }
```

Property Value

[int](#)

this[int, int]

```
public BakedGridSpace this[int x, int y] { get; }
```

Parameters

x [int](#)

y [int](#)

Property Value

[BakedGridSpace](#)

this[Vector2Int]

```
public BakedGridSpace this[Vector2Int position] { get; }
```

Parameters

position Vector2Int

Property Value

[BakedGridSpace](#)

Rooms

```
public List<BakedArea> Rooms { get; }
```

Property Value

[List](#) <[BakedArea](#)>

Waypoints

```
public List<Vector3> Waypoints { get; }
```

Property Value

[List](#) <[Vector3](#)>

transform

```
public Transform transform { get; }
```

Property Value

Transform

Methods

GetAllAreas()

```
public List<BakedArea> GetAllAreas()
```

Returns

[List](#) <[BakedArea](#)>

GetAllGridSpaces(BakedAreaType?)

```
public List<BakedGridSpace> GetAllGridSpaces(BakedAreaType? type = null)
```

Parameters

type [BakedAreaType](#)?

Returns

[List](#) <[BakedGridSpace](#)>

GetArea(BakedGridSpace)

```
public BakedArea GetArea(BakedGridSpace bakedGridSpace)
```

Parameters

bakedGridSpace [BakedGridSpace](#)

Returns

[BakedArea](#)

GetArea(Vector2Int)

```
public BakedArea GetArea(Vector2Int gridPosition)
```

Parameters

gridPosition Vector2Int

Returns

[BakedArea](#)

GetArea(Vector3)

```
public BakedArea GetArea(Vector3 worldPosition)
```

Parameters

worldPosition Vector3

Returns

[BakedArea](#)

GetAreas(AreaDefinitionSO)

```
public BakedArea[] GetAreas(AreaDefinitionSO areaDefinition)
```

Parameters

areaDefinition [AreaDefinitionSO](#)

Returns

[BakedArea\[\]](#)

GetAreas(string)

```
public BakedArea[] GetAreas(string areaDefinitionName)
```

Parameters

`areaDefinitionName` [string](#)

Returns

[BakedArea\[\]](#)

**GetConnectedNeighbors(BakedGridSpace,
HashSet<BakedGridSpace>, params EnvType[])**

```
public List<BakedGridSpace> GetConnectedNeighbors(BakedGridSpace context,  
HashSet<BakedGridSpace> exclude, params EnvType[] validTypes)
```

Parameters

`context` [BakedGridSpace](#)

`exclude` [HashSet](#)<[BakedGridSpace](#)>

`validTypes` [EnvType\[\]](#)

Returns

[List](#)<[BakedGridSpace](#)>

GetCorridor(AreaDefinitionSO)

```
public BakedArea GetCorridor(AreaDefinitionSO areaDefinition)
```

Parameters

`areaDefinition` [AreaDefinitionSO](#)

Returns

[BakedArea](#)

GetCorridor(int)

```
public BakedArea GetCorridor(int id)
```

Parameters

`id int`

Returns

[BakedArea](#)

GetCorridor(string)

```
public BakedArea GetCorridor(string areaDefinitionName)
```

Parameters

`areaDefinitionName string`

Returns

[BakedArea](#)

GetCorridors(AreaDefinitionSO)

```
public BakedArea[] GetCorridors(AreaDefinitionSO areaDefinition)
```

Parameters

`areaDefinition AreaDefinitionSO`

Returns

[BakedArea\[\]](#)

GetCorridors(string)

```
public BakedArea[] GetCorridors(string areaDefinitionName)
```

Parameters

areaDefinitionName [string](#)

Returns

[BakedArea\[\]](#)

GetCorridorsByIds(List<int>)

```
public List<BakedArea> GetCorridorsByIds(List<int> ids)
```

Parameters

ids [List](#)<[int](#)>

Returns

[List](#)<[BakedArea](#)>

GetGridSpace(Vector3)

```
public BakedGridSpace GetGridSpace(Vector3 worldPosition)
```

Parameters

worldPosition [Vector3](#)

Returns

[BakedGridSpace](#)

GetLocalPosition(Vector2Int)

```
public Vector3 GetLocalPosition(Vector2Int gridPosition)
```

Parameters

gridPosition Vector2Int

Returns

Vector3

GetLocalPosition(Vector3)

```
public Vector3 GetLocalPosition(Vector3 worldPosition)
```

Parameters

worldPosition Vector3

Returns

Vector3

GetNeighbor(BakedGridSpace, Edge)

```
public BakedGridSpace GetNeighbor(BakedGridSpace context, Edge neighborDirection)
```

Parameters

context [BakedGridSpace](#)

neighborDirection [Edge](#)

Returns

[BakedGridSpace](#)

GetNeighbor(Vector2Int, Edge)

```
public BakedGridSpace GetNeighbor(Vector2Int context, Edge neighborDirection)
```

Parameters

context Vector2Int

neighborDirection [Edge](#)

Returns

[BakedGridSpace](#)

GetNeighborsConnected(BakedGridSpace, Edge, out BakedGridSpace)

```
public bool GetNeighborsConnected(BakedGridSpace context, Edge neighborDirection, out  
BakedGridSpace neighbor)
```

Parameters

context [BakedGridSpace](#)

neighborDirection [Edge](#)

neighbor [BakedGridSpace](#)

Returns

[bool](#) ↗

GetNeighborsConnected(BakedGridSpace, Edge, out BakedGridSpace, params EnvType[])

```
public bool GetNeighborIsConnected(BakedGridSpace context, Edge neighborDirection, out  
BakedGridSpace neighbor, params EnvType[] validEnvTypes)
```

Parameters

context [BakedGridSpace](#)

neighborDirection [Edge](#)

neighbor [BakedGridSpace](#)

validEnvTypes [EnvType](#)[]

Returns

[bool](#)

GetNeighbors(BakedGridSpace)

```
public List<BakedGridSpace> GetNeighbors(BakedGridSpace context)
```

Parameters

context [BakedGridSpace](#)

Returns

[List](#)<[BakedGridSpace](#)>

GetNeighbors(Vector2Int)

```
public List<BakedGridSpace> GetNeighbors(Vector2Int context)
```

Parameters

context Vector2Int

Returns

[List](#) <[BakedGridSpace](#)>

GetRandomArea()

```
public BakedArea GetRandomArea()
```

Returns

[BakedArea](#)

GetRandomCorridor()

```
public BakedArea GetRandomCorridor()
```

Returns

[BakedArea](#)

GetRandomRoom()

```
public BakedArea GetRandomRoom()
```

Returns

[BakedArea](#)

GetRandomWaypoint()

```
public Vector3 GetRandomWaypoint()
```

Returns

GetRoom(AreaDefinitionSO)

```
public BakedArea GetRoom(AreaDefinitionSO areaDefinition)
```

Parameters

areaDefinition [AreaDefinitionSO](#)

Returns

[BakedArea](#)

GetRoom(int)

```
public BakedArea GetRoom(int id)
```

Parameters

id [int](#)

Returns

[BakedArea](#)

GetRoom(string)

```
public BakedArea GetRoom(string areaDefinitionName)
```

Parameters

areaDefinitionName [string](#)

Returns

[BakedArea](#)

GetRooms(AreaDefinitionSO)

```
public BakedArea[] GetRooms(AreaDefinitionSO areaDefinition)
```

Parameters

[areaDefinition](#) [AreaDefinitionSO](#)

Returns

[BakedArea\[\]](#)

GetRooms(string)

```
public BakedArea[] GetRooms(string areaDefinitionName)
```

Parameters

[areaDefinitionName](#) [string](#)

Returns

[BakedArea\[\]](#)

GetRoomsByIds(List<int>)

```
public List<BakedArea> GetRoomsByIds(List<int> ids)
```

Parameters

[ids](#) [List](#)<[int](#)>

Returns

GetWorldPosition(Vector2Int)

```
public Vector3 GetWorldPosition(Vector2Int gridPosition)
```

Parameters

gridPosition Vector2Int

Returns

Vector3

Class BakedMapGraphExtensions

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class BakedMapGraphExtensions
```

Inheritance

[object](#) ← BakedMapGraphExtensions

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

Find2FarthestAreas(BakedMap, List<BakedArea>, out
BakedArea, out BakedArea, Random, BakedAreaType?)

```
public static int Find2FarthestAreas(this BakedMap map, List<BakedArea> validAreas, out  
BakedArea a, out BakedArea b, Random rng, BakedAreaType? filter = null)
```

Parameters

map [BakedMap](#)

validAreas [List](#)<[BakedArea](#)>

a [BakedArea](#)

b [BakedArea](#)

rng [Random](#)

filter [BakedAreaType](#)?

Returns

[int](#)

Find2FarthestAreas(BakedMap, List<BakedArea>, out BakedArea, out BakedArea, string, BakedAreaType?)

```
public static int Find2FarthestAreas(this BakedMap map, List<BakedArea> validAreas, out BakedArea a, out BakedArea b, string seedKey, BakedAreaType? filter = null)
```

Parameters

map [BakedMap](#)

validAreas [List](#)<[BakedArea](#)>

a [BakedArea](#)

b [BakedArea](#)

seedKey [string](#)

filter [BakedAreaType](#)?

Returns

[int](#)

Find2FarthestGridSpaces(BakedMap, out BakedGridSpace, out BakedGridSpace, Random, BakedAreaType?)

```
public static int Find2FarthestGridSpaces(this BakedMap map, out BakedGridSpace a, out BakedGridSpace b, Random rng, BakedAreaType? filter = null)
```

Parameters

map [BakedMap](#)

a [BakedGridSpace](#)

b [BakedGridSpace](#)

`rng` [Random](#)

`filter` [BakedAreaType](#)?

Returns

[int](#)

`Find2FarthestGridSpaces(BakedMap, out BakedGridSpace, out BakedGridSpace, string, BakedAreaType?)`

```
public static int Find2FarthestGridSpaces(this BakedMap map, out BakedGridSpace a, out BakedGridSpace b, string seedKey, BakedAreaType? filter = null)
```

Parameters

`map` [BakedMap](#)

`a` [BakedGridSpace](#)

`b` [BakedGridSpace](#)

`seedKey` [string](#)

`filter` [BakedAreaType](#)?

Returns

[int](#)

`GetFarthestGridSpaces(BakedMap, BakedGridSpace, out List<BakedGridSpace>, BakedAreaType?)`

```
public static int GetFarthestGridSpaces(this BakedMap map, BakedGridSpace fromSpace, out List<BakedGridSpace> farthestGridSpaces, BakedAreaType? typeFilter = null)
```

Parameters

`map` [BakedMap](#)

`fromSpace` [BakedGridSpace](#)

`farthestGridSpaces` [List](#) <[BakedGridSpace](#)>

`typeFilter` [BakedAreaType](#)?

Returns

[int](#)

Class BakedMapPathExtensions

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class BakedMapPathExtensions
```

Inheritance

[object](#) ← BakedMapPathExtensions

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

TryGetPath(BakedMap, Vector2Int, Vector2Int, out SearchTile, params EnvType[])

```
public static bool TryGetPath(this BakedMap bakedMap, Vector2Int start, Vector2Int end, out  
SearchTile pathEnd, params EnvType[] validTypes)
```

Parameters

bakedMap [BakedMap](#)

start Vector2Int

end Vector2Int

pathEnd [SearchTile](#)

validTypes [EnvType](#)[]

Returns

[bool](#)

Class BasePropCategoriesEditor<T>

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public abstract class BasePropCategoriesEditor<T> : Editor where T : ScriptableObject
```

Type Parameters

T

Inheritance

[object](#) ← BasePropCategoriesEditor<T>

Derived

[AreaDefinitionsSOEditor](#), [TileModuleDefinitionsSOEditor](#)

Constructors

BasePropCategoriesEditor()

```
protected BasePropCategoriesEditor()
```

Fields

_propCategoriesUI

```
protected PropCategoriesEditorUI _propCategoriesUI
```

Field Value

[PropCategoriesEditorUI](#)

Methods

OnEnable()

```
public virtual void OnEnable()
```

OnInspectorGUI()

```
public override void OnInspectorGUI()
```

Enum BlueprintRotation

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public enum BlueprintRotation
```

Fields

c180 = 1

c90 = 0

cc90 = 2

none = 3

Class BoundsOverlapCalculator

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class BoundsOverlapCalculator
```

Inheritance

[object](#) ← BoundsOverlapCalculator

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

GetBoundsOverlap(Bounds, Bounds)

```
public static Vector3 GetBoundsOverlap(Bounds bounds1, Bounds bounds2)
```

Parameters

bounds1 Bounds

bounds2 Bounds

Returns

Vector3

GetLeastOverlap(Bounds, Bounds)

```
public static float GetLeastOverlap(Bounds bounds1, Bounds bounds2)
```

Parameters

bounds1 Bounds

bounds2 Bounds

Returns

[float](#) ↗

Class Chunk

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class Chunk
```

Inheritance

[object](#) ← Chunk

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

Chunk(Vector2Int, Vector2Int, Vector2Int)

```
public Chunk(Vector2Int chunkGridPos, Vector2Int xyStart, Vector2Int dimensions)
```

Parameters

chunkGridPos Vector2Int

xyStart Vector2Int

dimensions Vector2Int

Fields

chunkGridPos

```
public Vector2Int chunkGridPos
```

Field Value

Vector2Int

dimensions

`public Vector2Int dimensions`

Field Value

Vector2Int

isAssignedRoom

`public bool isAssignedRoom`

Field Value

[bool](#)

isMerged

`public bool isMerged`

Field Value

[bool](#)

mergedChunk

`public Chunk mergedChunk`

Field Value

[Chunk](#)

room

```
public Area room
```

Field Value

[Area](#)

xyStart

```
public Vector2Int xyStart
```

Field Value

Vector2Int

Methods

InfoString()

```
public string InfoString()
```

Returns

[string](#) ↗

Struct ChunkNeighborPair

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public struct ChunkNeighborPair
```

Inherited Members

[ValueType.Equals\(object\)](#) , [ValueType.GetHashCode\(\)](#) , [ValueType.ToString\(\)](#) , [object.GetType\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

ChunkNeighborPair(Chunk, Chunk)

```
public ChunkNeighborPair(Chunk chunkA, Chunk chunkB)
```

Parameters

chunkA [Chunk](#)

chunkB [Chunk](#)

Fields

chunkA

```
public Chunk chunkA
```

Field Value

[Chunk](#)

chunkB

```
public Chunk chunkB
```

Field Value

[Chunk](#)

Enum ChunkRoomConnectionMode

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public enum ChunkRoomConnectionMode
```

Fields

Neighbors = 0

Proximity = 1

Class Chunks

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class Chunks
```

Inheritance

[object](#) ← Chunks

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

GetChunkPairsNeighboringSet(Grid2D<Chunk>, List<Chunk>, List<Chunk>)

```
public static List<ChunkNeighborPair> GetChunkPairsNeighboringSet(Grid2D<Chunk> context,  
List<Chunk> chunksSet, List<Chunk> desiredNeighborSet)
```

Parameters

context [Grid2D<Chunk>](#)

chunksSet [List<Chunk>](#)

desiredNeighborSet [List<Chunk>](#)

Returns

[List<ChunkNeighborPair>](#)

GetChunksNeighboringSet(Grid2D<Chunk>, List<Chunk>, Chunk, List<Chunk>)

```
public static List<Chunk> GetChunksNeighboringSet(Grid2D<Chunk> context, List<Chunk>  
chunksSet, Chunk previousChunk, List<Chunk> desiredNeighborSet)
```

Parameters

context [Grid2D<Chunk>](#)

chunksSet [List<Chunk>](#)

previousChunk [Chunk](#)

desiredNeighborSet [List<Chunk>](#)

Returns

[List<Chunk>](#)

GetValidNextChunks(Grid2D<Chunk>, Chunk, Chunk,
List<Chunk>)

```
public static List<Chunk> GetValidNextChunks(Grid2D<Chunk> context, Chunk current, Chunk  
previous, List<Chunk> alreadyTouched)
```

Parameters

context [Grid2D<Chunk>](#)

current [Chunk](#)

previous [Chunk](#)

alreadyTouched [List<Chunk>](#)

Returns

[List<Chunk>](#)

Class ChunksStrategy

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class ChunksStrategy : IMapGenStrategy
```

Inheritance

[object](#) ← ChunksStrategy

Implements

[IMapGenStrategy](#)

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

ChunksStrategy()

