STUDENT REPORT

NTNU

NAME: Henrik Kjernmoen Gran			
DATE: 05/01/2025	COURSE CODE: IDATT2505	COURSE NAME: Extended Reality (XR)	# OF PAGES: 7

LECTURERS:

Thomas Holt, Alexander Holt

TITLE:

Report for portfolio, IDATA2505 Extended Reality (XR)

SUMMARY:

This report is regarding the portfolio submitted for evaluation in the Extended Reality (XR) course. It goes over the background for developing this project, as well as some of the technology & theory behind it, and finally a description of the implementation, with some evaluation & reflection.

This report is a submission by student(s) at NTNU.

TABLE OF CONTENTS

1 Summary 1
2 Introduction 1
2.1 Background 1
2.2 Restrictions 1
3 Theory, technology & architecture 1
3.1 Game engine 1
3.2 Separate model 2
3.3 Making textures 2
4 Requirements 3
5 Implementation 4
5.1 Simplified class diagram 4
5.2 Installation 4
5.3 How to play 5
5.4 Video demo 5
6 Evaluation 5
7 Conclusion & experiences 5
8 References 6
9 Attachments 7
9.1 Time table 7
9.2 GitHub repository 7

Figure list

Figure 1 Simplified class diagram of the entire game (Project docs/new class diagram.png) 4
Figure 2 Timetable (time table.txt) 7

1 SUMMARY

This report is regarding the portfolio submitted for evaluation in the Extended Reality (XR) course. It goes over the background for developing this project, as well as some of the technology & theory behind it, and finally a description of the implementation, with some evaluation & reflection.

2 INTRODUCTION

2.1 Background

The main purpose of this project is to cover the curriculum of the Extended Reality (XR) course, but since the actual requirements for this project were very open, it was also combined with an earlier passion project of mine, that being an infinite Minesweeper game. The earlier project was made without a proper game engine & with very little structure, so this new project became a sort of re-make, as well as an expansion of the original project into a roguelike, so it would be a "new" project & not just a pure remake of an existing one.

2.2 Restrictions

- One of the main restrictions of this project is time. Due to being quite an ambitious project, there's a lot of potential features that could be added to the game. Therefore, the limiting factor for many ideas is if it's worth the effort of implementing, or if that time is better spent working on something else.
- Certain relevant topics were a little outside the scope of this course's curriculum, like multithreading, sound design, UX design, Game design/theory, etc. These topics have therefore not been prioritized as much compared to their normal level of priority in game development.
- Due to the game being based on Minesweeper, a 3D implementation wasn't very feasible, as Minesweeper is not a game that translates well into 3D. While it would've been possible to make a 3D version of Minesweeper in VR or something like it, that would've been more a "proof of concept" rather than an actual enjoyable game. Therefore, the decision was made (with approval) to stick to 2D, for the sake of the game not being needlessly complicated & clunky.

3 THEORY, TECHNOLOGY & ARCHITECTURE

3.1 Game engine

The first question that needed to be answered for this project was what game engine to use. With many great options to choose from, Godot was chosen, due to being entirely free & open source, easy to set up, and simple & intuitive to use. The one drawback with Godot is that it does not follow the Model-View-Controller design pattern, instead splitting the game into a hierarchy of nodes, each with their own model, view & controller responsibilities. Although despite this unorthodox design, it does have its strengths as well, with a little bit of getting used to it.

3.2 Separate model

A separate model was created without dependence on Godot, leaving only the view & controller responsibilities to Godot in this project. The model was created with a single singleton entry point, that any part of the front-end (in this case, any Godot node script) can access. When changes in the model occur, these changes are then broadcasted to receivers in the front-end through an event system, allowing the UI to update accordingly

3.3 Making textures

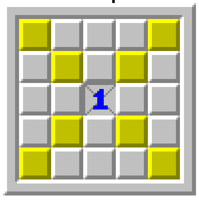
To easily make textures for the game, Aseprite was used. Aseprite is a paid external software that can be purchased on Steam, made specifically for creating pixel art. This software is in no way necessary to evaluate the project, as it just fills the purpose of an image editor & that’s it. Textures are stored in universally recognized image formats, and do not require Aseprite to be viewed or modified.

4 REQUIREMENTS

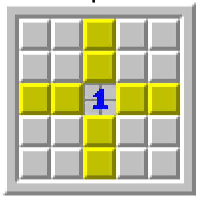
Since the project was very open without any "set in stone" requirements, a list was made early on with some requirements for the minimum viable product, all of which were fulfilled.

