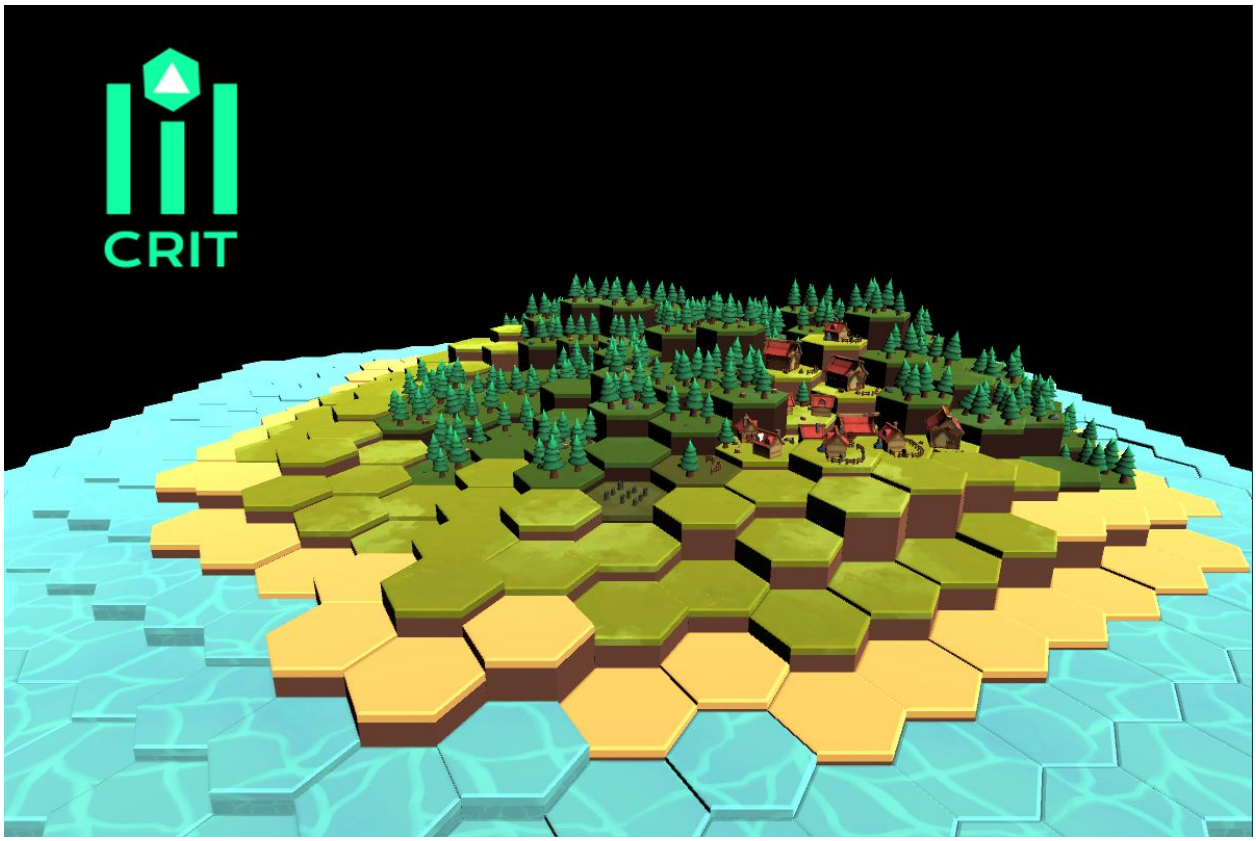
Hex Terrain Gen
V1.0



*** IMPORTANT ***

The hex terrain tool requires gizmos to be turned on for paint functionality to work properly, if you are trying to use the tool but you do not see the brush indicator or are unable to perform certain actions, check your gizmos.

This tool is designed to speed up the process of making Hexagon Floors for game levels. As such it has the ability to paint height, biome, and props onto the floor.

