

Qt 2D Tile Based Level Editor

Technical Design Document

**Year IV**

**Eoin Abbey-Maher**

**C00217717**

# 

Open-Book and Remote Assessment Cover Page

**Student Name:** Eoin Abbey-Maher

**Student Number:** C00217717

**Lecturer Name:** Mr Philip Bourke

**Module:** 4th Year Project

**Stage/Year:** 4th Year

**Date:** 28/04/2020

**Declaration**

**This examination/assessment will be submitted using Turnitin as the online submission tool. By submitting my examination/assessment to Turnitin, I am declaring that this examination/assessment is my own work. I understand that I may be required to orally defend any of my answers, to the lecturer, at a given time after the examination/assessment has been completed, as outlined in the student regulations.**

# I Table of Contents

[**I Table of Contents**](#_lsebbqnkr12m) **2**

[**1. Software Architecture**](#_i0emjial2clo) **4**

[Qt System Layout](#_v9fwt8ibgg8b) 4

[SFML System Layout](#_wguvlabpp294) 5

[**2. Features**](#_smy95sk75zgq) **6**

[2.1. Cross-Platform](#_8c25tbc47s8k) 6

[2.2. Cell-Based Selection](#_w1ppu7tyf1an) 6

[2.3. Custom User Loaded Tiles](#_xk1pifov28rc) 6

[2.4. Player and NPC Nodes](#_nh619lpcasgw) 6

[2.5. Export Map](#_pp3wmv84dwqu) 6

[2.6. Import Map](#_fleqanqih6xn) 7

[2.7. Rebuild Map Using SFML](#_pdeo3ftu585r) 7

[**3. CRC Cards**](#_oogbb9kticaa) **8**

[Qt Files](#_nd3lxesfaj3j) 8

[SFML Files](#_9xifi1n6d2h4) 10

[**4. References & Bibliography**](#_5cueun5b0exm) **12**

# 

# 1. Software Architecture

## Qt System Layout

## 

## 

## 

## SFML System Layout

# 

# 2. Features

## 2.1. Cross-Platform

* Install Qt and MinGW on Windows
* Set up a virtual machine
* Install a Linux image on the virtual machine
* Install Qt and MinGW on Linux
* Code full application on Windows and save to GitHub repo
* Build release Build on Windows
* Switch to Linux on Virtual Machine
* Set project properties and run the application again on Qt on Linux
* Use LinuxDeployerqt to make a Linux release build

## 2.2. Cell-Based Selection

* Create a table widget in Qt that has the dimensions of the user’s map
* Create a second table widget and populate it with each of the possible tile textures.
* When the player selects a cell/s on the map grid, fill the selected cells with the data of the selected tile on the Tile icon table

## 2.3. Custom User Loaded Tiles

* Create a “new tile” window
* Set up a folder selection Button that opens a file dialogue
* Create a text box for users to set the new tile name
* Create a dropdown menu to let users select tile type: Wall or Floor
* Create a confirm and a cancel button
* When the confirm button is pressed, append the new tile to the end of the icon table, save the tile name to the tiles vector and save the tile type to the new tile.

## 

## 2.4. Player and NPC Nodes

* Create a second grid for placement of the nodes
* Add a player node and an NPC node to the icon table
* When the node grid is selected and a tile is being placed check the type of tile
* If the tile is not a node, Do nothing
* If the tile is a node Check the node type
* If the node is a Player node, go through the table and remove the existing Player node if there is one
* Add the new player node
* If the node is an NPC node, add it to the Node Table

## 2.5. Export Map

* Create an XML Writer class to be called when the player exports the level
* Create a folder named Map\_Export
* Have the XML writer create a new XML File
* Write the number of textures and name of each texture to the file
* Write the data of each tile from the tile table and node table to include Position, Index, Tile Type, Texture name
* Export each image to an images folder within the Map\_Export folder

## 

## 2.6. Import Map

* Create a function to read the XML File
* Allow users to select the location of the Map Folder that holds the images and XML file
* Build the grids using the row and column size from the XML document
* Load each of the images and assign it to a pair with the texture name
* Load the tiles into the tile table basing the texture off the texture name in the XML and the index from the index value in the XML file.
* Load in the nodes and set their position based on the index value form the XML

## 2.7. Rebuild Map Using SFML

* Build a Room builder class to load in the XML file and assign values to the tiles
* Make a tile class using SFML to hold the texture, size, position and type of the tiles
* Using TinyXML read in the XML File, set the positions of the tiles, the texture of the tile and the tile type based off the XML file.
* If there has been players and NPCs passed to the function, set the player’s position and the NPC’s positions based on data from the nodes in the XML file.