(Taken from README.md [2])

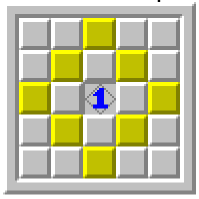
- Progressively generated board
- Progressively incrementing mine chance
- Additional lives, with squares or other items that can give them
- Squares or other items that reduce incrementing mine chance in some way
- Different types of number squares, that reveal mine info in different ways than a standard number square
    - Cross square

    - Plus square

    - Diamond square

- Save & load functionality, persistent storage
- Upgrades / power-ups
- Movable & zoomable camera
- Game can be restarted
- Basic functional UI
    - Main Menu / Title screen
    - Pause screen
    - Sidebar with…
        - Game info
        - Upgrade / power-up display

5 IMPLEMENTATION

5.1 Simplified class diagram

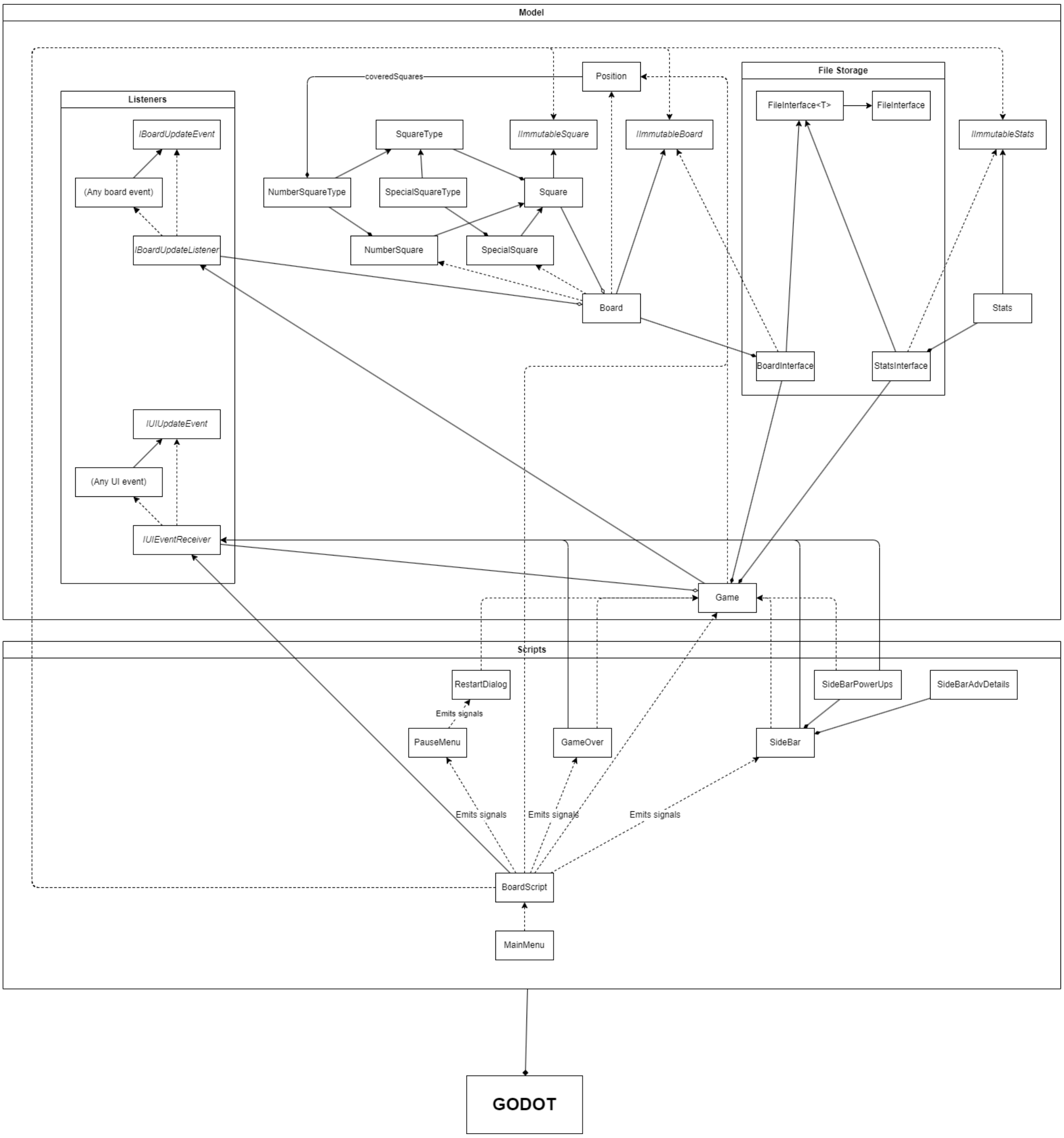


Figure 1 Simplified class diagram of the entire game (Project docs/new class diagram.png)

5.2 Installation

Launch the .exe included in the hand-in, or download a release from the GitHub repository [3]

5.3 How to play

(Taken from README.md [2])

All the standard rules of minesweeper apply in this game, you can click on squares to open them, you want to open as many squares as possible, while avoiding the bad squares. You can right click to place flags, and if you've found/marked every bad square around a number, you can middle click to instantly reveal the remaining squares around that number. However, there are also some new tiles you might encounter, and a full reference on every tile you can encounter is listed in the README.md file.

