



University Of Plymouth
School of Engineering,
Computing, and Mathematics

A Co-Operation Level Editor

Osbourne Laud Abraham Clark

10777267

BSc (Hons) Computer Science

Plymouth University

United Kingdom

??th April 2025

Acknowledgements

Abstract

Extra

Word Count:

Code Link:

Contents

Acknowledgements	1
Abstract	1
Extra	1
Contents	2
List of Figures	4
1 introduction ;TODO LOOK AT SOFTWAREDESIGN DOC IN REPO;	5
1.1 Background	5
1.2 Objectives	5
1.3 Deliverables	5
2 Legal, Social, Ethical and Professional Issues	7
3 Method Of Approach	7
3.1 technologies	7
3.1.1 unity	7
3.1.2 C#	7
3.1.3 Libraries	7
3.1.4 Co-Operation game	7
3.1.5 GitHub	8
3.2 functional requirements	8
3.3 non-functional requirements	8
3.4 UML	8
3.4.1 classes	8
3.4.2 control flow	8
4 project managment	8
5 Preparation	8
5.1 previous attempts	9
5.1.1 1 st attempt	9
5.1.2 2 nd attempt	9
6 implementation	9
6.1 unity	9
6.2 YAML	9

6.2.1	importing	9
6.2.2	exporting	9
6.3	Reverse Engineering of Level Layout	9
6.3.1	object placement	9
6.3.2	object orientation	9
6.4	cacheing for faster loading	9
7	Evaluation	9
8	Conclusion	9
8.1	further Work	9

List of Figures

1 introduction ; TODO LOOK AT SOFTWARE DESIGN DOC IN REPO ;

1.1 Background

this dissertation idea came about through a problem identified in my second academic year while developing a mod for the game Co-Operation created by MINDFEAST. the issue found was the tedious nature of creating levels. once levels became big or contain a lot of objects they start to become a mess, and attempts to quickly change a single object can become a massive effort. there are multiple method used for finding and changing object positioning:

- counting rows and columns in the game
- crawling through the file until you have found the correct object
- make continuous mental logs of where things have been placed

even these with these methods, level designing and iteration becomes slow and tedious. my aim with this project is to create a visual, easy to use tool to be used along side Co-Operation making level creation more fluid.

1.2 Objectives

the objective of this project is to create an easy to use visual level editing tool for both experience and inexperienced. it will be made for individuals in the Co-Operation community wanting to create levels easier. It will be aimed at people who have little to no understanding in modding and/or YAML. more experienced modders can utilise this tool to help create levels as they wont need to manually edit YAML files. it will also open up the creativity of modding to those who wouldn't normal attempt to make one.

1.3 Deliverables

the deliverable will be a piece of software that can display a close to 1:1 level representation, while facilitating a live tile manipulation solution to be able to edit the level directly. the users will be able to export their created levels to their respective folders, as well as import existing level files so that older levels can also be easily modified with the tool. object manipulation

- use the file format GLB (this format is also used in Co-Operation)
- create objects in the scene

- moved object around the scene before and after placement
- delete object from the scene

scene manipulation

- scroll wheel
 - used for zooming in and out of the scene changing the panning speed
 - upper and lower bound to zooming extent
- middle mouse click
 - hold click used for panning around the scene

importing and exporting of level data

- exporting of levels
 - files export to the YAML format provided by MindFeast for compatibility with the Co-Operation game
- importing of levels
 - import YAML files into the editor in Co-Operations format for both old and new levels
 - object definitions can be properly read to correct placement into the scene

quality of life

- have the object highlight when mouse is hovering over it
- know which object you are currently interacting with
- know which object is the next for placement
- middle mouse click to select currently hovering over object for next placement(creates new object not modifies new object)
-

2 Legal, Social, Ethical and Professional Issues

3 Method Of Approach

3.1 technologies

3.1.1 unity

to create this project i utilised unity for its easy to use graphics manipulation methods. From the beginning it was clear a 3D solution was required for the project. If 2D was chosen it would not have given the same feedback methods or easy usability to the user. By going 3D i could give the user a closer idea to what the levels will look like in game.

3.1.2 C#

i was required to code within C# as this is unity primary coding language . this was beneficial as i am well versed in it and have used it before(both unity and C#).

3.1.3 Libraries

importing GLB objects

i have chosen to use UnityGLTF to import GLB object, which are compatible with Co-Operation, to unity at runtime. this allowed for hot reloading objects without the need of re compilation of the entire unity project.

<https://github.com/KhronosGroup/UnityGLTF>

parsing YAML

to be able to import and export file in YAML format, without having to create my own parser from scratch, i have used YamlDotNet. This is a library built for C# and so worked easily with unity. this made importing and export much easier as i didn't need to manually parse or worry about correct spacing.

<https://github.com/aaubry/YamlDotNet>

3.1.4 Co-Operation game

Through out the development the actual game Co-Operation was used to valid my progress. without the use of the game i could not

- make sure that the object placement in the tool match the ones in game

- know if the exported file was valid to be used with in the game

3.1.5 GitHub

GitHub has been used for version control and storage of the project files and code. currently only two branches have been made and have now been merged, with the later branch being made more towards the end of development. while version control wasn't heavily used, it was extensively used for transporting the code.

3.2 functional requirements

functional requirements include details such as object CRUD as well as map panning and the importing and exporting of levels. these are the core vital aspects for the application to work. even with the bare bones of these features the tool would have made level creation easier visually.

3.3 non-functional requirements

non-functional requirements where the quality of life feature like:

- changing of pan speed based on zoom distance
- the on screen hints as to the current state of selected and next placements
- quick middle mouse click to select object for next placement

3.4 UML

3.4.1 classes

3.4.2 control flow

4 project managment

5 Preparation

the preparation for this project to start and be complete began when i first saw the issue arise. while creating the mod the least interesting part was level creation but this (in my opinion) was due to the mental strain of designing and then translating the level to the YAML format. I only ever had vague ideas on how it would look in game. and later on came the task of modifying a cell in the middle of the map with no easy to point out reference point.

5.1 previous attempts

the previous attempts were hacks. they were quickly made in an attempt to make the editing at that point easier. only one works, which happens to be the 2nd attempt, due to my shortcuts used when programming win32.

5.1.1 1st attempt

the first attempt used windows win32.h to create and show information to the users, it was entirely text based only showing buttons and input boxes to the user.

5.1.2 2nd attempt

the second attempt actually took notes from the first attempt, making it so each object has its own layer and so only one object is focused on at a time.

6 implementation

6.1 unity

6.2 YAML

6.2.1 importing

6.2.2 exporting

6.3 Reverse Engineering of Level Layout

6.3.1 object placement

6.3.2 object orientation

6.4 cacheing for faster loading

7 Evaluation

• What went well and what went badly? • Why was this the case? • To what extent was the aspect under consideration responsible (vs. other contributing factors, e.g., your own performance). • Was your experience in line with what might have been expected given the body of knowledge within the literature? • To what extent does the above cause you to reconsider the choices that you made in relation to the given aspect?

8 Conclusion

It is a brief summary of the project and its achievements. Therefore, you should relist your project's objectives and critically (and ruthlessly) evaluate whether you met the objectives

8.1 further Work