```
public ChunksStrategy()
```

Properties

NumChunks

```
public int NumChunks { get; }
```

Property Value

[int](#)

Methods

AddRoom(GameMapManager, RoomShapeSO, Vector2Int)

```
public static List<TileGridSpace> AddRoom(GameMapManager context, RoomShapeSO roomShape,  
Vector2Int startingCoords)
```

Parameters

context [GameMapManager](#)

roomShape [RoomShapeSO](#)

startingCoords Vector2Int

Returns

[List](#) <[TileGridSpace](#)>

CreateChunksGrid(GameMapManager)

```
public static Grid2D<Chunk> CreateChunksGrid(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Returns

[Grid2D](#) <[Chunk](#)>

GetMapGenCoroutine(GameMapManager)

```
public IEnumerator GetMapGenCoroutine(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Returns

[IEnumerator](#)

MergeChunks(GameMapManager, Grid2D<Chunk>)

```
public static void MergeChunks(GameMapManager context, Grid2D<Chunk> chunksGrid)
```

Parameters

context [GameMapManager](#)

chunksGrid [Grid2D<Chunk>](#)

PickRoomShapePerChunk(GameMapManager, Grid2D<Chunk>, List<TileGridSpace>, float)

```
public static IEnumerator PickRoomShapePerChunk(GameMapManager context, Grid2D<Chunk> chunksGrid, List<TileGridSpace> chunkTiles, float percentOfChunks)
```

Parameters

context [GameMapManager](#)

chunksGrid [Grid2D<Chunk>](#)

chunkTiles [List<TileGridSpace>](#)

percentOfChunks [float](#)

Returns

[IEnumerator](#)

Class CommonMapGen

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class CommonMapGen
```

Inheritance

[object](#) ← CommonMapGen

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

AssignIds(GameMap)

```
public static void AssignIds(GameMap context)
```

Parameters

context [GameMap](#)

CollapseExteriorCeiling(GameMapManager)

```
public static IEnumerator CollapseExteriorCeiling(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Returns

[IEnumerator](#)

CollapseRemainingTiles(GameMapManager)

```
public static IEnumerator CollapseRemainingTiles(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Returns

[IEnumerator](#)

CollapseTile(GameMapManager, RoomShapeSO, Vector2Int, Vector2Int)

```
public static TileGridSpace CollapseTile(GameMapManager context, RoomShapeSO roomShape, Vector2Int startingCoords, Vector2Int dimensions)
```

Parameters

context [GameMapManager](#)

roomShape [RoomShapeSO](#)

startingCoords Vector2Int

dimensions Vector2Int

Returns

[TileGridSpace](#)

CommonMapGenCoroutine(GameMapManager)

```
public static IEnumerator CommonMapGenCoroutine(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Returns

[IEnumerator](#)

ConnectAllRoomsRandom(GameMapManager)

```
public static IEnumerator ConnectAllRoomsRandom(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Returns

[IEnumerator](#)

FindAllValidDoorEdges(GameMap)

```
public static void FindAllValidDoorEdges(GameMap context)
```

Parameters

context [GameMap](#)

GetNearestConnectingTiles(Area, Area)

```
public static ConnectingTiles GetNearestConnectingTiles(Area startingRoom, Area nextRoom)
```

Parameters

startingRoom [Area](#)

nextRoom [Area](#)

Returns

[ConnectingTiles](#)

GetNearestConnectingTiles(Area, Area, ConnectingTiles[], List<PotentialConnection>)

```
public static ConnectingTiles GetNearestConnectingTiles(Area startingRoom, Area nextRoom,  
ConnectingTiles[] invalidConnections, List<PotentialConnection> possibleCombos )
```

Parameters

startingRoom [Area](#)

nextRoom [Area](#)

invalidConnections [ConnectingTiles\[\]](#)

possibleCombos [List<PotentialConnection>](#)

Returns

[ConnectingTiles](#)

GetNearestRooms(Area, List<Area>)

```
public static List<Area> GetNearestRooms(Area context, List<Area> validRooms )
```

Parameters

context [Area](#)

validRooms [List<Area>](#)

Returns

[List<Area>](#)

GetNearestRooms(Area, List<Area>, int)

```
public static List<Area> GetNearestRooms(Area context, List<Area> validRooms,  
int tieThreshold)
```

Parameters

context [Area](#)

validRooms [List](#)<[Area](#)>

tieThreshold [int](#)

Returns

[List](#)<[Area](#)>

GetNearestRooms(Area, List<Area>, out int)

```
public static List<Area> GetNearestRooms(Area context, List<Area> validRooms, out  
int distance)
```

Parameters

context [Area](#)

validRooms [List](#)<[Area](#)>

distance [int](#)

Returns

[List](#)<[Area](#)>

GetOrthoDistance(Vector2Int, Vector2Int)

```
public static int GetOrthoDistance(Vector2Int posA, Vector2Int posB)
```

Parameters

`posA` `Vector2Int`

`posB` `Vector2Int`

Returns

`int`

`GetRandomConnectingTiles(Area, Area, ConnectingTiles[], List<PotentialConnection>)`

```
public static ConnectingTiles GetRandomConnectingTiles(Area startingRoom, Area nextRoom,  
ConnectingTiles[] invalidConnections, List<PotentialConnection> possibleCombos)
```

Parameters

`startingRoom` [Area](#)

`nextRoom` [Area](#)

`invalidConnections` [ConnectingTiles\[\]](#)

`possibleCombos` [List](#)<[PotentialConnection](#)>

Returns

[ConnectingTiles](#)

`GetRoomDistance(Area, Area)`

```
public static int GetRoomDistance(Area context, Area otherRoom)
```

Parameters

`context` [Area](#)

`otherRoom` [Area](#)

Returns

[int](#)

GetRoomsWithinDistance(Area, List<Area>, int)

```
public static List<Area> GetRoomsWithinDistance(Area context, List<Area> validRooms,  
int distance)
```

Parameters

context [Area](#)

validRooms [List](#)<[Area](#)>

distance [int](#)

Returns

[List](#)<[Area](#)>

MakeConnectingPath(GameMapManager, ConnectingTiles, SearchTile)

```
public static void MakeConnectingPath(GameMapManager context, ConnectingTiles connection,  
SearchTile pathEnd)
```

Parameters

context [GameMapManager](#)

connection [ConnectingTiles](#)

pathEnd [SearchTile](#)

NarrowDownPathEdge(GameMap, Vector2Int, Edge)

```
public static void NarrowDownPathEdge(GameMap worldMap, Vector2Int gridPos, Edge edge)
```

Parameters

worldMap [GameMap](#)

gridPos Vector2Int

edge [Edge](#)

PairCorridorsToids(GameMapManager)

```
public static bool PairCorridorsToIds(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Returns

[bool](#) ↗

PairRoomsToids(GameMapManager)

```
public static bool PairRoomsToIds(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Returns

[bool](#) ↗

ReCollapseEdgeToType(TileGridSpace, Edge, EnvType)

```
public static void ReCollapseEdgeToType(TileGridSpace tile, Edge edge, EnvType type)
```

Parameters

tile [TileGridSpace](#)

edge [Edge](#)

type [EnvType](#)

ReCollapseTileCompletely(TileGridSpace, EnvType)

```
public static void ReCollapseTileCompletely(TileGridSpace tile, EnvType type)
```

Parameters

tile [TileGridSpace](#)

type [EnvType](#)

ReCollapseTileCompletely(TileGridSpace, EnvType[])

```
public static void ReCollapseTileCompletely(TileGridSpace tile, EnvType[] pixels)
```

Parameters

tile [TileGridSpace](#)

pixels [EnvType\[\]](#)

RemoveCorridors(GameMapManager)

```
public static IEnumerator RemoveCorridors(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Returns

[IEnumerator](#)

SpawnGameObjects(GameMapManager)

```
public static void SpawnGameObjects(GameMapManager context)
```

Parameters

context [GameMapManager](#)

StorePixelChanges(Dictionary<TileGridSpace, EnvType[]>, TileGridSpace, Corner, EnvType)

```
public static void StorePixelChanges(Dictionary<TileGridSpace, EnvType[]> store,  
TileGridSpace tile, Corner corner, EnvType toType)
```

Parameters

store [Dictionary](#)<TileGridSpace, EnvType[]>

tile [TileGridSpace](#)

corner [Corner](#)

toType [EnvType](#)

StorePixelChanges(Dictionary<TileGridSpace, EnvType[]>, TileGridSpace, Edge, EnvType)

```
public static void StorePixelChanges(Dictionary<TileGridSpace, EnvType[]> store,  
TileGridSpace tile, Edge edge, EnvType toType)
```

Parameters

store [Dictionary](#)<TileGridSpace, EnvType[]>

tile [TileGridSpace](#)

edge [Edge](#)

toType [EnvType](#)

StorePixelChanges(Dictionary<TileGridSpace, EnvType[]>, TileGridSpace, EnvType[])

```
public static void StorePixelChanges(Dictionary<TileGridSpace, EnvType[]> store,  
TileGridSpace tile, EnvType[] pixels)
```

Parameters

store [Dictionary](#)<TileGridSpace, EnvType[]>

tile [TileGridSpace](#)

pixels [EnvType](#)[]

TryGetConnectingPathAstar(GameGameManager, Area, Area, out SearchTile, out ConnectingTiles)

```
public static bool TryGetConnectingPathAstar(GameGameManager context, Area roomA, Area roomB,  
out SearchTile pathEnd, out ConnectingTiles connection)
```

Parameters

context [GameGameManager](#)

roomA [Area](#)

roomB [Area](#)

pathEnd [SearchTile](#)

`connection` [ConnectingTiles](#)

Returns

`bool` ↗

`TryMakeRandomValidConnection(Area, List<Area>, int, out SearchTile, out ConnectingTiles, out Area)`

```
public static bool TryMakeRandomValidConnection(Area context, List<Area> validRooms, int tieThreshold, out SearchTile pathEnd, out ConnectingTiles connection, out Area touchedRoom)
```

Parameters

`context` [Area](#)

`validRooms` [List](#) ↗<[Area](#)>

`tieThreshold` [int](#) ↗

`pathEnd` [SearchTile](#)

`connection` [ConnectingTiles](#)

`touchedRoom` [Area](#)

Returns

`bool` ↗

`UncollapseTile(TileGridSpace)`

```
public static void UncollapseTile(TileGridSpace tile)
```

Parameters

`tile` [TileGridSpace](#)

UncollapseTile(TileGridSpace, EnvType)

```
public static void UncollapseTile(TileGridSpace tile, EnvType resetNarrowingToType)
```

Parameters

tile [TileGridSpace](#)

resetNarrowingToType [EnvType](#)

UncollapseTile(TileGridSpace, EnvType[])

```
public static void UncollapseTile(TileGridSpace tile, EnvType[] resetNarrowingToTypes)
```

Parameters

tile [TileGridSpace](#)

resetNarrowingToTypes [EnvType\[\]](#)

ValidateMap(GameMapManager)

```
public static bool ValidateMap(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Returns

[bool](#) ↗

Class ComplexFloor

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class ComplexFloor : MonoBehaviour
```

Inheritance

[object](#) ← ComplexFloor

Constructors

ComplexFloor()

```
public ComplexFloor()
```

Methods

Initialize(MapTileBlueprint, int)

```
public void Initialize(MapTileBlueprint blueprint, int index)
```

Parameters

blueprint [MapTileBlueprint](#)

index [int](#)

Struct ConnectingTiles

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public struct ConnectingTiles
```

Inherited Members

[ValueType.Equals\(object\)](#) , [ValueType.GetHashCode\(\)](#) , [ValueType.ToString\(\)](#) , [object.GetType\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

ConnectingTiles(Vector2Int, Vector2Int)

```
public ConnectingTiles(Vector2Int start, Vector2Int end)
```

Parameters

start Vector2Int

end Vector2Int

Fields

endPosition

```
public Vector2Int endPosition
```

Field Value

Vector2Int

offset

```
public Vector2Int offset
```

Field Value

Vector2Int

startPosition

```
public Vector2Int startPosition
```

Field Value

Vector2Int

Enum Corner

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public enum Corner
```

Fields

BottomLeft = 0

BottomRight = 1

TopLeft = 2

TopRight = 3

Class DebugOutputWorldGenInput

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class DebugOutputWorldGenInput
```

Inheritance

[object](#) ← DebugOutputWorldGenInput

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

DebugOutputWorldGenInput()

```
public DebugOutputWorldGenInput()
```

Methods

OverwriteOutput(string)

```
public void OverwriteOutput(string log)
```

Parameters

log [string](#)

Class DivideIntoChunks

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class DivideIntoChunks
```

Inheritance

[object](#) ← DivideIntoChunks

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

SplitIntoChunks(int, int, int, bool)

```
public static List<int> SplitIntoChunks(int totalItems, int minChunkSize, int maxChunkSize,  
bool weighted)
```

Parameters

totalItems [int](#)

minChunkSize [int](#)

maxChunkSize [int](#)

weighted [bool](#)

Returns

[List](#)<[int](#)>

Enum DoorPlacementMode

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public enum DoorPlacementMode
```

Fields

HalfFrames = 1

WholeDoor = 0

Enum Edge

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public enum Edge
```

Fields

Bottom = 1

Left = 2

Right = 4

Top = 3

Uninitialized = 0

Enum EnvType

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public enum EnvType
```

Fields

Corridor = 2

Room = 1

Uninitialized = 0

Wall = 3

Class GameMap

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class GameMap
```

Inheritance

[object](#) ← GameMap

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

GameMap(GameMapManager)

```
public GameMap(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Fields

DoorObjectsByGridPosition

```
public readonly Dictionary<Vector2Int, List<GameObject>> DoorObjectsByGridPosition
```

Field Value

[Dictionary](#)<Vector2Int, [List](#)<GameObject>>

DoorPositions

```
public readonly List<Vector2Int> DoorPositions
```

Field Value

[List](#) <Vector2Int>

ExteriorCeiling

```
public readonly List<TileGridSpace> ExteriorCeiling
```

Field Value

[List](#) <[TileGridSpace](#)>

UnassignedTiles

```
public List<TileGridSpace> UnassignedTiles
```

Field Value

[List](#) <[TileGridSpace](#)>

UncollapsedCorridors

```
public readonly List<TileGridSpace> UncollapsedCorridors
```

Field Value

[List](#) <[TileGridSpace](#)>

UncollapsedNonCorridors

```
public readonly List<TileGridSpace> UncollapsedNonCorridors
```

Field Value

[List ↗ <TileGridSpace>](#)

UncollapsedTiles

```
public readonly List<TileGridSpace> UncollapsedTiles
```

Field Value

[List ↗ <TileGridSpace>](#)

UniqueCorridors

```
public readonly List<Area> UniqueCorridors
```

Field Value

[List ↗ <Area>](#)

UniqueRooms

```
public readonly List<Area> UniqueRooms
```

Field Value

[List ↗ <Area>](#)

Properties

GridAsArray

```
public TileGridSpace[] GridAsArray { get; }
```

Property Value

[TileGridSpace\[\]](#)

this[int, int]

```
public TileGridSpace this[int x, int y] { get; }
```

Parameters

x [int](#)

y [int](#)

Property Value

[TileGridSpace](#)

this[Vector2Int]

```
public TileGridSpace this[Vector2Int position] { get; }
```

Parameters

position Vector2Int

Property Value

[TileGridSpace](#)

Manager

```
public GameMapManager Manager { get; }
```

Property Value

[GameMapManager](#)

MapSize

```
public Vector2Int MapSize { get; }
```

Property Value

Vector2Int

PlayArea

```
public int PlayArea { get; }
```

Property Value

[int ↗](#)

PlaySize

```
public Vector2Int PlaySize { get; }
```

Property Value

Vector2Int

TotalArea

```
public int TotalArea { get; }
```

Property Value

Methods

GetAllAdjacentTilesFromSelection(List<TileGridSpace>, List<TileGridSpace>, bool)

```
public List<TileGridSpace> GetAllAdjacentTilesFromSelection(List<TileGridSpace> selection,  
List<TileGridSpace> possibleTiles, bool includeDiagonals)
```

Parameters

selection [List](#)<[TileGridSpace](#)>

possibleTiles [List](#)<[TileGridSpace](#)>

includeDiagonals [bool](#)

Returns

[List](#)<[TileGridSpace](#)>

GetAllCollapsedTilesByType(EnvType)

```
public List<TileGridSpace> GetAllCollapsedTilesByType(EnvType type)
```

