

# MINESWEEPER CRICKET

ARVIND YADAV

14 JUNE, 2023

## Contents

1	Introduction	2
2	Javascripts code	2
3	CSS Part	9
4	Html Part	13
5	Images	16

## 1 Introduction

Minesweeper Cricket is a fun and exciting game that combines the classic gameplay of Minesweeper and Cricket. In this project, I created an interactive user interface using HTML and CSS and incorporated dynamic functionality using JavaScript.

The game is based on a famous Indian cartoon "Chhota Bheem," in which the match will be between Kalia and Bheem.

The game will be played on a grid from 10\*10 to 4\*4 blocks, representing a cricket ground. Eleven fielders will be randomly distributed on the grid, and the player's goal is to click on blocks randomly to score runs while avoiding the fielders. If the player clicks on a block that does not contain a fielder, they score one run and can continue playing till they achieve more than Kalia's score, which is randomly allocated. However, if they click on a block that contains a fielder, the game ends if and their final score is displayed. This project is involved designing and implementing the game logic to create an engaging and enjoyable gaming experience.

## 2 Javascripts code

- I had made a single anonymous function that contains whole other javascript, which itself is called when Windows load

```
window.onload = function () {....}
```

- Next, I introduced a few universal variables and named them which represent their use. The first few variables will be used to play the audio effects. The rest are straightforward, except music\_count and count, which are used to check whether the user clicked the music button or not and to check whether kalia\_score\_determine function is run before passing kalia\_score to thegod() which is responsible for creating the board, respectively.

```
let bheem_audio = new Audio("Chota bheem.mp3");
let witch_audio = new Audio("Chota Bheem - Witch ! Dialogue.mp3");
let kalia_audio = new Audio("Chotta Bheem - Evil Laugh Sound.mp3");
let reveal_audio = new Audio("mixkit-arrow-shot-through-air-2771.wav");
let baller_audio = new Audio("Listion stump sound.mp3");
let background_audio = new Audio("Aylex - Too Hot.mp3");
let hover_audio = new Audio("mixkit-tech-break-fail- 2947.wav");
let click_audio = new Audio("mixkit-plastic-bubble-click-1124.wav")
//for click sound
var size = 10;
let count = 0;check whether kalia\_score\-\determine function is runned
var ballers = 11;
```

```
var kalia_score = 0;
let music_count = 0;//check whether user clicked the music button
```

3. Next I had added an eventlistner to music button, so if its clicked and music\_count is 0 then background\_audio is played in a loop from 3s and music\_count is made 1. Otherwise means background\_audio is already begin played so next part just removes that, and sets music\_count to 0 again so that on next click it again begins.

Next are some addition addeventlisterner for click to get sound effect. Next the function which is anonymously called for any click on start button, first its stores the value of boardsize selected from selction tag whose id is board\_size, then I had checked whether its a valid selection or not, meanwhile in those checkings I had also checked whether a valid difficulty level is selected. The difficulty level basically adjust the kalia\_score by running the function kalia\_score\_determine. The logic of kalia\_score\_determine is simple its picks a random number between two values which depend on size.

The next thing is that kalia\_score is passed as an parameter to thegod(). I used a count because suppose the user doesn't select the difficulty then in that case the kalia\_score will be 0(it was intialised 0), so just to ensure that kalia\_score\_determine has runned and reset the kalia\_score before thegod(), I set count to 1 in kalia\_score\_determine and then added a if condition to clear this. Meanwhile there are few commment that doesn't mean no more but I used them to debug and remake the code. And in need of this function I had removed the button.

```
document.getElementById("music").addEventListener("click", function music_handle()
if (music_count == 0) {
background_audio.currentTime = 3; background_audio.loop = true;
background_audio.play(); music_count++; }
else { background_audio.pause(); background_audio.currentTime = 0;
music_count = 0; });
document.getElementById("board_size").addEventListener("click", function ()
{ click_audio.play(); });
document.getElementById("level.setter").addEventListener("click", function ()
{ click_audio.play(); });
document.getElementById("start").addEventListener("click", function ()
{ click_audio.play(); });
document.getElementById("newgame").addEventListener("click", function ()
{ click_audio.play(); });
document.getElementById("music").addEventListener("click", function ()
{ click_audio.play(); });

document.getElementById("start").onclick = function ()
```

```

{
let w = document.getElementById("board_size")
size = w.options[w.selectedIndex].value;
let level.setter_element = document.getElementById("level.setter");
let difficulty =
level.setter_element.options[level.setter_element.selectedIndex].value;
//gets the selection
//console.log(size);
if (!(isNaN(size)) && !(isNaN(difficulty))) {//verifies the validity of selection
    document.getElementById("test").style.display = "none"; count = 0;
    kalia_score_determine();
}
if (isNaN(size) || isNaN(difficulty)) {//verifies the validity of selection
    let x = document.getElementById("test");
    x.style.color = "#bf3030";
    x.innerHTML = "abe select to kar";
    //count++;
}
function kalia_score_determine() {//set kalia's score
    //if (!(isNaN(difficulty))) {
    document.getElementById("test").style.display = "none"; count = 0;
    let logitics_factor_2 = 0.175;
    let logitics_factor_0 = 0.115;
    let logitics_factor_1 = 0.235;
    let logitics_factor_3 = 0.375;
    let max_score = size * size - ballers;
    count++;
    if (difficulty == 0) {
        kalia_score = max_score * logitics_factor_0;
        kalia_score = parseInt(Math.random() * kalia_score);
    }
    if (difficulty == 1) {
        kalia_score = max_score * logitics_factor_2;
        kalia_score =
        parseInt(Math.random() * ((max_score * logitics_factor_1)
        - kalia_score) + kalia_score);
    }
    if (difficulty == 2) {
        kalia_score = max_score * logitics_factor_3;
        kalia_score =
        parseInt(Math.random() * (kalia_score - (max_score * 0.285)) + max_score * 0.285);
    }
    //}
    /*if (isNaN(difficulty)) {
        let x = document.getElementById("test");
        x.style.color = "#bf3030";
    }
}

