Paper 2D

Steven Harris

TABLE OF CONTENTS

PAPER 2D

Paper 2D in Unreal Engine 4 are a suite of tools which allow you to create 2D or mixed 2D & 3D games, using 2D sprites.

1. CREATING A TILESET

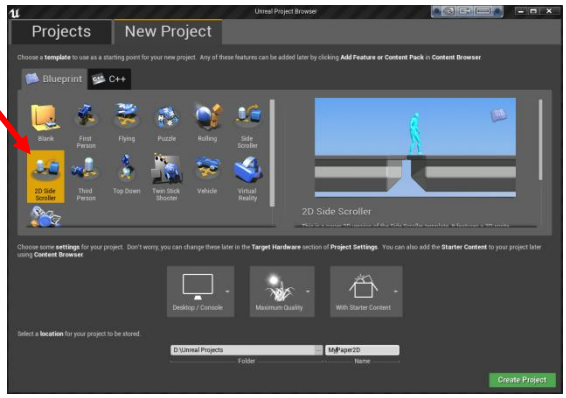
The Paper 2D system relies on 2D tilesets created in a separate art package like Photoshop. There are many free and commercial tilesets available. OpenGameArt.org provides a good source of tilesets, although the licence details for each tileset can vary.

NOTE: When creating a sprite sheet, all the individual sprites must be the same dimensions (32x32, 48x48 pixels etc). There should also be no gaps between individual sprites. These two points will be important later on. For the moment, we are using a pre-generated tileset which adheres to these rules.

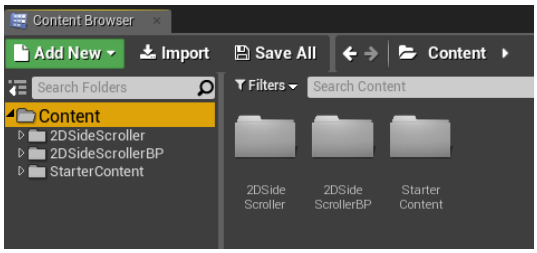
This tutorial will use the Generic Platformer: Tileset from author etqws3 https://goo.gl/XksY1m

1.1 SETING UP THE PROJECT

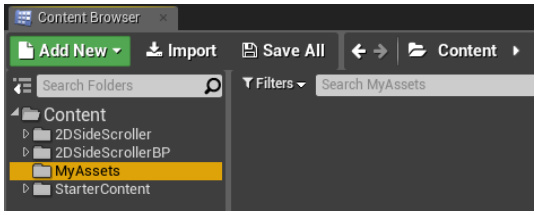
1. Open up Unreal Engine from the EPIC LAUNCHER. This tutorial was written with version 4.13.
2. In the UNREAL PROJECT BROWSER window which appears, select the NEW PROJECT tab and then the 2D SIDE SCROLLER template. Change the name and location of where the project is to be saved if required and click CREATE PROJECT.



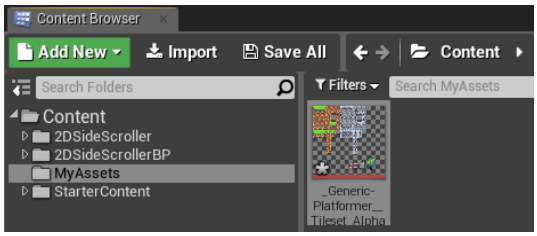
3. The template game will load and you can click the PLAY button to try out the template game. As you can see the basic mechanics of platforms, character movement and animation, as well as jumping are all set up. We will be adding our own background map into this level.
4. In the CONTENT BRWOSER window, make sure the folder view is visible. If it is not, click the folder button 



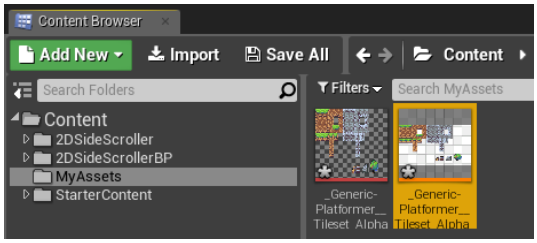
5. Right click on the CONTENT folder and select NEW FOLDER. Give it a name. This will be where we store our own assets for our game, as below.