Parameters

type [EnvType](#)

Returns

[List](#)<[TileGridSpace](#)>

GetCollapsedConnectedNeighbors(TileGridSpace, EnvType[]])

```
public List<TileGridSpace> GetCollapsedConnectedNeighbors(TileGridSpace context,  
EnvType[] connectingTypes)
```

Parameters

context [TileGridSpace](#)

connectingTypes [EnvType](#)[]

Returns

[List](#)<[TileGridSpace](#)>

GetCollapsedNeighborsOfEnvType(TileGridSpace, EnvType, bool)

```
public List<TileGridSpace> GetCollapsedNeighborsOfEnvType(TileGridSpace context, EnvType  
type, bool respectWalls)
```

Parameters

context [TileGridSpace](#)

type [EnvType](#)

respectWalls [bool](#)

Returns

[List](#)<[TileGridSpace](#)>

GetContiguousTilesOfEnvType(TileGridSpace, EnvType, bool, bool)

```
public List<TileGridSpace> GetContiguousTilesOfEnvType(TileGridSpace context, EnvType type,  
bool includeSelf, bool respectWalls)
```

Parameters

context [TileGridSpace](#)

type [EnvType](#)

includeSelf [bool](#)

respectWalls [bool](#)

Returns

[List](#) <[TileGridSpace](#)>

GetContiguousTilesOfEnvTypes(TileGridSpace, EnvType[], bool, bool)

```
public List<TileGridSpace> GetContiguousTilesOfEnvTypes(TileGridSpace context, EnvType[] types, bool includeSelf, bool respectWalls)
```

Parameters

context [TileGridSpace](#)

types [EnvType](#)[]

includeSelf [bool](#)

respectWalls [bool](#)

Returns

[List](#) <[TileGridSpace](#)>

GetContiguousUncollapsedTiles(TileGridSpace, int, bool)

```
public List<TileGridSpace> GetContiguousUncollapsedTiles(TileGridSpace startTile, int maxTiles, bool includeSelf)
```

Parameters

`startTile` [TileGridSpace](#)

`maxTiles` [int](#)

`includeSelf` [bool](#)

Returns

[List](#) <[TileGridSpace](#)>

GetCorridor(int)

`public Area GetCorridor(int id)`

Parameters

`id` [int](#)

Returns

[Area](#)

GetIsConnected(TileGridSpace, TileGridSpace, Edge)

`public bool GetIsConnected(TileGridSpace context, TileGridSpace neighbor, Edge direction)`

Parameters

`context` [TileGridSpace](#)

`neighbor` [TileGridSpace](#)

`direction` [Edge](#)

Returns

[bool](#)

GetIsConnected(TileGridSpace, TileGridSpace, Edge, EnvType)

```
public bool GetIsConnected(TileGridSpace context, TileGridSpace neighbor, Edge direction,  
EnvType desiredNeighborInteriorType)
```

Parameters

context [TileGridSpace](#)

neighbor [TileGridSpace](#)

direction [Edge](#)

desiredNeighborInteriorType [EnvType](#)

Returns

[bool](#) ↗

GetIsConnected(TileGridSpace, TileGridSpace, Edge, EnvType[])

```
public bool GetIsConnected(TileGridSpace context, TileGridSpace neighbor, Edge direction,  
EnvType[] validTypes)
```

Parameters

context [TileGridSpace](#)

neighbor [TileGridSpace](#)

direction [Edge](#)

validTypes [EnvType](#)[]

Returns

[bool](#) ↗

GetMostEntropic()

```
public TileGridSpace GetMostEntropic()
```

Returns

[TileGridSpace](#)

GetNeighbor(TileGridSpace, Corner)

```
public TileGridSpace GetNeighbor(TileGridSpace context, Corner neighbor)
```

Parameters

context [TileGridSpace](#)

neighbor [Corner](#)

Returns

[TileGridSpace](#)

GetNeighbor(TileGridSpace, Edge)

```
public TileGridSpace GetNeighbor(TileGridSpace context, Edge neighbor)
```

Parameters

context [TileGridSpace](#)

neighbor [Edge](#)

Returns

[TileGridSpace](#)

GetNeighbors(TileGridSpace, bool)

```
public List<TileGridSpace> GetNeighbors(TileGridSpace context, bool includeDiagonals)
```

Parameters

context [TileGridSpace](#)

includeDiagonals [bool](#)

Returns

[List](#)<[TileGridSpace](#)>

GetRoom(int)

```
public Area GetRoom(int id)
```

Parameters

id [int](#)

Returns

[Area](#)

GetRoomByDefinition(AreaDefinitionSO)

```
public Area GetRoomByDefinition(AreaDefinitionSO roomDefinition)
```

Parameters

roomDefinition [AreaDefinitionSO](#)

Returns

[Area](#)

GetRoomsByDefinition(AreaDefinitionSO)

```
public Area[] GetRoomsByDefinition(AreaDefinitionSO roomDefinition)
```

Parameters

roomDefinition [AreaDefinitionSO](#)

Returns

[Area\[\]](#)

GetSelectionFromContiguousTiles(List<TileGridSpace>, int)

```
public List<TileGridSpace> GetSelectionFromContiguousTiles(List<TileGridSpace>  
contiguousTiles, int count)
```

Parameters

contiguousTiles [List<TileGridSpace>](#)

count [int](#)

Returns

[List<TileGridSpace>](#)

GetUncollapsedNeighbors(TileGridSpace)

```
public List<TileGridSpace> GetUncollapsedNeighbors(TileGridSpace context)
```

Parameters

context [TileGridSpace](#)

Returns

[List](#) <TileGridSpace>

GetUncollapsedTilesNeighboringCollapsed()

```
public List<TileGridSpace> GetUncollapsedTilesNeighboringCollapsed()
```

Returns

[List](#) <TileGridSpace>

GetUncollapsedTilesNeighboringCollapsed(List<TileGridSpace>)

```
public List<TileGridSpace> GetUncollapsedTilesNeighboringCollapsed(List<TileGridSpace>  
include)
```

Parameters

include [List](#) <TileGridSpace>

Returns

[List](#) <TileGridSpace>

RegisterDoorObjectsToGridPosition(Vector2Int, List<GameObject>)

```
public void RegisterDoorObjectsToGridPosition(Vector2Int gridPos,  
List<GameObject> candidateDoorObjects)
```

Parameters

gridPos Vector2Int

candidateDoorObjects [List](#) <GameObject>

UpdateUncollapsedNeighbors(TileGridSpace)

```
public void UpdateUncollapsedNeighbors(TileGridSpace context)
```

Parameters

context [TileGridSpace](#)

Class GameMapManager

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class GameMapManager : MonoBehaviour
```

Inheritance

[object](#) ← GameMapManager

Constructors

GameMapManager()

```
public GameMapManager()
```

Fields

DontSpawnGameObjects

```
public bool DontSpawnGameObjects
```

Field Value

[bool](#)

GameMapSettings

```
public GameMapSettingsSO GameMapSettings
```

Field Value

[GameMapSettingsSO](#)

GenerationSucceeded

```
public bool GenerationSucceeded
```

Field Value

[bool](#)

MapObjects

```
public List<GameObject> MapObjects
```

Field Value

[List](#) <GameObject>

OnMapReady

```
public UnityEvent OnMapReady
```

Field Value

UnityEvent

ShowLogs

```
public static bool ShowLogs
```

Field Value

[bool](#)

Properties

BakedMap

```
public BakedMap BakedMap { get; }
```

Property Value

[BakedMap](#)

GameMap

```
public GameMap GameMap { get; }
```

Property Value

[GameMap](#)

GenerateOnAwake

```
public bool GenerateOnAwake { get; }
```

Property Value

[bool](#)

MapBaked

```
public bool MapBaked { get; }
```

Property Value

[bool](#)

MapGenStrategy

```
public IMapGenStrategy MapGenStrategy { get; }
```

Property Value

[IMapGenStrategy](#)

MapTileManager

```
public MapTileManager MapTileManager { get; }
```

Property Value

[MapTileManager](#)

Seed

```
public int Seed { get; }
```

Property Value

[int](#)

ShowWFCAnimated

```
public bool ShowWFCAnimated { get; }
```

Property Value

[bool](#)

SpawnSpriteTemplates

```
public bool SpawnSpriteTemplates { get; set; }
```

Property Value

[bool](#)

ValidateMapsLoop

```
public bool ValidateMapsLoop { get; }
```

Property Value

[bool](#)

Methods

BakePrefab()

```
public void BakePrefab()
```

ClearMap()

```
public bool ClearMap()
```

Returns

[bool](#)

GenerateMap()

```
public void GenerateMap()
```

GenerateMap(int)

```
public void GenerateMap(int seed)
```

Parameters

seed [int](#)

Log(string)

```
public static void Log(string message)
```

Parameters

message [string](#)

LogWarning(string)

```
public static void LogWarning(string message)
```

Parameters

message [string](#)

OnMapGenCoroutineDone()

```
public void OnMapGenCoroutineDone()
```

OnMapValidated()

```
public void OnMapValidated()
```

Class GameMapManagerEditor

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class GameMapManagerEditor : Editor
```

Inheritance

[object](#) ← GameMapManagerEditor

Constructors

GameMapManagerEditor()

```
public GameMapManagerEditor()
```

Methods

OnInspectorGUI()

```
public override void OnInspectorGUI()
```

Class GameMapSettingsEditor

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class GameMapSettingsEditor : Editor
```

Inheritance

[object](#) ← GameMapSettingsEditor

Constructors

GameMapSettingsEditor()

```
public GameMapSettingsEditor()
```

Methods

OnInspectorGUI()

```
public override void OnInspectorGUI()
```

Class GameMapSettingsSO

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class GameMapSettingsSO : ScriptableObject
```

Inheritance

[object](#) ← GameMapSettingsSO

Constructors

GameMapSettingsSO()

```
public GameMapSettingsSO()
```

Properties

AreaOptions

```
public AreaOptionsSO AreaOptions { get; }
```

Property Value

[AreaOptionsSO](#)

CFSPercent

```
public float CFSPercent { get; }
```

Property Value

[float](#)

ChunkMergesMinMax

```
public Vector2Int ChunkMergesMinMax { get; }
```

Property Value

Vector2Int

ChunkRoomConnectionMode

```
public ChunkRoomConnectionMode ChunkRoomConnectionMode { get; }
```

Property Value

[ChunkRoomConnectionMode](#)

ChunkSizeMinMax

```
public Vector2Int ChunkSizeMinMax { get; }
```

Property Value

Vector2Int

ClampArea

```
public bool ClampArea { get; }
```

Property Value

[bool](#) ↗

ConsolidatePaths

```
public bool ConsolidatePaths { get; }
```

Property Value

[bool](#) ↗

CorridorsFillSpace

```
public bool CorridorsFillSpace { get; }
```

Property Value

[bool](#) ↗

DeadEndsReductionFactor

```
public float DeadEndsReductionFactor { get; }
```

Property Value

[float](#) ↗

DirectRoutes

```
public bool DirectRoutes { get; }
```

Property Value

[bool](#) ↗

DoorPlacementMode

```
public DoorPlacementMode DoorPlacementMode { get; }
```

Property Value

[DoorPlacementMode](#)

ExactMapSize

```
public Vector2Int ExactMapSize { get; }
```

Property Value

Vector2Int

HybridChunksFactor

```
public float HybridChunksFactor { get; }
```

Property Value

[float](#)

LargeRoomSize

```
public int LargeRoomSize { get; }
```

Property Value

[int](#)

MapPlayAreaMinMax

```
public Vector2Int MapPlayAreaMinMax { get; }
```

Property Value

Vector2Int

MinCorridors

```
public int MinCorridors { get; }
```

Property Value

[int ↗](#)

MinRooms

```
public int MinRooms { get; }
```

Property Value

[int ↗](#)

NearestRoomTieThreshold

```
public int NearestRoomTieThreshold { get; }
```

Property Value

[int ↗](#)

RandomMapDimensionsMinMax

```
public Vector2Int RandomMapDimensionsMinMax { get; }
```

Property Value

Vector2Int

RandomSeed

```
public int RandomSeed { get; set; }
```

Property Value

[int](#)

RandomizeSeed

```
public bool RandomizeSeed { get; set; }
```

Property Value

[bool](#)

RemoveSmallRooms

```
public bool RemoveSmallRooms { get; }
```

Property Value

[bool](#)

RoomCreationMode

```
public RoomCreationMode RoomCreationMode { get; }
```

Property Value

[RoomCreationMode](#)

SaveAdjustedSizes

```
public bool SaveAdjustedSizes { get; }
```

Property Value

[bool](#)

SmallRoomSize

```
public int SmallRoomSize { get; }
```

Property Value

[int](#)

SplitLargeRooms

```
public bool SplitLargeRooms { get; }
```

Property Value

[bool](#)

UseExteriorCeiling

```
public bool UseExteriorCeiling { get; }
```

Property Value

[bool](#) ↗

UseInteriorCeiling

```
public bool UseInteriorCeiling { get; }
```

Property Value

[bool](#) ↗

UseRandomMapSize

```
public bool UseRandomMapSize { get; }
```

Property Value

[bool](#) ↗

Methods

DebugRandomizeAll()

```
public void DebugRandomizeAll()
```

GetActualMapSize()

```
public Vector2Int GetActualMapSize()
```

Returns

Vector2Int

GetSettingsInfoString()

```
public string GetSettingsInfoString()
```

Returns

string ↗

IncreaseMapSize()

```
public void IncreaseMapSize()
```

Randomize()

```
public void Randomize()
```

ResetTempSizes()

```
public void ResetTempSizes()
```

Class Grid2D<T>

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class Grid2D<T> : IEnumerable
```

Type Parameters

T

Inheritance

[object](#) ← Grid2D<T>

Implements

[IEnumerable](#)

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

Grid2D(Vector2Int)

```
public Grid2D(Vector2Int dimensions)
```

Parameters

dimensions Vector2Int

Properties

AsArray

```
public T[] AsArray { get; }
```

Property Value

T[]

Dimensions

```
public Vector2Int Dimensions { get; }
```

Property Value

Vector2Int

this[int, int]

```
public T this[int x, int y] { get; set; }
```

Parameters

x [int ↗](#)

y [int ↗](#)

Property Value

T

this[Vector2Int]

```
public T this[Vector2Int pos] { get; set; }
```

Parameters

pos Vector2Int

Property Value

Methods

GetEnumerator()

```
public IEnumrator GetEnumerator()
```

Returns

[IEnumrator](#)

GetIndex(Vector2Int)

```
public int GetIndex(Vector2Int pos)
```

Parameters

pos Vector2Int

Returns

[int](#)

InBounds(Vector2Int)

```
public bool InBounds(Vector2Int pos)
```

Parameters

pos Vector2Int

Returns

[bool](#)

Class HybridStrategy

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class HybridStrategy : IMapGenStrategy
```

Inheritance

[object](#) ← HybridStrategy

Implements

[IMapGenStrategy](#)

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

HybridStrategy()

```
public HybridStrategy()
```

Properties

NumChunks

```
public int NumChunks { get; }
```

Property Value

[int](#)

Methods

GetMapGenCoroutine(GameMapManager)

```
public IEnumerator GetMapGenCoroutine(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Returns

[IEnumerator](#)

Interface IMapGenPostProcess

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public interface IMapGenPostProcess
```

Methods

OnMapBaked(BakedMap)

```
void OnMapBaked(BakedMap bakedMap)
```

Parameters

bakedMap [BakedMap](#)

Reset()

```
void Reset()
```

Interface IMapGenStrategy

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public interface IMapGenStrategy
```

Methods

GetMapGenCoroutine(GameMapManager)

```
IEnumerator GetMapGenCoroutine(GameMapManager context)
```

Parameters

`context` [GameMapManager](#)

Returns

[IEnumerator](#)

Class MapGenConstants

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class MapGenConstants
```

Inheritance

[object](#) ← MapGenConstants

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Fields

DirectionOffsets

```
public static readonly Dictionary<Edge, Vector2Int> DirectionOffsets
```

Field Value

[Dictionary](#)<[Edge](#), [Vector2Int](#)>

Class MapGenInspectorLibrary

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class MapGenInspectorLibrary
```

Inheritance

[object](#) ← MapGenInspectorLibrary

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

CreateAssetField<T>(SerializedProperty)

