
By Ali Ghazniwal (788944)

# [2.1] Gameplay

Minesweeper is a game about finding and avoiding mines in a grid of cells. The player will start with an uncovered area and must use pattern recognition skills to figure out where bombs are and flag them. While doing so, the player must uncover the rest of the cells while avoiding the bombs. Failing to do so will result in a loss and the player will have to start over.

## [2.1a] Menu

The menu will have a title and a play button. There will also be a stats button to show the players fastests time and which modifiers were enabled. This will be a tinier window that pops up upon pressing the stats button.

## [2.1b] In Game

In the game there will be a 1:1 grid, this grid will be the field for the game. Above the grid will be 3 things. The players highscore, amount of flags left, and a timer. Under the grid will be 6 buttons. 3 of which will be difficulty related, 3 will be modifiers, and the remaining 1 button will be a restart button.

Here, the player can start the game by clicking a mine. From there, the game will place mines all over the grid randomly. However, it will avoid the area the player clicked to ensure the player starts with an open area to work with. From there, the player can right click to place a flag. If a flag was placed, the cell can no longer be clicked on until the player decides to remove it by right clicking it once more.

## [2.1c] Win / Lose Functionality

Upon clicking a mine (losing) or uncovering all cells that aren't mines (winning) a little window will pop up telling the player that they won and show their stats (time, modifiers, highscore, etc). The player will be able to click out of this pop up by pressing an X button at the top right. If the player reaches a new highscore, it will be saved as a local file.

# [2.2] Functionality

The game only takes in two inputs, right and left click. Meaning that the entire game will rely on the **mousePressed** event for updates. This means that I will have to do a lot of checks to see if a player is clicking in a certain area for my buttons functionality.

Without repetition, this process will require a lot of for loops for the grid. Which is why I can use math and for loops to check if the player clicked a cell in the grid and turn the **mouseX** and **mouseY** values into an X and Y coordinate on the grid, which can be used for registering a click said cell.

# [3.1] Graphics

The game's graphics will all be designed by me in a doodle-like style to make the game look more bright and simple. This will all be designed using Adobe Illustrator. Things like the buttons, flags, numbers, and title will be designed using Adobe Illustrator.

# [3.2] Gamemodes

## [3.2a] Green Light Red Light

At the top right of the screen will be an eye which will be watching for player movement. While the eye is closed, the player is free to do whatever on the grid. But when it is open, if the player makes any movements or inputs, the game will be over. The goal of this gamemode is to uncover all cells and find all minds before time runs out and avoid getting caught by the eye.

## [3.2b] Fog Of War

Upon uncovering a cell, the cell will only be visible for a little while before it is covered up again. Turning the game into both a pattern recognition and memory game. However, flags will still show but the numbers will not.

## [3.3c] Inverted

Cells that are uncovered will show an inverted number. As seen in the chart below.

