Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

QT Based 2D Tile Map Level Editor

Eoin Abbey-Maher

Abstract

Levels are the environment within which the game takes place and is played, they are used to set a scene, create an atmosphere for the players and guide them to their goal.

The 2D tile-based level editor is a standalone cross-platform piece of software which runs on both Linux and Windows, the software allows developers to quickly and graphically construct a tile-based level for their game rather than having to hand place every object using code.

The hope is that this project will tackle long development times for creating new levels by allowing developers to quickly see how their levels look and feel, the level editor will also allow artists and members of the design theme to quickly put together a mock-up of the level without having any code or development experience.

The level editor could also be packaged as a program within a game that allows players to build their own custom levels and play them or share them with friends and other members of the game community.

Project Description

The finalised project was completed with many of the specification and design details that were laid out at the beginning of the project. It featured many of the initial Ideas and was able to be successfully demonstrated with the user being able to design and export their map from my application and import it into an SFML project.

Tile Editor Features

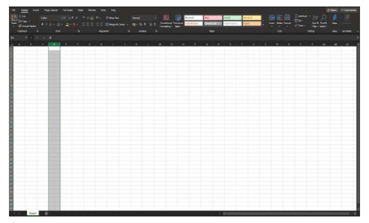
- Cross Platform on Windows 10 and Linux Ubuntu 18.04.4 LTS
- Grid-Based Layout to place your tiles and nodes.
- Player and NPC positioning nodes for easy layout of The games player and enemies.
- User-Generated Map Sizes.
- The ability for users to import their own images to use as Tiles.
- Ability to Export User map to an XML file for easy import into the users SFML project.
- Ability to Import pre-existing maps to be edited and changed.

Literature Review

For my Literature review I wanted to familiarise myself with numerous concepts that would be used as part of my project development. As part of the literature review, I hoped to learn about cross-platform development and issues and advantages of cross-platform programs. I hoped to learn about the QT Development and Design Environment which I am using for the main development of my program and look at other cross-platform development tools and what they have to offer to developers. I also hope to learn about the important features to consider when creating a level for a video game and the history of level editors both standalone and within video games.

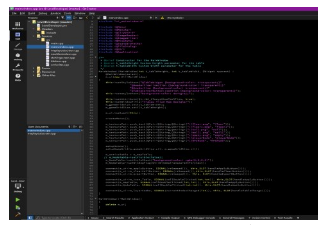
Background Research

For my background research I looked at pre-existing tile editors and other programs that work with a grid layout, I used these programs to try and narrow down an idea of what functionality and features my own program should have.


Microsoft Excel Layout

I used Microsoft Excel to see how I could consider doing multiple and single-cell selection and create a way to select what colour and texture that cell can be assigned. I also researched a pre-existing tile editor known as "Tiled", whilst I knew it wouldn't be possible for me to recreate this editor in the 7 months we had to do this project it gave me great inspiration and an idea of the shape I'd like my tile editor to take.

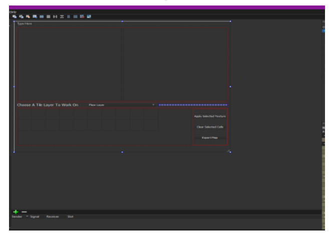
The Qt Development Environment


Qt Development Environment | Code

"Qt Creator is a cross-platform integrated development environment (IDE) built for the maximum developer experience. Qt Creator runs on Windows, Linux, and macOS desktop operating systems, and allows developers to create applications across desktop, mobile, and embedded platforms. "

The Qt Environment uses signals and slots for event tracking rather than an event loop , these signals and slots are used for objects to communicate with each other and to register user interactions on the Application Window.

The Qt Environment offers the ability to both work directly with the code and also using a GUI style editor to layout your application in a quick and easy way.


Qt Development Environment | GUI

Testing and Results

For my testing and study section of the project, I wanted to test if a level editor really did make it faster to build a level rather than hand placing everything by code.