```
public static void CreateAssetField<T>(SerializedProperty assetProperty)
```

Parameters

assetProperty SerializedProperty

Type Parameters

T

DrawHorizontalLine(Color, GUIStyle)

```
public static void DrawHorizontalLine(Color color, GUIStyle guiStyle)
```

Parameters

color Color

`guiStyle` GUIStyle

DrawMultiAxisDistributionGraphs(Rect, IReadOnlyList<GraphAxisData>, float, float)

```
public static void DrawMultiAxisDistributionGraphs(Rect totalRect,  
IReadOnlyList<MapGenInspectorLibrary.GraphAxisData> axes, float graphHeight = 40, float  
padding = 2)
```

Parameters

`totalRect` Rect

`axes` [IReadOnlyList](#)<[MapGenInspectorLibrary.GraphAxisData](#)>

`graphHeight` [float](#)

`padding` [float](#)

DrawPropOffsetsGraphs(PropOffsets)

```
public static void DrawPropOffsetsGraphs(PropOffsets propOffsets)
```

Parameters

`propOffsets` [PropOffsets](#)

DrawSampledDistributionGraph(Rect, Vector2, Func<float>, Color, int)

```
public static void DrawSampledDistributionGraph(Rect rect, Vector2 minMax, Func<float>  
getSample, Color color, int sampleCount = 200)
```

Parameters

`rect` Rect

```
minMax Vector2  
  
getSample Func<float>  
  
color Color  
  
sampleCount int
```

GetHorizontalLineStyle(float)

```
public static GUIStyle GetHorizontalLineStyle(float thickness = 1)
```

Parameters

thickness float

Returns

GUIStyle

ShowDisabledScriptReference(MonoBehaviour)

```
public static void ShowDisabledScriptReference(MonoBehaviour targetCastToType)
```

Parameters

targetCastToType MonoBehaviour

ShowDisabledScriptReference(ScriptableObject)

```
public static void ShowDisabledScriptReference(ScriptableObject targetCastToType)
```

Parameters

targetCastToType ScriptableObject

Struct MapGenInspectorLibrary.GraphAxisData

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public struct MapGenInspectorLibrary.GraphAxisData
```

Inherited Members

[ValueType.Equals\(object\)](#) , [ValueType.GetHashCode\(\)](#) , [ValueType.ToString\(\)](#) , [object.GetType\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

GraphAxisData(Vector2, Color, Func<float>, string)

```
public GraphAxisData(Vector2 minMax, Color color, Func<float> sampler, string label = null)
```

Parameters

minMax Vector2

color Color

sampler [Func<float>](#)

label [string](#)

Fields

Color

```
public Color Color
```

Field Value

Color

Label

```
public string Label
```

Field Value

[string](#)

MinMax

```
public Vector2 MinMax
```

Field Value

Vector2

Sampler

```
public Func<float> Sampler
```

Field Value

[Func](#) <[float](#)>

Class MapGenStrategies

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class MapGenStrategies
```

Inheritance

[object](#) ← MapGenStrategies

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

Get(RoomCreationMode)

```
public static IMapGenStrategy Get(RoomCreationMode mode)
```

Parameters

mode [RoomCreationMode](#)

Returns

[IMapGenStrategy](#)

Class MapSizeUtility

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class MapSizeUtility
```

Inheritance

[object](#) ← MapSizeUtility

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Fields

MIN_DIMENSION

```
public const int MIN_DIMENSION = 4
```

Field Value

[int](#)

Methods

GetDefaultUnclampedAreaFromDimensions(Vector2Int)

```
public static Vector2Int GetDefaultUnclampedAreaFromDimensions(Vector2Int dimensions)
```

Parameters

dimensions Vector2Int

Returns

Vector2Int

GetValidMapSizes(Vector2Int, Vector2Int)

```
public static List<Vector2Int> GetValidMapSizes(Vector2Int dimMinMax, Vector2Int areaMinMax)
```

Parameters

dimMinMax Vector2Int

areaMinMax Vector2Int

Returns

[List](#) <Vector2Int>

GetValidRandomArea(Vector2Int, Vector2Int, bool, int)

```
public static Vector2Int GetValidRandomArea(Vector2Int proposedMinMaxArea, Vector2Int minMaxDimensions, bool clampArea, int safeMaxArea)
```

Parameters

proposedMinMaxArea Vector2Int

minMaxDimensions Vector2Int

clampArea [bool](#)

safeMaxArea [int](#)

Returns

Vector2Int

GetValidRandomDimensions(Vector2Int, int)

```
public static Vector2Int GetValidRandomDimensions(Vector2Int proposedMinMaxDimensions,  
int safeMaxArea)
```

Parameters

proposedMinMaxDimensions Vector2Int

safeMaxArea [int](#)

Returns

Vector2Int

PickRandomMapSize(List<Vector2Int>, IRandomProvider, bool)

```
public static Vector2Int PickRandomMapSize(List<Vector2Int> validSizes, IRandomProvider rng,  
bool allowSwap = true)
```

Parameters

validSizes [List](#)<Vector2Int>

rng [IRandomProvider](#)

allowSwap [bool](#)

Returns

Vector2Int

Class MapTileBlueprint

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class MapTileBlueprint
```

Inheritance

[object](#) ← MapTileBlueprint

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

MapTileBlueprint(Sprite)

```
public MapTileBlueprint(Sprite template)
```

Parameters

template Sprite

Fields

AccumulatedWeight

```
public float AccumulatedWeight
```

Field Value

[float](#)

OddsWeight

```
public float OddsWeight
```

Field Value

[float](#)

PixelEnvTypes

```
public EnvType[] PixelEnvTypes
```

Field Value

[EnvType](#)[]

SpriteTemplate

```
public Sprite SpriteTemplate
```

Field Value

Sprite

Properties

InteriorType

```
public EnvType InteriorType { get; }
```

Property Value

[EnvType](#)

name

```
public string name { get; }
```

Property Value

[string](#)

Methods

AddDoorSpot(Edge, Vector4)

```
public void AddDoorSpot(Edge edge, Vector4 posAndRot)
```

Parameters

edge [Edge](#)

posAndRot Vector4

GetDoorSpots()

```
public List<Tuple<Edge, Vector4>> GetDoorSpots()
```

Returns

[List](#)<[Tuple](#)<Edge, Vector4>>

GetEdgeType(Edge)

```
public EnvType GetEdgeType(Edge edge)
```

Parameters

edge [Edge](#)

Returns

[EnvType](#)

GetEdgeTypes(Edge)

public EnvType[] GetEdgeTypes(Edge edge)

Parameters

edge [Edge](#)

Returns

[EnvType\[\]](#)

GetEdgesByCenterType(EnvType)

public Edge[] GetEdgesByCenterType(EnvType desiredType)

Parameters

desiredType [EnvType](#)

Returns

[Edge\[\]](#)

GetModules()

public Module[] GetModules()

Returns

[Module\[\]](#)

SetModules(List<Module>)

```
public void SetModules(List<Module> modules)
```

Parameters

modules [List](#)<[Module](#)>

SetPixelEnvTypes(EnvType[])

```
public void SetPixelEnvTypes(EnvType[] types)
```

Parameters

types [EnvType](#)[]

Class MapTileFactory

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class MapTileFactory
```

Inheritance

[object](#) ← MapTileFactory

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

BuildNewMapTile(MapTileBlueprint, Vector2Int, Vector2, bool)

```
public static GameObject BuildNewMapTile(MapTileBlueprint blueprint, Vector2Int gridPos,  
Vector2 tileSize, bool spawnSpriteTemplate)
```

Parameters

blueprint [MapTileBlueprint](#)

gridPos Vector2Int

tileSize Vector2

spawnSpriteTemplate [bool](#)

Returns

GameObject

CompleteWorldTile(TileGridSpace, AreaDefinitionSO)

```
public static void CompleteWorldTile(TileGridSpace tile, AreaDefinitionSO areaDefinition)
```

Parameters

tile [TileGridSpace](#)

areaDefinition [AreaDefinitionSO](#)

Class MapTileManager

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class MapTileManager
```

Inheritance

[object](#) ← MapTileManager

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

MapTileManager()

```
public MapTileManager()
```

Fields

BackupBlueprints

```
public static List<MapTileBlueprint> BackupBlueprints
```

Field Value

[List](#)<[MapTileBlueprint](#)>

Blueprints

```
public static List<MapTileBlueprint> Blueprints
```

Field Value

[List ↗](#) <MapTileBlueprint>

BottomCenterPixelIndex

```
public const int BottomCenterPixelIndex = 1
```

Field Value

[int ↗](#)

BottomLeftPixelIndex

```
public static readonly int BottomLeftPixelIndex
```

Field Value

[int ↗](#)

BottomPixelIndexes

```
public static readonly int[] BottomPixelIndexes
```

Field Value

[int ↗ \[\]](#)

BottomRightPixelIndex

```
public static readonly int BottomRightPixelIndex
```

Field Value

[int](#)

ColorConstantsDict

```
public static Dictionary<Color, EnvType> ColorConstantsDict
```

Field Value

[Dictionary](#)<Color, [EnvType](#)>

Corners

```
public static Corner[] Corners
```

Field Value

[Corner](#)[]

EdgeBlueprint

```
public static MapTileBlueprint EdgeBlueprint
```

Field Value

[MapTileBlueprint](#)

EdgeDoorPlacements

```
public static Dictionary<Edge, Vector4> EdgeDoorPlacements
```

Field Value

[Dictionary](#)<[Edge](#), [Vector4](#)>

Edges

```
public static Edge[] Edges
```

Field Value

[Edge\[\]](#)

LeftCenterPixelIndex

```
public const int LeftCenterPixelIndex = 3
```

Field Value

[int↗](#)

LeftPixelIndexes

```
public static readonly int[] LeftPixelIndexes
```

Field Value

[int↗\[\]](#)

RightCenterPixelIndex

```
public const int RightCenterPixelIndex = 5
```

Field Value

[int↗](#)

RightPixelIndexes

```
public static readonly int[] RightPixelIndexes
```

Field Value

[int\[\]](#)

SubTilePositions

```
public static Vector2Int[] SubTilePositions
```

Field Value

[Vector2Int\[\]](#)

TileModuleEdgeSize

```
public int TileModuleEdgeSize
```

Field Value

[int](#)

TileSize

```
public Vector2 TileSize
```

Field Value

[Vector2](#)

TopCenterPixelIndex

```
public const int TopCenterPixelIndex = 7
```

Field Value

[int ↗](#)

TopLeftPixelIndex

```
public static readonly int TopLeftPixelIndex
```

Field Value

[int ↗](#)

TopPixelIndexes

```
public static readonly int[] TopPixelIndexes
```

Field Value

[int\[\] ↗](#)

TopRightPixelIndex

```
public static readonly int TopRightPixelIndex
```

Field Value

[int ↗](#)

Methods

GenerateBackupBlueprint(EnvType[])

```
public MapTileBlueprint GenerateBackupBlueprint(EnvType[] pixelEnvTypes)
```

Parameters

pixelEnvTypes [EnvType\[\]](#)

Returns

[MapTileBlueprint](#)

GetBackupBlueprint(EnvType[])

```
public static MapTileBlueprint GetBackupBlueprint(EnvType[] pixelEnvTypes)
```

Parameters

pixelEnvTypes [EnvType\[\]](#)

Returns

[MapTileBlueprint](#)

GetBlueprint(EnvType[])

```
public static MapTileBlueprint GetBlueprint(EnvType[] pixelEnvTypes)
```

Parameters

pixelEnvTypes [EnvType\[\]](#)

Returns

[MapTileBlueprint](#)

GetBlueprint(Sprite)

```
public static MapTileBlueprint GetBlueprint(Sprite template)
```

Parameters

template Sprite

Returns

[MapTileBlueprint](#)

GetBlueprints(EnvType)

```
public static List<MapTileBlueprint> GetBlueprints(EnvType envType)
```

Parameters

envType [EnvType](#)

Returns

[List](#) <[MapTileBlueprint](#)>

GetBlueprints(EnvType[])

```
public static List<MapTileBlueprint> GetBlueprints(EnvType[] envTypes)
```

Parameters

envTypes [EnvType](#)[]

Returns

[List](#) <[MapTileBlueprint](#)>

GetEdgeByDirectionOffset(Vector2Int)

```
public static Edge GetEdgeByDirectionOffset(Vector2Int direcitonOffset)
```

Parameters

direcitonOffset Vector2Int

Returns

[Edge](#)

GetEnvTypes(Sprite)

```
public static EnvType[] GetEnvTypes(Sprite sprite)
```

Parameters

sprite Sprite

Returns

[EnvType\[\]](#)

GetEnvTypes(Sprite, out Color[])

```
public static EnvType[] GetEnvTypes(Sprite sprite, out Color[] pixels)
```

Parameters

sprite Sprite

pixels Color[]

Returns

[EnvType\[\]](#)

GetFacingDirectionAsQuaternion(Edge)

```
public static Quaternion GetFacingDirectionAsQuaternion(Edge facing)
```

Parameters

facing [Edge](#)

Returns

Quaternion

GetFacingDirectionAsVector3(Edge)

```
public static Vector3 GetFacingDirectionAsVector3(Edge facing)
```

Parameters

facing [Edge](#)

Returns

Vector3

GetFullPixelArrayFromCorner(Corner, EnvType)

```
public static EnvType[] GetFullPixelArrayFromCorner(Corner corner, EnvType toType)
```

Parameters

corner [Corner](#)

toType [EnvType](#)

Returns

[EnvType](#)[]

GetFullPixelArrayFromEdge(Edge, EnvType)

```
public static EnvType[] GetFullPixelArrayFromEdge(Edge edge, EnvType type)
```

Parameters

edge [Edge](#)

type [EnvType](#)

Returns

[EnvType](#)[]

GetFullPixelArrayOfType(EnvType)

```
public static EnvType[] GetFullPixelArrayOfType(EnvType type)
```

Parameters

type [EnvType](#)

Returns

[EnvType](#)[]

GetOppositeCorner(Corner)

```
public static Corner GetOppositeCorner(Corner corner)
```

Parameters

corner [Corner](#)

Returns

[Corner](#)

GetOppositeEdge(Edge)

```
public static Edge GetOppositeEdge(Edge edge)
```

Parameters

[edge](#) [Edge](#)

Returns

[Edge](#)

GetPixelEnvTypesAsString(EnvType[])

```
public static string GetPixelEnvTypesAsString(EnvType[] pixelEnvTypes)
```

Parameters

[pixelEnvTypes](#) [EnvType](#)[]

Returns

[string](#) ↗

GetPixelIndexByCorner(Corner)

```
public static int GetPixelIndexByCorner(Corner corner)
```

Parameters

[corner](#) [Corner](#)

Returns

[int](#)

GetPixelIndexesByEdge(Edge)

```
public static int[] GetPixelIndexesByEdge(Edge edge)
```

Parameters

[edge](#) [Edge](#)

Returns

[int](#)[]

GetRotatedBlueprint(MapTileBlueprint, BlueprintRotation)

```
public static MapTileBlueprint GetRotatedBlueprint(MapTileBlueprint blueprint,  
BlueprintRotation rotation)
```

Parameters

[blueprint](#) [MapTileBlueprint](#)

[rotation](#) [BlueprintRotation](#)

Returns

[MapTileBlueprint](#)

GetRotatedBlueprint(MapTileBlueprint, int)

```
public static MapTileBlueprint GetRotatedBlueprint(MapTileBlueprint blueprint,  
int numRotations)
```

Parameters

`blueprint` [MapTileBlueprint](#)

`numRotations` [int↗](#)

Returns

[MapTileBlueprint](#)

Initialize(GameMapManager, int, AdvancedSettings)

```
public void Initialize(GameMapManager worldMapManager, int tileModuleEdgeSize,  
AdvancedSettings advancedSettings)
```

Parameters

`worldMapManager` [GameMapManager](#)

`tileModuleEdgeSize` [int↗](#)

`advancedSettings` [AdvancedSettings](#)

Class Module

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class Module
```

Inheritance

[object](#) ← Module

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

Module(ModuleType, Vector2Int, EnvType, int, float, bool)

```
public Module(ModuleType moduleType, Vector2Int subGridPos, EnvType envType, int arrayIndex,  
float rotation, bool isFloor)
```

Parameters

moduleType [ModuleType](#)

subGridPos Vector2Int

envType [EnvType](#)

arrayIndex [int](#)

rotation [float](#)

isFloor [bool](#)