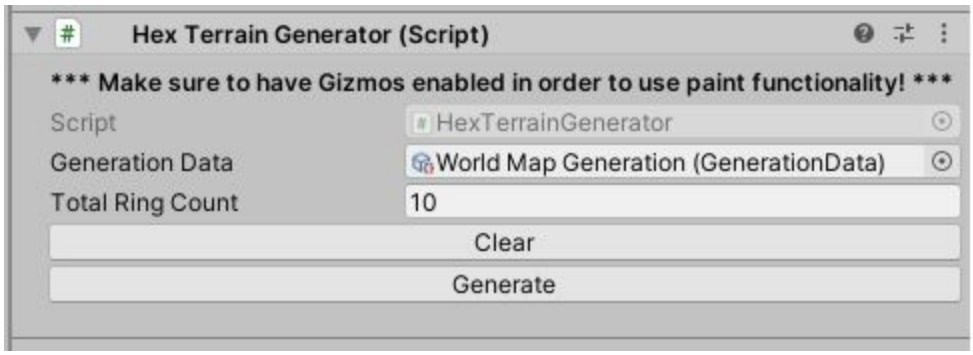
Quick Start

1. Drag in the **Hex Terrain Generator** prefab into the scene
2. Choose the **Generation Data** scriptable object that this generator will use for hex sizes, biome data, and more.
    a. Included in this asset is a premade Generation Data at ***"Data/Generation Data/World Map Generation"***, but you can make new ones as your game needs.
3. Enter the **Total Ring Count** you want for your hex floor. This determines the radius of the hex grid.
4. Click the **Generate** button to create the initial terrain.
5. From here, new options will appear allowing you to select the different options (biome, height, props, etc) and paint them onto the terrain (see details below).

Tool Overview

Hex Terrain Generator GUI

Pre Generation



Important Note: as the note at the top mentions if you don't have gizmos enabled the paint functionality won't work properly.

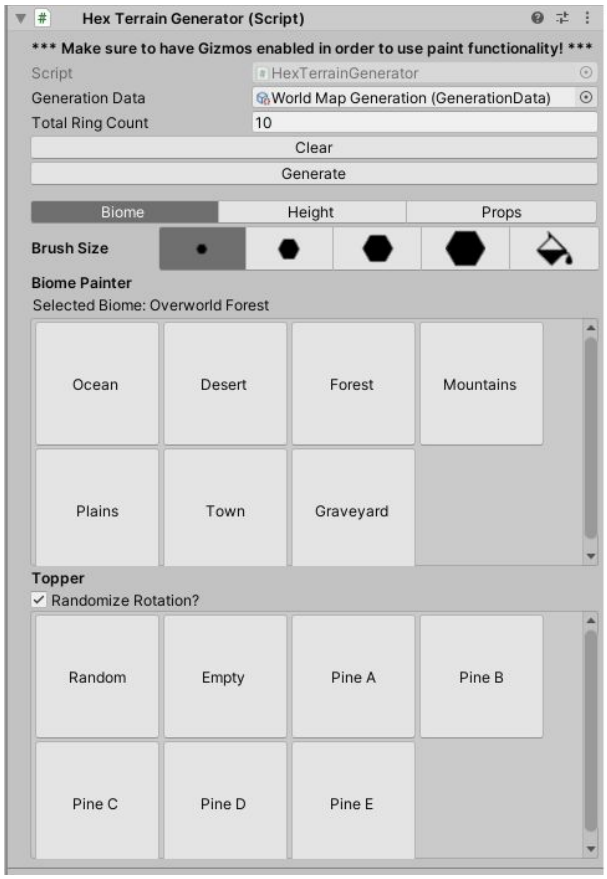
Generation Data: Needs to be set so the tool has the necessary data to create the terrain as well as the tool options for biome and props. A generation data is included in the asset.

Total Ring Count: The radius of rings of hexes that will be generated for the terrain. While all hexes and toppers are static by default, keep your target platform specs in mind. Higher ring counts will eventually mean heavier RAM usage, load times, or rendering depending how you use the tool.

Clear: Removes the current generation and all data in the scene.

Generate: Creates a new, blank terrain of hexes, with all hexes being set to the default biome of the generation data.

