FRONT PAGE

- Module Code: CS2JA16
- Assignment report Title: Android Game
- Student Number (e.g., 25098635): 29015642
- Date (when the work is completed): 23/03/22
- Actual hrs spent on the assignment: 25
- Assignment evaluation (3 key points):
    1. I really enjoyed this assignment.
    2. Learning android studio was good and was a good challenge for me
    3. Professor was really helpful

Minesweeper

Abstract

This project is based on the recreation of the original windows XP game Minesweeper. This report will underly the steps and procedures I have used to in order to create Minesweeper as an android game. As well as all the improvements and extensions I have added to the game.

1. Introduction and Showcase

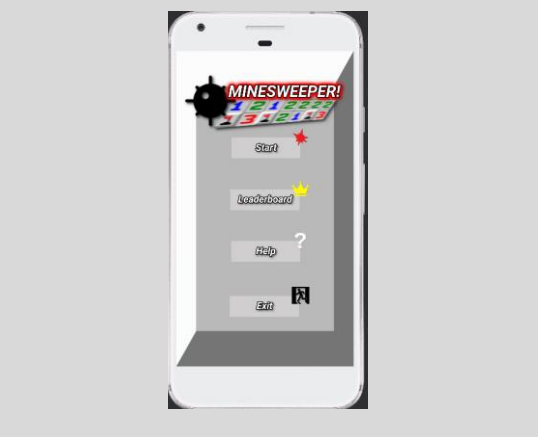


Fig. 1: Main menu page with selection of different sections of the app. Including professional UI, images and colourful titles to improve player experience

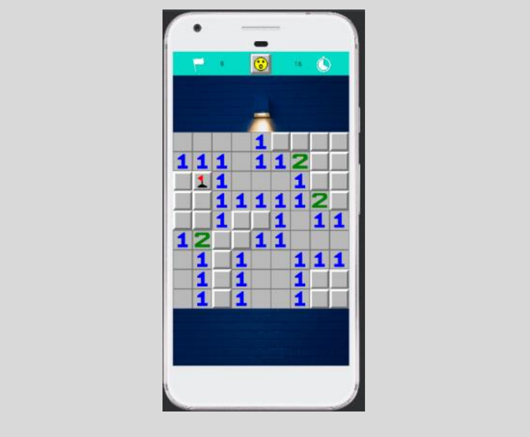
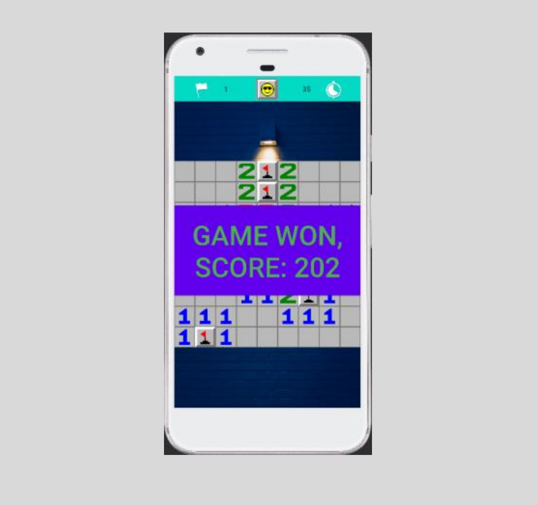
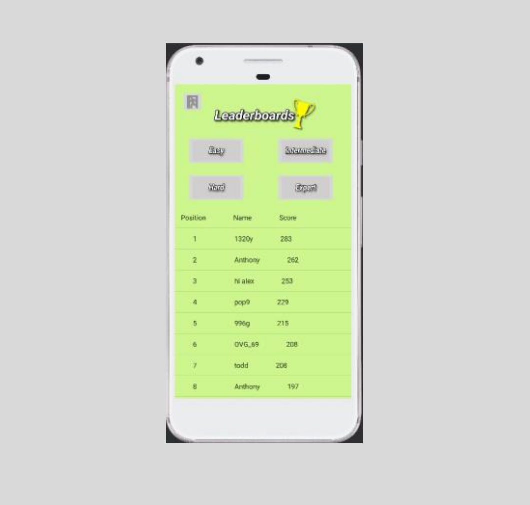


Fig. 2: Minesweeper game playing, player flagged a bomb and has most of the cells unrevealed. There is a timer and a flag counter on the top of the screen that update accordingly.



GAME WON, SCORE: 202

Fig. 3: Minesweeper game won. All bombs were flagged, and every other cell was unrevealed. Game has ended and the timer stops, score is generated and is saved to the database



Position	Name	Score
1	1320y	283
2	Anthony	262
3	hi alex	253
4	pop9	229
5	996g	215
6	OVG_69	208
7	todd	208
8	Anthony	197

Fig. 4: Leaderboard menu from firebase database and configured to be displayed in a ‘leaderboard’ manner. Displays each player with their position, which is related to the highest score.