6. Now open WINDOWS EXPLORER and locate your tileset. Drag the tileset file into your newly created folder in the CONTENT BROWSER. This will create a texture asset, as below.

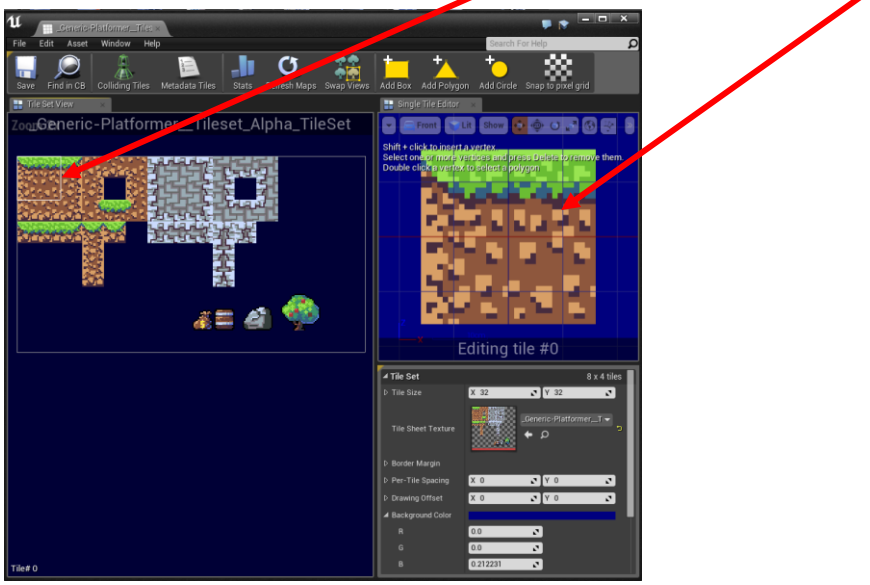


7. Now right click on the new texture and select SPRITE ACTION then APPLY PAPER2D TEXTURE SETTINGS. This applies settings which make the material more compatible with Paper 2D.
   7.1. One of these settings is to examine the colours in the PNG file to see which colour has been setup as the transparency (or alpha), effectively removing the background from the image. This will allow you to place once smaller graphics over another larger graphic so we can see the graphic underneath, called layering.
   7.2. This does assume the artist has already defined one of the colours in the file as transparent.
8. With the new texture selected, right click on it and select SPRITE ACTIONS then CREATE TILESET. You should see a new TILESET asset in the CONTENT BROWSER, as below.



1.2 CREATING SPRITES FROM A SPRITE SHEET

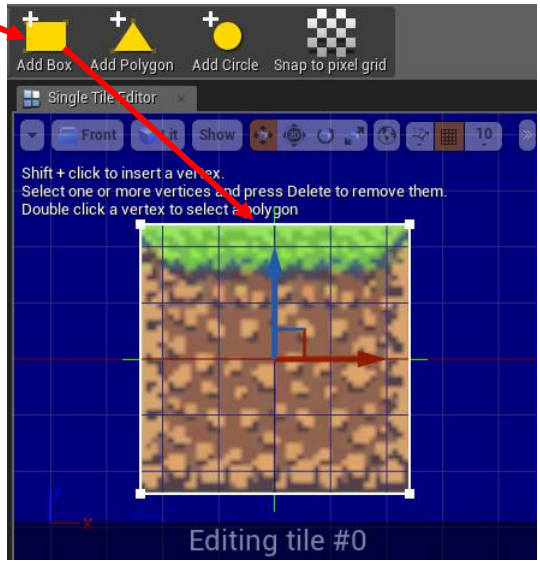
1. Now double click the new TILESET asset to open it. You will see the SPRITESHEET EDITOR window below. If you click once on the TILE SET VIEW tab, a white box appears. This is a single sprite. The SINGLE TILE EDITOR window shows you a magnified version of that tile.



