Super Platformer Editor Manual

Creative Spore

Support: creativespore@gmail.com

Web: http://www.creativespore.com

Unity Forums: user CreativeSpore

Twitter: @CreativeSpore

Index

1 Introduction

This is the user manual for the Unity asset Super Platformer Editor.

Super Platformer Editor is a bundle including (**Super Tilemap Editor** and **Smart Platform Colliders**) to create platformer games using the best of both tools.

This manual will teach you how to use both tools to create a platform game.

2 Getting Started

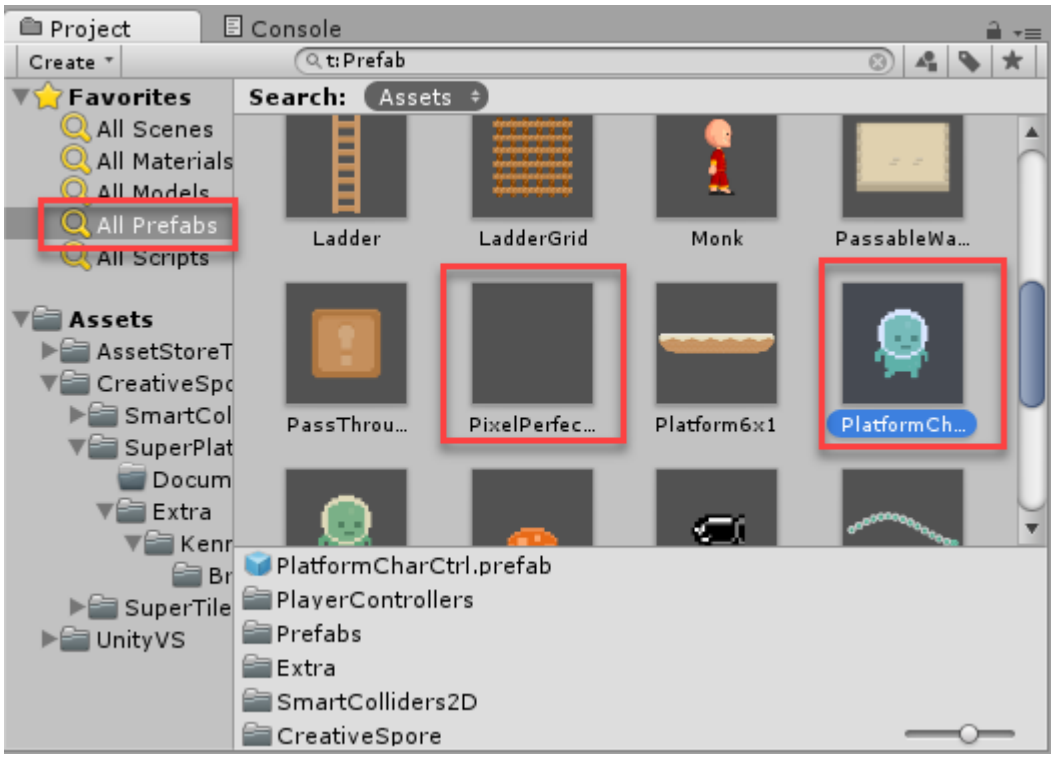
This section will explain how to create a demo scene with a Platform Character Controller and create the level using Super Tilemap Editor. All you need and the final scene can be found in the folder "**Assets\CreativeSpore\SuperPlatformerEditor\Extra\Kenneys - Platformer**"