Fields

arrayIndex

```
public int arrayIndex
```

Field Value

[int ↗](#)

envType

```
public EnvType envType
```

Field Value

[EnvType](#)

isFloor

```
public bool isFloor
```

Field Value

[bool ↗](#)

moduleType

```
public ModuleType moduleType
```

Field Value

[ModuleType](#)

rotation

```
public float rotation
```

Field Value

[float ↗](#)

subGridPos

```
public Vector2Int subGridPos
```

Field Value

Vector2Int

Properties

Position

```
public Vector3 Position { get; }
```

Property Value

Vector3

Enum ModuleType

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public enum ModuleType
```

Fields

ExteriorCeiling = 5

FloorA = 0

FloorB = 1

InteriorCeiling = 6

WallCornerInside = 3

WallCornerOutside = 4

WallStraight = 2

Struct PotentialConnection

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public struct PotentialConnection
```

Inherited Members

[ValueType.Equals\(object\)](#) , [ValueType.GetHashCode\(\)](#) , [ValueType.ToString\(\)](#) , [object.GetType\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

PotentialConnection(int, int, Vector2Int)

```
public PotentialConnection(int a, int b, Vector2Int offset)
```

Parameters

a [int](#)

b [int](#)

offset Vector2Int

Fields

a

```
public int a
```

Field Value

[int](#)

b

```
public int b
```

Field Value

[int ↗](#)

offset

```
public Vector2Int offset
```

Field Value

Vector2Int

Class PrefabSelectionBase

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class PrefabSelectionBase : MonoBehaviour
```

Inheritance

[object](#) ← PrefabSelectionBase

Constructors

PrefabSelectionBase()

```
public PrefabSelectionBase()
```

Class PropCategoriesEditorUI

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class PropCategoriesEditorUI
```

Inheritance

[object](#) ← PropCategoriesEditorUI

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

PropCategoriesEditorUI(SerializedProperty)

```
public PropCategoriesEditorUI(SerializedProperty propCategories)
```

Parameters

propCategories SerializedProperty

Methods

Draw()

```
public void Draw()
```

Class RandomStrategy

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class RandomStrategy : IMapGenStrategy
```

Inheritance

[object](#) ← RandomStrategy

Implements

[IMapGenStrategy](#)

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

RandomStrategy()

```
public RandomStrategy()
```

Methods

GetMapGenCoroutine(GameMapManager)

```
public IEnumerator GetMapGenCoroutine(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Returns

[IEnumerator](#)

RemoveSmallRooms(GameMapManager, List<TileGridSpace>)

```
public static IEnumerator RemoveSmallRooms(GameMapManager context, List<TileGridSpace>  
tilesToOmit = null)
```

Parameters

`context` [GameMapManager](#)
`tilesToOmit` [List](#)<[TileGridSpace](#)>

Returns

[IEnumerator](#)

SplitLargeRooms(GameMapManager, List<TileGridSpace>)

```
public static IEnumerator SplitLargeRooms(GameMapManager context, List<TileGridSpace>  
tilesToOmit = null)
```

Parameters

`context` [GameMapManager](#)
`tilesToOmit` [List](#)<[TileGridSpace](#)>

Returns

[IEnumerator](#)

Class ReflectionProbePlacement

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class ReflectionProbePlacement : MonoBehaviour, IMapGenPostProcess
```

Inheritance

[object](#) ← ReflectionProbePlacement

Implements

[IMapGenPostProcess](#)

Constructors

ReflectionProbePlacement()

```
public ReflectionProbePlacement()
```

Methods

OnMapBaked(BakedMap)

```
public void OnMapBaked(BakedMap bakedMap)
```

Parameters

bakedMap [BakedMap](#)

Reset()

```
public void Reset()
```

Enum RoomCreationMode

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public enum RoomCreationMode
```

Fields

Chunks = 1

Hybrid = 2

Random = 0

Class RoomShapeManager

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class RoomShapeManager
```

Inheritance

[object](#) ← RoomShapeManager

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

RoomShapeManager()

```
public RoomShapeManager()
```

Fields

AllRoomShapes

```
public static List<RoomShapeSO> AllRoomShapes
```

Field Value

[List](#)<[RoomShapeSO](#)>

Methods

GetRandomRoomShape(Vector2Int)

```
public static RoomShapeSO GetRandomRoomShape(Vector2Int maxDimensions)
```

Parameters

maxDimensions Vector2Int

Returns

[RoomShapeSO](#)

GetRandomRoomShape(Vector2Int, Vector2Int)

```
public static RoomShapeSO GetRandomRoomShape(Vector2Int minDimensions,  
Vector2Int maxDimensions)
```

Parameters

minDimensions Vector2Int

maxDimensions Vector2Int

Returns

[RoomShapeSO](#)

PrepareAllRoomShapes()

```
public void PrepareAllRoomShapes()
```

Class RoomShapeSO

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class RoomShapeSO : ScriptableObject
```

Inheritance

[Object](#) ← RoomShapeSO

Constructors

RoomShapeSO()

```
public RoomShapeSO()
```

Properties

Dimensions

```
public Vector2Int Dimensions { get; }
```

Property Value

Vector2Int

Sprites

```
public Sprite[] Sprites { get; }
```

Property Value

Sprite[]

Methods

GetSprite(Vector2Int)

```
public Sprite GetSprite(Vector2Int coordinates)
```

Parameters

coordinates Vector2Int

Returns

Sprite

Init(Vector2Int, Sprite[])

```
public void Init(Vector2Int dimensions, Sprite[] sprites)
```

Parameters

dimensions Vector2Int

sprites Sprite[]

Class RoomShapeSOEditor

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class RoomShapeSOEditor : Editor
```

Inheritance

[object](#) ← RoomShapeSOEditor

Constructors

RoomShapeSOEditor()

```
public RoomShapeSOEditor()
```

Methods

OnInspectorGUI()

```
public override void OnInspectorGUI()
```

Class SearchTile

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class SearchTile
```

Inheritance

[object](#) ← SearchTile

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Extension Methods

[SearchTileExtensions.GetPathPositions\(SearchTile\)](#) ,
[SearchTileExtensions.GetPathPositionsAsArray\(SearchTile\)](#) ,
[SearchTileExtensions.GetPathPositionsAsList\(SearchTile\)](#) ,
[SearchTileExtensions.GetPathPositionsFromEnd\(SearchTile\)](#) ,
[SearchTileExtensions.GetPathStepPairs\(SearchTile\)](#)

Constructors

SearchTile(Vector2Int)

```
public SearchTile(Vector2Int position)
```

Parameters

position Vector2Int

SearchTile(Vector2Int, SearchTile)

```
public SearchTile(Vector2Int position, SearchTile parent)
```

Parameters

position Vector2Int

parent [SearchTile](#)

Fields

g

`public float g`

Field Value

[float](#) ↗

h

`public float h`

Field Value

[float](#) ↗

parent

`public SearchTile parent`

Field Value

[SearchTile](#)

postion

```
public Vector2Int position
```

Field Value

Vector2Int

Properties

f

```
public float f { get; }
```

Property Value

[float](#)

Class SearchTileExtensions

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class SearchTileExtensions
```

Inheritance

[object](#) ← SearchTileExtensions

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

GetPathPositions(SearchTile)

```
public static IEnumerable<Vector2Int> GetPathPositions(this SearchTile endTile)
```

Parameters

endTile [SearchTile](#)

Returns

[IEnumerable](#)<Vector2Int>

GetPathPositionsAsArray(SearchTile)

```
public static Vector2Int[] GetPathPositionsAsArray(this SearchTile endTile)
```

Parameters

endTile [SearchTile](#)

Returns

Vector2Int[]

GetPathPositionsAsList(SearchTile)

```
public static List<Vector2Int> GetPathPositionsAsList(this SearchTile endTile)
```

Parameters

endTile [SearchTile](#)

Returns

[List](#)<Vector2Int>

GetPathPositionsFromEnd(SearchTile)

```
public static IEnumerable<Vector2Int> GetPathPositionsFromEnd(this SearchTile endTile)
```

Parameters

endTile [SearchTile](#)

Returns

[IEnumerable](#)<Vector2Int>

GetPathStepPairs(SearchTile)

```
public static IEnumerable<(Vector2Int from, Vector2Int to)> GetPathStepPairs(this SearchTile endTile)
```

Parameters

endTile [SearchTile](#)

Returns

[IEnumerable](#)<(Vector2Int [from](#), Vector2Int [to](#))>

Class SerializableDictionaryEntryDrawer

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class SerializableDictionaryEntryDrawer : PropertyDrawer
```

Inheritance

[object](#) ← SerializableDictionaryEntryDrawer

Constructors

SerializableDictionaryEntryDrawer()

```
public SerializableDictionaryEntryDrawer()
```

Methods

GetPropertyHeight(SerializedProperty, GUIContent)

```
public override float GetPropertyHeight(SerializedProperty property, GUIContent label)
```

Parameters

property SerializedProperty

label GUIContent

Returns

[float](#)

OnGUI(Rect, SerializedProperty, GUIContent)

```
public override void OnGUI(Rect position, SerializedProperty property, GUIContent label)
```

Parameters

position Rect

property SerializedProperty

label GUIContent

Class SerializableDictionaryEntry<TKey, TValue>

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class SerializableDictionaryEntry<TKey, TValue>
```

Type Parameters

TKey

TValue

Inheritance

[object](#) ← SerializableDictionaryEntry<TKey, TValue>

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

SerializableDictionaryEntry()

```
public SerializableDictionaryEntry()
```

SerializableDictionaryEntry(TKey, TValue)

```
public SerializableDictionaryEntry(TKey key, TValue value)
```

Parameters

key TKey

value TValue

Fields

key

```
public TKey key
```

Field Value

TKey

value

```
public TValue value
```

Field Value

TValue

Class SerializableDictionary<TKey, TValue>

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class SerializableDictionary<TKey, TValue> : ISerializationCallbackReceiver,  
IEnumerable
```

Type Parameters

TKey

TValue

Inheritance

[object](#) ← SerializableDictionary<TKey, TValue>

Implements

ISerializationCallbackReceiver, [IEnumerable](#)

Inherited Members

[object.GetType\(\)](#), [object.MemberwiseClone\(\)](#), [object.ToString\(\)](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.ReferenceEquals\(object, object\)](#), [object.GetHashCode\(\)](#)

Constructors

SerializableDictionary()

```
public SerializableDictionary()
```

Properties

Count

```
public int Count { get; }
```

Property Value

[int](#)

this[TKey]

```
public TValue this[TKey key] { get; set; }
```

Parameters

key TKey

Property Value

TValue

Keys

```
public Dictionary<TKey, TValue>.KeyCollection Keys { get; }
```

Property Value

[Dictionary](#)<TKey, TValue>.KeyCollection

Values

```
public Dictionary<TKey, TValue>.ValueCollection Values { get; }
```

Property Value

[Dictionary](#)<TKey, TValue>.ValueCollection

Methods

Add(TKey, TValue)

```
public void Add(TKey key, TValue value)
```

Parameters

key TKey

value TValue

Clear()

```
public void Clear()
```

ContainsKey(TKey)

```
public bool ContainsKey(TKey key)
```

Parameters

key TKey

Returns

bool ↗

OnAfterDeserialize()

```
public void OnAfterDeserialize()
```

OnBeforeSerialize()

```
public void OnBeforeSerialize()
```

Remove(TKey)

```
public bool Remove(TKey key)
```

Parameters

key TKey

Returns

bool ↗

TryGetValue(TKey, out TValue)

```
public bool TryGetValue(TKey key, out TValue value)
```

Parameters

key TKey

value TValue

Returns

bool ↗

Class SmartPrefabUtility

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public static class SmartPrefabUtility
```

Inheritance

[object](#) ← SmartPrefabUtility

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

InstantiatePrefab(Object)

```
public static Object InstantiatePrefab(Object assetComponentOrGameObject)
```

Parameters

assetComponentOrGameObject Object

Returns

Object

InstantiatePrefab(Object, Transform)

```
public static Object InstantiatePrefab(Object assetComponentOrGameObject, Transform parent)
```

Parameters

assetComponentOrGameObject Object

parent Transform

Returns

Object

InstantiatePrefab(Object, Vector3, Quaternion, Transform)

```
public static Object InstantiatePrefab(Object assetComponentOrGameObject, Vector3  
worldPosition, Quaternion rotation, Transform parent)
```

Parameters

assetComponentOrGameObject Object

worldPosition Vector3

rotation Quaternion

parent Transform

Returns

Object

Class TestOverlap

Namespace: [MoraGames.MapGen](#)

Assembly: Assembly-CSharp.dll

```
public class TestOverlap : MonoBehaviour
```

Inheritance

[object](#) ← TestOverlap

Constructors

TestOverlap()

```
public TestOverlap()
```

Methods

Test()

```
public void Test()
```

Struct TileEdgePair

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public struct TileEdgePair
```

Inherited Members

[ValueType.Equals\(object\)](#) , [ValueType.GetHashCode\(\)](#) , [ValueType.ToString\(\)](#) , [object.GetType\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

TileEdgePair(TileGridSpace, Edge)

```
public TileEdgePair(TileGridSpace tile, Edge edge)
```

Parameters

tile [TileGridSpace](#)

edge [Edge](#)

Fields

edge

```
public Edge edge
```

Field Value

[Edge](#)

tile

```
public TileGridSpace tile
```

Field Value

[TileGridSpace](#)

Class TileGridSpace

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class TileGridSpace
```

Inheritance

[object](#) ← TileGridSpace

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

TileGridSpace(GameMap, Vector2Int)

```
public TileGridSpace(GameMap context, Vector2Int pos)
```

Parameters

context [GameMap](#)

pos Vector2Int

Fields

ConnectsToStart

```
public bool ConnectsToStart
```

Field Value

[bool](#)

Properties

AreaType

```
public EnvType AreaType { get; }
```

Property Value

[EnvType](#)

Blueprint

```
public MapTileBlueprint Blueprint { get; }
```

Property Value

[MapTileBlueprint](#)

GridPos

```
public Vector2Int GridPos { get; }
```

Property Value

[Vector2Int](#)

IsCollapsed

```
public bool IsCollapsed { get; }
```

Property Value

[bool](#) ↗

Map

```
public GameMap Map { get; }
```

Property Value

[GameMap](#)

PropSpots

```
public List<PropSpot> PropSpots { get; }
```

Property Value

[List](#) <[PropSpot](#)>

RoomId

```
public int RoomId { get; }
```

Property Value

[int](#)

RootObject

```
public GameObject RootObject { get; }
```

Property Value

GameObject

ValidBlueprints

```
public List<MapTileBlueprint> ValidBlueprints { get; }
```

Property Value

[List](#) <[MapTileBlueprint](#)>

Methods

AddPropSpot(PropSpot)

```
public void AddPropSpot(PropSpot propSpot)
```

Parameters

propSpot [PropSpot](#)

Collapse(GameMapManager)

```
public void Collapse(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Collapse(GameMapManager, MapTileBlueprint)

```
public void Collapse(GameMapManager context, MapTileBlueprint blueprint)
```

Parameters

context [GameMapManager](#)

blueprint [MapTileBlueprint](#)

Collapse(GameMapManager, Sprite)

```
public void Collapse(GameMapManager context, Sprite template)
```

Parameters

context [GameMapManager](#)

template [Sprite](#)

GetWorldPos()

```
public Vector3 GetWorldPos()
```

Returns

[Vector3](#)

NarrowDown(EnvType[])

```
public void NarrowDown(EnvType[] requiredEnvTypes)
```

Parameters

requiredEnvTypes [EnvType](#)[]

NarrowDown(EnvType[], Edge)

```
public void NarrowDown(EnvType[] theirEdge, Edge edge)
```

Parameters

theirEdge [EnvType](#)[]

edge [Edge](#)

NarrowDown(GameMapManager)

```
public void NarrowDown(GameMapManager context)
```

Parameters

context [GameMapManager](#)

Recollapse(MapTileBlueprint)

```
public void Recollapse(MapTileBlueprint newBlueprint)
```

Parameters

newBlueprint [MapTileBlueprint](#)

ResetNarrowing()

```
public void ResetNarrowing()
```

ResetNarrowing(EnvType)

```
public void ResetNarrowing(EnvType envType)
```

Parameters

envType [EnvType](#)

ResetNarrowing(EnvType[])

```
public void ResetNarrowing(EnvType[] envTypes)
```

