Terrain Engine 2D
A 2D Block Engine for Unity
Out now on the Unity Asset Store

BUY NOW!

Textures

This page explains how to setup your art textures and tilemaps for use in the engine.

Table of Contents

The Basics

Terrain Engine 2D uses tilesets in order to render the beautiful terrain you see in the examples. In general these tilesets are very easy to setup, although there are a few rules you must follow in order to get everything working properly.

The terrain is made up of layers and each layer has its own set of block types. The layers determine the order in which blocks are rendered. Generally you will have a background layer, a main layer, and a foreground layer.

Each layer has its own tileset containing textures for all of the blocks which that layer contains. These are stored as Materials in Unity.

The very first thing you will want to figure out when planning your game is how many layers you will have, and what kind of blocks you will want in each layer. Although don't worry too much about getting down every single block and layer right at the start, as it is easy to make changes and add more blocks in later on.

Creating the Artwork

Once you have decided on your layers and have some ideas of the blocks you want to create, you can get started on creating your first tileset.

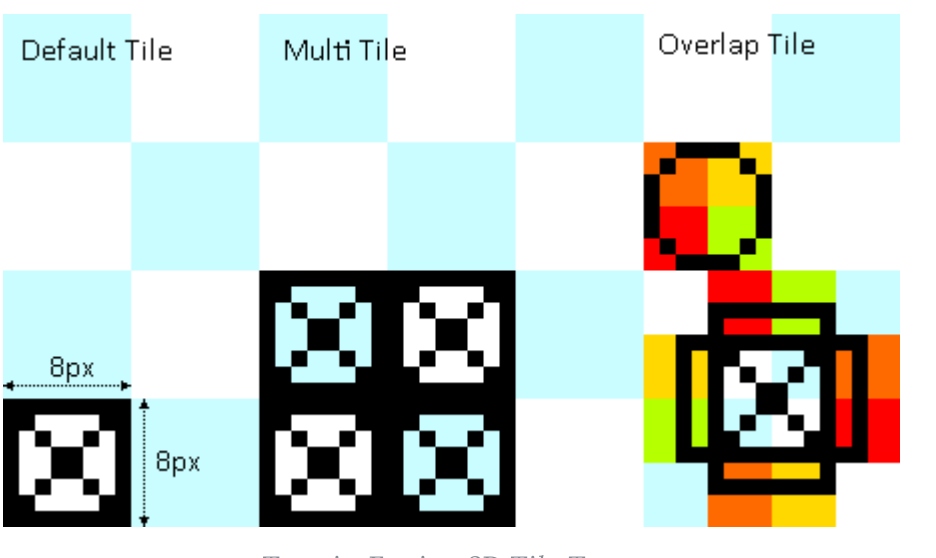
The tilesets used for Terrain Engine 2D are pretty straight forward, and there are plenty of examples included with the engine to help you out.

Now before you start creating tilesets and artwork to use with the block engine, you first need to decide what your **Pixels Per Block** ratio is going to be. This is the side length of a single tile in pixels. We recommend you stick with **powers of 2** (8, 16, 32, etc). Generally the higher ratio you use, the more detailed your blocks are going to have to be. In the included example we went with a Pixels Per Block ratio of 8.

Once you know what your Pixels Per Block ratio is, you can begin by creating your first block.