2.1 First Steps

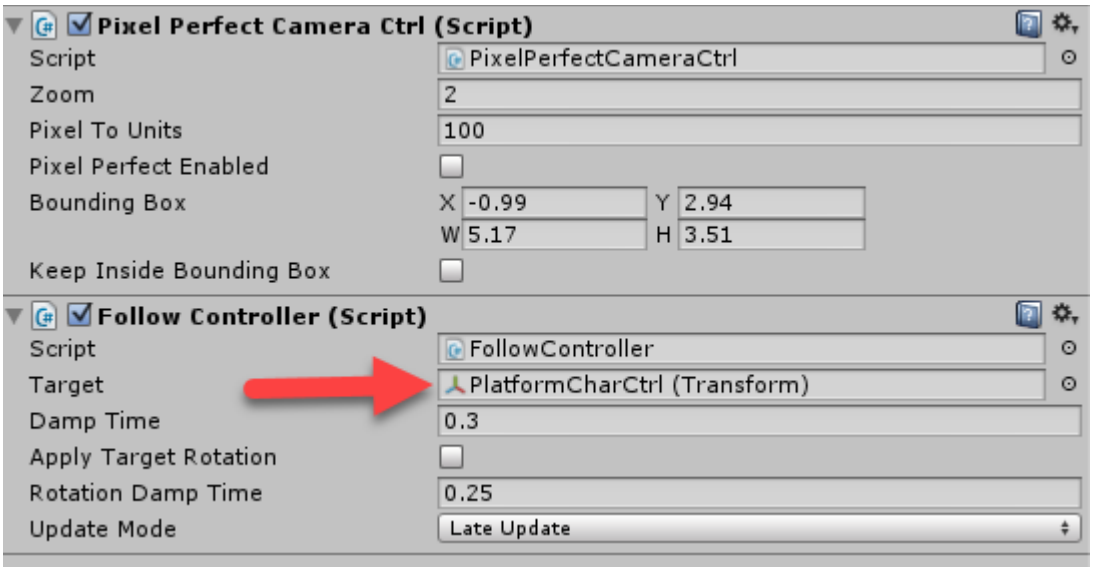
First of all, you need to setup the project to create some extra layers and configure the physics.

Go to the menu "**Smart2DColliders->Samples->Setup Project**".

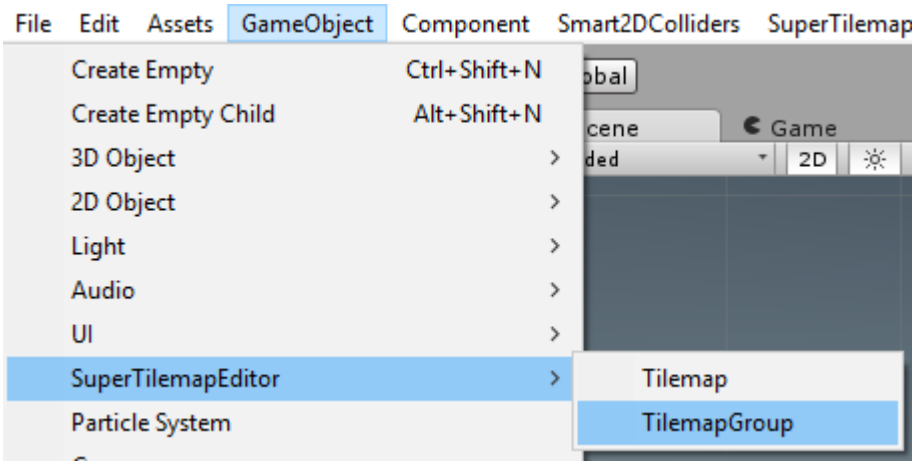
Now add to the scene a **PixelPerfectCamera** and a **PlatformCharacterController** prefab.



