General Statement

As a group, we wanted to create our own version of the game, Minesweeper. Minesweeper is a single-player, puzzle style game in which the user is trying to find a number of mines hidden beneath a grid of squares. The game begins with a mostly empty grid, along with a few numbers between one and four. The numbers are indicators as to how many bombs that spot is touching.

In order to create our own version of the game in java we needed to begin from a base level. We first needed to understand the logic behind how bombs are placed randomly and how we could use a program to search the remaining board for bombs.

When deciding to create a version of the game Minesweeper, we wanted to come up with something that was different from the original game. One way we did this was creating another level, called “Insane”. This change provides the user with four levels of difficulty to choose from, easy, medium, hard, and insane. Another way that we changed our game from the original is that in the original game you can go over the amount of flags and place as many as you want, then you have to get rid of the extra ones at the end. Instead, we don’t allow the player to add more flags then there are bombs.

The final stage of the program involved creating a colorful and interactive game board. We wanted to create the same visual appeal that a normal game of minesweeper provides the users. We thought about how the visual aspect of a game is very important to a user and wanted to make it as appealing as possible. One way we did this was by giving the user an option between light and dark mode, so give the user a choice in what their screen looks like. In order to do this, we had to add another style sheet.

Introduction-

Minesweeper is a computer game, which uses strategy and the logic of the player. The game starts with a blank grid of squares. Once the player clicks on a location, bombs are randomly placed at different spots on the grid. The point of the game is to determine where all the bombs are located by flagging the square they are in. The only guidance the player gets is if a square is bordering a bomb and the player clicks on it, it will say how many bombs it is bordering. However, once the player clicks on a square with a bomb, the game is lost.

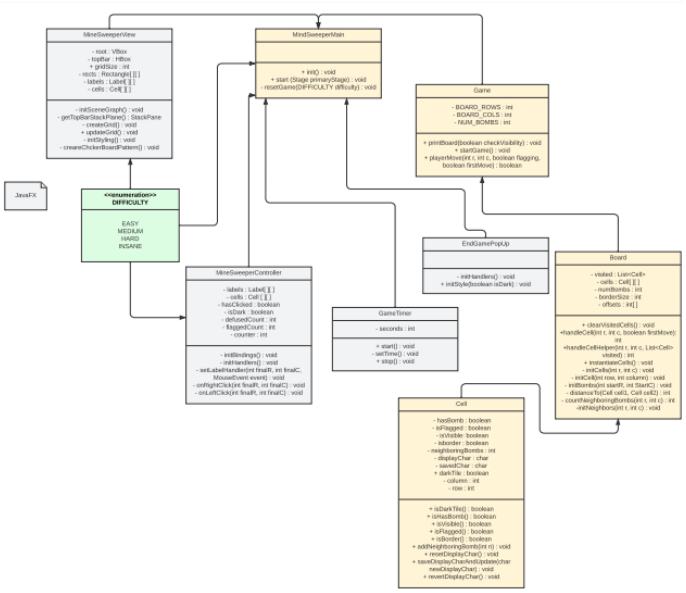
The problem that our group chose was to create the game Minesweeper, and to be inclusive of people's different ability and skill levels we created multiple levels. This way if the player has never played the game or is struggling, they can go to an easier level. The levels are easy, medium, hard and insane, and as the levels get harder, the game will have more bombs. We also wanted to be more inclusive for different people playing and if they like dark mode or light mode better on their computers and phones. This is why we added the option for the player to choose either.

Background/Motivation-

Minesweeper is an interesting puzzle game that requires significant problem solving skills to solve quickly. The game consists of finding bombs based on adjacent tiles. Tiles may consits of a 1-4 based on how many bombs are left,right, up, down or diagonal to that tile. The players first click will not consist of a bomb. The game is over when the user correctly flags all the bombs.

The player loses if they click on a bomb, misidentifying it as a regular tile. In order to create this game, there will need to be an individual cell, this cell will hold various information about that individual cell, like how many bombs are near it, a board which consists of these cells and various mechanics to allow the player to play it.

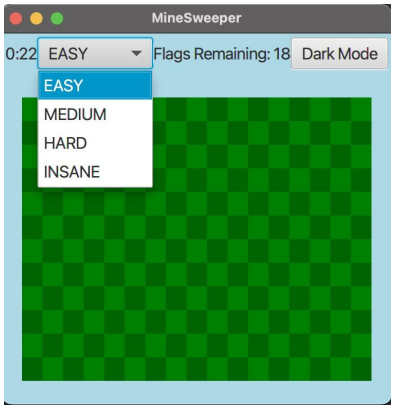
UML Class Diagram:



Instructions-

The object of the game is to determine where all the bombs are, using the numbers on the tiles.

Once the game is running, players should select the difficulty level they want to play on in the top right corner from the dropdown menu. Once the difficulty is selected, the game will automatically reset starting with that difficulty level.

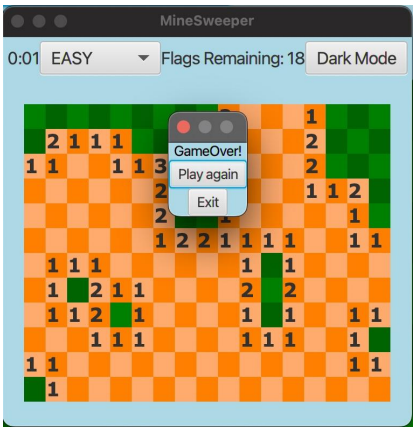


Use the left click button on the mouse to select a space on the grid. Tiles will then be removed revealing empty tiles and tiles with numbers on them. The numbers on the board represent how many bombs are adjacent to a square. For example, if a square has a “2” on it, there are 2 bombs next to that square. The bombs could be above, below, right, left, or diagonal to that square.

If you have located a bomb, right click on it to flag it. Flags remaining demonstrates how many flags the player has left.

Once the flags remaining counter reaches 0, the game will end if all flags are correctly placed on bombs, however if a flag is misplaced, the player should pick up the incorrect flag by using right click on that tile.

Once all bombs have been successfully flagged, the game ends resulting in a player win. However, if they click on a bomb, the game ends, resulting in a player loss.



If the player wants to play again they should click on the play again button, however if they want to exit they should click on the exit button.