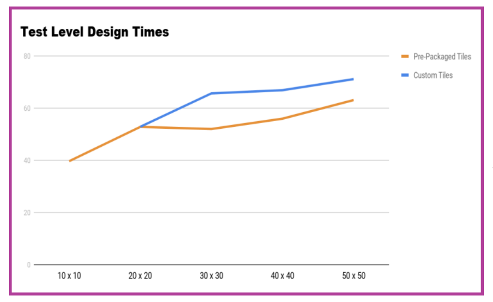
To test out my level editor I wanted to test the amount of time it took me to build a varied amount of map sizes and move them from my level editor and into my SFML test game that I have test built.

To ensure that the times are fair I tried to keep each map the same design despite the difference in map sizes.

Each map featured the player position node and 3 NPC nodes.
I also wanted to test the difference in the time it took me to build the map using the tiles that come pre-packaged with the level editor versus loading in a single custom file and using that as the floor in the level. I have also included a screenshot of the 2 "10 x 10 maps" built with the pre-packaged tiles and the custom user tile.

It took approximately an extra 10 seconds to build the level when loading in a single extra tile, due to these tests being carried out by a human rather than the computer there is an extra margin of error that we have to take into account due to the tests been carried out by a human.

Map Size	Custom Tile	Time Taken
10 x 10	YES	45.36s
10 x 10	NO	39.61s
20 x 20	YES	52.85s
20 x 20	NO	52.82s
30 x 30	YES	1m 05s
30 x 30	NO	52s
40 x 40	YES	1m 06s
40 x 40	NO	55.96s
50 x 50	YES	1m 11s
50 x 50	NO	1m 03s



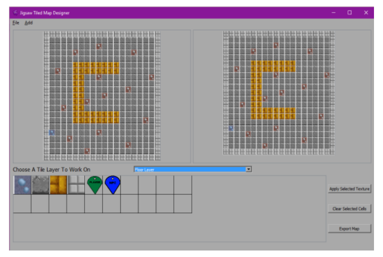
During testing I found that as the size of the grid got larger it the program was slightly slower to fill all the grid slots with the selected tile, I believe this may have occurred due to the whole table being filled through a single "For Loop".

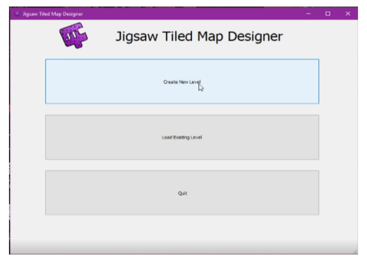
A way to tackle this problem and perhaps make it more efficient would be to separate the tables out into a quarter of the grid which would allow for the fill algorithm to only have to check a smaller amount of tiles and therefore take less time to assign the tiles to them.

Finished Project and Conclusions

I feel that in the end a large amount of the project went how I had intended and hoped for when I started. At the beginning of the project, I ran into many problems whilst setting up my development environments and trying to learn how all the new IDE's and software I was using worked. Due to my inexperience in using the Qt development environment, I restarted my project three times in the early stages as I felt that the foundation I had built would make things more difficult to advance on the further I got into the Project.

I feel that starting out I would have spent more time to study the Qt IDE and planning how each part worked before starting on it, rather than learning how to do each part on the fly as I was doing it.





I found that having completed the level editor it made the level design much faster and much more efficient rather than trying to hand place all of the tiles along with the NPCs and the player.

I feel that I had many personal and technical achievements and learnings whilst working on this project. The personal learning in this is that I was willing to keep myself motivated when the Qt environment wasn't working or when I was having issues with learn-

References

Referenced Publication	Citation	Reference
Website		Techopedia. Cross-Platform Development [Online] (https://www.techopedia.com/definition/30026/cross-platform-
Website		Rishabh Software. (29th January 2019) Pros And Cons Of Cross-Platform Mobile Application Development[Online] (https://www.rishabhsoft.com/blog/pros-cons-cross-platform-mobile-app-development). (Accessed 5th December 2019).
Book	(David and Russel 2009)	David P. and Russel D. (2009). David Perry on Game Design: A Brainstorming Toolbox. Boston. Course Technology CENGAGE Learning.
Website		Qt, Development Tools [online] (https://www.qt.io/development-tools). (Accessed