

WAND AND CIRCLES ISOMETRIC TILES CREATOR

DOCUMENTATION

1. Introduction

The Isometric Tiles Creator is a tool for Unity that allows developers to easily generate isometric maps in their projects. The asset includes over 2300 prefabs, which are visually divided into 6 different sets: tiles, environment objects, nature objects, chests, coins, roads and bridges, stairs, portals, and more. Additionally, the asset includes ready-to-use Points of Interest (POI) prefabs, particle systems, HDRP lighting and volume prefabs, and patches for different render pipelines.

NOTE: Asset doesn't include nature shader which we use in our trailer videos on Asset Store page. We are using a free shader from Asset Store, and include instructions on how to add this shader to your project, also in asset included patch which will replace prefabs of nature objects with new prefabs, on which already attached material with this shader.

2. Getting Started

Before using "WNC - Isometric Tiles Creator", it is important to set up your project correctly. This includes installing the asset package and setting up your render pipeline.

Asset Installation and Setup

To install the asset package, you can use Unity's Package Manager. The package can be found in the Unity Asset Store, and can be easily imported into your project by following these steps:

- Open your Unity project and go to the "Window" menu.
- Select "Package Manager" from the menu.
- In the "All" tab, search for "WNC - Isometric Tiles Creator"
- Click the "Import" button to import the package into your project.

Render Pipeline Patch

To set up your render pipeline, you can use the included patches. The patches are located in the "Wand and Circles/Isometric Tiles Creator/Patch" folder and can be applied to your project by following the instructions in the README file.

Also in the patch folder there is a patch that updates the prefabs of our asset and adds materials with a shader for nature objects to the project. Read the README carefully to add the assets in the correct order.

ITC - Settings

It is also important to ensure that the settings file is located by the path:

Assets/Wand and Circles/Isometric Tiles Creator/Editor/Resources/WNC_ITC/Settings/Settings.asset

The settings are there by default. Just don't move them anywhere.

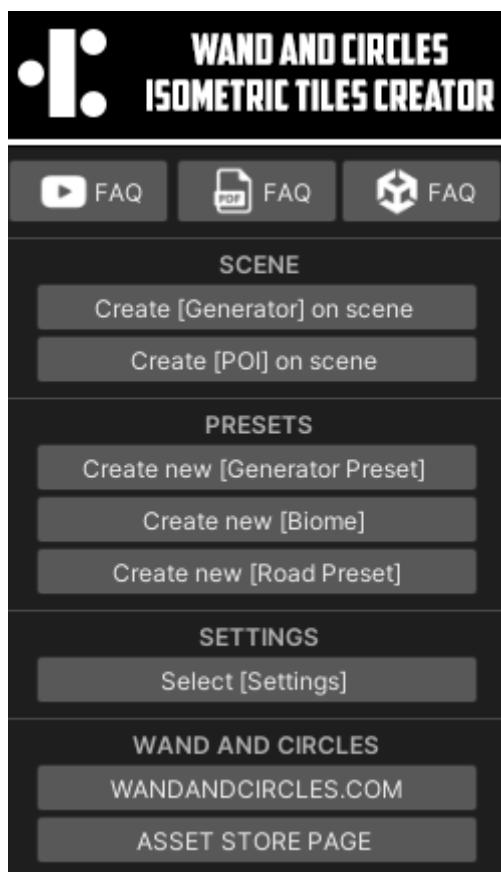
Namespace

All components are in the namespace: **using WNC.ITC;**

3. Components

The Isometric Tiles Creator asset is built on a number of different components, each with their own specific functionality. In this section, we will take a closer look at the key components of the asset and how they interact with each other to generate the maps. The Generator script and the Isometric Tiles Creator script work together to provide the functionality for generating maps, and the Biome, Road Preset, POI components provide the necessary prefabs and settings for the Generator to use during the generation process.

Isometric Tiles Creator Menu



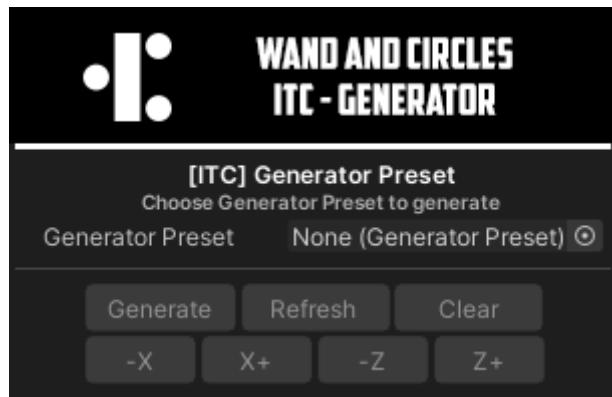
[Isometric Tiles Creator Menu]

The Isometric Tiles Creator Menu is an interface that provides access to all of the features of the asset. It can be opened from the top of the Unity editor by going to:

Tools > Wand and Circles > [Isometric Tiles Creator]

From this menu, you can easily create new generators, POIs, presets, or choose a settings file. It also includes a link to the FAQ and FAQ in Unity, as well as a YouTube tutorial.

