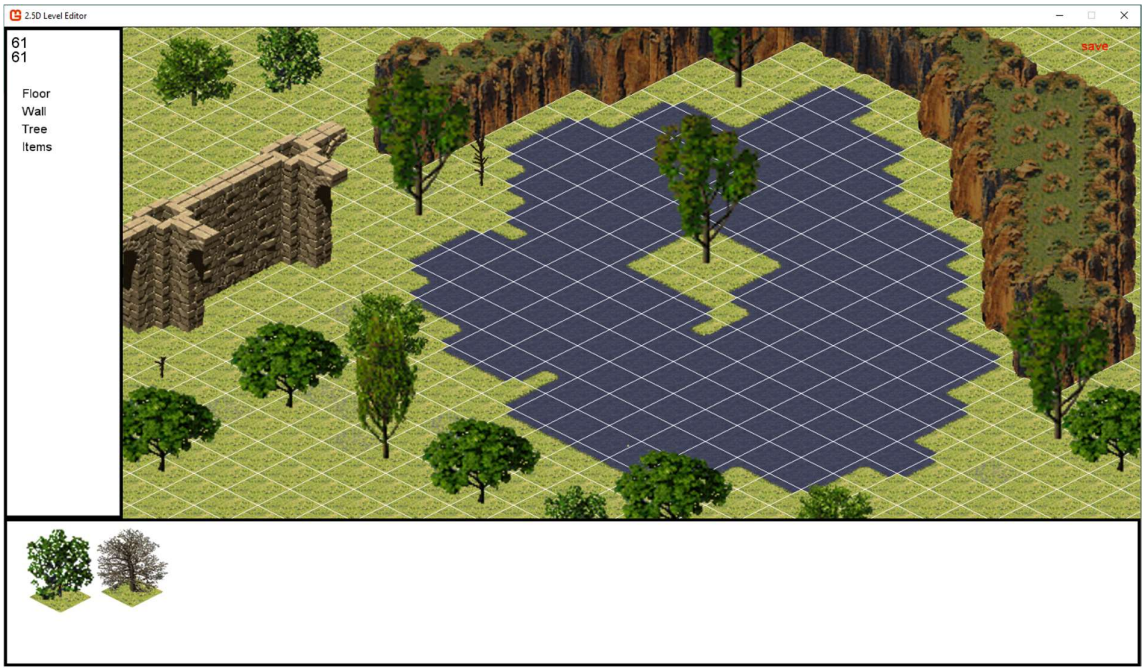
Valkan

Lourens Verhage
6762115

Level Editor



The level editor will be a essential part of the game. It will give the ability to create levels to be played in the game.

Technical design

1. Mouse

One of the most essential parts of the level editor is the mouse. The mouse is the interface trough which the level editor is used. The mouse also serves as the object to change tiles in the grid. The mouse will save the current selected object and give this information to the grid when a tile is selected. The grid will than change the tile to the new correct tile. When the mouse is pressed in the menu the save data from the current selected object is changed to the object that has been selected in the menu

2. Auto-tile

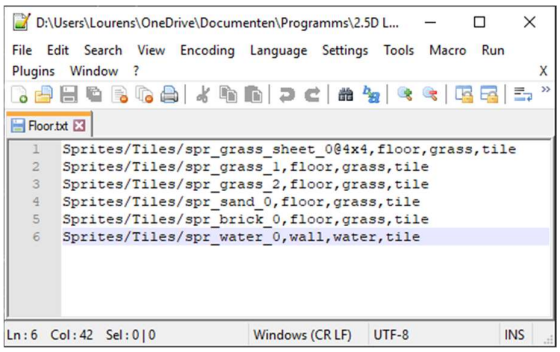
128	1	2
64	Tile	4
32	16	8

To make sure that the correct tiles are drawn when next to each other, like walls that need to connect, a auto-tiling algorithm is used. The auto-tiling algorithm makes use of binary calculations to give every possible combination a unique number. The algorithm checks every tile around itself and gives it a number. When the tile meets the desired parameters of the algorithm the number is added. When all the tiles around the tile are checked the numbers are added together and that gives the combination it's unique number. Based on that number the correct sprite is chosen from the spritesheet. The spritesheet will have a sprite for every possible tile, when multiple combinations use the same tile this will be handled in the program.

