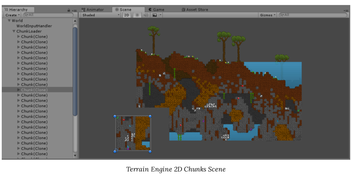
Chunks

In depth information on how terrain is rendered in the engine.

Table of Contents

General

Typically your world is going to be much larger than the size of your screen. Since at any time you will only be seeing a portion of the world, there is no reason for the whole world to be loaded into scene at all times, this only slows down your game. In order to solve this problem Terrain Engine 2D renders your world in chunks, so that only the world within view of the camera is loaded into the scene.



Terrain Engine 2D Chunks Scene

How Terrain is Rendered

Each chunk acts as a puzzle piece which can be put together to make the visible portion of the terrain. A chunk GameObject is made up of a Mesh, Chunk script, ColliderGenerator script and PolygonCollider2D. When a Chunk is instantiated, the Chunk script generates a mesh using the terrain data from its relative position in the world. Submeshes are used to generate the different layers of terrain, where each submesh uses the textured material of the layer it has generated. The terrain blocks of each layer combine to form each submesh which acts as a grid for which the block textures are mapped to.

The ColliderGenerator script generates the colliders for the terrain of that chunk. It does this by creating the PolygonCollider2D paths that trace along the edges of the terrain of that chunk. This is further explained in Terrain.

The ChunkLoader

Creation of these chunks is handled by the ChunkLoader. The ChunkLoader works by loading and unloading chunks based on their position relative to the **LoadTransform**. There is a distance threshold from the center position of the LoadTransform, which when crossed causes a row/column of chunks to be unloaded from one side of the terrain and reloaded at the other. The chunks are pooled so that they can be reused over and over again.

Terrain Engine 2D Chunk Loading

The ChunkLoader also handles updating the chunk graphics which must occur whenever a change is made to the terrain data. Depending on the location of the change and the type of change (block added or removed) one or more chunks will have their meshes regenerated.

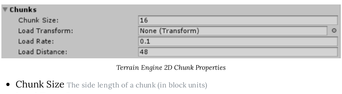
There are a few key properties and functions in the ChunkLoader which might be useful to you when creating your game:

- `OriginLoadedChunks` - This property stores the position of the bottom left block of the current loaded world
- `EndPointLoadedChunks` - This property stores the position of the top right block of the current loaded world
- `LoadChunksAtPosition` - This function allows you to specify a specific position to load chunks at. Which can be useful for teleporting to different parts of the terrain

The ChunkLoader also runs an event whenever chunks are loaded/unloaded: `OnChunksLoaded`. This can be very useful, and is used to keep the lighting and fluid simulation up to date with the currently loaded terrain information.

More information about the properties and functions of the ChunkLoader can be found in the API.

Chunk Properties



Terrain Engine 2D Chunk Properties

- Chunk Size The side length of a chunk (in block units)
- Load Transform The Transform of the GameOject where chunks will be loaded
- Load Rate The rate at which chunks are checked and loaded into the scene
- Load Distance The horizontal distance from the object which chunks will load in

The World inspector has a section where you can control some of the **Chunks** properties. The first property is the Chunk Size, which represents the side length of a single square chunk. A smaller chunk size means that your game will need more chunks to render a portion of your terrain compared to a larger chunk size, and chunks will need to be loaded more often, but the chunks will load in faster. A larger chunk size means chunks won't need to be loaded/unloaded as often, but it will take longer to load/unload chunks. Keep this in mind when choosing your chunk size (we recommend 16 by default). This needs to be an even number (generally a factor of 2) and has to be a factor of the width and height (if it is not the engine will round it for you).

The Load Transform is the GameObject whos position will be used to load chunks. It needs to be set, or else it will default to the Main Camera.

The Load Rate will determine how often the game will load chunks in seconds.

The Load Distance will determine how many chunks the game will load horizontally away from the Load Transform. It should be a multiple of the ChunkSize. This value is also used to calculate the vertical load distance, which will be proportionally equivalent.