Generator



[Generator Component]

The Generator script is responsible for generating maps based on a specified preset. It contains several functions that can be used to generate, refresh, and clear maps. The functions include:

public void Generate()

This function generates a new map based on the preset selected in the Generator component

public void Refresh()

This function refreshes the current map, updating it based on any changes made to the preset or other settings, also, if you refresh generated map - the generated landscape will remain, and only the filling will be regenerated

void ResetGenerator()

This function resets the map, removing all generated objects and set all variables to default

public void Clear()

This function clears the current map, removing all generated objects

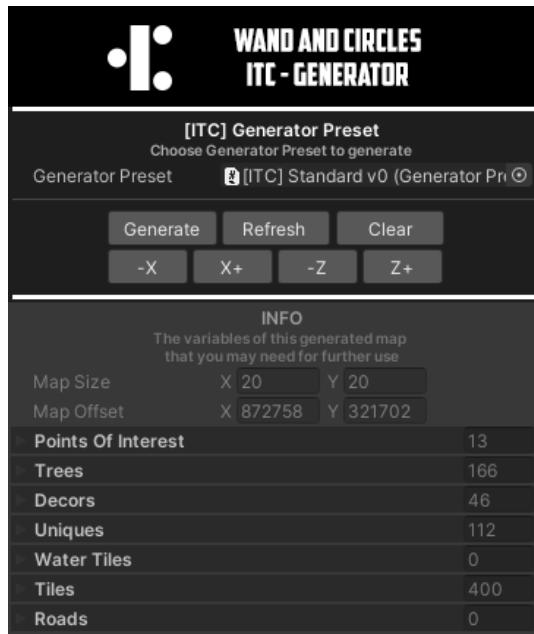
public void GenerateLeft()

public void GenerateRight()

public void GenerateDown()

public void GenerateUp()

These functions create new generators and generate new maps in the specified direction with GeneratorPreset which attached to this Generator component. Mostly, I added this to the generator demonstrate how to create adjacent seamless maps.



[Generator Component after Generate() function]

The following is the list of variables available in the Generator script after generation:

```
public bool mapGenerated
```

This variable indicates whether the map has been generated or not, it's used for UI of Generator component

```
public Vector2Int mapSize
```

This variable stores the size of the generated map

```
public Vector2Int mapOffset
```

This variable stores the noise offset of the generated map

```
public List<POI> pointsOfInterest
```

This variable stores the list of Points of Interest (POIs) that have been spawned during the map generation

```
public List<GameObject> trees
```

This variable stores the list of tree GameObjects that have been spawned during the map generation

```
public List<GameObject> decors
```

This variable stores the list of decor GameObjects that have been spawned during the map generation

```
public List<GameObject> uniques
```

This variable stores the list of unique GameObjects that have been spawned during the map generation

```
public List<GameObject> waterTiles
```

This variable stores the list of water tile GameObjects that have been spawned during the map generation

```
public List<GameObject> tiles
```

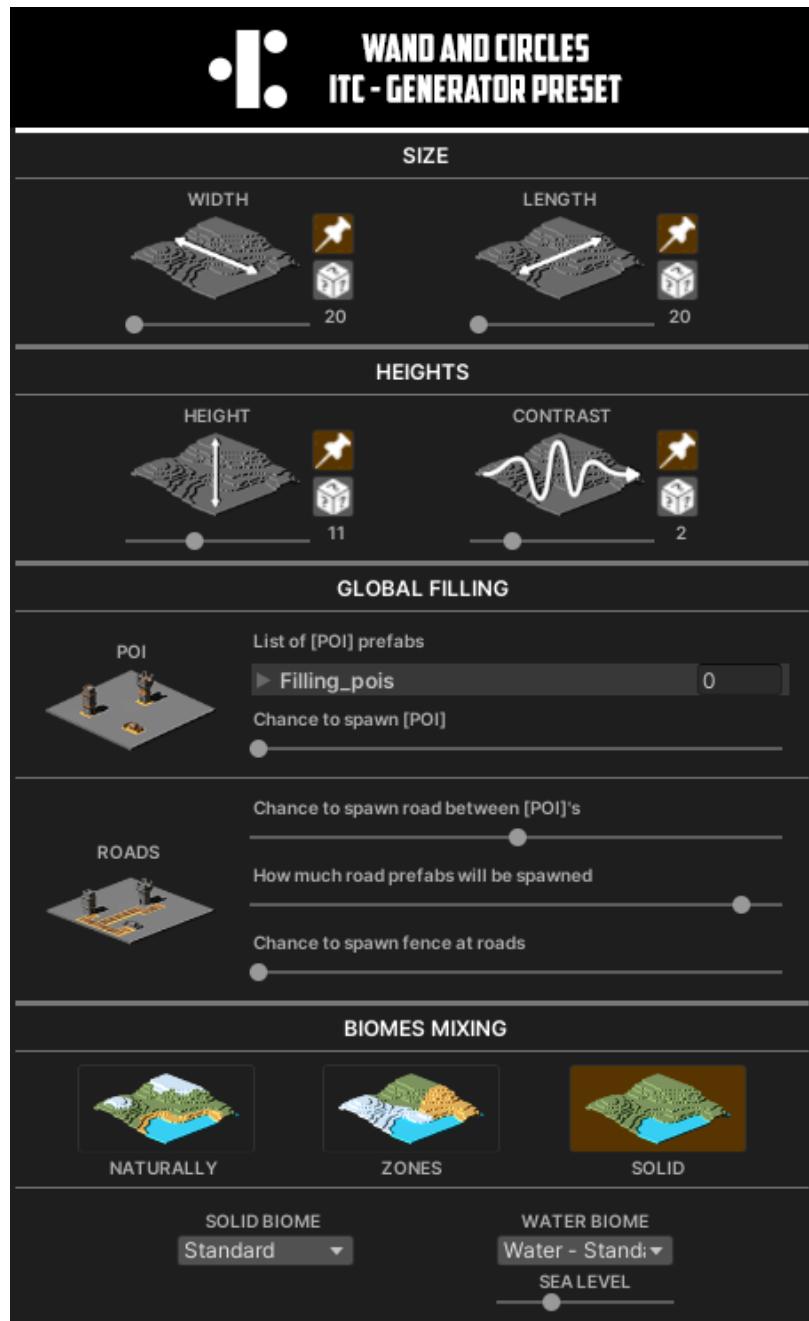
This variable stores the list of tile GameObjects that have been spawned during the map generation

```
public List<GameObject> roads
```

