2.5D Tile Editor package for Unity

Ward Dehairs
`http://wardddev.com`

Abstract

The 2.5D Tile Editor is a package to be used in the Unity editor. It is a simple tool for creating 3D-tile based levels in a 2D plane.
The editor comes with a few premade tilesets and a the following features:

- Tiles can be placed one by one or in rectangular chunks
- Tilesets include backgrounds, and functionality for special tiles, such as staircases and ramps
- Tile levels can be scaled and rotated reliably, without losing functionality
- Tilesets can easily be edited or even swapped out, while carrying trough changes troughout the existing level
- Tilesets can consist of straightforward cubes or more advanced shapes

Contents

1 How does it work

The 2.5D Tile Editor works by having a template tile for each possible neighbour configuration. In other words, you have a tile that is used when there are no neighbours, one that is used when there is only an above neighbour, one if there is a neighbour at both sides etc. There are 4 neighbours that are taken into account which leads to $2^4 = 16$ different tiles. Expanded on top of this are background tiles, which do not take into account neighbours, as well as special tiles, which fill up a slot like a normal tile and can be used for special cases like ramps or staircases.
A template (prefab) of each tile is stored and managed in a "TileSet". The "TileEditor" copies tiles from the "TileSet" and places them in the correct position in space, parented under special holder objects under the "TileEditor".

2 Implementation details

In order to ensure efficient neighbour searching the "TileEditor" will build a hash map of tile coordinates of all existing tiles. This map is build as soon as the editor is selected and ensures high efficiency when placing or removing many tiles at once. The bottle neck for performance is always Unity's own rendering, which tends to limit practical sizes from $10,000$ to $100,000$ tiles.
In order to facilitate undo and redo, the editor will also keep track of changes made in an undo stack. This stack grows indefinitely and is reset when the tile editor loses focus in the Unity editor. Note that these delta objects are fairly light weight, allowing for millions of tile changes before any large amount of memory is needed.

3 Setting up an editor with a pre made tile set

After downloading the package and including it in your project, open the scene where you would like to use tiles. Follow these steps to get working with a pre made tile set.

Create a blank game object in the editor and attach the "TileEditor" script. Look at this script in the inspector and you will find it reporting its current status. Firstly, the editor needs 2 childs to act as parents for the copied tiles. Press the appropriate buttons in the inspector to add them automatically.

Now, locate a "TileSet" prefab somewhere from the "2.5dTileEditor" folder. Drag this prefab into your scene to instantiate it. Drag the new instance under the game object from earlier, so that it becomes a sibling of the tile parent game objects. Make sure that the tile set is deactivated if you don't want it to show up in the scene.

The editor should now be functional and ready to use.

4 Basic tile editing

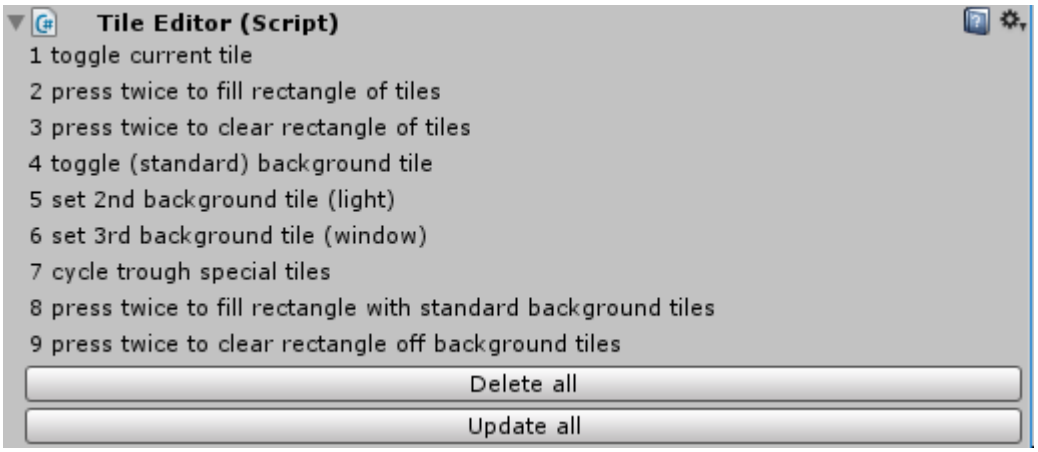

Figure 1: Normal inspector view of a Tile Editor script

When you have a TileEditor selected in Unity, its functionality automatically activates. You will see a yellow outline indicating the tile slot that is currently selected by your cursor. Move the cursor to select different slots. In the editor, you will see a list of hotkeys that can be used to edit the tiles you have selected, with the following functionality.

