

CS104 Project Report

Minesweeper Cricket

Deeksha Dhiwakar

June 2023

Contents

1	Project Description	2
2	Implementation	2
2.1	Game Setup	2
2.2	Playing	2
2.3	Instructions	3
2.4	Customize	3
2.5	Leaderboard	4
2.6	Settings	4
3	Customization	5
4	Purpose of each File/ Folder	6
5	Modifications to Reference Code	6
6	Compiling the Code	6

1 Project Description

I have created a game that is a simple combination of the two classic games, minesweeper and cricket. I have used HTML and CSS to create a responsive and interactive user interface, and incorporated dynamic functionality using JavaScript.

The ultimate goal of the game is to maximize the player's score (number of runs). The player scores runs by uncovering cells on the grid of the playing field. The field also contains fielders, randomly distributed and hidden. If the player clicks on a cell containing a fielder, the game is terminated and the score thus far is final. To 'win', the player must uncover all cells except those containing fielders.

2 Implementation

A brief summary of the game logic is as follows:

2.1 Game Setup

When the server is launched, a function is called to generate the grid, based on the values of field size and number of fielders. Depending on the field size, the function also decides the number of cells containing power-ups and 1, 2 and 3 runs. Three successive functions are then called, which distribute fielders and power-ups and then assign run values to the remaining cells. The logic behind the distribution is to maintain an array of all possible unoccupied cell coordinates, shuffle it using random function implementation and pop off coordinates when we wish to add a fielder/ power-up/ run.

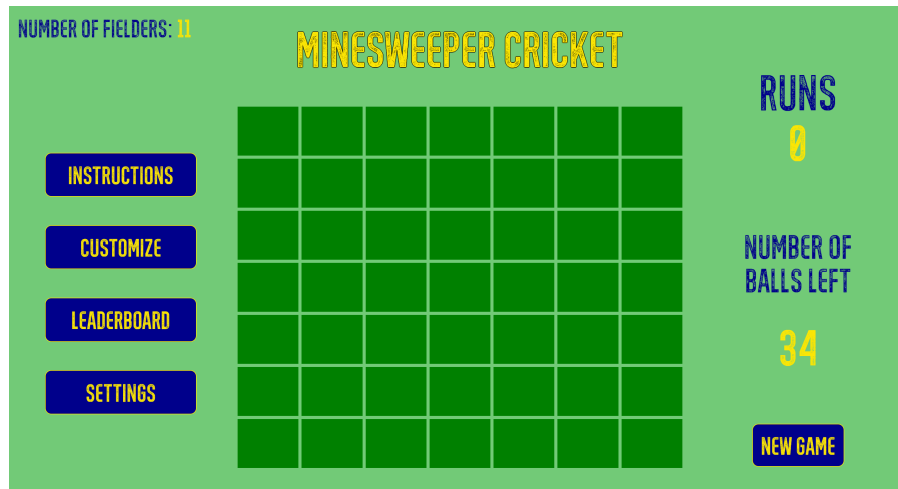


Figure 1: Field on Startup

2.2 Playing

When the player clicks on a cell, the code first checks if the cell contains a fielder or not. If it does, the game ends (assuming the player does not have an extra life, if they do the game goes on) and all the fielders' positions are revealed, and the final score is displayed. If the cell does not contain a fielder, a function is called to check if the cell contains a power-up. If it does, appropriate

variables are updated and the power-up's image is displayed. If it doesn't, the function checks for the number of runs the cell holds. The value is displayed and the current score is updated.

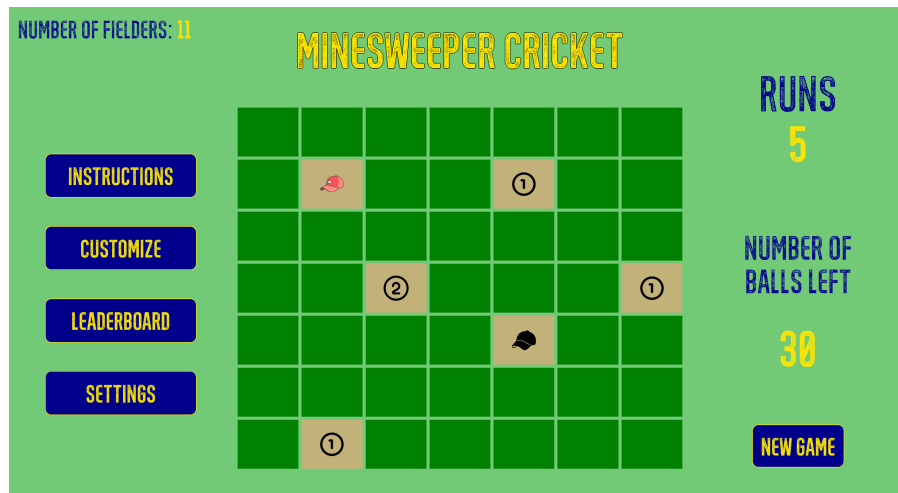


Figure 2: Field while Playing

2.3 Instructions

When the player clicks on this button, the game rules are displayed.