Post Generation - Biome

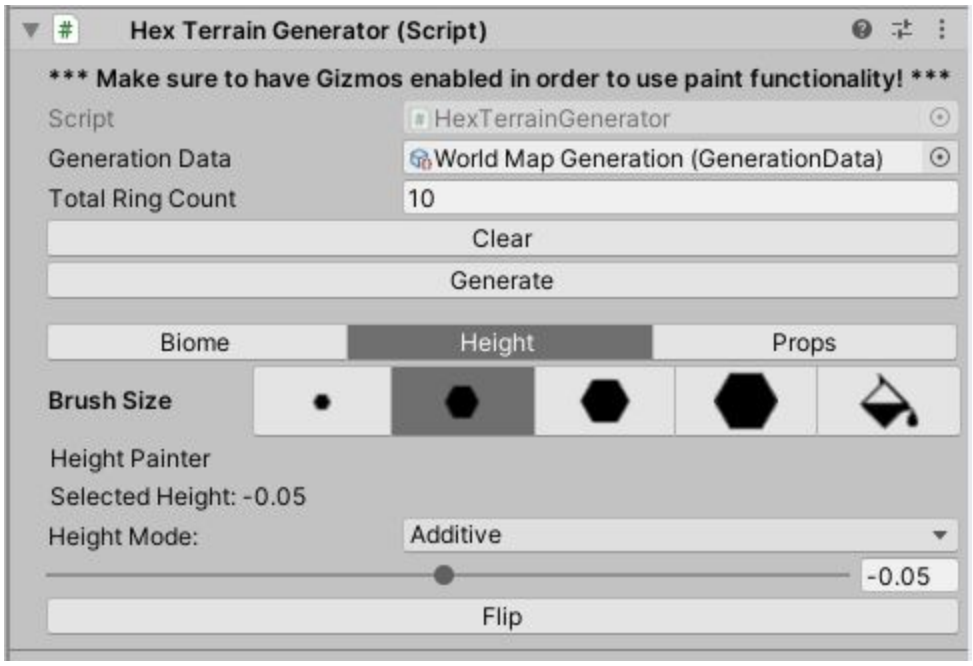


Brush Size: This is the current size set for the brush. Sizes go from a single hex to a fill option. The brush size will be reflected by the highlighted area when hovering over hexes in the scene view.

Selected Biome: The buttons in the first section are the different biomes you have set up in the generation data. Selecting one will allow you to paint that biome onto the hexes in the scene.

Topper: This allows you to select any potential toppers to be painted on top of the biome while you paint. The options you see are set in each of the biome data separate scriptables, as well as the option to randomly select toppers from the list as you paint, or empty, to not paint any toppers with the biome. Optionally, you can randomize the rotation of toppers as you paint to create some easy variation.

Post Generation - Height



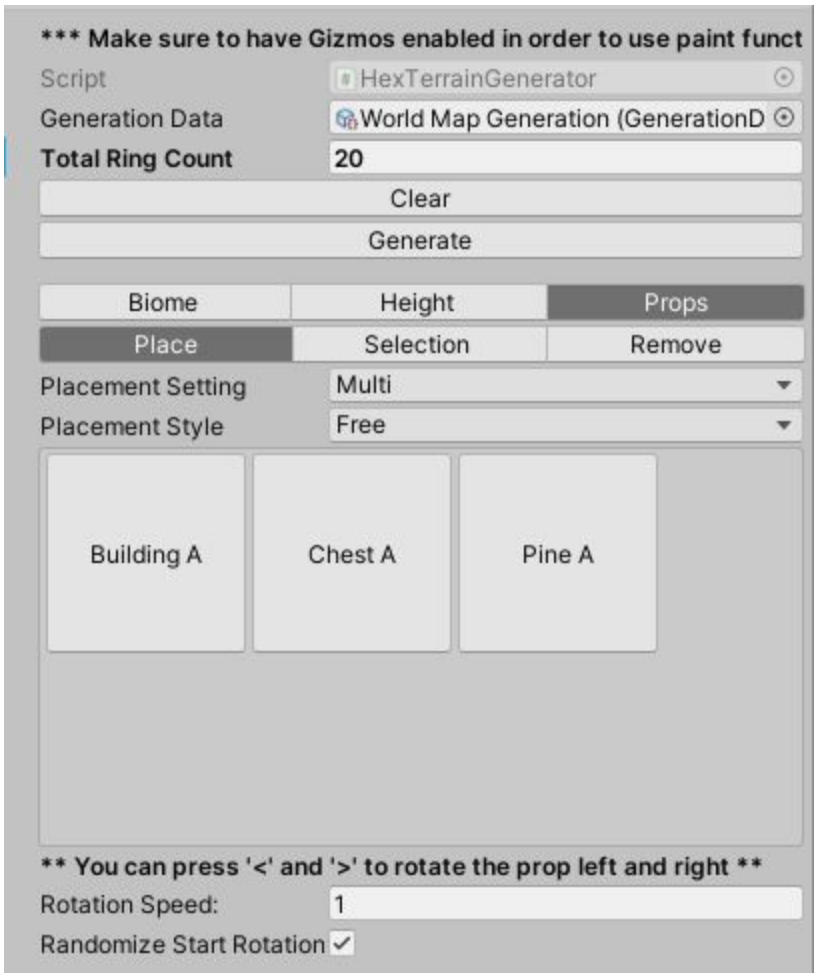
Height Mode: This option lets you choose between a few different modes for adding height.