1. Adds a basic tile if the selected slot is empty. Removes it otherwise.
2. Allows you to select a rectangular region; the first hit selects one corner, the second hit selects the second corner and fills a rectangle between the 2 corners with basic tiles.
3. Works similarly to 2, but clears the selection of tiles in stead.
4. Add or remove a background tile. (Background tiles exist independently of basic tiles.)
5. Cycle trough different types of background tiles.
6. Remove background tile.
7. Insert a special tile in the current slot. Press repeatedly to cycle trough all types of special tiles. Special tiles are listed in the tile set. They usually contain ramps and staircase tiles among other things. Special tiles fill up a basic tile slot, forcing surrounding tiles to attempt to connect to it.
8. Works similarly as 2, but fills with basic background tiles in stead.
9. Same as 8, but clears backgrounds.

While the default hotkeys are just digits ranging from 1 to 9, you can set custom hotkeys in an included TileEditorSettings window. To open this window, go the your top menu bar in Unity and look trough Window > 2.5D Tile Editor settings. Note that the keys in the tile editor are really just typed characters, and not keyboard keys. As a result, caps lock and keyboard layout can affect the hot keys, since for example an upper case D is distinct from a lower case d.

4.1 Editor prefabs

From `Unity 2018.3` onwards editor scripts can no longer delete gameObjects under a prefab during normal scene editing. This blocks normal functioning of the tile editor severely, which is why one needs to be careful when making a prefab from a tile editor and its child tiles. If you must do this, note that you can no longer reliably edit the tiles directly. In stead, you must enter the prefab editing mode (by double clicking the prefab in question in the assets) or temporarily unpack the

prefab. If you do not do this proper functioning cannot be guaranteed, though the editor is now designed to log appropriate error messages.

5 Advanced tile editing

The Delete All function is self explanatory, but you might be wondering why there is an Update All button. This button will force changes made in the tile set to propagate trough to the actual tiles. For example, you might decide to swap out your tile set for a different one, only to notice that the level itself has not changed. Simply press Update All to fix this.

The 2.5D tile editor can consistently make use of rotations and scales, although this functionality is a little bit confusing. What follows is a detailed explanation.
First of all, the tile editor object itself can be rotated and scaled freely without losing its functionality. Tiles will be rotated and stretched accordingly and update as expected.
Parents of the "TileEditor" should always have default scale, since the editor will not take global scaling into account. The reason for this is that there is no simple way to account for non-axis-aligned scaling, therefore ignoring global scale altogether is the safest solution.
The scale and rotation of the "TileSet" object itself are ignored and can be changed freely.
The tile objects in a tile set can be scaled and rotated. These changes will propagate to the tiles used in the editor. Note that for proper functioning, the tiles should always be scaled and rotated into a $2 \times 1 \times 1$ unit format (Unity units, or meters, $x \times y \times z$). Say that you want to make use of $2 \times 2 \times 2$ tiles. In such a case, the tile objects under the tile set should be scaled to $1 \times .5 \times .5$. This scaling will reduce the tiles to the required size mentioned above. The tile editor object should then be scaled back to the appropriate size, meaning a scale of $1 \times 2 \times 2$. This will yield a tile editor that makes use of the original tiles, in a grid with cubic elements of $2 \times 2 \times 2$ units.

6 Connectors

New in the 2.5D Tile Editor is the option to add connector pieces. Connector pieces are special blocks that are added to normal tiles in order to ensure proper connections. This is sometimes necessary since the composition of a tile can depend on tiles diagonally across from them, while the regular tileset only depends on horizontal and vertical neighbours.
Connector pieces can be used in different layouts, although currently there is only the option to use "None" or "Inverted Corners". Inverted corners are used for hollow tiles that make up an environment, in other words the case where a single tile is a hollow block with a ceiling, 2 walls and a floor. See the "LaboEnvironmentTileSet" as an example.

Connector type	Number of connectors	order
None	0	-
InvertedCorners	4	TopLeft, TopRight, BottomLeft, BottomRight

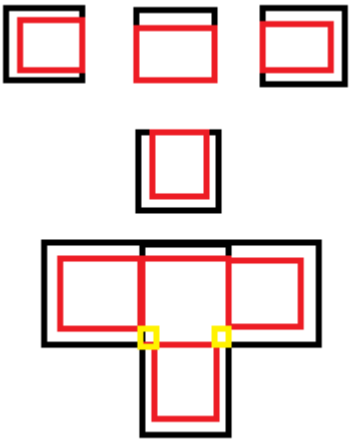

Figure 2: Diagram showing the use case of ”InvertedCorner” connectors.

7 Making a new tile set

If you want to make a new tile set with custom models, you need to build 16 different tiles, for each possible configuration. The tiles are usually arranged in a single 3D model file (.blend, .FBX) containing 16 meshes arranged from top left to bottom right in the following order (altough any order will work at this point):

HoverBlock	HoverPiece	HoverCorner_L	HoverCorner_R
FloorBlock	FloorPiece	FloorCorner_L	FloorCorner_R
WallBlock	WallPiece	WallCorner_L	WallCorner_R
CeilingBlock	CeilingPiece	CeilingCorner_L	CeilingCorner_R

”Hover” refers to there being no top or bottom neighbours, ”Floor” means bottom neighbour but no top, ”Ceiling” means top but no neighbour and ”Wall” has both a top and a bottom neighbour. Similarly, a ”Block” is free horizontally, a ”Piece” is sandwiched horizontally, and the 2 corner types have one neighbour left or right.
As a last example, a ”FloorCorner_L” has a neighbour below and to the right, but has no neighbours above and to the left.