# 3. CRC Cards

## Qt Main Tile Editor Cards

|  |  |
| --- | --- |
| **Class Name:** mainWindow |  |
| **Superclasses:** QMainWindow |  |
| **Responsibilities** | **Collaborators** |
| Construct Game Grid and Node Grid | Table Height, Table Width |
| Handle Tile Selection Button |  |
| Handle Exporting Button | xmlWriter |
| Handle Key Presses | Key Event |
| Handle Mouse Wheel Events | Mouse Wheel Event |
| Add and Remove rows and columns from the game table |  |
| Launch New Tile Window | newTileWindow |
| Return to the main menu | startingScreen |

# 

# 

|  |  |
| --- | --- |
| **Class Name**: mapLayoutScreen |  |
| **Superclasses:** QMainWindow |  |
| **Responsibilities** | **Collaborators** |
| Set the row and column size of the new map | mainWindow |
| Cancel and return to the main menu | startingScreen |

# 

|  |  |
| --- | --- |
| **Class Name:** newTileWindow |  |
| **Superclasses**: QMainWindow |  |
| **Responsibilities** | **Collaborators** |
| Access the New Tile Location |  |
| Set the new tile name |  |
| Set the new tiles type |  |
| Create the new tile | texture names pair, icon table, |

# 

# 

|  |  |
| --- | --- |
| **Class Name:** startingScreen |  |
| **Superclasses:** QMainWindow |  |
| **Responsibilities** | **Collaborators** |
| Enter map size selection screen | mapLayoutScreen |
| Load in a pre-existing map | xmlwriter |

# 

# 

|  |  |
| --- | --- |
| **Class Name:** xmlwriter |  |
| **Superclasses**: QWidget |  |
| **Responsibilities** | **Collaborators** |
| Write files to XML | map table, node table, texture names pair |
| Open existing level from XML | mainWindow |

# 

|  |  |
| --- | --- |
| **Class Name**:tileItem |  |
| **Subclasses:** |  |
| **Superclasses:** QTableWidgetItem |  |
| **Responsibilities** | **Collaborators** |
| Construct the tile item | Index value, Texture image |
| Set The tile texture | Texture image |
| Set if the tile is collidable | True or false |
| Set the index of the tile | Index value |
| Get the tile image |  |

# 

## SFML Test Level Cards

# 

|  |  |
| --- | --- |
| **Class Name**: Tile |  |
| **Responsibilities** | **Collaborators** |
| Set the texture of the tiles | Texture image |
| Initialise the tile | Texture map with the name and image |
| Draw the tile | Render window |

## 

|  |  |
| --- | --- |
| **Class Name**: Room Builder |  |
| **Responsibilities** | **Collaborators** |
| Load in the room file | Player, NPC, Room Name |
| Draw the window |  |
| Build the room | Render Window |

# 

# 

|  |  |
| --- | --- |
| **Class Name**: Player |  |
| **Responsibilities** | **Collaborators** |
| Build The Player | Render Window |
| Update the player | Delta time, RoomBuilder |
| Move Player | acceleration |
| Check Boundaries | RoomBuilder |

# 

# 

|  |  |
| --- | --- |
| **Class Name**: Game |  |
| **Responsibilities** | **Collaborators** |
| Check for key and mouse presses | Event |
| Update Game | Delta time |
| Run the program |  |
| Render everything |  |

# 

# 

|  |  |
| --- | --- |
| **Class Name**: Player |  |
| **Responsibilities** | **Collaborators** |
| Build The Enemy | Texture, position |
| Update the player | RoomBuilder, Player |
| Draw the enemy | Render Window |
| Check Boundaries | RoomBuilder |

# 

# 

# 4. References & Bibliography

|  |  |
| --- | --- |
| Website | Umletino[online](<https://www.umletino.com/umletino.html>).(Accessed 2nd May). |