- **Additive**: As you hold down the left mouse button, the height continually increases or decreases by the set value.
- **Override**: Height is set to the given value.
- **Variation Additive**: Height is continually adjusted by an amount between the min and max values.
- **Variation Override**: Height is set to a value between the min and max values.
- **Flatten**: Tiles within the brush radius are set to the same height as the tile at the center of the brush.

Value: Below height mode, you will set the height values to be used for the above options. The value input will change based on the height mode selected. This will typically be a slider or multiple input fields.

Flip: the flip button is a quick way to change the value to the opposite you have set.

Post Generation - Props



Place / Selection / Remove: these buttons switch the prop tool between placing new props, selecting existing props to move, or removing props.

Post Generation - Props - Place

Placement Setting: Determines the number of props to place in a row.

- Single: Place a single prop, after which another prop needs to be selected to continue placing.
- Multiple: Continue to place the selected prop until a new prop is chosen.

Placement Style: Choose between the snap option, forcing props to be placed at the center of the selected hex, or the free option, allowing for props to be placed anywhere on the grid.

Prop Selection: all the props that are set up in the generation data appear here for the user to select what prop they want to place.

Rotation Options: The prop can be rotated before and after placement with the '<' and '>' keys to rotate left and right.

Rotation Speed: Determines the speed at which the prop rotates.

Randomize Start Rotation: Allows props to be randomly rotated when placed.

Post Generation - Props - Selection

Selecting: Select any placed props by clicking their center. This is generally the hex where the prop was placed. With the proper selected, move it across the hex grid and place it by clicking the left mouse button.

Rotation: As with placement, the prop can be rotated before and after placement with the '<' and '>' keys to rotate left and right. The speed can be determined by the **Rotation Speed** value.

Post Generation - Props - Remove

Removing: With this option selected, click the center of the prop, generally the hex that the prop is placed on, to remove it from the grid.

Interacting with the Tool (Examples)

This tool is designed to be open ended, allowing for users to interact with and get data from the tool as needed. In this section we outline some use cases to give an idea of how you might use some of the tool properties to your advantage. Some of these cases may seem obvious, but if you are looking for specific information, this section may help guide you in what you are looking for.

Remember you can always add to or expand the functionality of this tool as you see fit, or feel free to simply use the data and perform any calculations in your own code externally from the tool.

Case 1 - Pathfinding

Something you will likely wish to do with this tool is find a path from point A to point B. This may be to determine a players movement across the map or for AI to roam from hex to hex.

On each hex is a script called ***floor tile***. This component contains a list of all adjacent hexes, topper information, prop, etc. You can add properties to this component should you need more information. The important information right now is the list of adjacent hexes. We don't provide a path finding algorithm with this tool, however you can use any path finding algorithm such as A* to create a path from one hex to another using distance as a weight and these lists to create a path for your characters or other purposes.

You can retrieve these components through many methods. For example, if you are using mouse clicks, you can set the hex prefabs to a specific layer and send a raycast from the screen and capture the hex the player clicked and retrieve the component that way. You could also maintain the floor tile component on your character's data components or vice versa for quick retrieval.