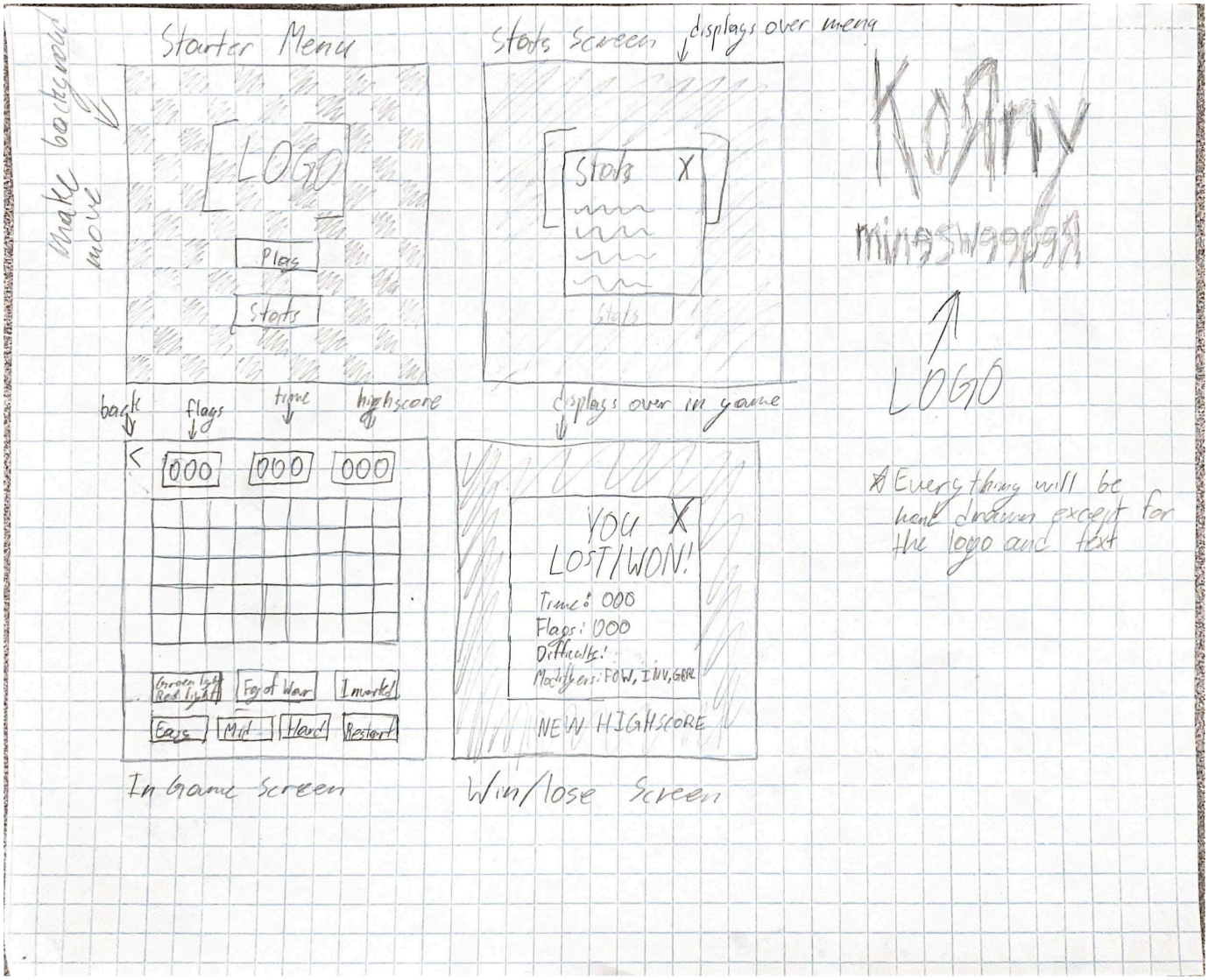
| Original Number | Inverted Number |
|---|---|
| 1 | 8 |
| 2 | 7 |
| 3 | 6 |
| 4 | 5 |
| 5 | 4 |
| 6 | 3 |
| 7 | 2 |
| 8 | 1 |

The cells number will be inverted as seen in the chart above

# [4.1] Timeline

| Game Design Plan | | Week 1 | | | | | | | | | | Week 2 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | M | | T | | W | | T | | F | | M | | T | | W | | T | | F | |
| **Activity** | **Duration** | | | | | | | | | | | | | | | | | | | | |
| Design | | | | | | | | | | | | | | | | | | | | | |
| Planning | 1 | | | | | | | | | | | | | | | | | | | | |
| Timeline | 0.5 | | | | | | | | | | | | | | | | | | | | |
| Design UI | 0.5 | | | | | | | | | | | | | | | | | | | | - |
| Pseudocode / IPO | 1.5 | | | | | | | | | | | | | | | | | | | | |
| Assets (UI, SFX) | 1 | | | | | | | | | | | | | | | | | | | | |
| Programming | | | | | | | | | | | | | | | | | | | | | |
| Start Menu & Stats | 1 | | | | | | | | | | | | | | | | | | | | |
| In Game UI & Death/Win UI | 1 | | | | | | | | | | | | | | | | | | | | |
| Main Gameplay | 1 | | | | | | | | | | | | | | | | | | | | |
| Fog Of War | 1 | | | | | | | | | | | | | | | | | | | | |
| Green Light, Red Light | 1 | | | | | | | | | | | | | | | | | | | | |
| Inverted | 1 | | | | | | | | | | | | | | | | | | | | |