Parameters

envTypes [EnvType\[\]](#)

SetRoomId(int, EnvType)

```
public void SetRoomId(int id, EnvType roomType)
```

Parameters

id [int](#)

roomType [EnvType](#)

SpawnGameObjects(AreaDefinitionSO)

```
public void SpawnGameObjects(AreaDefinitionSO definition)
```

Parameters

definition [AreaDefinitionSO](#)

Uncollapse()

```
public void Uncollapse()
```

Class TileModuleDefinitionsSO

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class TileModuleDefinitionsSO : ScriptableObject
```

Inheritance

[object](#) ← TileModuleDefinitionsSO

Constructors

TileModuleDefinitionsSO()

```
public TileModuleDefinitionsSO()
```

Properties

PropCategories

```
public PropCategories PropCategories { get; }
```

Property Value

[PropCategories](#)

Methods

GetRandByType(ModuleType)

```
public GameObject GetRandByType(ModuleType moduleType)
```

Parameters

`moduleType` [ModuleType](#)

Returns

GameObject

GetRandDoor(DoorPlacementMode)

`public GameObject GetRandDoor(DoorPlacementMode mode)`

Parameters

`mode` [DoorPlacementMode](#)

Returns

GameObject

GetRandExteriorCeiling()

`public GameObject GetRandExteriorCeiling()`

Returns

GameObject

GetRandFloorA()

`public GameObject GetRandFloorA()`

Returns

GameObject

GetRandFloorB()

```
public GameObject GetRandFloorB()
```

Returns

GameObject

GetRandHalfDoorFrame()

```
public GameObject GetRandHalfDoorFrame()
```

Returns

GameObject

GetRandHalfDoorPanel()

```
public GameObject GetRandHalfDoorPanel()
```

Returns

GameObject

GetRandInteriorCeiling()

```
public GameObject GetRandInteriorCeiling()
```

Returns

GameObject

GetRandWallCornerInside()

```
public GameObject GetRandWallCornerInside()
```

Returns

GameObject

GetRandWallCornerOutside()

```
public GameObject GetRandWallCornerOutside()
```

Returns

GameObject

GetRandWallStraight()

```
public GameObject GetRandWallStraight()
```

Returns

GameObject

GetRandWholeDoor()

```
public GameObject GetRandWholeDoor()
```

Returns

GameObject

Class TileModuleDefinitionsSOEditor

Namespace: [MoraGames.MapGen](#)

Assembly: MoraMapGen.dll

```
public class TileModuleDefinitionsSOEditor :  
BasePropCategoriesEditor<TileModuleDefinitionsSO>
```

Inheritance

[object](#) ← [BasePropCategoriesEditor<TileModuleDefinitionsSO>](#) ← [TileModuleDefinitionsSOEditor](#)

Inherited Members

[BasePropCategoriesEditor<TileModuleDefinitionsSO>..propCategoriesUI](#)

Constructors

TileModuleDefinitionsSOEditor()

```
public TileModuleDefinitionsSOEditor()
```

Methods

OnEnable()

```
public override void OnEnable()
```

OnInspectorGUI()

```
public override void OnInspectorGUI()
```

Namespace MoraGames.MapGen.Demo

Classes

[Demo3rdPersonCamRig](#)

[Demo3rdPersonPlayer](#)

[DemoDungeonChest](#)

[DemoDungeonKey](#)

[DemoDungeonLockAndSwitch](#)

[DemoDungeonPlayer](#)

[DemoDungeonPrompt](#)

[DemoFPSPlayer](#)

[DemoFindFarthestAreas](#)

[DemoFindFarthestGridSpaces](#)

[DemoGridPathfinding](#)

[DemoLocationTrackingUI](#)

[DemoNPC](#)

[DemoNPCSpawner](#)

[DemoOpenDoors](#)

Enums

[MarkingMode](#)

Class Demo3rdPersonCamRig

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class Demo3rdPersonCamRig : MonoBehaviour
```

Inheritance

[object](#) ← Demo3rdPersonCamRig

Constructors

Demo3rdPersonCamRig()

```
public Demo3rdPersonCamRig()
```

Methods

Initialize(Transform)

```
public void Initialize(Transform follow)
```

Parameters

follow Transform

Class Demo3rdPersonPlayer

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class Demo3rdPersonPlayer : MonoBehaviour
```

Inheritance

[object](#) ← Demo3rdPersonPlayer

Constructors

Demo3rdPersonPlayer()

```
public Demo3rdPersonPlayer()
```

Methods

OnMapReady()

```
public void OnMapReady()
```

OnMove(CallbackContext)

```
public void OnMove(InputAction.CallbackContext context)
```

Parameters

context InputAction.CallbackContext

UpdateCamRelativity()

```
public void UpdateCamRelativity()
```

Class DemoDungeonChest

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoDungeonChest : MonoBehaviour
```

Inheritance

[object](#) ← DemoDungeonChest

Constructors

DemoDungeonChest()

```
public DemoDungeonChest()
```

Events

OnChestCollected

```
public static event Action OnChestCollected
```

Event Type

[Action](#)

Class DemoDungeonKey

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoDungeonKey : MonoBehaviour
```

Inheritance

[object](#) ← DemoDungeonKey

Constructors

DemoDungeonKey()

```
public DemoDungeonKey()
```

Events

OnKeyCollected

```
public static event Action OnKeyCollected
```

Event Type

[Action](#)

Class DemoDungeonLockAndSwitch

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoDungeonLockAndSwitch : MonoBehaviour, IMapGenPostProcess
```

Inheritance

[object](#) ← DemoDungeonLockAndSwitch

Implements

[IMapGenPostProcess](#)

Constructors

DemoDungeonLockAndSwitch()

```
public DemoDungeonLockAndSwitch()
```

Methods

OnMapBaked(BakedMap)

```
public void OnMapBaked(BakedMap bakedMap)
```

Parameters

bakedMap [BakedMap](#)

Reset()

```
public void Reset()
```

Class DemoDungeonPlayer

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoDungeonPlayer : MonoBehaviour
```

Inheritance

[object](#) ← DemoDungeonPlayer

Constructors

DemoDungeonPlayer()

```
public DemoDungeonPlayer()
```

Methods

Initialize(BakedMap, BakedArea)

```
public void Initialize(BakedMap bakedMap, BakedArea startingArea)
```

Parameters

bakedMap [BakedMap](#)

startingArea [BakedArea](#)

OnMove(CallbackContext)

```
public void OnMove(InputAction.CallbackContext context)
```

Parameters

context InputAction.CallbackContext

Events

OnAttemptMoveIntoLockedDoor

```
public static event Action OnAttemptMoveIntoLockedDoor
```

Event Type

[Action](#)

Class DemoDungeonPrompt

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoDungeonPrompt : MonoBehaviour
```

Inheritance

[object](#) ← DemoDungeonPrompt

Constructors

DemoDungeonPrompt()

```
public DemoDungeonPrompt()
```

Class DemoFPSPlayer

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoFPSPlayer : MonoBehaviour
```

Inheritance

[object](#) ← DemoFPSPlayer

Constructors

DemoFPSPlayer()

```
public DemoFPSPlayer()
```

Methods

OnAim(CallbackContext)

```
public void OnAim(InputAction.CallbackContext context)
```

Parameters

context InputAction.CallbackContext

OnMapReady()

```
public void OnMapReady()
```

OnMove(CallbackContext)

```
public void OnMove(InputAction.CallbackContext context)
```

Parameters

context InputAction.CallbackContext

UpdateCamRelativity()

```
public void UpdateCamRelativity()
```

Class DemoFindFarthestAreas

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoFindFarthestAreas : MonoBehaviour, IMapGenPostProcess
```

Inheritance

[object](#) ← DemoFindFarthestAreas

Implements

[IMapGenPostProcess](#)

Constructors

DemoFindFarthestAreas()

```
public DemoFindFarthestAreas()
```

Methods

OnMapBaked(BakedMap)

```
public void OnMapBaked(BakedMap bakedMap)
```

Parameters

bakedMap [BakedMap](#)

Reset()

```
public void Reset()
```

Class DemoFindFarthestGridSpaces

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoFindFarthestGridSpaces : MonoBehaviour, IMapGenPostProcess
```

Inheritance

[object](#) ← DemoFindFarthestGridSpaces

Implements

[IMapGenPostProcess](#)

Constructors

DemoFindFarthestGridSpaces()

```
public DemoFindFarthestGridSpaces()
```

Methods

OnMapBaked(BakedMap)

```
public void OnMapBaked(BakedMap bakedMap)
```

Parameters

bakedMap [BakedMap](#)

Reset()

```
public void Reset()
```

Class DemoGridPathfinding

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoGridPathfinding : MonoBehaviour
```

Inheritance

[object](#) ← DemoGridPathfinding

Constructors

DemoGridPathfinding()

```
public DemoGridPathfinding()
```

Methods

GetCurrentPath()

```
public void GetCurrentPath()
```

GetRandomPath()

```
public void GetRandomPath()
```

OnMapReady()

```
public void OnMapReady()
```

Class DemoLocationTrackingUI

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoLocationTrackingUI : MonoBehaviour
```

Inheritance

[object](#) ← DemoLocationTrackingUI

Constructors

DemoLocationTrackingUI()

```
public DemoLocationTrackingUI()
```

Class DemoNPC

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoNPC : MonoBehaviour
```

Inheritance

[object](#) ← DemoNPC

Constructors

DemoNPC()

```
public DemoNPC()
```

Class DemoNPCSpawner

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoNPCSpawner : MonoBehaviour
```

Inheritance

[object](#) ← DemoNPCSpawner

Constructors

DemoNPCSpawner()

```
public DemoNPCSpawner()
```

Methods

OnMapReady()

```
public void OnMapReady()
```

Class DemoOpenDoors

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoOpenDoors : MonoBehaviour
```

Inheritance

[object](#) ← DemoOpenDoors

Constructors

DemoOpenDoors()

```
public DemoOpenDoors()
```

Enum MarkingMode

Namespace: [MoraGames.MapGen.Demo](#)

Assembly: MoraMapGen.dll

```
public enum MarkingMode
```

Fields

Both = 2

Corridors = 1

Rooms = 0

Namespace MoraGames.MapGen.Props

Classes

[ProceduralPropSpot](#)

[ProceduralPropSpotOffsetsSO](#)

[ProceduralPropSpotOffsetsSOEditor](#)

[PropAnchorOverride](#)

[PropAnchorOverrideEditor](#)

[PropCategories](#)

[PropChannel](#)

[PropDefinition](#)

[PropOptionDefinition](#)

[PropOptions](#)

[PropPathBlock](#)

[PropPlacement](#)

[PropPlacementTypeInt](#)

[PropPlacementTypeIntDrawer](#)

[PropPlacementVolume](#)

[PropPlacementVolumeEditor](#)

[PropRotationSettings](#)

[PropRotationSettingsEditor](#)

[PropRotationSettingsExtensions](#)

[PropSpot](#)

[PropSpotEditor](#)

Interfaces

[IPropSpot](#)

Enums

[PropPlacementType](#)

Interface IPropSpot

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public interface IPropSpot
```

Properties

PropChannel

PropChannel PropChannel { **get**; }

Property Value

[PropChannel](#)

PropPlacementType

PropPlacementType PropPlacementType { **get**; }

Property Value

[PropPlacementType](#)

Methods

CheckIfUsed()

bool CheckIfUsed()

Returns

[bool](#)

GetRandomOffset()

`Vector3 GetRandomOffset()`

Returns

`Vector3`

GetTransform()

`Transform GetTransform()`

Returns

`Transform`

Use(GameObject)

`void Use(GameObject prop)`

Parameters

`prop GameObject`

Class ProceduralPropSpot

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class ProceduralPropSpot : IPropSpot
```

Inheritance

[object](#) ← ProceduralPropSpot

Implements

[IPropSpot](#)

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

ProceduralPropSpot(Vector3, Quaternion, PropPlacementType, Transform)

```
public ProceduralPropSpot(Vector3 pos, Quaternion rot, PropPlacementType type,  
Transform parent)
```

Parameters

pos Vector3

rot Quaternion

type [PropPlacementType](#)

parent Transform

Properties

Position

```
public Vector3 Position { get; }
```

Property Value

Vector3

PropChannel

```
public PropChannel PropChannel { get; }
```

Property Value

[PropChannel](#)

PropPlacementType

```
public PropPlacementType PropPlacementType { get; }
```

Property Value

[PropPlacementType](#)

Rotation

```
public Quaternion Rotation { get; }
```

Property Value

Quaternion

Methods

CheckIfUsed()

```
public bool CheckIfUsed()
```

Returns

[bool](#)

GetRandomOffset()

```
public Vector3 GetRandomOffset()
```

Returns

Vector3

GetTransform()

```
public Transform GetTransform()
```

Returns

Transform

SetOffsets(PropOffsets)

```
public void SetOffsets(PropOffsets offsets)
```

Parameters

offsets [PropOffsets](#)

Use(GameObject)

```
public void Use(GameObject prop)
```

Parameters

prop GameObject

Class ProceduralPropSpotOffsetsSO

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class ProceduralPropSpotOffsetsSO : ScriptableObject
```

Inheritance

[object](#) ← ProceduralPropSpotOffsetsSO

Constructors

ProceduralPropSpotOffsetsSO()

```
public ProceduralPropSpotOffsetsSO()
```

Properties

AllProceduralPropSpots

```
public PropOffsets AllProceduralPropSpots { get; }
```

Property Value

[PropOffsets](#)

OnCorridorCorners

```
public PropOffsets OnCorridorCorners { get; }
```

Property Value

[PropOffsets](#)

OnCorridorIntersections

```
public PropOffsets OnCorridorIntersections { get; }
```

Property Value

[PropOffsets](#)

OnCorridorStraights

```
public PropOffsets OnCorridorStraights { get; }
```

Property Value

[PropOffsets](#)

OnDeadEnds

```
public PropOffsets OnDeadEnds { get; }
```

Property Value

[PropOffsets](#)

OnDoorSpaces

```
public PropOffsets OnDoorSpaces { get; }
```

Property Value

[PropOffsets](#)

OnRoomCenters

```
public PropOffsets OnRoomCenters { get; }
```

Property Value

[PropOffsets](#)

OnRoomCorners

```
public PropOffsets OnRoomCorners { get; }
```

Property Value

[PropOffsets](#)

Methods

GetOffsets(PropPlacementType)

```
public PropOffsets GetOffsets(PropPlacementType propPlacementType)
```

Parameters

propPlacementType [PropPlacementType](#)

Returns

[PropOffsets](#)

Class ProceduralPropSpotOffsetsSOEditor

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class ProceduralPropSpotOffsetsSOEditor : Editor
```

Inheritance

[object](#) ← ProceduralPropSpotOffsetsSOEditor

Constructors

ProceduralPropSpotOffsetsSOEditor()

```
public ProceduralPropSpotOffsetsSOEditor()
```

Methods

OnInspectorGUI()

```
public override void OnInspectorGUI()
```

Class PropAnchorOverride

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropAnchorOverride : MonoBehaviour
```

Inheritance

[object](#) ← PropAnchorOverride

Constructors

PropAnchorOverride()

```
public PropAnchorOverride()
```

Properties

Channel

```
public PropChannel Channel { get; }
```

Property Value

[PropChannel](#)

LocalPosition

```
public Vector3 LocalPosition { get; }
```

Property Value

Vector3

LocalRotation

```
public Quaternion LocalRotation { get; }
```

Property Value

Quaternion

PropPlacementType

```
public PropPlacementType PropPlacementType { get; }
```

Property Value

[PropPlacementType](#)

RotationPivot

```
public Transform RotationPivot { get; }
```

Property Value

Transform

Class PropAnchorOverrideEditor

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropAnchorOverrideEditor : Editor
```

Inheritance

[object](#) ← PropAnchorOverrideEditor

Constructors

PropAnchorOverrideEditor()

```
public PropAnchorOverrideEditor()
```

Methods

OnInspectorGUI()

```
public override void OnInspectorGUI()
```

Class PropCategories

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropCategories
```

Inheritance

[object](#) ← PropCategories

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

PropCategories()

```
public PropCategories()
```

Fields

AlwaysSpawnProps

```
public PropOptions[] AlwaysSpawnProps
```

Field Value

[PropOptions\[\]](#)

OptionalCommonProps

```
public PropDefinition[] OptionalCommonProps
```

Field Value

[PropDefinition\[\]](#)

OptionalUniqueProps