Case 2 - Hex Location Information

In most cases, you will want some of your hexes to mean something. Maybe a house on a tile represents an inn, or a special rock is a shrine that when stepped on grants certain powers, or maybe just a chest with bonus treasure. In any case, you will want to identify that such a location exists on a hex when your character enters it.

Again, the ***floor tile*** component located on every hex will be our starting point. Your special location can be identified by either a prop or a topper. This is your preference. You will see the floor tile component contains the data scriptables for the prop and topper that are placed on it, and can be accessed through here.

The topper is simply the prefab of the topper placed upon that hex. Since the prefab is custom made by you, you may store any information in it that you see fit and you can access it through here.

Alternatively, props always contain the ***prop base*** component, which is directly referenced by the floor tile script. While the prop is also a prefab, it may be more efficient for you to derive scripts from the prop base that include special functionality you can trigger directly through this chain of references. For example, you could derive a ***prop treasure*** script and place this on your custom prop instead. This script could then include methods for granting the player treasure, accessible entirely from the hex you place it on.

Ultimately how you provide the functionality is entirely up to you, but this is an example how you can find your functionality from a specific hex in your map.

Data Types

The hex terrain tool uses scriptable objects to hold the data for all parts of this tool. Below are the different types of data and how to use them.

Generation Data

The generation data is the main holder for all generation settings and data. It is placed directly into the Hex Terrain Generator component to fill out the inspector.

Fields

- Hex Offset
    - This list is used in the generation of the hex floor to give the proper separation positioning for each adjacent hex.
    - This shouldn't need to change unless you create your own hex model that does not match the size of the provided hex mesh.
    - An easy way to get these numbers for your new hex is to place six hexes in a position around a central hex and enter their transform position numbers into the hex offset list. Do this with the central hex positioned at 0, 0, 0.
- Hex Scale
    - Depending on the style of game, you may want to change these numbers to best suit your characters, toppers, and props. You may end up scaling your toppers and props down instead.
    - You should always leave the y as 1 and change the X and Z to the same number
- Default Biome Data
    - This is the default biome that will be used when generating the map, all tiles will be initially set to this.
- Biome Sets
    - These are all the biomes that can be painted once the map is generated.
    - You may want to have seperate sets of generation data with different biomes. For example, you may have a world map that is broken down into different regions represented by biomes, but also a town map that has a more specific set of biomes.
        - Biomes are a material for the ground and a list of toppers that can spawn on the tile. For the town example they could be a road tile or

a park tile, where a world map may have a desert tile or a forest tile.

- Props
    - These are a list of props that can be placed with this generator.
    - Props differ from tile toppers in that they can be placed over multiple tiles. You may use these for special objects in your world.

Biome Data

The biome data is where the appearance and toppers of a biome are set. A biome can be used to generalize a whole region, such as a forest, or as a specific kind of tile, such as a street or sidewalk.

Fields

- Biome ID:
    - An identifier for this biome. This is not used by the generator, but may be used for any specific identifier needs.
- Display Name:
    - What is shown in the hex terrain generator custom inspector.
- Display Texture
    - Texture that is shown on the button for this biome for easy identification.
- Biome Tiles
    - List of tile data for this biome. You may have one or multiple depending on preference. The floor tile data keeps the material that the tile will use. You could have multiple tile data in this list to randomize the material appears to vary how the ground of a region looks.
- Biome Toppers
    - This holds the topper list data for the toppers that can be placed on tiles of this biome.

Floor Tile Data

The floor tile data holds the data and information needed to properly initialize a hex tile.

Fields

- Biome
    - A naming field to identify the biome this tile belongs to.
- Floor Audio ID
    - Unused by the generator, but available should users wish to attach custom sounds for individual floor tiles.
- Buildable
    - Unused by the generator, but available should there be a building system and the need to know if a tile is able to be built upon.
- Surface Material
    - The material used for the top of a tile.
- Body Material
    - The material used for the sides of the tile.
- Tile Prefab
    - The prefab for this tile, which includes a hex mesh and a Floor Tile script attached. All floor tiles can be set to the same prefab, but can be replaced with a more custom hex to make certain biomes more interesting.