3. Multiple layers

The level editor will consist of multiple layers (currently there is only one layer). This will give the possibility to add multiple objects on top of each other, for instance items on top of the ground. In the save data there will also be more grids, one grid for every layer.

4. Menu

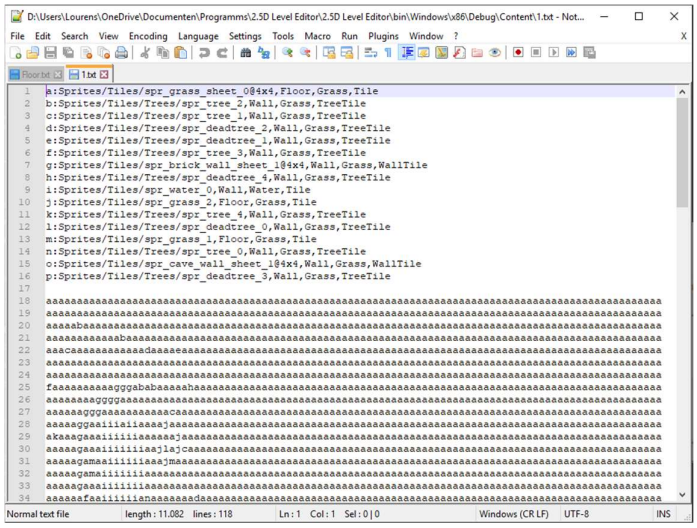


In the menu can be chosen which tile will be selected. When a tile is selected in the menu the data in the mouse is changed to the selected tile. The menu is filled up with tiles. To avoid having to code every individual tile the tiles are read from a .txt file. The program reads out the file and will make a different button for every line. This makes it very easy to add new tiles. An example of a tile would be:

Sprites/Tiles/spr_grass_1,floor,grass,tile

The line is read out in its entirety and then split across the ','. The first part says which sprite the tile should use, and where it is located, Sprites/Tiles/spr_grass_1. The second part says which tiletype it is, floor. This is used in the auto-tiling algorithm for the walls. The third part says which texturetype the tile is, grass. This is used in the auto-tiling algorithm for the floor tiles. The last part says which object should be created, tile.
All the information is saved in the button, and is given to the mouse when the mouse presses the button. The mouse then gives it to the grid when it presses a tile on the grid. The grid then changes that tile according to the data.

5. Save/Loading levels



The level is currently saved in a .txt file. To avoid having to code every tile and how it should be saved the level editor checks if it has seen the tile before and if not gives it a unique character to save it with. The meaning of the characters is saved in the top of the level file, underneath the character data is the actual level grid. When a level is loaded the program will first read the character data and will then kwon what the characters in the grid are supposed to be. This makes it easier to add tiles, because new tiles do not have to be given a unique character to identify them. Instead the program does everything on its own. The character data is almost the same as the data for the buttons and the mouse, the only difference is the inclusion of a unique identifier character before the data.

j:Sprites/Tiles/spr_grass_2,Floor,Grass,Tile

currently there is only one grid in the level file, but in the future there will be more grids that will be stacked on top of eachother.

Improvements/Future development

1. Auto-tile

Currently the auto-tiling algorithm is primarily used on walls to ensure they line up and have the right sprite. The auto-tiling is also used on the grass tiles to blend over in to the water tiles. Being able to have the grass blend over in more different tiles would be a nice addition to the level editor.

2. Saving

Currently the level is saved in a specific file in the bin folder. When a new level is created and saved the old level is overridden and is lost. Being able to save multiple levels is a essential upgrade to the level editor. Being able to choose the save directory would also be an nice upgrade to the level editor but would not be essential. Being able to chose the level name would also be a nice addition to the level editor.

3. Creating new levels

Currently the new levels that are created are always of the same size and have the same beginning tile. It would be a nice addition to be able to choose the size of the field and also the beginning tile when a new level is created.

4. Multiple layers

Like mentioned earlier it is essential for the level editor to have multiple layers to be able to have multiple objects on top of each other.