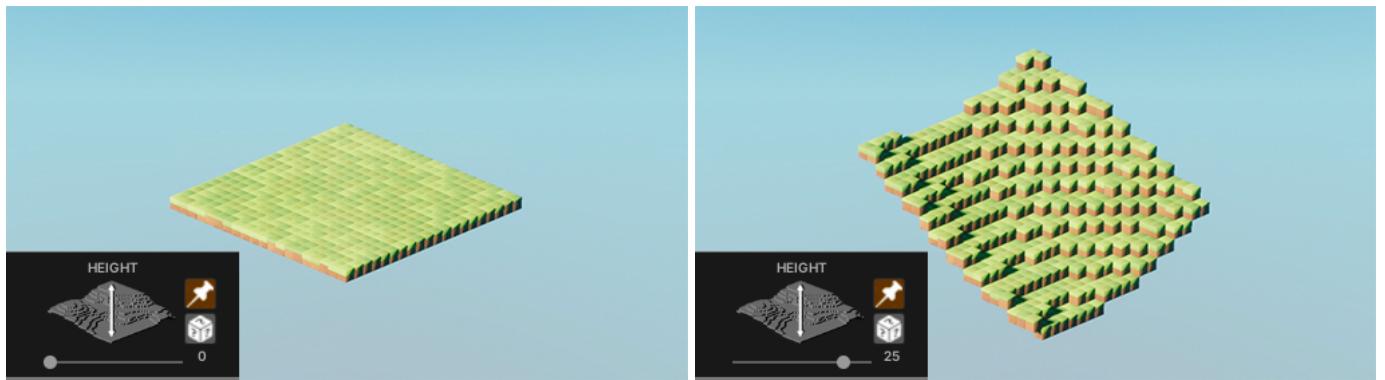
This variable stores the list of road GameObjects that have been spawned during the map generation

Generator Preset

The GeneratorPreset is a scriptable object that contains the settings for generating maps. It is used by the Generator script to determine the parameters for generating maps, such as size, max height, landscape contrast, global POI settings, road generating settings, and biome mixing.



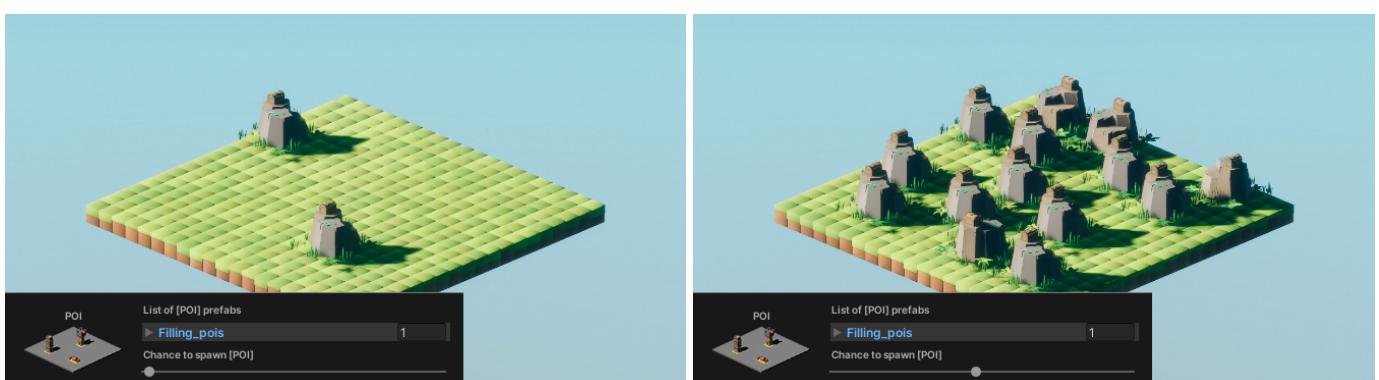
[Generator Preset Component]



This parameter controls the maximum possible height of the map. If height or contrast equal zero - map will be flat