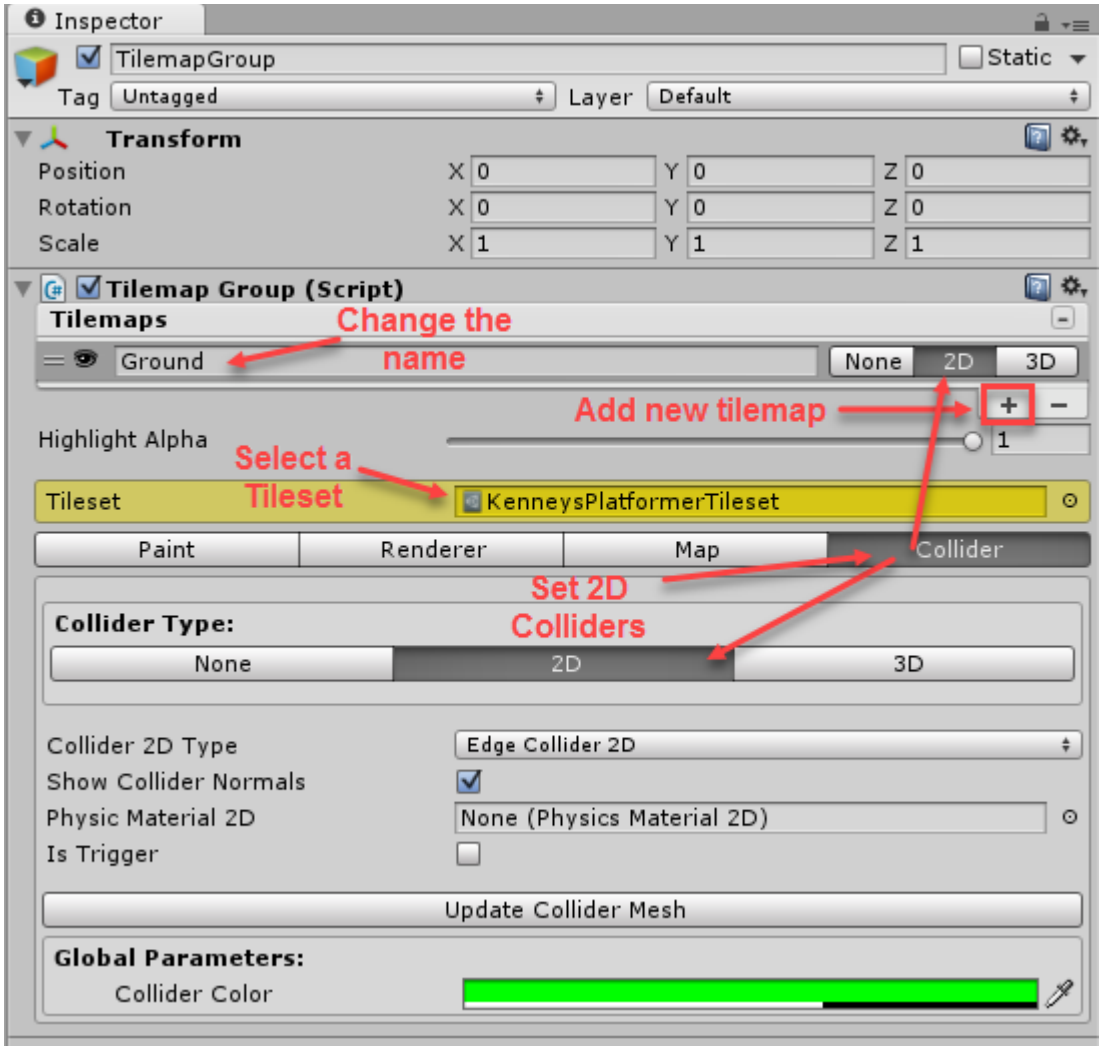
Attach the PlarformCharacterController to the FollowerController of the camera.



Now, create a new **TilemapGroup** to manage the tilemaps.



Add a new tilemap, called Ground, for example, and select a Tileset to start painting. Also set the colliders to 2D.