Power-ups are a new addition to this game, to spice up the gameplay & to add some more strategy. These are active-use effects that can be selected & used on a single tile. To use one, simply select the power-up to use, then left click on a tile on the board to use it there. Right clicking with a power-up selected deselects it.

5.4 Video demo

https://www.youtube.com/watch?v=HuUEWtcVFm0 [4]

6 EVALUATION

Overall, the project achieved all the minimum requirements set out at the start of development, and without much time to spare. The gameplay is addictive, which was discovered early on in development, as certain playtesting sessions became a lot longer than they needed to be. Although if anything, this only means the project succeeded at everything it set out to do.

The one point of criticism that does exist is the slight lack of identity as a roguelike. The game feels closer to just infinite Minesweeper than it does to a Minesweeper roguelike, although with time & more development, extra features can be added to solve that problem.

The project is also lacking in code documentation. While some documentation exists, there are large parts of the code that are still undocumented, which is mainly because of this project being developed solo. ~~Although this problem is somewhat easy to fix, it's worth considering that the developer of this project got a little tired, and is currently sleep deprived & writing this at 2:59 with only 9 hours until the project deadline~~

7 CONCLUSION & EXPERIENCES

Finally, for some personal opinions, I'm happy with how this project has turned out, and I've learned a lot about the use of game engines. The project had a little bit of a slow start, mainly because of me knowing next to nothing about the game engine I chose, as well as the programming language (C#). This has also led to some inconsistent coding styles regarding things that are new/different in C# compared to what I'm used to (Java), though these are minor issues that'll get fixed as development continues.

For future development, I'm looking forward to developing this further, as this is likely not the last time I touch this project. Developing it further after the deadline was always the plan since I started development, and I plan on following through with that

8 REFERENCES

[1] GodotSweeper-Infinite, https://github.com/SmileyFace799/GodotSweeper-Infinite

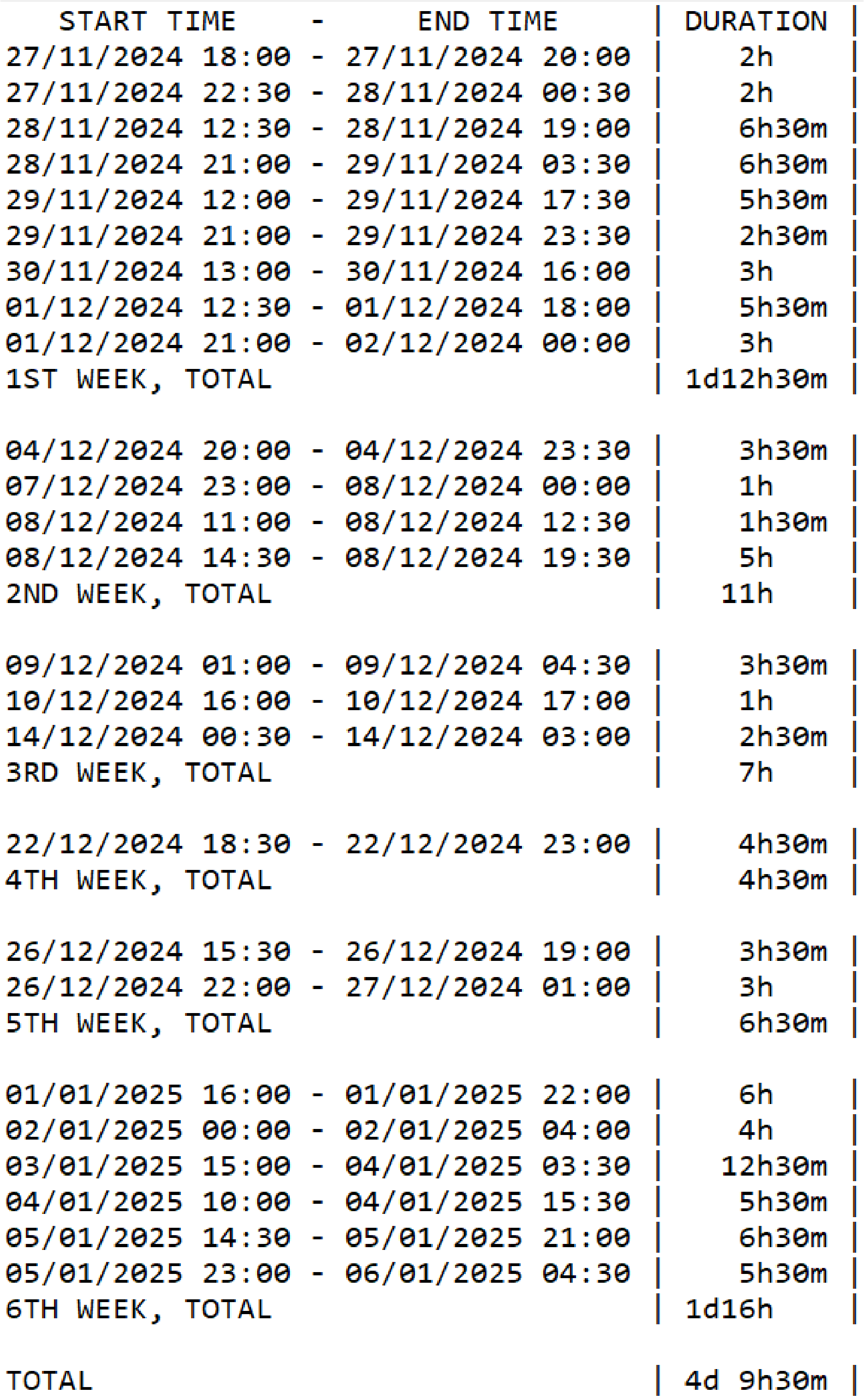
[2] GodotSweeper-Infinite: README.md, https://github.com/SmileyFace799/GodotSweeper-Infinite/blob/main/README.md