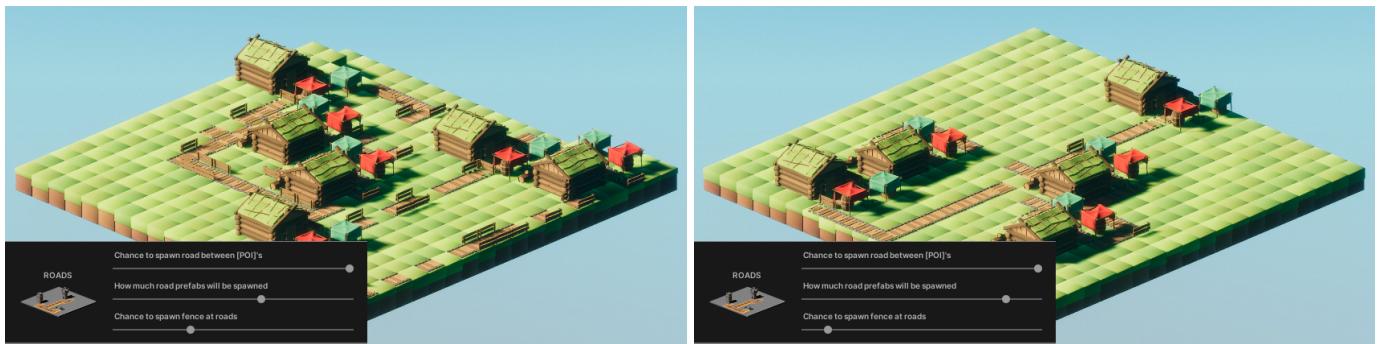


This parameter controls the difference in height between the highest and lowest points on the map. If the contrast is set to zero, the map will be flat



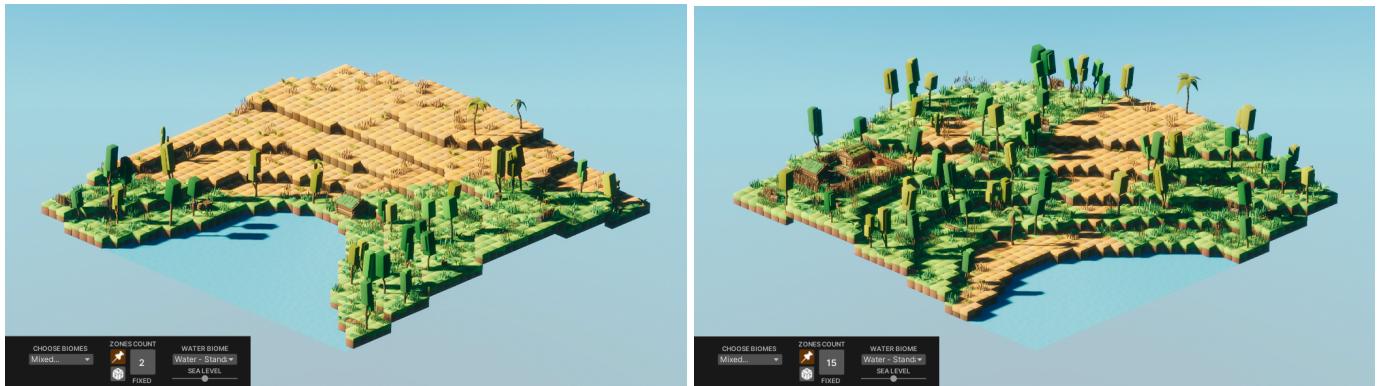
This section allows the user to add global Points of Interest (POIs) to the map

NOTE: The difference between *poi* that you add in biomes and *poi* that you add to global filling is that *poi* from global filling spawn regardless of the tile biome, and can be spawned for example at the junction of two biomes. *POIs* that in biomes will only spawn on tiles of their own biome.



This section allows the user to set the road generation settings, such as the width and curviness of the roads

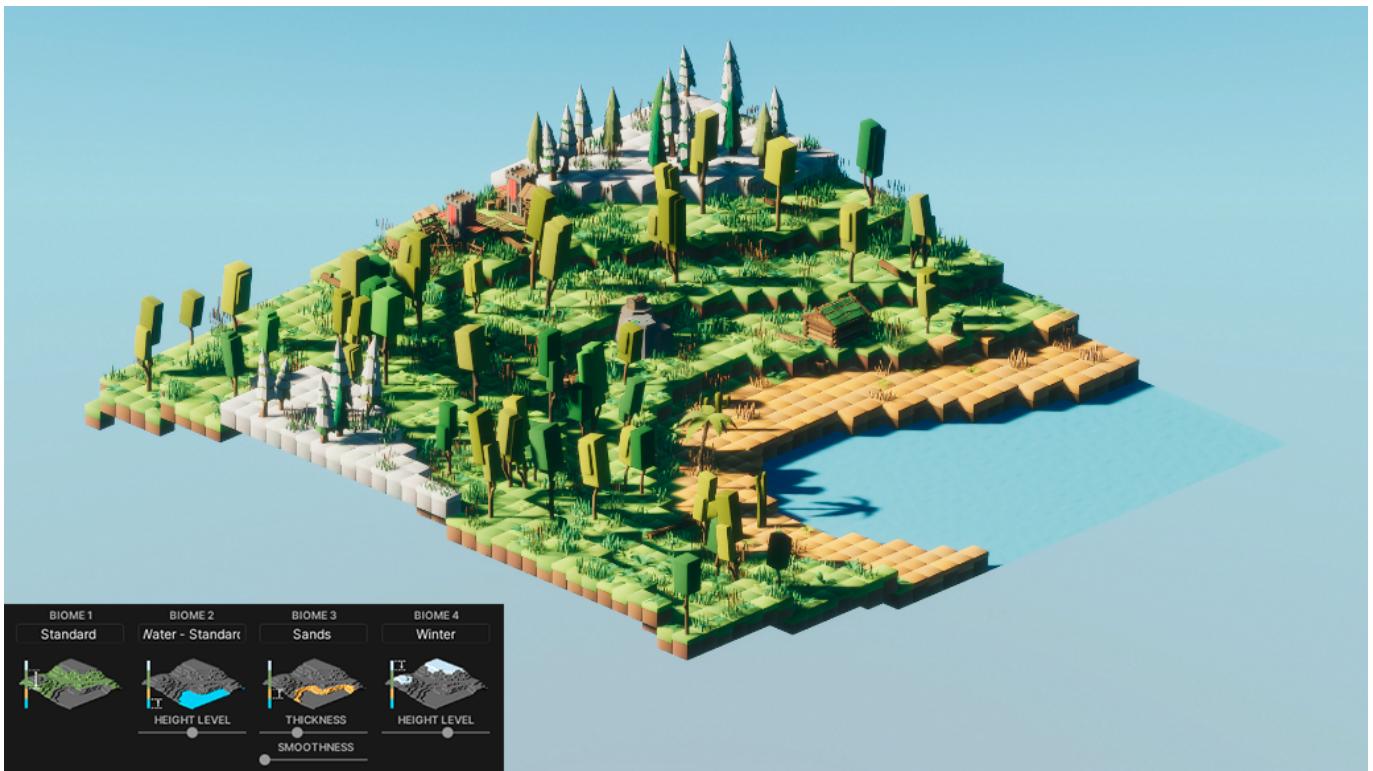
NOTE: All mixing types have a tab for selecting a water biome. If you don't want the generated map to have water, set the sea level slider to zero. The water biome can also be filled with poi and other objects, but no roads are built over the water biome.