You can follow a template layed out by the ”TileSetTemplate” object included in the package.

When modelling your tile set, you need to take good care that tiles that are meant to fit together will actually fit together. Keeping to the $2 \times 1 \times 1$ rectangular format makes this easy, but you can be a bit creative here. Note that you should prevent overlaps by sticking to the following diagram.

It’s a bit tricky to get the hang of making interlocking tiles, especially when it comes to textures, so take your time to practice and test the sets you make.

Background tiles are fairly straightforward, they usually include a ”Background”, ”LightBackground” and ”WindowBackground”. Backgrounds can be added indefinitely to a tile set, but do note that you have to manually cycle to the background you want in the editor. If you need to have large pieces of distinct backgrounds, like say different kinds of wallpaper, it may be better to use multiple tilesets, with one background each.

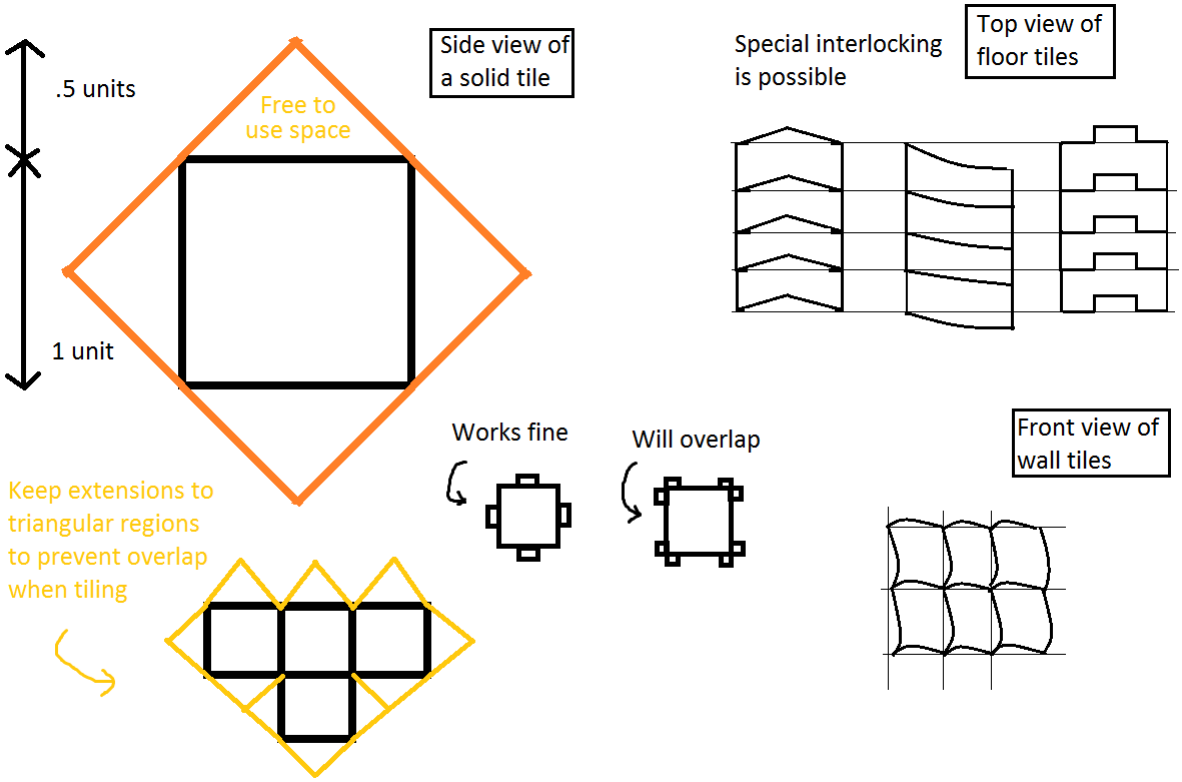

Figure 3: Diagram giving some tips on making custom tile sets.

Special tiles, such as ramps and staircases can also be added to a tile set indefinitely. Special tiles should connect and interlock to surrounding tiles if they are expected to be placed next to other tiles. If a special tile is supposed to connect to its environment in a specific way, the user is responsible for correctly setting up the environment of that tile.

Once a tile set is complete, drag it into Unity and add a "TileSet" script to the parent holding all the tiles. If you have followed the naming convention above, you can assign all 16 basic tiles and 3 background tiles at the click of a button, leaving you to assign any special tiles and select the correct corner configuration with its GameObject references.

8 Expansions and new use cases

If you have other suggestions or special use cases you feel should be included in the standard package, make sure to contact the developer at `info@WARddDev.com`.