```
public PropDefinition[] OptionalUniqueProps
```

Field Value

[PropDefinition\[\]](#)

RequiredCommonProps

```
public PropDefinition[] RequiredCommonProps
```

Field Value

[PropDefinition\[\]](#)

RequiredUniqueProps

```
public PropDefinition[] RequiredUniqueProps
```

Field Value

[PropDefinition\[\]](#)

Class PropChannel

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropChannel : ScriptableObject
```

Inheritance

[object](#) ← PropChannel

Constructors

PropChannel()

```
public PropChannel()
```

Class PropDefinition

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropDefinition
```

Inheritance

[object](#) ← PropDefinition

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

PropDefinition()

```
public PropDefinition()
```

Fields

Prefab

```
public GameObject Prefab
```

Field Value

GameObject

ProceduralOffsetsOverride

```
public ProceduralPropSpotOffsetsSO ProceduralOffsetsOverride
```

Field Value

[ProceduralPropSpotOffsetsSO](#)

PropChannel

`public PropChannel PropChannel`

Field Value

[PropChannel](#)

RotationSettings

`public PropRotationSettings RotationSettings`

Field Value

[PropRotationSettings](#)

Weight

`public float Weight`

Field Value

[float](#)

Properties

Placement

`public PropPlacementType Placement { get; set; }`

Property Value

[PropPlacementType](#)

Class PropOptionDefinition

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropOptionDefinition
```

Inheritance

[object](#) ← PropOptionDefinition

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

PropOptionDefinition()

```
public PropOptionDefinition()
```

Fields

OptionalRotation

```
public PropRotationSettings OptionalRotation
```

Field Value

[PropRotationSettings](#)

Prefab

```
public GameObject Prefab
```

Field Value

GameObject

Weight

```
public float Weight
```

Field Value

[float](#) ↗

Class PropOptions

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropOptions
```

Inheritance

[object](#) ← PropOptions

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

PropOptions()

```
public PropOptions()
```

Fields

Channel

```
public PropChannel Channel
```

Field Value

[PropChannel](#)

Options

```
public PropOptionDefinition[] Options
```

Field Value

[PropOptionDefinition\[\]](#)

ProceduralOffsetsOverride

```
public ProceduralPropSpotOffsetsSO ProceduralOffsetsOverride
```

Field Value

[ProceduralPropSpotOffsetsSO](#)

Properties

Placement

```
public PropPlacementType Placement { get; set; }
```

Property Value

[PropPlacementType](#)

Class PropPathBlock

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropPathBlock
```

Inheritance

[object](#) ← PropPathBlock

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

PropPathBlock()

```
public PropPathBlock()
```

Properties

Colliders

```
public HashSet<Collider> Colliders { get; }
```

Property Value

[HashSet](#)<Collider>

Methods

Process(PropPlacement)

```
public void Process(PropPlacement propPlacement)
```

Parameters

propPlacement [PropPlacement](#)

Reset()

```
public void Reset()
```

Class PropPlacement

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropPlacement : MonoBehaviour, IMapGenPostProcess
```

Inheritance

[object](#) ← PropPlacement

Implements

[IMapGenPostProcess](#)

Constructors

PropPlacement()

```
public PropPlacement()
```

Properties

BakedMap

```
public BakedMap BakedMap { get; }
```

Property Value

[BakedMap](#)

PathHeight

```
public float PathHeight { get; }
```

Property Value

[float](#)

PathThickness

```
public float PathThickness { get; }
```

Property Value

[float](#)

ProceduralOffsetDefaults

```
public ProceduralPropSpotOffsetsSO ProceduralOffsetDefaults { get; }
```

Property Value

[ProceduralPropSpotOffsetsSO](#)

Methods

GetAllColliders(GameObject)

```
public static List<Collider> GetAllColliders(GameObject obj)
```

Parameters

obj GameObject

Returns

[List](#)<Collider>

GetEncapsulatingBounds(List<Collider>)

```
public static Bounds GetEncapsulatingBounds(List<Collider> allColliders)
```

Parameters

allColliders [List](#)<Collider>

Returns

Bounds

OnMapBaked(BakedMap)

```
public void OnMapBaked(BakedMap bakedMap)
```

Parameters

bakedMap [BakedMap](#)

Reset()

```
public void Reset()
```

Enum PropPlacementType

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
[Flags]  
public enum PropPlacementType
```

Fields

None = 0

OnCorridorCorners = 8

OnCorridorIntersections = 16

OnCorridorStraights = 4

OnDeadEnds = 128

OnDoorSpaces = 2

OnPropSpot = 1

OnRoomCenters = 64

OnRoomCorners = 32

Class PropPlacementTypeInt

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropPlacementTypeInt
```

Inheritance

[object](#) ← PropPlacementTypeInt

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

PropPlacementTypeInt()

```
public PropPlacementTypeInt()
```

Fields

value

```
public int value
```

Field Value

[int](#)

Class PropPlacementTypeIntDrawer

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropPlacementTypeIntDrawer : PropertyDrawer
```

Inheritance

[object](#) ← PropPlacementTypeIntDrawer

Constructors

PropPlacementTypeIntDrawer()

```
public PropPlacementTypeIntDrawer()
```

Methods

OnGUI(Rect, SerializedProperty, GUIContent)

```
public override void OnGUI(Rect position, SerializedProperty property, GUIContent label)
```

Parameters

position Rect

property SerializedProperty

label GUIContent

Class PropPlacementVolume

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropPlacementVolume : MonoBehaviour
```

Inheritance

[object](#) ← PropPlacementVolume

Constructors

PropPlacementVolume()

```
public PropPlacementVolume()
```

Class PropPlacementVolumeEditor

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropPlacementVolumeEditor : Editor
```

Inheritance

[object](#) ← PropPlacementVolumeEditor

Constructors

PropPlacementVolumeEditor()

```
public PropPlacementVolumeEditor()
```

Methods

OnInspectorGUI()

```
public override void OnInspectorGUI()
```

Class PropRotationSettings

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropRotationSettings : ScriptableObject
```

Inheritance

[object](#) ← PropRotationSettings

Extension Methods

[PropRotationSettingsExtensions.GetRandomRotation\(PropRotationSettings\)](#)

Constructors

PropRotationSettings()

```
public PropRotationSettings()
```

Properties

PlusOrMinus

```
public Vector3 PlusOrMinus { get; }
```

Property Value

Vector3

RotationIncrement

```
public float RotationIncrement { get; }
```

Property Value

[float](#) ↗

RotationOffsets

```
public PropOffsets RotationOffsets { get; }
```

Property Value

[PropOffsets](#)

XMax

```
public float XMax { get; }
```

Property Value

[float](#) ↗

XMin

```
public float XMin { get; }
```

Property Value

[float](#) ↗

YMax

```
public float YMax { get; }
```

Property Value

[float](#) ↗

YMin

```
public float YMin { get; }
```

Property Value

[float](#) ↗

ZMax

```
public float ZMax { get; }
```

Property Value

[float](#) ↗

ZMin

```
public float ZMin { get; }
```

Property Value

[float](#) ↗

Class PropRotationSettingsEditor

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropRotationSettingsEditor : Editor
```

Inheritance

[object](#) ← PropRotationSettingsEditor

Constructors

PropRotationSettingsEditor()

```
public PropRotationSettingsEditor()
```

Methods

OnInspectorGUI()

```
public override void OnInspectorGUI()
```

Class PropRotationSettingsExtensions

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public static class PropRotationSettingsExtensions
```

Inheritance

[object](#) ← PropRotationSettingsExtensions

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

GetRandomRotation(PropRotationSettings)

```
public static Quaternion GetRandomRotation(this PropRotationSettings settings)
```

Parameters

[settings](#) [PropRotationSettings](#)

Returns

Quaternion

Class PropSpot

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropSpot : MonoBehaviour, IPropSpot
```

Inheritance

[object](#) ← PropSpot

Implements

[IPropSpot](#)

Constructors

PropSpot()

```
public PropSpot()
```

Properties

EnvType

```
public EnvType EnvType { get; }
```

Property Value

[EnvType](#)

Offsets

```
public PropOffsets Offsets { get; }
```

Property Value

[PropOffsets](#)

PropChannel

```
public PropChannel PropChannel { get; }
```

Property Value

[PropChannel](#)

PropPlacementType

```
public PropPlacementType PropPlacementType { get; }
```

Property Value

[PropPlacementType](#)

Methods

CheckIfUsed()

```
public bool CheckIfUsed()
```

Returns

[bool](#)

GetRandomOffset()

```
public Vector3 GetRandomOffset()
```

Returns

Vector3

GetTransform()

```
public Transform GetTransform()
```

Returns

Transform

SetEnvType(EnvType)

```
public void SetEnvType(EnvType type)
```

Parameters

type [EnvType](#)

Use(GameObject)

```
public void Use(GameObject prop)
```

Parameters

prop GameObject

Class PropSpotEditor

Namespace: [MoraGames.MapGen.Props](#)

Assembly: MoraMapGen.dll

```
public class PropSpotEditor : Editor
```

Inheritance

[object](#) ← PropSpotEditor

Constructors

PropSpotEditor()

```
public PropSpotEditor()
```

Methods

OnInspectorGUI()

```
public override void OnInspectorGUI()
```

Namespace MoraGames.MapGen.RNG

Classes

[DistributionParams](#)

[DistributionParamsDrawer](#)

[DistributionParamsExtensions](#)

[FixedRandomProvider](#)

[MapGenRandom](#)

[MapGenRandomProvider](#)

[PropOffsets](#)

[PropOffsetsDrawer](#)

[PropOffsetsExtensions](#)

[SeedKeys](#)

[SeedManager](#)

Interfaces

[IRandomProvider](#)

Enums

[DistributionMode](#)

Enum DistributionMode

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public enum DistributionMode
```

Fields

Biased = 2

Normal = 1

Uniform = 0

Class DistributionParams

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public class DistributionParams
```

Inheritance

[object](#) ← DistributionParams

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Extension Methods

[DistributionParamsExtensions.Validate\(DistributionParams\)](#).

Constructors

DistributionParams()

```
public DistributionParams()
```

DistributionParams(DistributionMode)

```
public DistributionParams(DistributionMode mode)
```

Parameters

mode [DistributionMode](#)

DistributionParams(float, bool)

```
public DistributionParams(float biasStrength = 1, bool flipBias = false)
```

Parameters

biasStrength [float](#)

flipBias [bool](#)

DistributionParams(float, float, float, int)

```
public DistributionParams(float mean = 0, float stdDev = 1, float spreadFactor = 0.2, int maxRerolls = 10)
```

Parameters

mean [float](#)

stdDev [float](#)

spreadFactor [float](#)

maxRerolls [int](#)

Fields

BiasStrength

```
public float BiasStrength
```

Field Value

[float](#)

FlipBias

```
public bool FlipBias
```

Field Value

[bool](#)

Mode

`public DistributionMode Mode`

Field Value

[DistributionMode](#)

NormalMaxRerolls

`public int NormalMaxRerolls`

Field Value

[int](#)

NormalMean

`public float NormalMean`

Field Value

[float](#)

NormalSpreadFactor

`public float NormalSpreadFactor`

Field Value

[float](#)

NormalStdDev

```
public float NormalStdDev
```

Field Value

[float](#)

Methods

Biased(float, bool)

```
public static DistributionParams Biased(float biasStrength = 1, bool flipBias = false)
```

Parameters

biasStrength [float](#)

flipBias [bool](#)

Returns

[DistributionParams](#)

Normal(float, float, float, int)

```
public static DistributionParams Normal(float mean = 0, float stdDev = 1, float spreadFactor = 0.2, int maxRerolls = 10)
```

Parameters

mean [float](#)

stdDev [float](#)

spreadFactor [float](#)

`maxRerolls` [int ↗](#)

Returns

[DistributionParams](#)

Uniform()

`public static DistributionParams Uniform()`

Returns

[DistributionParams](#)

Class DistributionParamsDrawer

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public class DistributionParamsDrawer : PropertyDrawer
```

Inheritance

[object](#) ← DistributionParamsDrawer

Constructors

DistributionParamsDrawer()

```
public DistributionParamsDrawer()
```

Methods

GetPropertyHeight(SerializedProperty, GUIContent)

```
public override float GetPropertyHeight(SerializedProperty property, GUIContent label)
```

Parameters

property SerializedProperty

label GUIContent

Returns

[float](#)

OnGUI(Rect, SerializedProperty, GUIContent)

```
public override void OnGUI(Rect position, SerializedProperty property, GUIContent label)
```

Parameters

position Rect

property SerializedProperty

label GUIContent

Class DistributionParamsExtensions

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public static class DistributionParamsExtensions
```

Inheritance

[object](#) ← DistributionParamsExtensions

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

Validate(DistributionParams)

```
public static void Validate(this DistributionParams dps)
```

Parameters

dps [DistributionParams](#)

Class FixedRandomProvider

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public class FixedRandomProvider : IRandomProvider
```

Inheritance

[object](#) ← FixedRandomProvider

Implements

[IRandomProvider](#)

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

FixedRandomProvider(int, float)

```
public FixedRandomProvider(int fixedIndex, float fixedValue = 0)
```

Parameters

fixedIndex [int](#)

fixedValue [float](#)

Methods

Range(int, int)

```
public int Range(int minInclusive, int maxExclusive)
```

Parameters

`minInclusive` [int↗](#)

`maxExclusive` [int↗](#)

Returns

[int↗](#)

Value()

`public float Value()`

Returns

[float↗](#)

Interface IRandomProvider

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public interface IRandomProvider
```

Methods

Range(int, int)

```
int Range(int minInclusive, int maxExclusive)
```

Parameters

minInclusive [int↗](#)

maxExclusive [int↗](#)

Returns

[int↗](#)

Value()

```
float Value()
```

Returns

[float↗](#)

Class MapGenRandom

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public static class MapGenRandom
```

Inheritance

[object](#) ← MapGenRandom

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Properties

UnityValue

```
public static float UnityValue { get; }
```

Property Value

[float](#)

Methods

BiasedPick<T>(Random, IList<T>, float, bool)

```
public static T BiasedPick<T>(Random rng, IList<T> list, float biasStrength = 3, bool flipBias = false)
```

Parameters

rng [Random](#)

list [IList](#)<T>

biasStrength [float](#)

flipBias [bool](#)

Returns

T

Type Parameters

T

BiasedPick<T>(string, IList<T>, float, bool)

```
public static T BiasedPick<T>(string key, IList<T> list, float biasStrength = 3, bool flipBias = false)
```

Parameters

key [string](#)

list [IList](#)<T>

biasStrength [float](#)

flipBias [bool](#)

Returns

T

Type Parameters

T

BiasedRange(Random, int, int, float, bool)

```
public static int BiasedRange(Random rng, int minInclusive, int maxExclusive, float biasStrength, bool flipBias = false)
```

Parameters

`rng` [Random](#)

`minInclusive` [int](#)

`maxExclusive` [int](#)

`biasStrength` [float](#)

`flipBias` [bool](#)

Returns

[int](#)

BiasedRange(Random, float, float, float, bool)

```
public static float BiasedRange(Random rng, float min, float max, float biasStrength, bool flipBias = false)
```

Parameters

`rng` [Random](#)

`min` [float](#)

`max` [float](#)

`biasStrength` [float](#)

`flipBias` [bool](#)

Returns

[float](#)

BiasedRange(string, int, int, float, bool)

```
public static int BiasedRange(string key, int minInclusive, int maxExclusive, float biasStrength, bool flipBias = false)
```

Parameters

key [string](#)

minInclusive [int](#)

maxExclusive [int](#)

biasStrength [float](#)

flipBias [bool](#)

Returns

[int](#)

BiasedRange(string, float, float, float, bool)

```
public static float BiasedRange(string key, float min, float max, float biasStrength, bool flipBias = false)
```

Parameters

key [string](#)

min [float](#)

max [float](#)

biasStrength [float](#)

flipBias [bool](#)

Returns

[float](#)

BiasedValue01(Random, float, bool)

```
public static float BiasedValue01(Random rng, float biasStrength, bool flipBias = false)
```

Parameters

`rng` [Random](#)

`biasStrength` [float](#)

`flipBias` [bool](#)

Returns

[float](#)

Chance(Random, float)

```
public static bool Chance(Random rng, float probability)
```

Parameters

`rng` [Random](#)

`probability` [float](#)

Returns

[bool](#)

Chance(string, float)