Zones type of Biomes Mixing. Zones mixing type in the GeneratorPreset allows the user to specify a specific number of zones in which the map will be divided and set a different biome for each zone. You can specify the number of zones by entering a specific number or by setting a minimum and maximum value for a random number of zones. With this type, you can choose any number of biomes that will be used in the generation



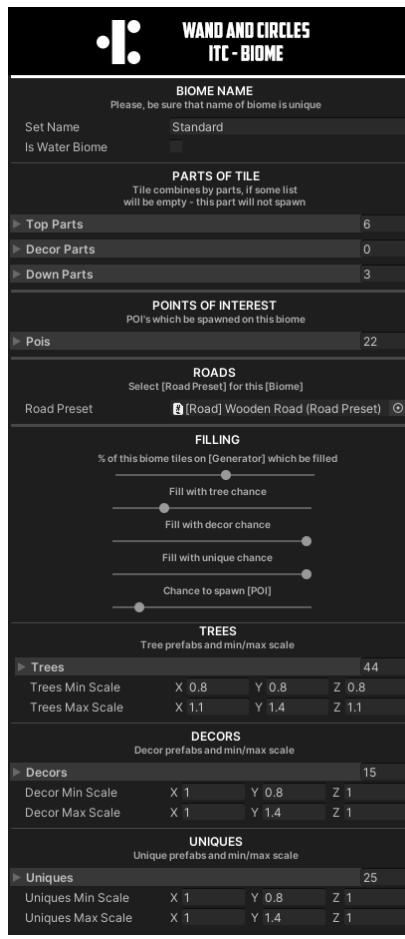
Solid mixing type in the GeneratorPreset allows the user to specify a single biome for the entire map. This means that all the tiles on the map will be generated using the selected biome.



Natural mixing type in the GeneratorPreset allows the user to specify up to four biomes to be used in generating the map, with one of the biomes being designated as a water biome. This type of mixing is meant to emulate natural terrain and ecosystems, where different types of biomes would be adjacent to each other. With this type of mixing, one biome is used as the main biome. The second one is located around the water bodies and you can adjust its width and smoothness of blending with the standard one. The third biome will be located above a certain height level, which you can also assign in the preset.

Biome

The Biome is a scriptable object that contains the settings for generating specific types of environment. It includes the prefabs of tiles, Points of Interest (POIs), road preset, and environment objects such as trees, decorations, and unique objects that are used to fill the tiles during map generation.



[Biome Component]

NOTE: Top, Decor, Down parts of the tile are used to generate the tile in that biome. You do not need to complete all three lists. When generating one tile, one prefab is randomly taken from each list and spawned in one place with a random rotation by y.

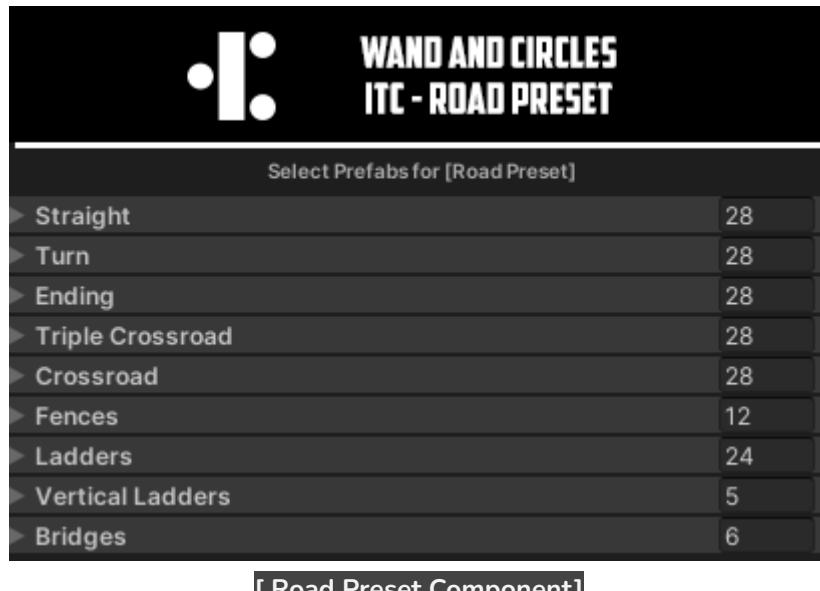
NOTE: The road preset will be used if a road needs to be spawned on this biome.