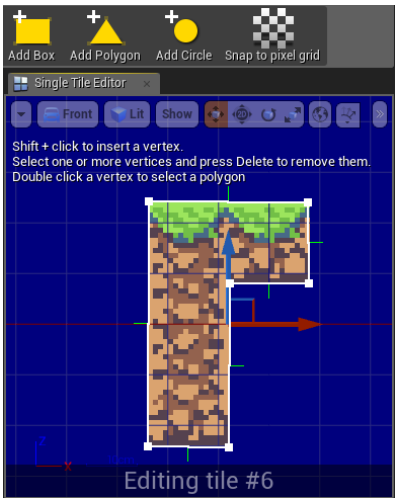
2. If the SINGLE TILE EDITOR window does not show the full time, as above, then move the mouse over the window and use the mouse scroll button to zoom in or out. This also works in the TILE SET VIEW. To pan in either viewport, right click and drag the mouse.
3. In the above screenshot, you can see that the white square in the TILE SET VIEW is not the same size as the tiles. This size is determined by the TILE SIZE setting in the TILE SET tab, which is currently 32x32 pixels. The tile size in the tileset we are using is 48x48. If we change these setting to this dimension the white box will match the tile size in this tileset. This will now define a 48x48 grid across the sprite sheet in which each adjacent 48x48 square has a separate sprite. You can see that this organisation takes some planning by the artist to make sure he/she is creating each sprite to the right size and in the correct location on the grid.

1.3 ADDING COLLISION TO THE SPRITES

1. To add collision detection around a sprite, select it in the TILE SET VIEW tab, so that it is displayed in the SINGLE TILE EDITOR tab, as below.
2. For square tiles, as above, you can add a BOX collision shape over the sprite. To do this, click the BOX icon in the toolbar. A WHITE collision box is placed around the sprite, see below. This will stop objects moving into this sprite in the game.



3. For more complicated shapes, as below, you can use the POLYGON tool to add a custom collision. Click the POLYGON button and click at each corner of the shape in a clockwise direction to place a vertex of the collision shape. The first three clicks create a triangle, with subsequent clicks adding extra sides to that triangle. Press RETURN to complete the polygon.

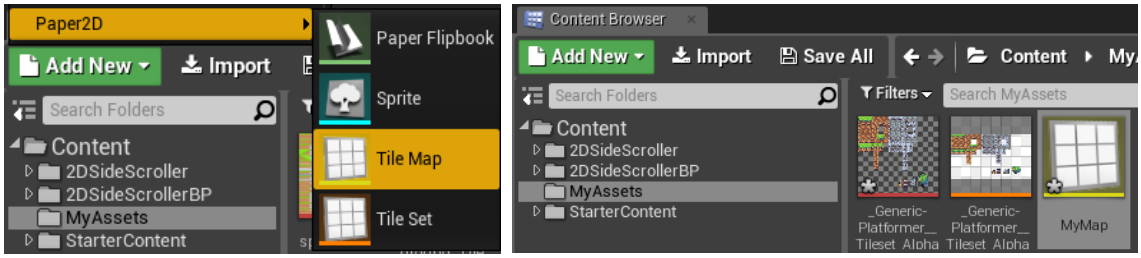


4. In some circumstances, the polygon edges may loop back on itself, which is to be avoided. If this happens, try to add the polygon vertices in the opposite direction.
5. Now add collision shapes for all the tiles in the tile sheet which require them.
6. The tileset is now setup and ready to be used to draw a map with.

2. CREATING A TILEMAP

To create map for our 2D side scrolling game, we need to create an assets to store the 2D level.

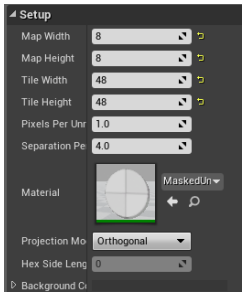
1. In the CONTENT BROWSER, click ADD NEW, then PAPER2D, then TILE MAP, as below. Give the TILE MAP asset a name, then double click it to open the tile map.