```
public static bool Chance(string key, float probability)
```

Parameters

`key` [string](#)

probability [float](#)

Returns

[bool](#)

GetRng(string)

```
public static Random GetRng(string key)
```

Parameters

key [string](#)

Returns

[Random](#)

GetWeightedRandom<T>(IEnumerable<T>, Func<T, float>, string)

```
public static T GetWeightedRandom<T>(IEnumerable<T> options, Func<T, float> getWeight,  
string randomKey)
```

Parameters

options [IEnumerable](#)<T>

getWeight [Func](#)<T, [float](#)>

randomKey [string](#)

Returns

T

Type Parameters

T

Init(int)

```
public static void Init(int seed)
```

Parameters

seed [int](#)

NormalPick<T>(Random, IList<T>, float, int)

```
public static T NormalPick<T>(Random rng, IList<T> list, float spreadFactor = 0.2, int maxRerolls = 10)
```

Parameters

rng [Random](#)

list [IList](#)<T>

spreadFactor [float](#)

maxRerolls [int](#)

Returns

T

Type Parameters

T

NormalPick<T>(string, IList<T>, float, int)

```
public static T NormalPick<T>(string key, IList<T> list, float spreadFactor = 0.2, int
```

```
maxRerolls = 10)
```

Parameters

key [string](#)

list [IList](#)<T>

spreadFactor [float](#)

maxRerolls [int](#)

Returns

T

Type Parameters

T

NormalRange(Random, int, int, float, int)

```
public static int NormalRange(Random rng, int minInclusive, int maxExclusive, float  
spreadFactor = 0.2, int maxRerolls = 10)
```

Parameters

rng [Random](#)

minInclusive [int](#)

maxExclusive [int](#)

spreadFactor [float](#)

maxRerolls [int](#)

Returns

[int](#)

NormalRange(Random, float, float, float, int)

```
public static float NormalRange(Random rng, float min, float max, float spreadFactor = 0.2,
int maxRerolls = 10)
```

Parameters

`rng` [Random](#)

`min` [float](#)

`max` [float](#)

`spreadFactor` [float](#)

`maxRerolls` [int](#)

Returns

[float](#)

NormalRange(string, int, int, float, int)

```
public static int NormalRange(string key, int minInclusive, int maxInclusive, float
spreadFactor = 0.2, int maxRerolls = 10)
```

Parameters

`key` [string](#)

`minInclusive` [int](#)

`maxInclusive` [int](#)

`spreadFactor` [float](#)

`maxRerolls` [int](#)

Returns

[int](#)

NormalRange(string, float, float, float, int)

```
public static float NormalRange(string key, float min, float max, float spreadFactor = 0.2,  
int maxRerolls = 10)
```

Parameters

key [string](#)

min [float](#)

max [float](#)

spreadFactor [float](#)

maxRerolls [int](#)

Returns

[float](#)

NormalValue(Random, float, float)

```
public static float NormalValue(Random rng, float mean = 0, float stdDev = 1)
```

Parameters

rng [Random](#)

mean [float](#)

stdDev [float](#)

Returns

[float](#)

NormalValue(string, float, float)

```
public static float NormalValue(string key, float mean = 0, float stdDev = 1)
```

Parameters

key [string](#)

mean [float](#)

stdDev [float](#)

Returns

[float](#)

Pick<T>(Random, DistributionMode, IList<T>, float, int, float, bool)

```
public static T Pick<T>(Random rng, DistributionMode mode, IList<T> list, float  
normalSpreadFactor = 0.2, int maxRerolls = 10, float biasStrength = 3, bool flipBias  
= false)
```

Parameters

rng [Random](#)

mode [DistributionMode](#)

list [IList](#)<T>

normalSpreadFactor [float](#)

maxRerolls [int](#)

biasStrength [float](#)

flipBias [bool](#)

Returns

T

Type Parameters

T

Pick<T>(Random, DistributionParams, IList<T>)

```
public static T Pick<T>(Random rng, DistributionParams distributionParams, IList<T> list)
```

Parameters

`rng` [Random](#)

`distributionParams` [DistributionParams](#)

`list` [IList](#)<T>

Returns

T

Type Parameters

T

Pick<T>(string, DistributionMode, IList<T>, float, int, float, bool)

```
public static T Pick<T>(string key, DistributionMode mode, IList<T> list, float  
normalSpreadFactor = 0.2, int maxRerolls = 10, float biasStrength = 3, bool flipBias  
= false)
```

Parameters

`key` [string](#)

`mode` [DistributionMode](#)

`list` [IList](#)<T>

`normalSpreadFactor` [float](#)

`maxRerolls` [int](#)

`biasStrength` [float](#)

`flipBias` [bool](#)

Returns

`T`

Type Parameters

`T`

Pick<T>(string, DistributionParams, IList<T>)

```
public static T Pick<T>(string key, DistributionParams distributionParams, IList<T> list)
```

Parameters

`key` [string](#)

`distributionParams` [DistributionParams](#)

`list` [IList](#)<T>

Returns

`T`

Type Parameters

`T`

Range(Random, DistributionMode, int, int, float, int, float, bool)

```
public static int Range(Random rng, DistributionMode mode, int minInclusive, int maxExclusive, float normalSpreadFactor = 0.2, int normalMaxRerolls = 10, float biasStrength = 3, bool flipBias = false)
```

Parameters

`rng` [Random](#)

`mode` [DistributionMode](#)

`minInclusive` [int](#)

`maxExclusive` [int](#)

`normalSpreadFactor` [float](#)

`normalMaxRerolls` [int](#)

`biasStrength` [float](#)

`flipBias` [bool](#)

Returns

[int](#)

`Range(Random, DistributionMode, float, float, float, int, float, bool)`

```
public static float Range(Random rng, DistributionMode mode, float min, float max, float normalSpreadFactor = 0.2, int normalMaxRerolls = 10, float biasStrength = 3, bool flipBias = false)
```

Parameters

`rng` [Random](#)

`mode` [DistributionMode](#)

`min` [float](#)

`max` [float](#)

`normalSpreadFactor` [float](#)

`normalMaxRerolls` [int](#)

`biasStrength` [float](#)

`flipBias` [bool](#)

Returns

[float](#)

Range(Random, DistributionParams, float, float)

```
public static float Range(Random rng, DistributionParams distributionParams, float min,  
float max)
```

Parameters

`rng` [Random](#)

`distributionParams` [DistributionParams](#)

`min` [float](#)

`max` [float](#)

Returns

[float](#)

Range(string, DistributionMode, int, int, float, int, float, bool)

```
public static int Range(string key, DistributionMode mode, int minInclusive, int  
maxExclusive, float normalSpreadFactor = 0.2, int normalMaxRerolls = 10, float biasStrength  
= 3, bool flipBias = false)
```

Parameters

key [string](#)

mode [DistributionMode](#)

minInclusive [int](#)

maxExclusive [int](#)

normalSpreadFactor [float](#)

normalMaxRerolls [int](#)

biasStrength [float](#)

flipBias [bool](#)

Returns

[int](#)

Range(string, DistributionMode, float, float, float, int, float, bool)

```
public static float Range(string key, DistributionMode mode, float min, float max, float  
normalSpreadFactor = 0.2, int normalMaxRerolls = 10, float biasStrength = 3, bool flipBias  
= false)
```

Parameters

key [string](#)

mode [DistributionMode](#)

min [float](#)

max [float](#)

normalSpreadFactor [float](#)

normalMaxRerolls [int](#)

biasStrength [float](#)

`flipBias` [bool](#)

Returns

[float](#)

Range(string, DistributionParams, float, float)

```
public static float Range(string key, DistributionParams distributionParams, float min,  
float max)
```

Parameters

`key` [string](#)

`distributionParams` [DistributionParams](#)

`min` [float](#)

`max` [float](#)

Returns

[float](#)

Shuffle<T>(List<T>, string)

```
public static void Shuffle<T>(List<T> list, string randomKey)
```

Parameters

`list` [List](#)<T>

`randomKey` [string](#)

Type Parameters

T

UniformPick<T>(Random, IList<T>)

```
public static T UniformPick<T>(Random rng, IList<T> list)
```

Parameters

rng [Random](#)

list [IList](#)<T>

Returns

T

Type Parameters

T

UniformPick<T>(string, IList<T>)

```
public static T UniformPick<T>(string key, IList<T> list)
```

Parameters

key [string](#)

list [IList](#)<T>

Returns

T

Type Parameters

T

UniformRange(Random, int, int)

```
public static int UniformRange(Random rng, int minInclusive, int maxExclusive)
```

Parameters

`rng` [Random](#)

`minInclusive` [int](#)

`maxExclusive` [int](#)

Returns

[int](#)

UniformRange(Random, float, float)

```
public static float UniformRange(Random rng, float minInclusive, float maxInclusive)
```

Parameters

`rng` [Random](#)

`minInclusive` [float](#)

`maxInclusive` [float](#)

Returns

[float](#)

UniformRange(string, int, int)

```
public static int UniformRange(string key, int minInclusive, int maxExclusive)
```

Parameters

`key` [string](#)

`minInclusive` [int](#)

`maxExclusive` [int](#)

Returns

[int](#)

UniformRange(string, float, float)

```
public static float UniformRange(string key, float minInclusive, float maxInclusive)
```

Parameters

`key` [string](#)

`minInclusive` [float](#)

`maxInclusive` [float](#)

Returns

[float](#)

UniformValue(Random)

```
public static float UniformValue(Random rng)
```

Parameters

`rng` [Random](#)

Returns

[float](#)

UniformValue(string)

```
public static float UniformValue(string key)
```

Parameters

key [string](#)

Returns

[float](#)

UnityRange(int, int)

```
public static int UnityRange(int minInclusive, int maxExclusive)
```

Parameters

minInclusive [int](#)

maxExclusive [int](#)

Returns

[int](#)

UnityRange(float, float)

```
public static float UnityRange(float minInclusive, float maxInclusive)
```

Parameters

minInclusive [float](#)

maxInclusive [float](#)

Returns

[float](#)

Value(Random, DistributionMode, float, float, float, bool)

```
public static float Value(Random rng, DistributionMode mode, float normalMean = 0, float  
normalStdDev = 1, float biasStrength = 3, bool flipBias = false)
```

Parameters

`rng` [Random](#)

`mode` [DistributionMode](#)

`normalMean` [float](#)

`normalStdDev` [float](#)

`biasStrength` [float](#)

`flipBias` [bool](#)

Returns

[float](#)

Value(Random, DistributionParams)

```
public static float Value(Random rng, DistributionParams distributionParams)
```

Parameters

`rng` [Random](#)

`distributionParams` [DistributionParams](#)

Returns

[float](#)

Value(string, DistributionMode, float, float, float)

```
public static float Value(string key, DistributionMode mode, float normalMean = 0, float  
normalStdDev = 1, float biasStrength = 3)
```

Parameters

key [string](#)

mode [DistributionMode](#)

normalMean [float](#)

normalStdDev [float](#)

biasStrength [float](#)

Returns

[float](#)

Value(string, DistributionParams)

```
public static float Value(string key, DistributionParams distributionParams)
```

Parameters

key [string](#)

distributionParams [DistributionParams](#)

Returns

[float](#)

Class MapGenRandomProvider

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public class MapGenRandomProvider : IRandomProvider
```

Inheritance

[object](#) ← MapGenRandomProvider

Implements

[IRandomProvider](#)

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

MapGenRandomProvider(Random)

```
public MapGenRandomProvider(Random rng)
```

Parameters

rng [Random](#)

Methods

Range(int, int)

```
public int Range(int minInclusive, int maxExclusive)
```

Parameters

minInclusive [int](#)

`maxExclusive` [int ↗](#)

Returns

[int ↗](#)

`Value()`

`public float Value()`

Returns

[float ↗](#)

Class PropOffsets

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public class PropOffsets
```

Inheritance

[object](#) ← PropOffsets

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Extension Methods

[PropOffsetsExtensions.Validate\(PropOffsets\)](#).

Constructors

PropOffsets()

```
public PropOffsets()
```

Fields

_xDistribution

```
public DistributionParams _xDistribution
```

Field Value

[DistributionParams](#)

_xOffsetMinMax

```
public Vector2 _xOffsetMinMax
```

Field Value

Vector2

_yDistribution

```
public DistributionParams _yDistribution
```

Field Value

[DistributionParams](#)

_yOffsetMinMax

```
public Vector2 _yOffsetMinMax
```

Field Value

Vector2

_zDistribution

```
public DistributionParams _zDistribution
```

Field Value

[DistributionParams](#)

_zOffsetMinMax

```
public Vector2 _zOffsetMinMax
```

Field Value

Vector2

Methods

GetCenter()

```
public Vector3 GetCenter()
```

Returns

Vector3

GetRandomOffsets()

```
public Vector3 GetRandomOffsets()
```

Returns

Vector3

GetRandomXOffset()

```
public float GetRandomXOffset()
```

Returns

[float](#)

GetRandomYOffset()

```
public float GetRandomYOffset()
```

Returns

[float](#)

GetRandomZOffset()

```
public float GetRandomZOffset()
```

Returns

[float](#)

GetSize()

```
public Vector3 GetSize()
```

Returns

Vector3

Class PropOffsetsDrawer

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public class PropOffsetsDrawer : PropertyDrawer
```

Inheritance

[object](#) ← PropOffsetsDrawer

Constructors

PropOffsetsDrawer()

```
public PropOffsetsDrawer()
```

Methods

GetPropertyHeight(SerializedProperty, GUIContent)

```
public override float GetPropertyHeight(SerializedProperty property, GUIContent label)
```

Parameters

property SerializedProperty

label GUIContent

Returns

[float](#)

OnGUI(Rect, SerializedProperty, GUIContent)

```
public override void OnGUI(Rect position, SerializedProperty property, GUIContent label)
```

Parameters

position Rect

property SerializedProperty

label GUIContent

Class PropOffsetsExtensions

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public static class PropOffsetsExtensions
```

Inheritance

[object](#) ← PropOffsetsExtensions

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Methods

Validate(PropOffsets)

```
public static void Validate(this PropOffsets offsets)
```

Parameters

offsets [PropOffsets](#)

Class SeedKeys

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public static class SeedKeys
```

Inheritance

[object](#) ← SeedKeys

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Fields

MapGenAssignAreas

```
public const string MapGenAssignAreas = "MapGenAssignAreas"
```

Field Value

[string](#)

MapGenMain

```
public const string MapGenMain = "MapGenMain"
```

Field Value

[string](#)

MapGenSkin

```
public const string MapGenSkin = "MapGenSkin"
```

Field Value

[string](#)

PostGen

```
public const string PostGen = "PostGen"
```

Field Value

[string](#)

PropPlacement

```
public const string PropPlacement = "PropPlacement"
```

Field Value

[string](#)

PropVariance

```
public const string PropVariance = "PropVariance"
```

Field Value

[string](#)

Class SeedManager

Namespace: [MoraGames.MapGen.RNG](#)

Assembly: MoraMapGen.dll

```
public class SeedManager
```

Inheritance

[object](#) ← SeedManager

Inherited Members

[object.GetType\(\)](#) , [object.MemberwiseClone\(\)](#) , [object.ToString\(\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.GetHashCode\(\)](#)

Constructors

SeedManager(int)

```
public SeedManager(int seed)
```

Parameters

seed [int](#)

Methods

GetRng(string)

```
public Random GetRng(string key)
```

Parameters

key [string](#)

Returns

[Random](#)

Namespace MoraGames.MapGen.RNG.Demo

Classes

[DemoMapGenRandom](#)

[DemoMapGenRandomEditor](#)

Class DemoMapGenRandom

Namespace: [MoraGames.MapGen.RNG.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoMapGenRandom : MonoBehaviour
```

Inheritance

[object](#) ← DemoMapGenRandom

Constructors

DemoMapGenRandom()

```
public DemoMapGenRandom()
```

Methods

Process()

```
public void Process()
```

Class DemoMapGenRandomEditor

Namespace: [MoraGames.MapGen.RNG.Demo](#)

Assembly: MoraMapGen.dll

```
public class DemoMapGenRandomEditor : Editor
```

Inheritance

[object](#) ← DemoMapGenRandomEditor

Constructors

DemoMapGenRandomEditor()

```
public DemoMapGenRandomEditor()
```

Methods

OnInspectorGUI()

```
public override void OnInspectorGUI()
```