2.2 The Tileset

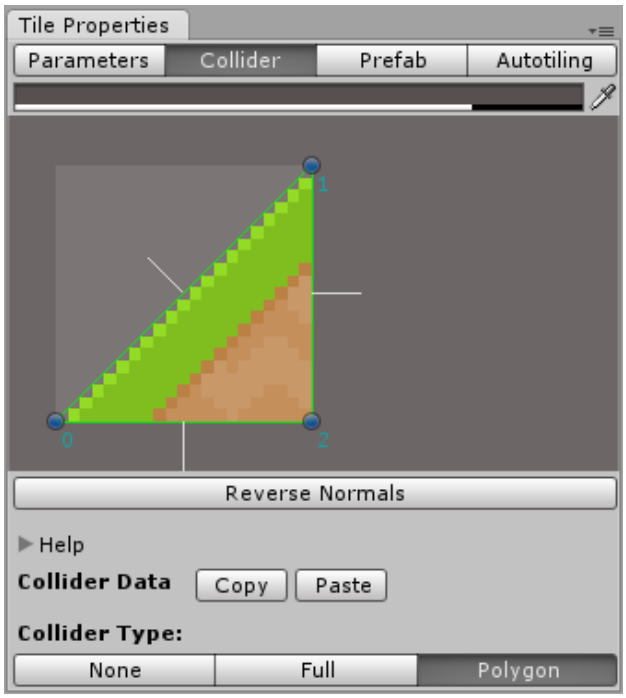
For this example I have used the **KenneysPlatformerTileset**. This tileset has been configured to use the Kenney's tileset for pixel platformer.

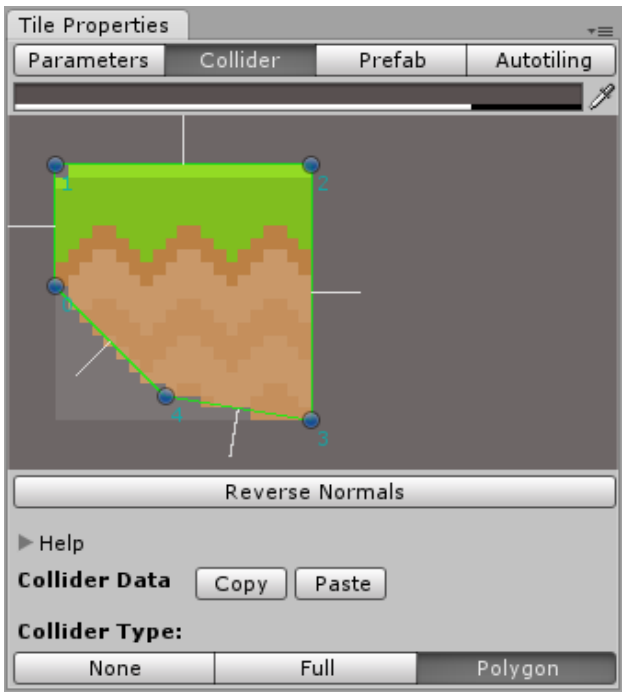
I am going to explain here all the steps necessary to setup the tileset properly, once the tileset has been created slicing the atlas texture.

2.2.1 Tile Colliders

To setup the colliders for a tile, you can open the tile property window, by right clicking over any tile in the palette (you need to select a tilemap or tilemap group and select the Pain tab to see the tile palette).

Here you can set the colliders for each tile. If you need more help setting up the colliders, check the Super Tilemap Manual.