2. The TILE MAP EDITOR window will appear, as below.

3. The first thing to do is select a tile sheet to use to draw your level. To do this, click the yellow text at the bottom left of the screen PICK A TILE SET. This will open a window which allows you to select any of the tile sheets you have created in the CONTENT BROWSER. Select the tile sheet we created in the previous steps. This will load the tiles into the left pane of the TILE MAP EDITOR.
4. You can change this tileset anytime by selecting the grid icon next to ACTIVE TILE SHEET at the top left of the editor, as below.

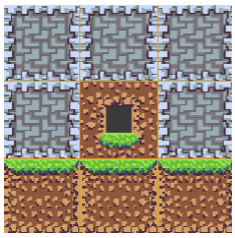
5. We now need to tell Unreal Engine how big the level is going to be. We do this by defining the number of horizontal tiles and the number of vertical tiles to define the length and height of the map. You can do this in the DETAILS TAB, in the SETUP section and change the MAP WIDTH and MAP HEIGHT values to an appropriate number for your level.
6. You should also tell Unreal Engine how big your individual tiles are. In our case, our tiles are 48 pixels wide and 48 pixels high. These values should be entered into the TILE WIDTH and TILE HEIGHT values, as below.



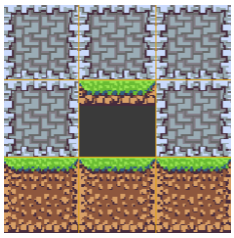
7. You can move around both the tile sheet and the map grid by using a right click and hold to pan across the tiles or map, and by using the middle mouse scroll button to zoom in and out.
8. A quick way to get started is to fill the entire map with one tile. To do this, select the tile on the tile sheet, click the FILL icon above and click in one tile on the map to fill the entire map.
9. You can now use the ERASE icon to remove any tiles in the map. The PAINT icon will allow you to draw the currently selected tile onto the map. Click and drag to paint several tiles continuously.

2.1 LAYERS

10. Some tiles, for example platforms, are designed to be placed over other tiles. IF you attempt to place a platform tile onto a background tile, the platform tile will remove the background tile and replace it. See the 'before and 'after' example below.

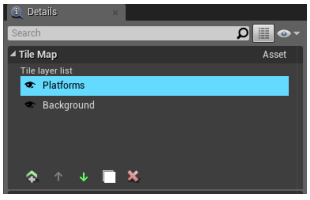


Before

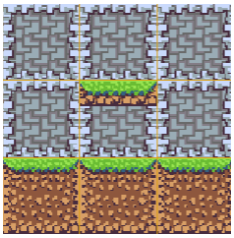


After

11. To avoid this you can create a new layer onto which you draw new tiles. Layers work like sheets of acetate placed on top of each other. So that tiles on the top layer are drawn over tiles on the layer underneath, so that lower tiles are not erased.
12. You can see the list of layers in the DETAILS pane, as below.