[3] GodotSweeper-Infinite: Releases, https://github.com/SmileyFace799/GodotSweeper-Infinite/releases

[4] RogueSweeper – Demo video, https://www.youtube.com/watch?v=HuUEWtcVFm0

9 ATTACHMENTS

9.1 Time table



```
   START TIME    -     END TIME    | DURATION |
27/11/2024 18:00 - 27/11/2024 20:00 |    2h     |
27/11/2024 22:30 - 28/11/2024 00:30 |    2h     |
28/11/2024 12:30 - 28/11/2024 19:00 |    6h30m  |
28/11/2024 21:00 - 29/11/2024 03:30 |    6h30m  |
29/11/2024 12:00 - 29/11/2024 17:30 |    5h30m  |
29/11/2024 21:00 - 29/11/2024 23:30 |    2h30m  |
30/11/2024 13:00 - 30/11/2024 16:00 |    3h     |
01/12/2024 12:30 - 01/12/2024 18:00 |    5h30m  |
01/12/2024 21:00 - 02/12/2024 00:00 |    3h     |
1ST WEEK, TOTAL                     | 1d12h30m  |

04/12/2024 20:00 - 04/12/2024 23:30 |    3h30m  |
07/12/2024 23:00 - 08/12/2024 00:00 |    1h     |
08/12/2024 11:00 - 08/12/2024 12:30 |    1h30m  |
08/12/2024 14:30 - 08/12/2024 19:30 |    5h     |
2ND WEEK, TOTAL                     |   11h     |

09/12/2024 01:00 - 09/12/2024 04:30 |    3h30m  |
10/12/2024 16:00 - 10/12/2024 17:00 |    1h     |
14/12/2024 00:30 - 14/12/2024 03:00 |    2h30m  |
3RD WEEK, TOTAL                     |    7h     |

22/12/2024 18:30 - 22/12/2024 23:00 |    4h30m  |
4TH WEEK, TOTAL                     |    4h30m  |

26/12/2024 15:30 - 26/12/2024 19:00 |    3h30m  |
26/12/2024 22:00 - 27/12/2024 01:00 |    3h     |
5TH WEEK, TOTAL                     |    6h30m  |

01/01/2025 16:00 - 01/01/2025 22:00 |    6h     |
02/01/2025 00:00 - 02/01/2025 04:00 |    4h     |
03/01/2025 15:00 - 04/01/2025 03:30 |   12h30m  |
04/01/2025 10:00 - 04/01/2025 15:30 |    5h30m  |
05/01/2025 14:30 - 05/01/2025 21:00 |    6h30m  |
05/01/2025 23:00 - 06/01/2025 04:30 |    5h30m  |
6TH WEEK, TOTAL                     | 1d16h     |

TOTAL                               | 4d 9h30m  |
```

Figure 2 Timetable
(time table.txt)

9.2 GitHub repository

https://github.com/SmileyFace799/GodotSweeper-Infinite [1]