

CS104 Project Report

Ananya Rao

June 14, 2023

Contents

1	Project Description	2
2	Codes and Files	2
2.1	index.html	2
2.2	game1.css and game2.css	2
2.3	game.js	2
2.4	Other files	3
3	How does the Game work?	3
3.1	Instructions	3
3.2	Playing	3
3.3	Quit Game	3
4	JavaScript	4
4.1	Taking input from user	4
4.2	Generating Grid	4
4.3	Cell Visuals	5
4.4	Runs and Lifelines	5
4.5	Change theme	5
4.6	Turn audio on/off	6
5	CSS	6
6	Customizations	6
7	Compilation	7

1 Project Description

This is a **Classic Game** page made as the final project of *CS104(SSL)* course. It's an amalgamation of already available minesweeper and cricket games. Like cricket, it has 11 players(but all fielders :p), so one will be playing against them. The player has to click on any one of the boxes of his/her choice and the game goes on according to the type of box revealed. It's a fun game with appealing features and surprises.

The JavaScript codebase was taken from the internet and modified and customized accordingly. The HTML and CSS codes were made single-handedly with a little help from [w3schools](#) and other related websites.



Figure 1: Grid Page

2 Codes and Files

2.1 index.html

There is only one **HTML** file named *index.html* which includes all the html code. This is a one-page game therefore just one html code suffices. All the *CSS* and *JS* code files are linked to *index.html* file [3].

2.2 game1.css and game2.css

There are two css files for two different themes. The CSS files get linked one at a time only, game1.css is by default linked to the html file. The linked file changes on clicking the "Change Theme" button according to what was linked earlier.

2.3 game.js

game.js code file contains all JavaScript code required for responsive page and functioning of the game.

2.4 Other files

- **grass.jpg** - Background image of playground.
- **bomb.png** - This is the image used to show a bomb in a cell
- **player.png** - This is the image used to symbolize a fielder in cell
- **failed.png** - Used as a close button in settings overlay during game
- **bombed.mp3** - Sound effect on getting bombed
- **four.mp3** - Sound effect on scoring a four
- **six.mp3** - Sound effect on scoring a six
- **lost_lifeline.mp3** - Sound effect on losing a lifeline
- **pop2-84862 (1).mp3** - Sound effect on scoring 1 or 2 runs.

3 How does the Game work?

This game **Minesweeper Cricket** is a blend of minesweeper and cricket games. Let us see how it works!

3.1 Instructions

The instructions for this game is showed up when **Instructions** button is clicked.

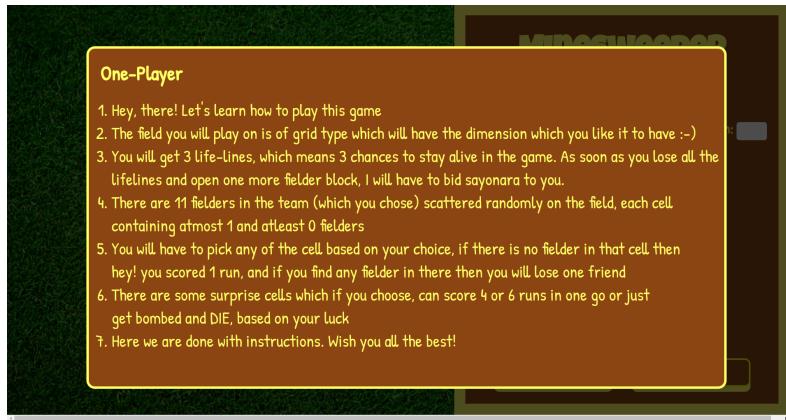


Figure 2: Instruction Overlay

3.2 Playing

Since we have read the instructions, now it's very easy to play this game. One just has to click on cells randomly and the game proceeds according to that.

3.3 Quit Game

There is a button at the bottom of the scoreboard. The player is given 3 options when that button is clicked. Restart, Continue and Homepage. On restarting, the game restarts with the same dimension. *Homepage* refreshes the page.



Figure 3: Quit Game button

4 JavaScript

The game.js file contains javascript code written for this game. I have used a codebase [4] from the internet and modified it according to my required features.

4.1 Taking input from user

There is an input box which takes the dimensions of the grid as input from the user. On clicking the button "Get Set Ready!", the functions "game()" "generateGrid()" are called. game() makes all the visuals other than the grid and scoreboard, invisible and pops up a wish for the player!

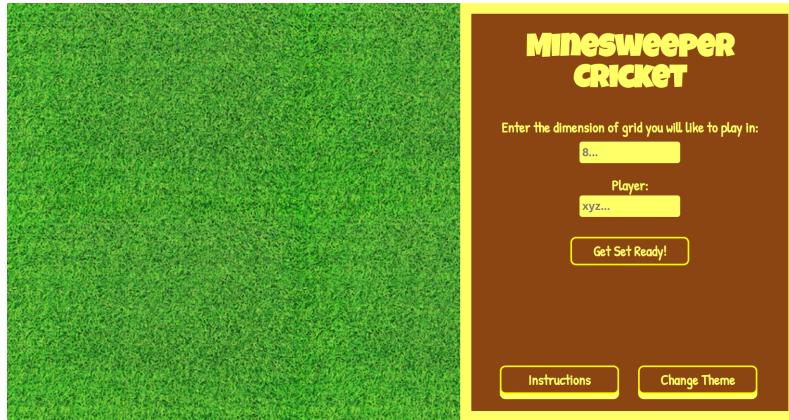


Figure 4: Home page, giving inputs

4.2 Generating Grid

All the values are initialised in the generateGrid() function and grids= cells are added. The functions which are used in the functioning of the game are getting called in this function. New attributes like fielders, bombs, and runs are getting created and assigned to the cells with an initial value of false. Now further on, random cells are chosen for each of the attributes and the value is set to true.