Topper List Data

The topper list data is used to hold a group of likened toppers together in an easy to find and use data set. Biomes can hold multiple of these lists, so organizing these lists in such a way that you can reuse your lists across multiple biomes is a good strategy.

Fields

- Tile Toppers
    - A list of tile topper data.

Tile Topper Data

The Tile Topper Data Holds all the data needed to place a tile topper on top of a Hex Tile

Fields

- ID

    - An identifying string to help identify a specific topper.
- Display Name
    - The name that is shown in the custom inspector for the button.
- Editor Icon
    - The icon that is shown in the custom inspector for the button.
- Topper Prefab
    - The prefab that is spawned on top of the hex tiles when placed.

Prop List Data

The prop list data is used to hold multiple prop data together in an easy to find list. Like topper lists, these are used to organize similar props together for ease of use.

Fields

- Props
    - A list of prop data.

Prop Data

The prop data is the information needed for spawning and placing props.

Fields

- ID
    - A string identifier used to identify a specific prop.
- Display Name
    - The name shown in the custom inspector.
- Flatten Radius
    - The radius around the prop where the ground will get flattened. Useful props that extend across multiple tiles.
- Editor Icon
    - The texture that will get displayed in the custom inspector for easy identification.
- Prop Prefab
    - The prefab of the prop that contains a Prop Base component.

Advanced

Adding New Models

Hexes

Hex model prefabs are set in the Floor Tile Data scriptable objects. If you want a specific floor tile to have a different hex prefab, it is set in this scriptable.

Using our standard floor tile as an example (at ***Assets/Prefab/Tiles/Overworld Floor Tile***), you will see the prefab is scaled to a 1, 1, 1 unit size, and contains a Floor Tile script on the highest parent.

The properties of the Floor Tile script that must be set are:

- Hex Parent: This is typically the FBX of the hex.
- Hex Body: A hex model should ideally be broken into two parts, a surface and a body. The body mesh is the sides of the hex. This value must be set as the body is what is stretched when the height of the hex is changed through the generator.
- Surface and Body Renderer: These renderers are set to guarantee that the materials you have set in the biome data can be applied to the hex properly.
- Selection Object: This is what is used by the generator to show the brush in the scene view. We use a simple hex outline, but this could be changed with any art you wish.

Hex Materials (Surface & Body)

To set new materials that will be used on the top and sides of hexes for each biome, you will need to go to the floor tile data using your hex prefab described above.

There are two fields; surface material and body material. These fields will set the materials of a hex when the Floor Tile Data is applied to it while painting with the generator.

Remember multiple floor tile data can be created for each biome, so you can have multiple variations of hexes for each biome.

Toppers

To add new toppers to the tool, you will need to create a new Tile Topper Data scriptable. This is where you will set the prefab for the topper and any display information (such as name and icon) for the tool to easily identify your new topper in the generator.

The prefab you set to this data will be what is spawned directly onto a hex when generating or painting your biomes. No special components are needed in the prefab. The prefab is a good place to also scale your assets to fit the hexes. In our samples, we have the parent at 1, 1, 1 scale and shrunk our assets originally scaled to match a standard character to fit our overworld map.

After finishing your new Tile Topper Data, you will either need to add it to an existing Topper List Data or create a new one. These are used to hold groups of toppers that will be commonly used together so that biomes can be set up with a short list of topper lists rather than a long unruly list of the toppers themselves. You can then mix and match lists of toppers into your biomes instead of constantly re-adding the same individual toppers multiple times.

Props

To add new Props to the tool, you will need to create a new Prop Data scriptable object. This is where you will set the prefab and other data for the Prop, similar to a topper.

The prefab is where you will want to scale your props to match your hex grid. The prefab must include a Prop Base component on the uppermost parent of the prefab. There are no values that need to be set on this component.

After creating the new Prop Data you will need to add it to a Prop Data List scriptable to be grouped with similar props. Prop Data Lists are set directly in the Generation Data scriptable.

Adding to the Code

[Possibly break into more sub sections. We should explain where to add more to the scriptables, recommendations on where to add expansions people may want, and where in code is best to add certain properties]