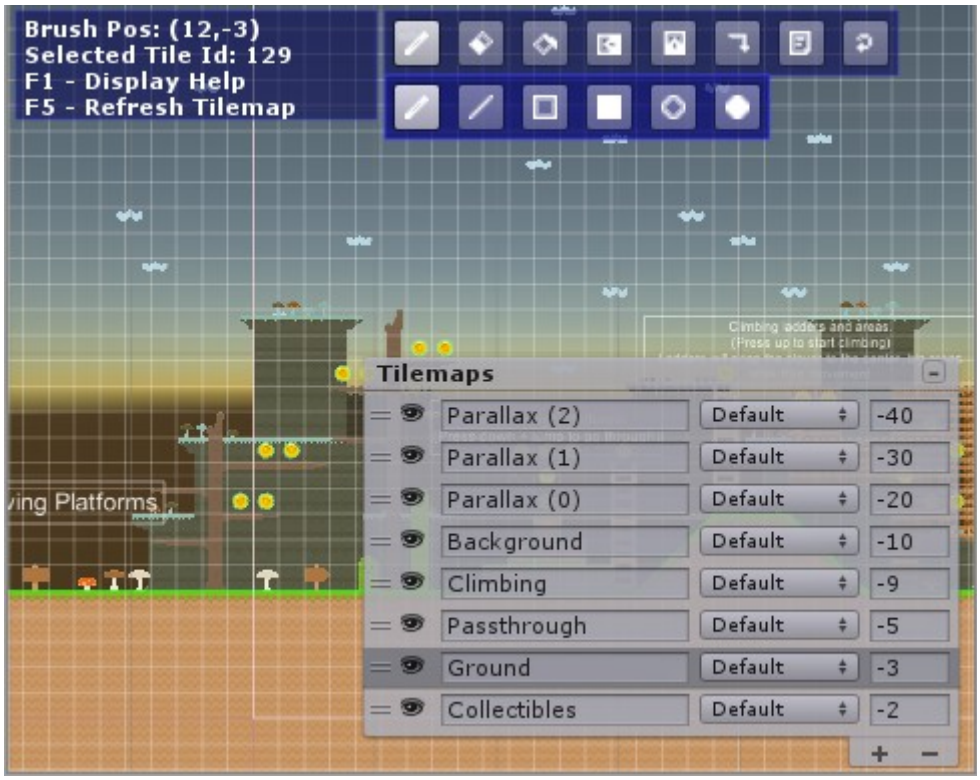
Once all the ground tiles have colliders, you can start painting a level.

You can now test the level with the player using the keyboard (Arrows to move and Space to jump) or the XBOX gamepad.

2.2.2 Tilemaps

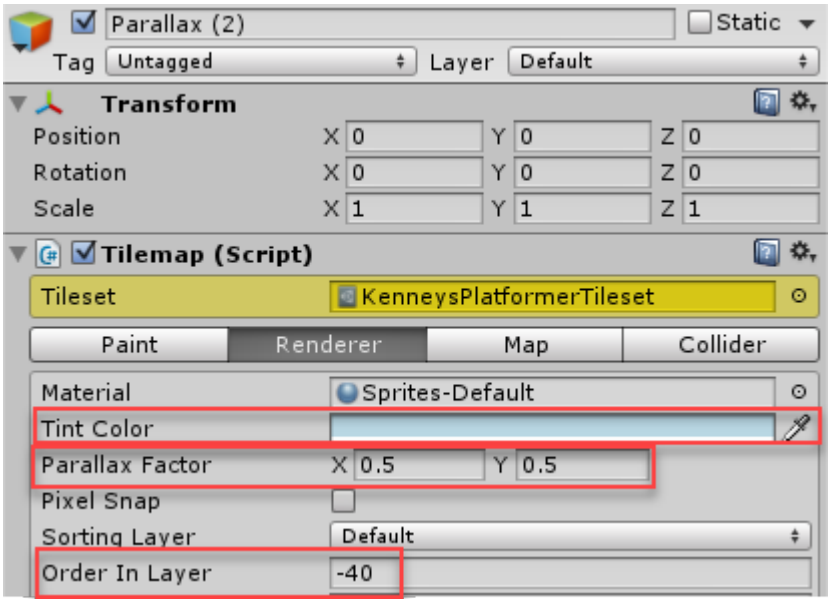
If you check the demo scene **Demo Super Platformer Editor - 01**, and you select the gameobject TILEMAPS containing the TilemapGroup, you

will see there are a lot of tilemaps in this level.



Separating the level in several tilemaps, you can create different elements with different properties.

For example, the **Parallax** tilemaps are used for the background, using a smaller parallax factor to make it looks like far elements. Also, the Tint Color has been changed to make them looks darker and increase the distance effect.