Figure 5: Newly generated grid

4.3 Cell Visuals

There is a function "clickCell(cell)" which gets called whenever a cell is clicked. This function checks the attribute value for each attribute for a cell and assigns it a class as per the value. The visuals of the grid cell are controlled by the class it has been assigned which is done based upon the true attribute. There are several audio effects based on the attribute's value for any cell.



Figure 6: Cell Visuals

4.4 Runs and Lifelines

I have declared two variables for runs and lifelines. Their value changes as per the type of cell revealed and accordingly the innerHTML of *Score* and *Lifelines left* gets updated.

4.5 Change theme

I have added an extra feature of theme option. There are two css files for this page. As one clicks the "Change Theme" button, if game1.css is linked then game2.css will get linked and its css will get applied. If game2.css is linked then game1.css will get linked and its css will be applied.

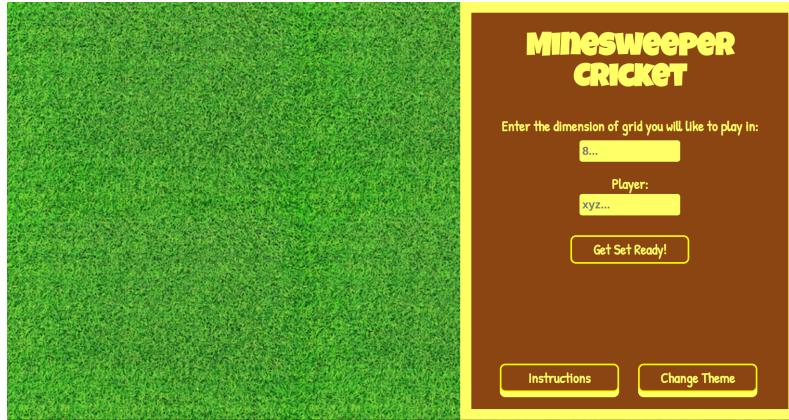


Figure 7: Theme 1 (Default)

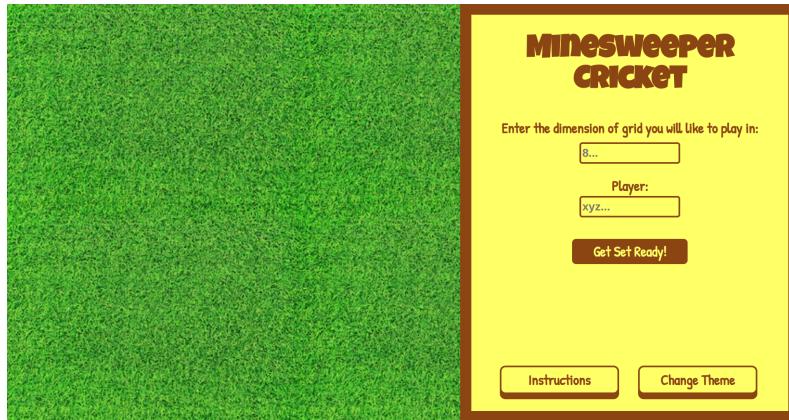


Figure 8: Theme 2

4.6 Turn audio on/off

Audio can be easily turned on or off using the audio button in settings. I declared a function called "audio()" which is being called on clicking the "Turn on/off audio" button. Depending on the current status of the audio, it will either turn on or off. There are four variables for the 4 audios, their value is set as true when audio is to be kept on and false in another case. These values are validated before calling .play() function.

5 CSS

Designing this page is done using the CSS features available on the internet [1]. The font style [2] is taken from Google fonts

6 Customizations

Some unique features of my game project:

- **Dimension of grid** can be chosen by the player as per choice. Although there is a recommended range of 6 to 10

- **Audio Effects** are there for few selected steps like scoring 4s, 6s, losing lifeline or getting bombed.
- **Theme change** is one of the coolest features. There are two themes. We can toggle between them using buttons.
- **Life-lines** are available. Its number decreases by one each time when caught by a fielder and the player is saved.
- **Settings**. One can turn off the audio effects using setting features.
- **Bombs** are planted in the cells. If a bomb cell is clicked, the game is over immediately.
- **Validation of input**. There are a few constraints on inputs. They must be fulfilled in order to proceed further.
- **4s and 6s** are there for the player to become happy in his/her gloomy time *-*
- **Random Runs**. Runs like 1, 2 and 3 can also be scored.

7 Compilation

These codes don't need any extra steps to compile. Following are the steps to be followed for running these codes.

- Download and extract the zip file.
- Open the folder in any IDE (for ex. VS Code).
- Open the index.html file and right-click on it.
- If you are using Linux, you must be seeing a "Show Preview" option. If not then download the extension for the live server and try again.
- If you are using Windows then you must see "Open with live server" option. Click on it and you are ready to play.
- **All the best!**

References

- [1] Url: <https://www.w3schools.com/css/default.asp>.
- [2] Url: https://fonts.googleapis.com/specimen/Patrick+Hand?preview.text=patrick%20hand&preview.text_type=custom&query=patr.
- [3] Url: <https://www.w3schools.com/html/default.asp>.
- [4] Url: <https://www.101computing.net/minesweeper-in-javascript/>.