

# CS104 Project

## Minesweeper Cricket

Satyankar Chandra  
22B0967

### Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Overview</b>	<b>2</b>
<b>3</b>	<b>Basic Logic</b>	<b>3</b>
<b>4</b>	<b>Customization</b>	<b>3</b>
<b>5</b>	<b>Syntax</b>	<b>4</b>
<b>6</b>	<b>Source Code</b>	<b>4</b>
<b>7</b>	<b>Bibiliography</b>	<b>4</b>

## 1 Introduction

The aim of this project was to implement a combination of two classical games - minesweeper and cricket. The game is built in HTML and CSS, with the interactive functionality added through JavaScript. I used Git as a version control system throughout this project to effectively manage multiple files of code.

The game incorporates the grid generation and cell numbering of minesweeper, with the scoring system and wickets of cricket. The scoring mechanics also provide an incentive to play as fast as possible and maximise the scoring streak. There is a 1 minute timer in which the player has to score maximum runs.

The game ends if either the time runs out, all the non-fielder cells are revealed, or the player gets all-out.

## 2 Overview

The basic minesweeper code was obtained from [GitHub repository](#).

As per the basic requirements, I modified the JS code to not reveal multiple cells together, and fixed the number of fielders (bombs) to 11. I rewrote the bomb placement function, the scoring function, and the click handling functions. I also added more customizable variables in the *components* dictionary to account for the streak, wickets and timer mechanism.

I modified the HTML to add a navbar, scoreboard, game stats and a [confetti](#) effect after completion of the game.

I heavily edited the CSS file, changed fonts and colours, and made the grid cells size dynamically.

I created 2 more pages, one for the **game settings** and one for the **instructions** and their related CSS and JS.

### 3 Basic Logic

The game is implemented using the following ideas:

- Grid Generation - The grid is generated using JS by creating table.
- Bomb placement - Done using *sets* and *Math.random()* so that bombs don't repeat positions.
- Scoring - The streak system is implemented using *time* library, a function checks whether 2 moves were made close enough to maintain the streak and increases the score based on it.
- Start and end functions - Called once, the start functions sets several variables in *components* from the browser **localStorage**. The end functions account for different outcomes.

### 4 Customization

There are several customizations:

- Cell numbering is done via the **minesweeper algorithm**, which makes the game more skill inclined and reduces randomness
- One minute **timer** to complete the game
- **Scoring streaks** motivate the player to play faster as higher streak adds more to the score
- Customizable grid size and number of wickets across different *html* pages using browser **localStorage**
- **Game stats** at the end of game showing the individual player scores
- Sliders and **confetti effect** implemented in JS and styled via CSS
- Added **navbar** and other components

## 5 Syntax

Open the *settings.html* file to set the game parameters.

Then navigate to the *game.html* file to play the game.

## 6 Source Code

The source code of the website can be found in my [GitHub Repo](#) and has also been submitted.

## 7 Bibliography

All the references are cited below, from the *report.bib* file.

□ □ □ □

## References

- *Basic Minesweeper Code*. URL: <https://arghac14.github.io/Minesweeper/>.
- *Confetti effect*. URL: <https://www.cssscript.com/confetti-falling-animation/>.
- *Navbar from W3Schools*. URL: [https://www.w3schools.com/howto/howto\\_js\\_navbar\\_sticky.asp](https://www.w3schools.com/howto/howto_js_navbar_sticky.asp).
- *Sliders from W3Schools*. URL: [https://www.w3schools.com/howto/howto\\_js\\_rangeslider.asp](https://www.w3schools.com/howto/howto_js_rangeslider.asp).