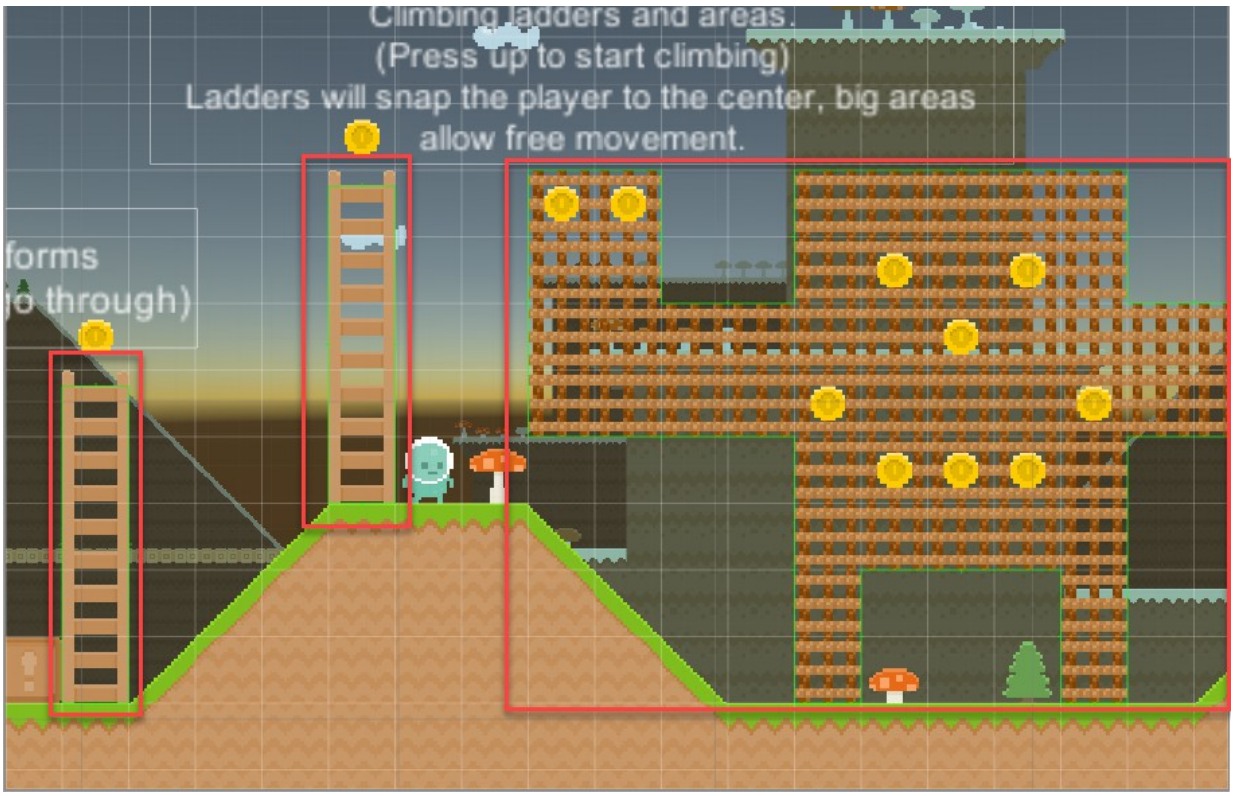


All the layers has been placed in the same sorting Layer (Default), but the sorting order has been changed to render them in the right order.

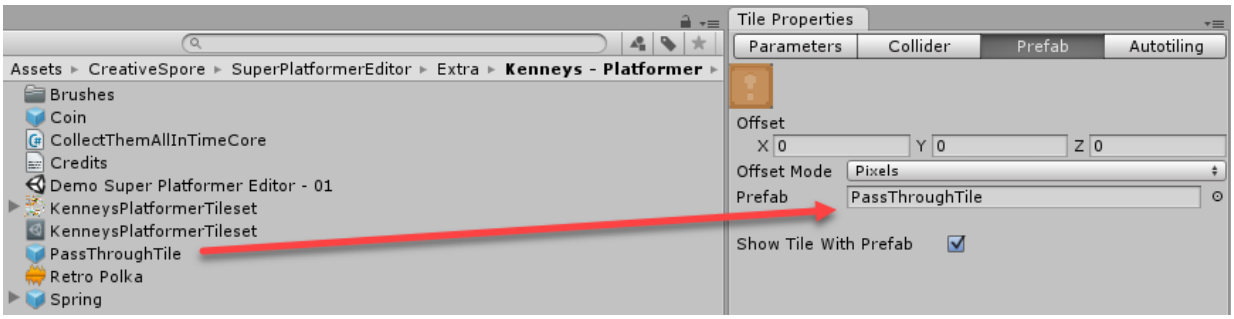
The player will be in the sorting layer Default - 0 by default.