Block Types

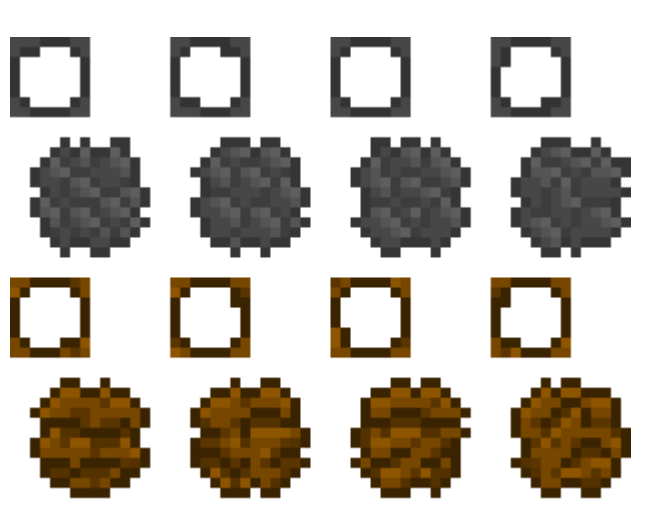
In Terrain Engine 2D we currently support three types of blocks. The **Default** block which is just a simple single block tile, this is used for any blocks which only take up one tile and don't require any kind of special blending. The **Multi Tile** block is similar to the Default block, except it takes up multiple tiles per block, this kind of block will likely be used for decoration and large objects. Lastly we have what we call the **Overlap Block**, this block is special in that it uses **Bitmasking** to render transitions and blend with the surrounding blocks. This block will likely be used for your terrain.