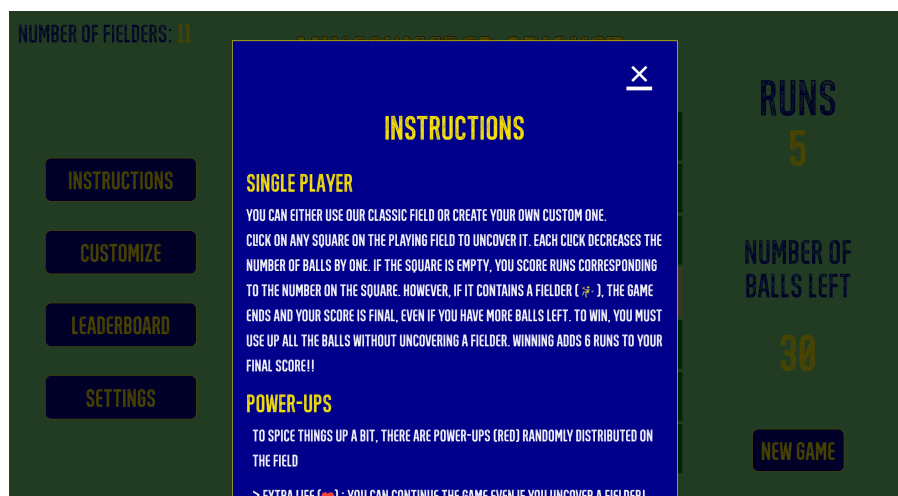


Figure 3: Instructions

2.4 Customize

By clicking on this button, the player has the option to set custom field size and number of fielders, or to use the custom field. JavaScript variables store these values and regenerate the grid with the updated parameters.

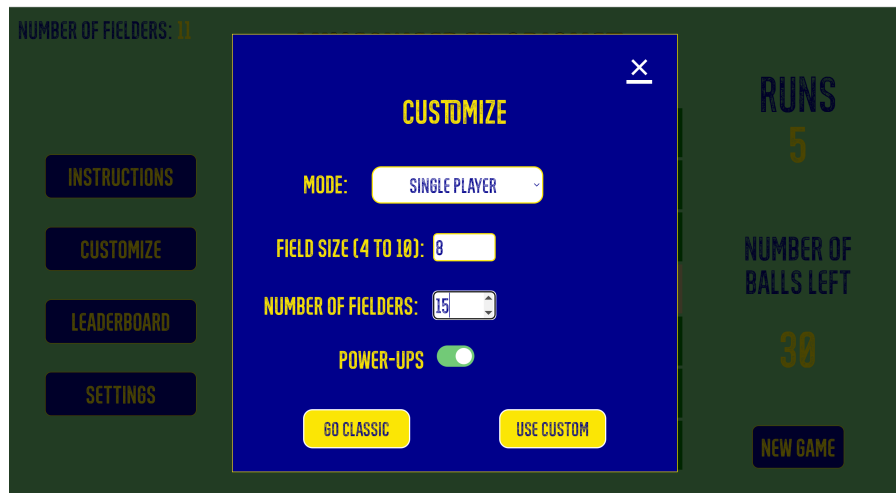


Figure 4: Customize

2.5 Leaderboard

Once a player has finished playing on the classic field, they have the option to save their score to the leaderboard. If their score is among the top five scores, it is added to the leaderboard. I have implemented this by maintaining an array of objects, each object containing the player's name and score. If the player chooses to save their score, their name is taken as input and added to the map along with their score. Then a separate function is called to set the leaderboard. It sorts the map in descending order of players' scores and then populates the leaderboard with the top five scorers.

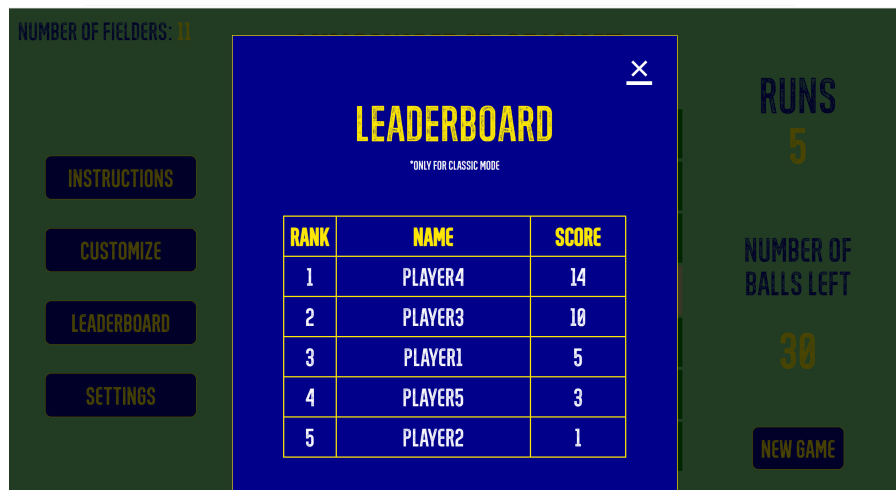


Figure 5: Leaderboard

2.6 Settings

By clicking on this button, the player can choose to enable/ disable background music and sound effects.