The **Background** layer is similar to the Parallax tilemaps but it has a parallax factor of 1, so it moves at the same speed with the camera.

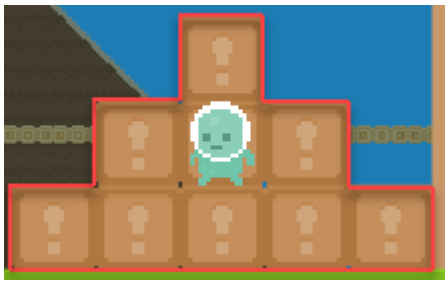
The **Climbing** tilemap is special, because the climbing colliders need to be in a special Layer called Climbing. This layer also have 2D colliders, and it's used to paint the ladders and climbing areas.



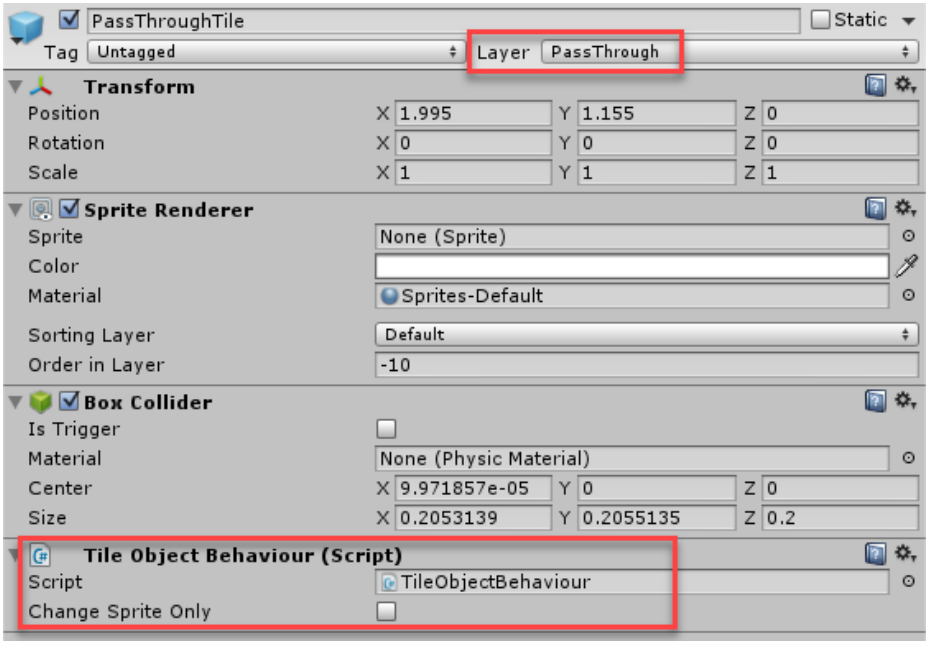
The **Passtrough** tilemap has a similar purpose, but in this case, for passthorugh colliders, only stopping the player when moving down. This is used with the boxes, but in this case, instead of adding a collider using the tile, a prefab has been attached to the box tile.



The reason of this is, if you use the tile to create the colliders, the colliders inside the boxes will be removed, only creating a collider line around all the boxes. And the player couldn't be over a box inside a pile of boxes like here.



The prefab used to draw the boxes can be used with any tile, because it will replicate the tile used to paint it thanks to the component **TileObjectbehaviour** attached to it.



As you can see, you only need to set the Layer to PassThrough, add a BoxCollider with the size of a tile and that's all.