NOTE: Lists with trees, decors and uniques work on the same principle. % of filling of all tiles determines the number of tiles that will be filled with one of the objects from these three lists. The chance of spawning an object from a list is calculated as follows - all three chances of spawning of each type are summed up and taken as 100%, and the chance of each object spawning is equal to their chance of the total sum.

NOTE: The spawn chance of a poi is the number of tiles out of the total tiles of that biome that will check whether any poi from the poi list can spawn.

Road Preset

The RoadPreset is a scriptable object that contains the settings for generating roads and paths between Points of Interest (POIs) and biomes. It includes the prefabs of various road, ladder and bridge tiles that are used to generate paths during map generation.

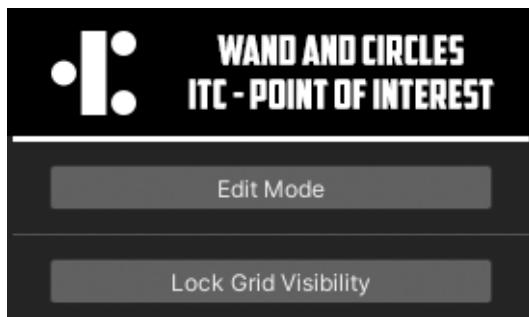


[Road Preset Component]

NOTE: If you will be creating your custom road presets from 3rd party prefabs, use the existing ones as a reference, and make sure your prefabs have the correct rotation (According to those prefabs that are already used in the included presets).

Point of Interest (POI)

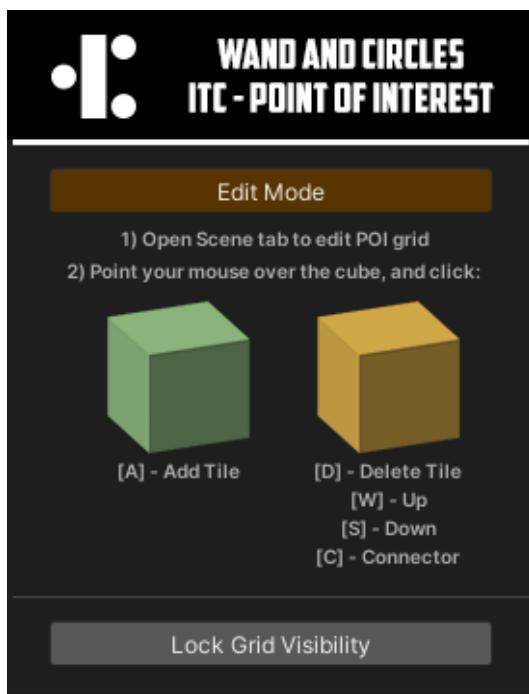
The Points of Interest (POI) component is a powerful tool for adding unique and large-scale objects to your map. POIs are used to specify the coordinates of tiles for correct spawning and can be used to embed large-scale and unique prefabs in your map.



[Point of Interest Component]

Point of Interest (POI) - How to create a new POI?

To create a new POI, you can open the **Isometric Tiles Creator Menu** and press the "**[Create new POI]**" button. This will add a new POI component to the scene.



[Point of Interest Component in Edit Mode]

Point of Interest (POI) - How to edit POI?

Once you have added a POI to the scene, you can enter "Edit Mode" by pressing the "**Edit Mode**" button in the **inspector window**. In the scene view, you will now see a grid that represents the POI's area. Over your mouse over POI grid, and press "A", "D", "W", "S" to edit the grid.

Point of Interest (POI) - How to add roads to POI?

In [Edit Mode] you can use the C key to add connectors to the grid. These connectors are used by the Generator to generate roads leading to the POI.

NOTE: *These connectors are used when generating a map for building roads. All connectors are taken from all poi spawned during generation, and connectors are randomly selected from them that will be connected by the road.*