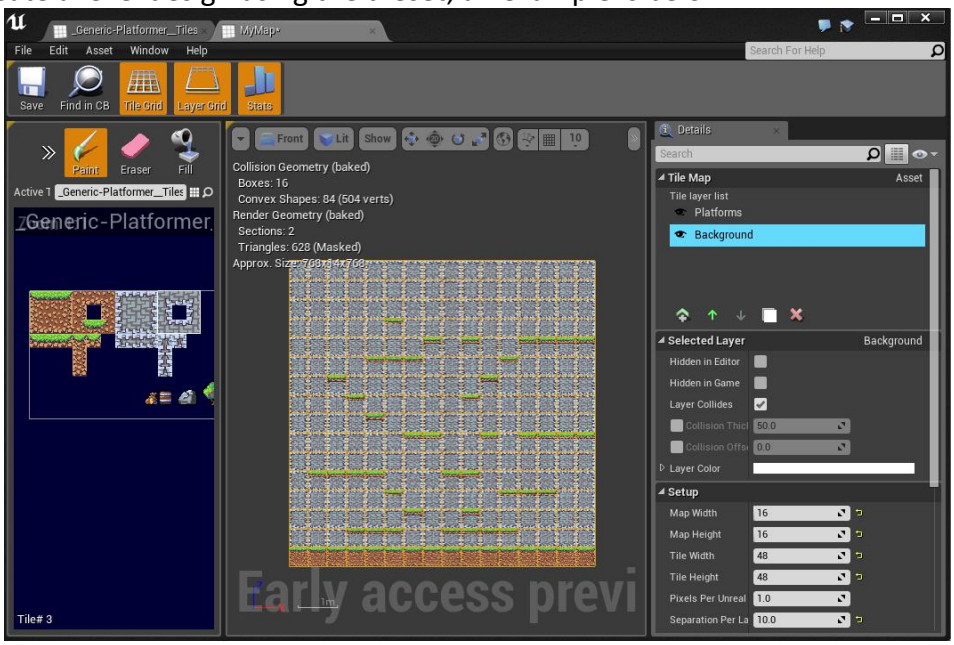


13. Click the ADD LAYER icon  to add a new layer. It is good practice to rename the layers so you do not get confused, and use each layer to draw a specific type of item into the level. In the example above you can see that there is a layer for background tiles and one for platforms. As long as the PLATFORM layer is above the BACKGROUND layer in the list, tiles in the PLATFORM layer will be drawn on top of the underlying BACKGROUND tile, as below.



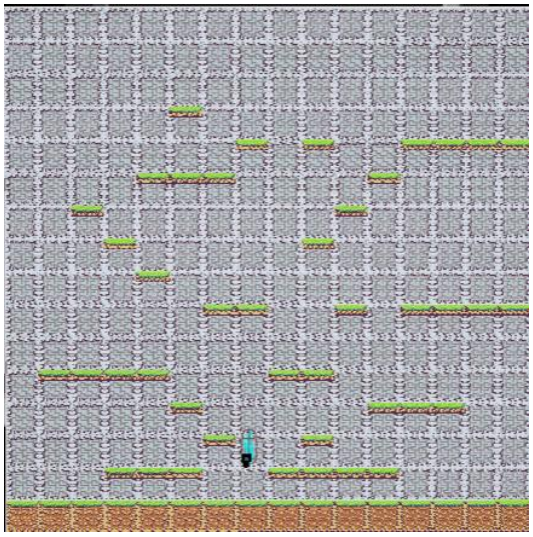
Using Layers

14. Double click a layer to rename it. Use the arrow icons to reorder the currently selected layer. The duplicate icon makes a copy of the currently selected layer and the delete icon deletes the currently selected layer.
15. Now create a level design using the tileset, an example is below.

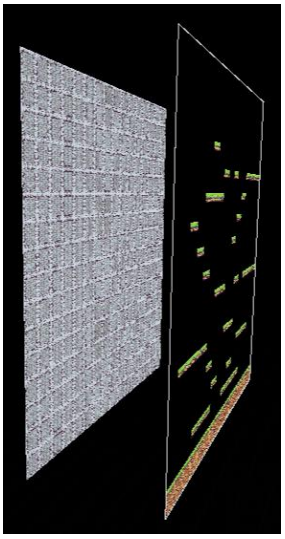


16. Now would be a good time to save your level.
17. Before we add the map into the editor, there is an important aspect you need to understand relating to depth. Although you are creating a 2D game, you are actually constructing it in a 3D environment. Each layer in you 2D map can will occupy a different depth in the 3D world. The distance between each layer is controlled by the SEPERATION PER LAYER option in the map editor window. In the above screenshot you can see that this value is set to 10. This means that there will be 10 units of empty space between the BACKGROUND and PLATFORMS layers. With the BACKGROUND layer being 10 units behind the PLATFORMS layer. Why is this important?
18. It is possible to have collision detection work backwards into other layers. This setting is the COLLISION THICKNESS value. You will need to do some basic arithmetic here. For example, in the above screenshot, there are 10 units between the layers, but collision thickness is set to 5 units. That is not enough depth for an object in one layer to reach the layer behind it so the object will only collide with objects in its own layer (assuming those objects have had collision setup in the tileset). If the collision thickness is set to 10, then this is enough thickness for objects in one layer to reach the other layer. Objects in one layer can then collide with objects in layer in front and behind.