# [4.2] UI Design


Starter Menu
make background move
LOGO
Play
Stats
Stats Screen
displays over menu
Stats
X
Kostny
minesweeper
LOGO
back
flags
time
highscore
000
000
000
displays over in game
Green light Red light
Fog of War
Inverted
Easy
Mid
Hard
Restart
YOU X
LOST/WON!
Time: 000
Flags: 000
Difficulty:
Modifiers: FOW, INV, GOR
NEW HIGHSCORE
In Game Screen
Win/lose Screen
* Everything will be hand drawn except for the logo and text

# [5.1] Psuedocode

```
// void draw()
START
	drawBackground()

	IF player is in menu
		drawMenu()

		IF stats page is open
			drawStats()
	ELSE
		drawGame() // draw game will draw all the elements inside of the in game screen

		/*
			This includes:
				- drawGameStats() // this will draw the timer, amount of flags, and highscore
				- drawGrid() // will render the field
				- drawButton() // this will be called multiple times to create the buttons under the field
		*/

		IF dead or won
			drawEndScreen()
END
```

# [5.2] IPO Chart

| Name & Description | Input | Process | Output |
|---|---|---|---|
| drawBackground() | | Draw an image that will move around using sin and cos | |
| drawMenu() | | Draws menu elements<br>- Play button<br>- Stats button<br>- Logo | |
| drawLogo() | X: int<br>Y: int<br>Scale: int | Draws a logo at said location with said scale | |

| | | | |
|---|---|---|---|
| drawStats() | | Draws stats menu over menu with the background elements shaded and will draw<br>- The players fastest time<br>- Hardest modifiers<br>- Hardest difficulty | |
| drawGame | | Draws the buttons and numbers on the in game scren | |
| drawGameStats | | Draws the timer, flags left, and highscore above the grid | |
| drawGrid<br>// Renders the minesweeper cells | | Renders the minesweeper cells | |
| drawButton<br>// Draws a button (used to reduce lines | text: string<br>X: Int<br>Y: Int<br>sizeX: Int<br>sizeY: Int<br>imageID_1: Int<br>imageID_2: Int | Creates a button given on the input and the images<br><br>The frame of the button will swap between the two image IDs | |
| mouseButtonClick<br>// checks if the mouse is over the given area | X: Int<br>Y: Int<br>sX: Int<br>sY: Int | mouseX >= X && mouseX <= X + sX && mouseY >= Y && mouseX <= Y + sY | Boolean |
| clickCell | X: int<br>Y: int<br>Chording: boolean | Will register a click on the given cell on the minesweeper grid | |
| findOpenCells | X: int<br>Y: int | Will find all blank cells in an area and returns them | Array |
| flagCell | X: int<br>Y: int | Will register a flag on the given cell on the minesweeper grid | |
| restart | | Restarts the game | |
| handleGridClicks | Input: mouseButton | Will find where the player clicked on the grid and register a click (checks if its a | |

| | | | |
|---|---|---|---|
| | | left or right click too). Iterates through a for loop that checks every cell with the mouseButtonClick method | |
| setDifficulty | difficulty: string | Changes the games difficulty | |
| saveData | JSON: JSONObject | Saves the JSON as a file<br><br>REFER TO https://processing.org/reference/JSONObject.html | Boolean (if it errors or not) |
| readData | | Returns saved player data | JSONObject (the player data) |
| playSound | Name: string | Plays the given sound<br><br>REFER TO https://processing.org/reference/libraries/sound/SoundFile_play_.html | Boolean (if it errors or not) |
| handleMenuButtons | | Registers clicks on the menu buttons (start and stats) | |
| handlePostGameButtons | | Registers clicks on the death/win screen | |
| handleStatsButtons | | Registers clicks on the stats screen | |
| handleInGameButtons | | Registers clicks on the in game screen | |