Terrain Engine 2D Tile Types

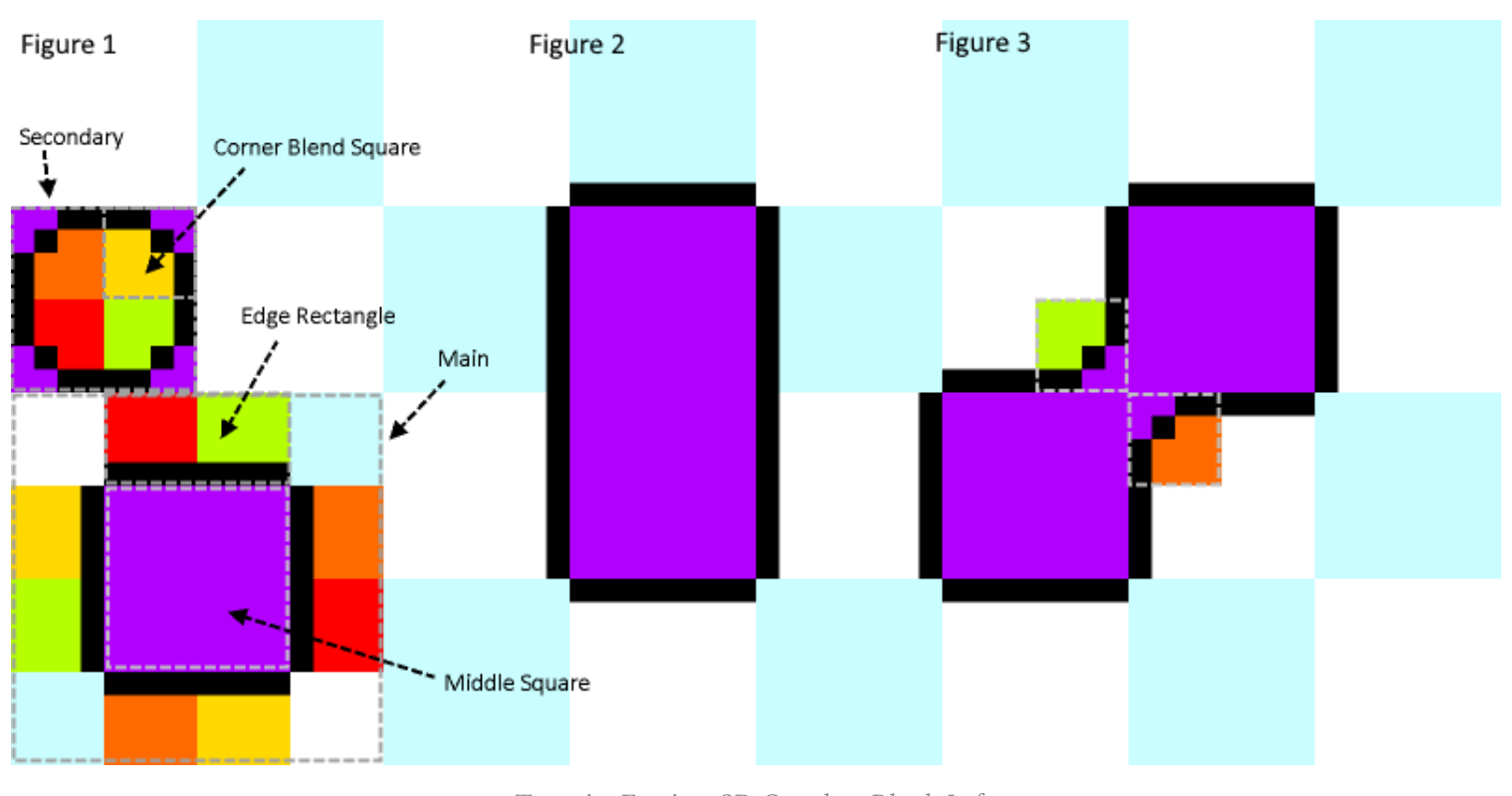
Overlap Block

Overlap Block's are unique in that certain portions of the texture are rendered depending on the surrounding blocks. They will overlap into other tiles to allow for nice transitions (or blending) between different block types.



Terrain Engine 2D Overlap Block Example

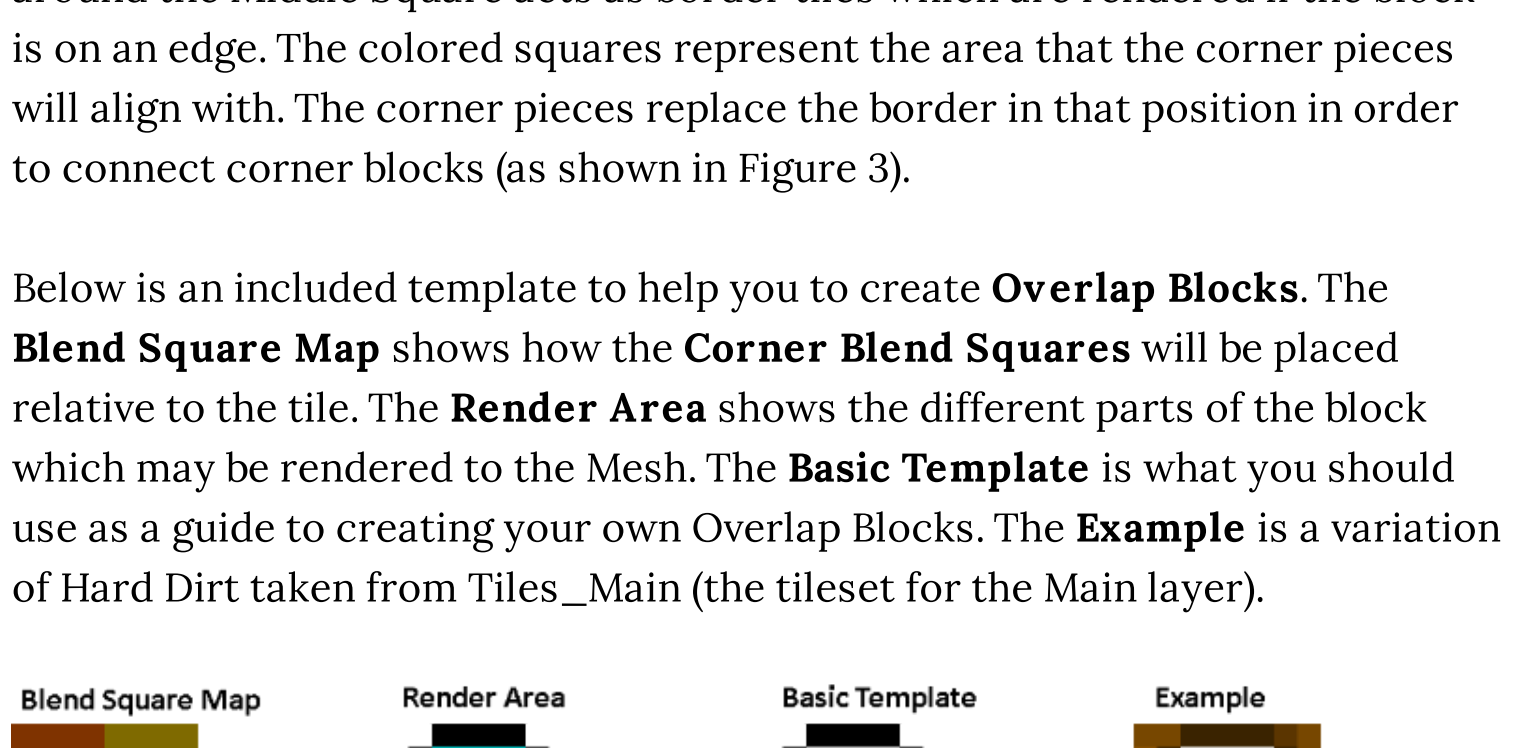
Overlap Blocks are 3 tiles in height and 2 tiles in width. Each Overlap Block is made up of two sections; the **Main** section which holds the main tile graphic, and the **Secondary** section which holds the **Corner Blend Squares**. The Middle Square of the **Main** section is always rendered. The **Edge Rectangles** are only rendered if there is no adjacent block of the same type at that edge. The **Corner Blend Squares** are rendered on top of the **Edge Rectangles** if there is an adjacent corner block. The **Corner Blend Squares** are mapped to the proper position as show by the color in the image below.