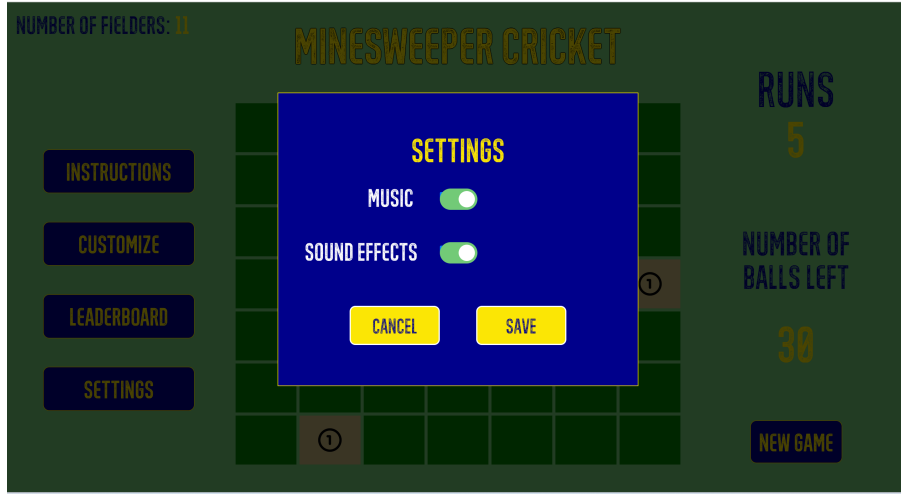


Figure 6: Settings

3 Customization

Some of the custom features I have added to the game are

- **Custom field size:** Players have the option to set the size of the grid.
- **Custom number of fielders:** Players can also choose the number of fielders to play against.
- **Classic mode:** Instead of setting the grid size and number of fielders themselves, players can simply choose Classic mode to play on a 7 by 7 grid with 11 fielders.
- **Leaderboard:** Players have the option to save their scores in classic mode to a leaderboard, which displays the top five scores across games.
- **Random runs:** The cells contain different number of runs (1, 2 or 3).
- **Background music and Sound effects:** I have added background music to the game as well as different sound effects for each of the power-ups, runs, winning and losing.
- **Power-ups and Power-downs:** I have added two power-ups and two power-downs that are randomly distributed in the field along with the runs and fielders. The power-ups are Extra Life and Removing a Fielder and the power-downs are Minus 2 Runs and Adding a Fielder.
- **New Game Option:** At any point of time, the player can start a new game by simply clicking the New Game button at the bottom right corner. The custom settings are also retained.
- **Winning:** The player 'wins' when they have uncovered all cells except those containing fielders. Winning adds 6 runs to the player's final score.
- **Settings Menu:** Users have the option to enable or disable background music and sound effects.

4 Purpose of each File/ Folder

- **game.html:** Contains the HTML code for the game.
- **style.css:** Contains the CSS that styles the HTML elements.
- **script.js:** Contains the JavaScript code that implements the game functionality.
- **fonts:** Contains the custom fonts I have downloaded from the internet [1].
- **images:** Contains all the images I have used in the game, downloaded from the internet [2].
- **sound:** Contains all the sounds I have used in the game, downloaded from the internet [3].

5 Modifications to Reference Code

I created the grid and implemented the basic JavaScript functionality using the code from [4]. I used [5] as reference for advanced HTML and CSS elements (like transitions, [6]) and also for some specific JavaScript functions.

To implement the leaderboard functionality, I referred to [7].

I downloaded custom fonts from [1], images from [2] and sounds from [3].

6 Compiling the Code

The game requires no compilation. The player simply has to double click on *game.html* to begin playing the game.

In order to generate *report.pdf* from *report.tex* and *report.bib*, run the following commands sequentially on Terminal:

```
pdflatex report
bibtex report
pdflatex report
pdflatex report
```

References

- [1] [Online]. Available: <https://www.fontspace.com/>
- [2] [Online]. Available: <https://www.flaticon.com/>
- [3] [Online]. Available: <https://mixkit.co/>
- [4] [Online]. Available: <https://www.101computing.net/minesweeper-in-javascript/>
- [5] [Online]. Available: <https://www.w3schools.com/>
- [6] [Online]. Available: https://www.w3schools.com/css/css3_transitions.asp
- [7] [Online]. Available: <https://stackoverflow.com/questions/47918195/creating-a-basic-html-javascript-leaderboard>