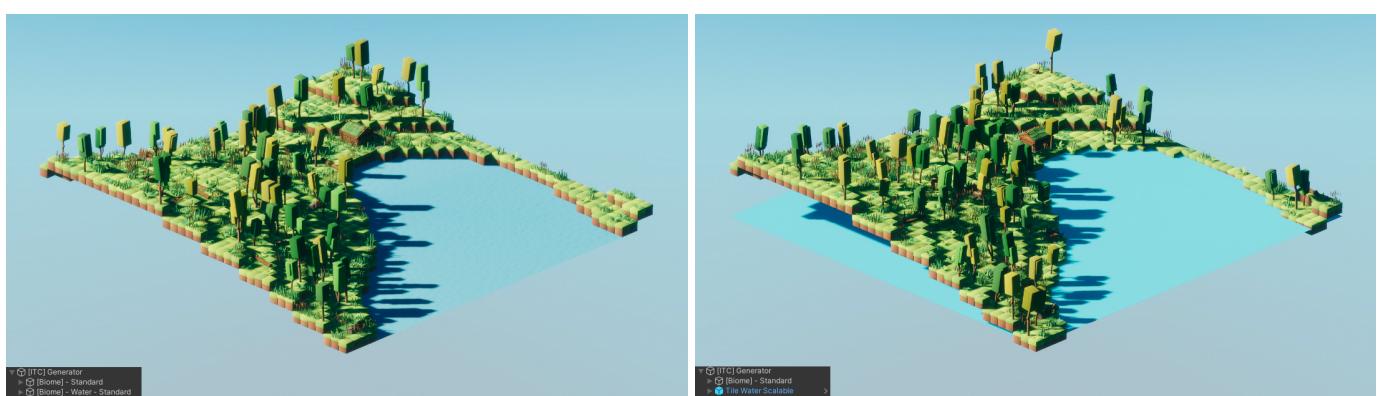
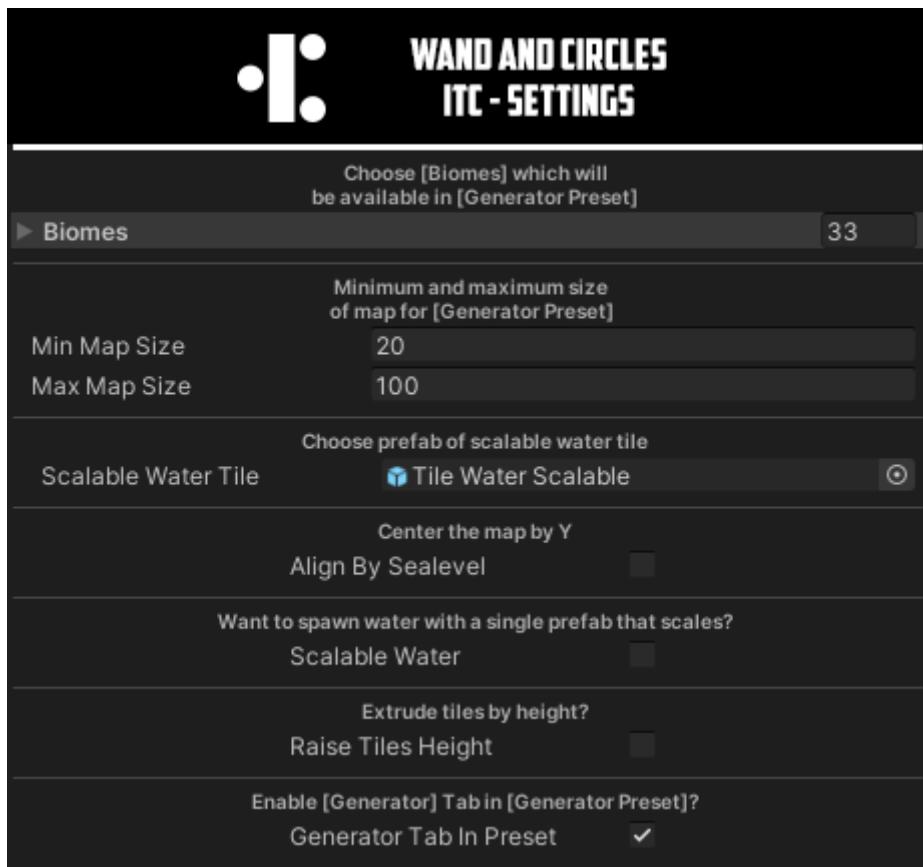
Point of Interest (POI) - How to add POI to map?

Once you have finished editing the POI (for example, we added different prefabs, particle systems, lighting sources to your POI, as on our prefabs included in the POI asset), you can save it as a prefab. You can then add the prefab to the necessary biome or generator preset so that it will be used in the map generation process.

NOTE: *It is important to note that the POIs can be used in both biomes and generator presets, so you can use them in different ways in your map.*

Settings

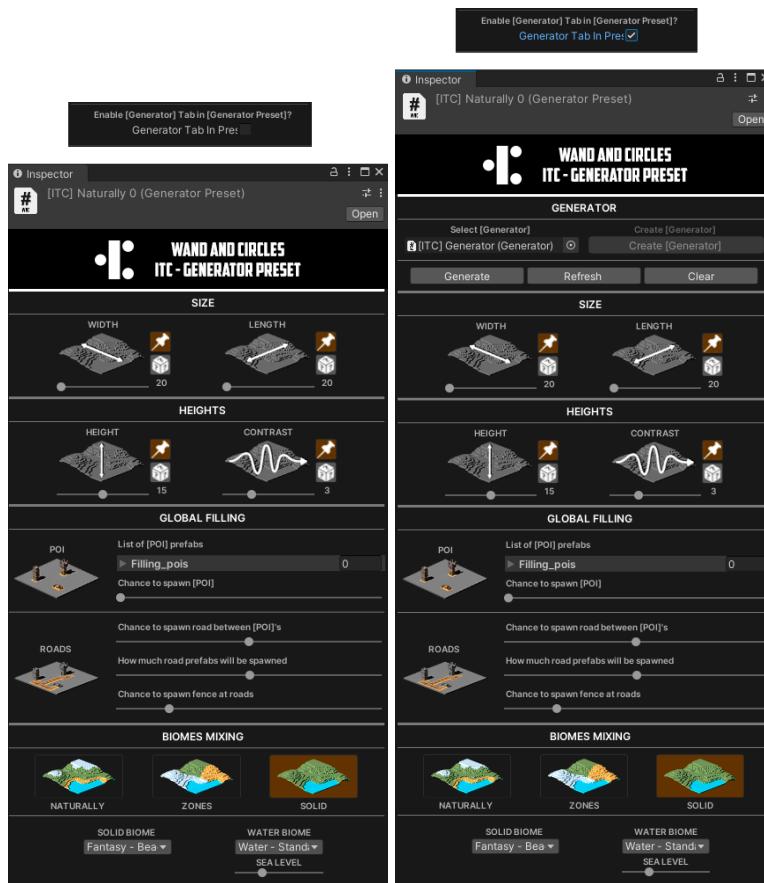
The Points of Interest (POI) component is a powerful tool for adding unique and large-scale objects to your map. POIs are used to specify the coordinates of tiles for correct spawning and can be used to embed large-scale and unique prefabs in your map.