2. OOP design

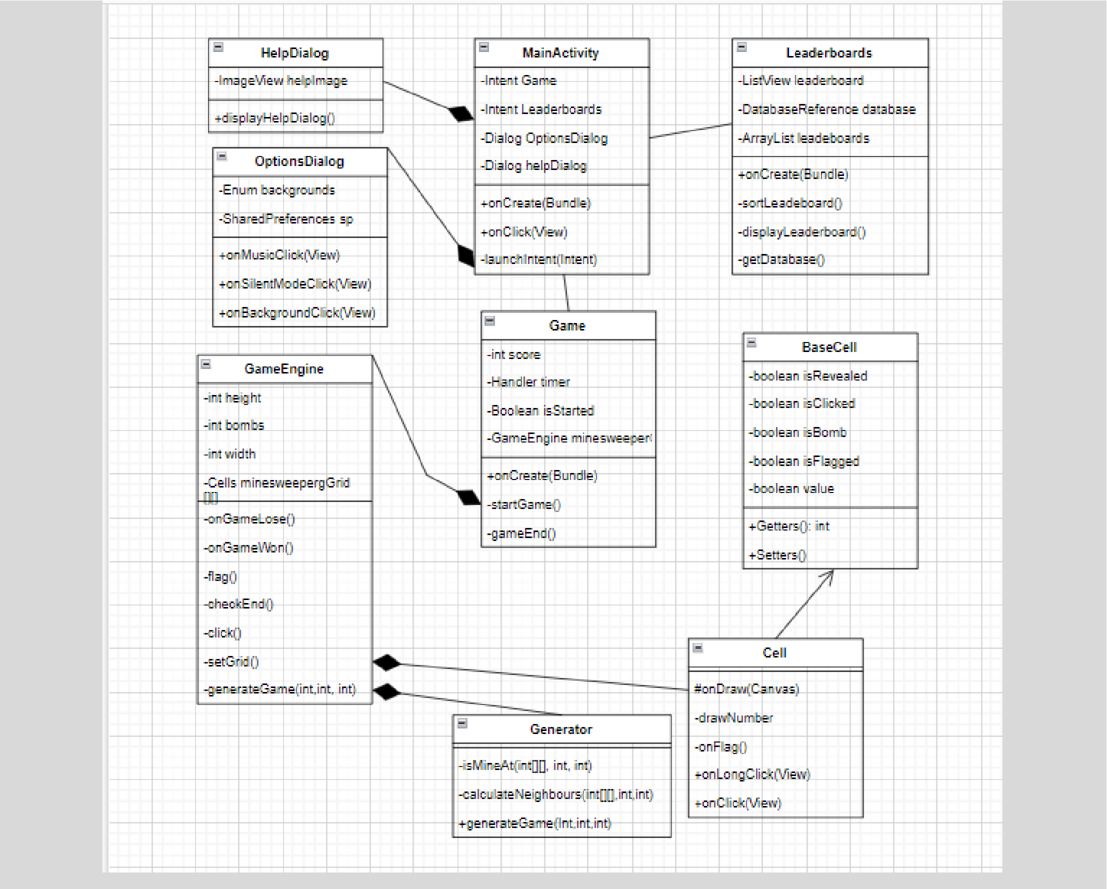


Fig 5. Caption

Fig 5, as shown above, is the class diagram at its early stage of development. The diagram contains incomplete and missing classes. This is because it was purely used for design purposes, hence aiding me throughout my development of the game itself.

This class diagram really helped me understand how all the objects and classes are going to be linked and associated with one another. Throughout the development process of Minesweeper, I referred to my class diagram many times to help me structure the creation of my classes. The various structural relationships between each class, gave me an idea on which classes will have a part of, has a, or inherits from relationships. For example, the class Cell is designed to inherit from BaseCell. This shows that all the procedures, functions and variables that are in BaseCell are now in Cell too. This means that Cell can call the getters and setters that BaseCell has.

Unfortunately, I did not use the optionsDialog in my project at all due to time constraints, however the design for it was prosperous. On the other hand, I did add a few more classes to my project that were not in my design. A significant one was the Menu class. This Menu class was used for level selection and custom level creation. If I were to add the Menu class into my design, it would fit in between MainActivity and Game.

3. Memory usage and Speed improvements

A short description (a few sentences) about the improvement of memory efficiency and speed efficiency for your game

To improve the memory usage and speed, I have implemented an optimisation that removes unused intents. Android applications work like a stack. This means that when a new intent is created, the new intent is put pushed on top of the stack. Over time, this will start to become a problem as player would play multiple games of minesweeper in one instance of the application. Therefore, the application takes up more storage. This issue was solved by implementing intent flags when creating new intents [1]. These flags clear all previous intents so that memory can be freed. I tested this by using a profiler. Without the flags, launching a new intent would increase total memory usage. However, when flags are included, the total memory usage reduces dramatically, especially after had launched many minesweeper games.

4. Improvements/extensions

As an extension, I have added an animated face while the game is running. This face will change its image based on the users moves. If the user flags a cell, then the face will look shocked, if they reveal a cell, then the face will smile, if they click on a bomb, it will go dead and if they win the face will have sunglasses. This feature was added to make the user experience more exciting and adventurous.

To make the game more entertaining, I added music and sound. These were added by using Android’s API, MediaPlayer. MediaPlayer allowed me to start, stop and pause any audio within my source folder. Music plays in the background while a minesweeper game is playing, this is to make the gameplay more intense and interactive. I downloaded music straight from YouTube [3]. Most of the sounds I created myself with my microphone, for example, at the end of each game, if the user has a new highscore, then an audio with my voice will play “new highscore”.

I have implemented a fully functional leaderboard system. I took big inspiration from an article on stack overflow to create this. [2] I wanted to create a familiar and easy to use leaderboard system so that the user can see in real time, their scores compared to other users playing the game.

5. Conclusions

Using my OOP design, external references and additional help, I have created a professional recreation of Minesweeper. Additional features and vital memory optimisation have been added to provide a smooth and fast game to give the best user experience possible. I have really enjoyed working with android studio.

References

[1]https://stackoverflow.com/questions/12947916/android-remove-all-the-previous-activities-from-the-back-stack

[2] https://stackoverflow.com/questions/64264915/firebase-android-leaderboard-system

[3] https://www.youtube.com/watch?v=7I9E6Enw1R0

(END OF REPORT)