19. As an example, the screen shots below are from a map where the COLLISION THICKNESS is set to 1000 so you can see the physical separation of the BACKGROUND and PLATFORMS layers.

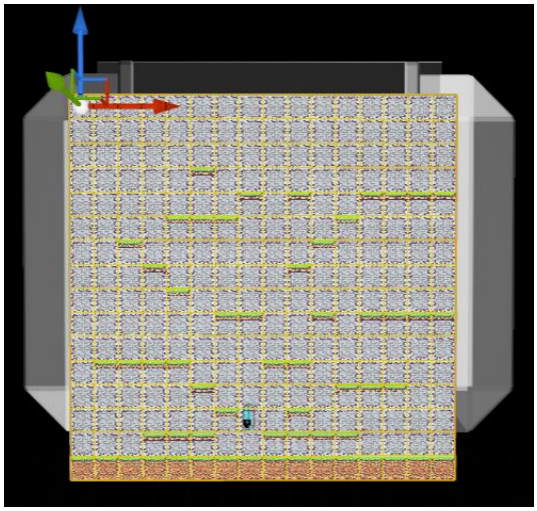


Front View

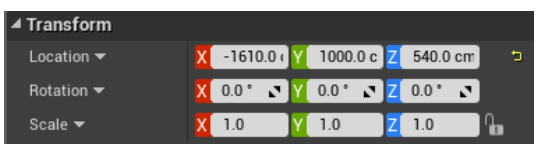


Side View with
collision thickness = 1000

20. For simplicity and speed, we will place our new level in front of the existing template level.
21. Drag the 2D map into the editor. And place it in front of the template level. Rescale it if necessary to fill over the old level.

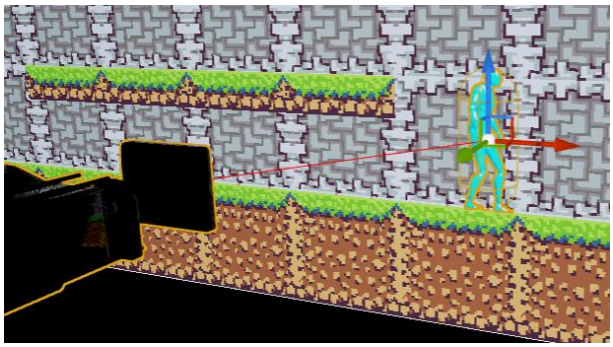


22. With it still selected, change the Y location value, in the TRANSFORM box, to 5000. The X and Z values are not important. This will set the BACKGROUND layer at depth 5000 and the PLATFORMS layer to 5010, as they are automatically separated out to a distance of 10 units, as above.



23. In the WORLD OUTLINER, single click the 2DSIDESCROLLERCHARACTER. And change the Y value in the LOCATION box to 5010. This sets the player character to the same depth as the PLATFORMS level. Now position the player character on the lower ground level by moving it

on the X and Z axis only. To do this move the mouse into the small box aligned with the Z and X axis so that it turn yellow, as below, then click and drag. Do not change its position on the Y axis as you will push it out of alignment with the PLATFORMS level. If you do this, then change its Y value back to 5010 in the DETAILS pane.



24. This process has now loaded up the map into UE4. However there are many other issues to address in terms of platform collision, pickups etc.
25. You can now click the PLAY button 

. You may also need to rescale the player character for the map. You may need to click in the game window to activate keyboard into the game to play it.
26. If you find your character colliding with invisible objects, check that your COLLISION THICKNESS is not too large that collision is occurring with layers behind. Also check for stray objects which may be within this collision thickness value.