Settings - Scalable Water

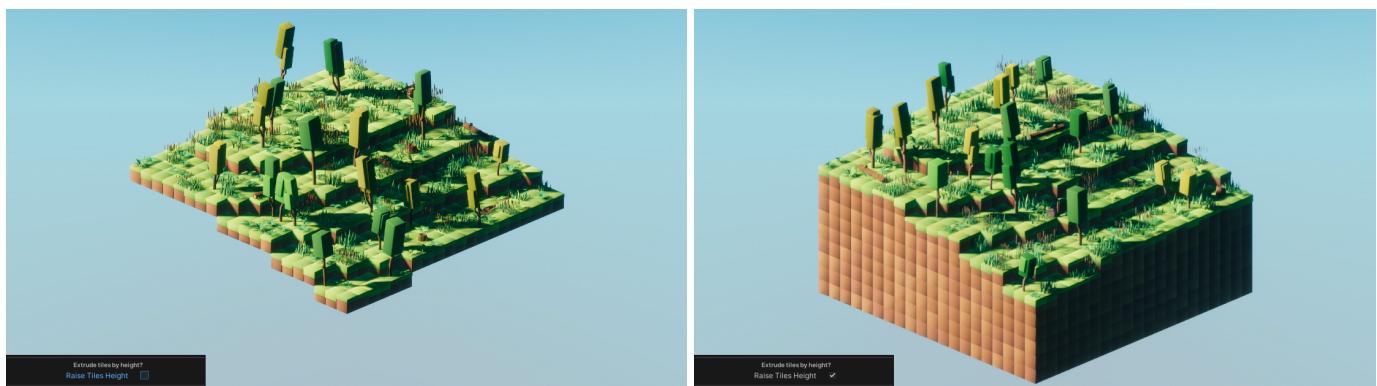
With the Scalable Water setting enabled, the generated maps will spawn the Scalable Water Prefab (Scalable Water Tile) and scale to the map size, instead of spawning a water biome.

NOTE: If you decide to replace the scaleable water prefab with your own, make sure that initially its scale matches our example.



Settings - Generator Tab in [Generator Preset]

“Generator Tab in Preset” allows you to enable an additional tab through which you can immediately generate a map with this preset on scene.



Settings - Raise Tiles Height

This setting slightly changes the principle of spawning tiles, and in each tile it will spawn additional bot parts in an amount that corresponds to the height of this tile on the map.

TileInfo

This component represents the tile and contains the information about them:

```
public Vector2Int tileCoordinates
```

This function generates a new map based on the preset selected in the Generator component

```
public Vector3 pos
```

Local position of tile

```
public string biome
```

Name of biome

```
public int distanceToWater
```

Distance to near tile of water biome

```
public bool blocked
```

Is Tile blocked by other object (like a POI, road or filling)

```
public bool connector
```

Is Tile a connector of POI

```
public bool road
```

Is Tile a part of road

Isometric Tiles Creator

A component that contains methods for generating a map. If you want to call generation with preset from code, you should use this function:
WNC.ITC.IsometricTilesCreator.GenerateMap(...)

You can use functions of component if you want to generate a map that goes beyond what you can do with a preset. In this case, we recommend paying attention to the GenerateMap() method in this component, you can use it as a good example with comments.

Methods that are implemented for generations:

```
public static TileInfo[,] NoiseForTileMap(...)
>Returns a TilePos[] with generated heights by noise
public static TileInfo[,] SetBiome(...)
Set one biome for all tiles from tilemap
public static TileInfo[,] InsertWater(...)
Insert a water biome in the tilemap at a given height
public static TileInfo[,] InsertBiomeAroundWater(...)
Insert a biome around the water biome to the tilemap, you can specify the thickness around the water and
the smoothness of the transition with bordering normal biomes
public static TileInfo[,] InsertBiomeOnHeights(...)
Insert in tilemap a biome located on the tops of the map, embedded at a given height
public static TileInfo[,] SplitInZones(...)
Split the tilemap into the specified number of zones, each of which is assigned a random biome from the
given array
public static TileInfo[,] SplitInZonesAlternative(...)
Split the tilemap into the specified number of zones, each of which is assigned a random biome from the
given array. Alternative method, breaks into zones of rectangular shapes
public static void CreateChildsStructure(...)
Creates empty GameObjects in child objects of the generator to structure and sort future spawned objects
public static void SpawnTiles(...)
Spawns all tiles for the tilemap
public static GameObject GenerateTile(...)
Returns GameObject of spawned tile
public static TileInfo[,] SpawnPOIs(...)
Calculates positions and spawns POI from the given array of POI Prefabs, with the specified spawn chance
If the biome is null, then POI will spawn on any biome, except water type biomes
public static TileInfo[,] InsertPOI(...)
Spawns on the tilemap given POI, on the specified biome, provided that there is a free and suitable place
for it on the map. If the biome is null, then POI will spawn on any biome, except water type biomes
public static TileInfo[,] CreateRoadsBetweenPOI(...)
Adds roads to the tilemap that are laid between the connectors of POI
public static void SpawnRoads(...)
Spawns all roads that are in the tilemap
public static void SpawnFilling(...)
Spawns all filling objects for the specified biome according to the settings specified in this biome
```