Terrain Engine 2D Overlap Block Info

The purple pixels represent the inside of the block and the black pixels act as a border to the outside. The Middle Square is the main texture of the block, it should not contain any border within it as it will be tiled. The surrounding area around the Middle Square acts as border tiles which are rendered if the block is on an edge. The colored squares represent the area that the corner pieces will align with. The corner pieces replace the border in that position in order to connect corner blocks (as shown in Figure 3).

Below is an included template to help you to create **Overlap Blocks**. The **Blend Square Map** shows how the **Corner Blend Squares** will be placed relative to the tile. The **Render Area** shows the different parts of the block which may be rendered to the Mesh. The **Basic Template** is what you should use as a guide to creating your own Overlap Blocks. The **Example** is a variation of Hard Dirt taken from Tiles_Main (the tileset for the Main layer).



Terrain Engine 2D Overlap Block Template

Variations

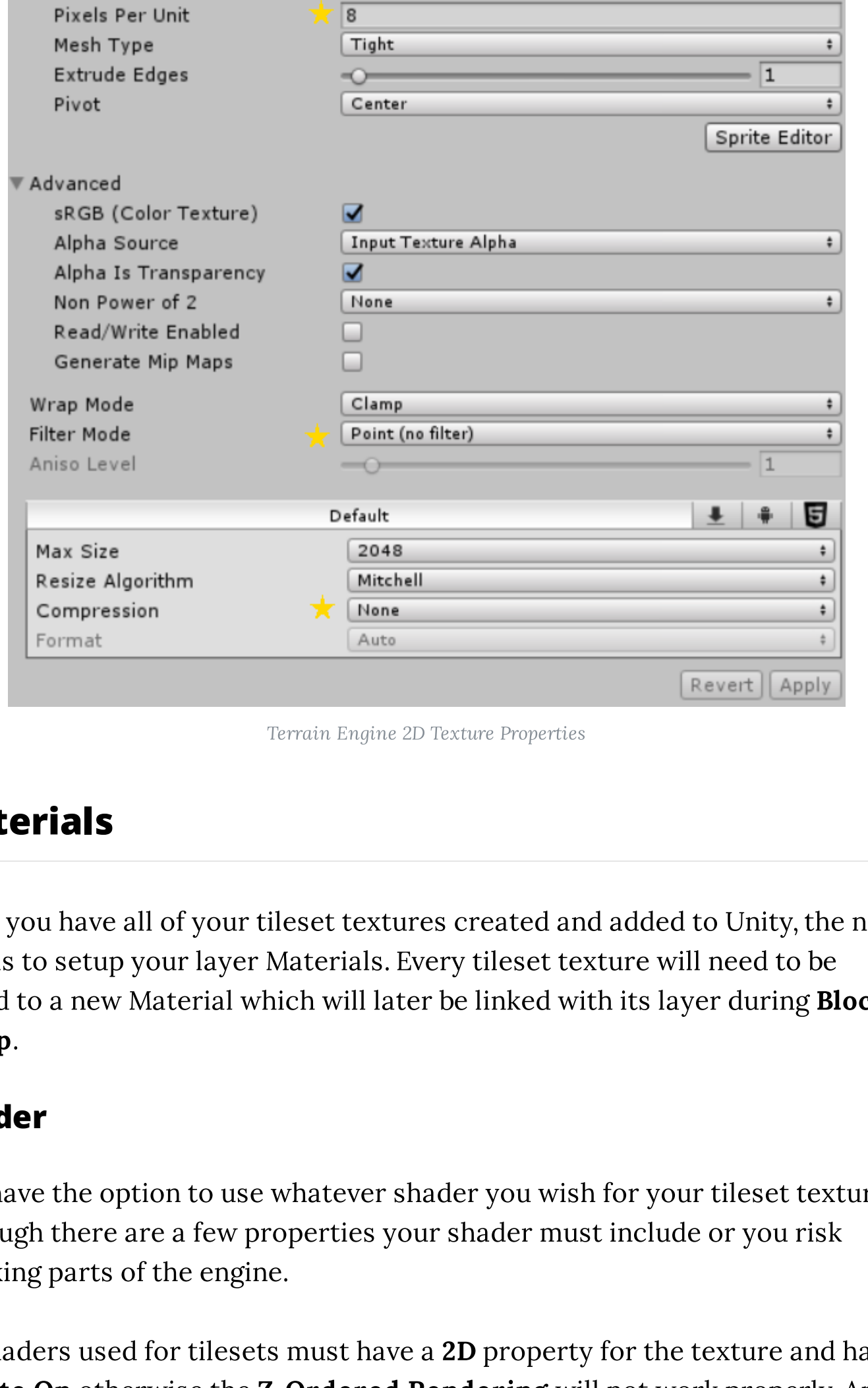
Each block can have as many variations as you would like. This allows you to further randomize your terrain and prevents any patterns from forming. We recommend you have at least 3 variations for any block used for terrain. To add a variation to your block simply create a variant of your block artwork in the adjacent right tile.

Terrain Engine 2D Variations Example

Textures in Unity

When adding your textures in Unity you should use the Recommended Settings for Textures found in Setup.

The recommended property settings for your tileset textures are shown below. Just be sure that your Pixels Per Unit is the same as your chosen Pixels Per Block ratio. These settings will give you a nice sharp pixel art style.



Terrain Engine 2D Texture Properties

Materials

Once you have all of your tileset textures created and added to Unity, the next step is to setup your layer Materials. Every tileset texture will need to be added to a new Material which will later be linked with its layer during **Block Setup**.

Shader

You have the option to use whatever shader you wish for your tileset textures, although there are a few properties your shader must include or you risk breaking parts of the engine.

All shaders used for tilesets must have a **2D** property for the texture and have **ZWrite On** otherwise the **Z-Ordered Rendering** will not work properly. Any tilesets using **Overlap Blocks** also need to use a **Transparent Cutout** shader. The recommended shader for tilesets is the: **TerrainEngine2D/Terrain** shader.

Material Setup

1. Start by creating a new Material in the Project Window of Unity.
2. Select your desired shader.
3. Add your tileset texture to the Material's Texture parameter.
4. Setup the shaders parameters.