```

```

        x.innerHTML = "abe select to kar";
    }*/
}
//document.getElementById("test").innerHTML=w.options[w.selectedIndex].value;
if (count == 1) {
    thegod(kalia_score);
    //background_audio.pause();
    //background_audio.currentTime = 0;
    document.getElementById("start").style.display = "none";
}
};

```

4. Next, is the thegod() function which sets the boards, size and backgrounds etc.

```
function thegod(kalia_score){....}
```

5. In this part I had first initialised bheem\_score to 0, then prints the kalia's scores. And made a kind global variable board for the thegod() function which is basically the table tag. Next is a variable showallballer\_called which remained used for hover thing(see later). Anyways, next part of code generates the board by using createElement(), appendChild, and introduced to new attributes basically the number and column of a block. Next thing I added a new attribute to baller\_block which basically whethers its baller or not.

```

var bheem_score = 0;
document.getElementById("score2").innerHTML = kalia_score;
let board = document.getElementById("board");
var showallballer_called = 1;
//this will check if showallballer function invoked or not
//making the board
for (let i = 0; i < size; i++) {
    let row = document.createElement('tr')
    for (let j = 0; j < size; j++) {
        let block = document.createElement('td')
        block.dataset.row = i;
        block.dataset.col = j;
        block.dataset.isballer = 0;
        row.appendChild(block);
    }
    board.appendChild(row);
}

```

6. Next I placed the ballers on the board randomly, placing a counter for number ballers placed, to ensure that there 11 ballers. Here the main problem was that like some i,j comes to be one previous value (due floor function) than we will placing at same positions which results in decrement in number of ballers, so check if the i,j block is baller already or not. Again there some comment that mean anything it was just for testing marked\*.

```
//now lessgo to fix the ballers positions
//function baller_place(){ /*to fix baller error
var baller_palced = 0;
while (baller_palced < ballers) {
    let i = Math.floor(Math.random() * size);
    let j = Math.floor(Math.random() * size);
    let placementblock = board.querySelector(`[data-row="${i}"] [data-col="${j}"]`);
    let x = parseInt(placementblock.getAttribute('data-isballer'));
    if (x === 0) { //if the i,j block is baller already or not.
        placementblock.setAttribute('data-isballer', 1);
        baller_palced++;
    }
}
/*}
```

7. Next is showallballer() function that a parameter, which is called when either bheem's score & kalia's score or user hits one of the baller. I used querySelector to get the first element which is baller and then set isballer to 2 so that it doesn't get selected in next loop and change the background to cricketball. Now depending on x being 0,1,2 means loss, draw, win and changed the background and text in element with id "teams" which basically the lower heading. In win case two randomly backgrounds are set, by generating a either 0 or 1 randomly. And end of each case it removes the evenlistener from the table tag. I had comment some styling that wasn't good when applied (not in sense of bug).

```
function showallballer(x) {
for (let i = 0; i < ballers; i++) {
    let baller_block = board.querySelector(`[data-isballer="1"]`);
    baller_block.style.backgroundSize = "contain";
    baller_block.style.backgroundImage = "url('./cricketball.svg')";
    baller_block.setAttribute('data-isballer', 2);
    showallballer_called = 0;
}
if (x == 0) { //lost
    document.getElementById("thegod").style.backgroundImage = "url('series_8.jpg')";
    board.removeEventListener("click", reveal);
```

```

        document.getElementById("teams").innerHTML =
        "Kalia Won, Better Luck Next Time!"
        // *kalia_audio.play();
    }
    if (x == 1) { //win
        document.getElementById("thegod").style.backgroundSize = "cover";
        let bg_rnd = Math.random() * 2;
        if (bg_rnd == 0) document.getElementById("thegod").style.backgroundImage =
        "url('series_6.jpg')";
        else document.getElementById("thegod").style.backgroundImage =
        "url('Bheem_the_fighter.jpg')";
        //document.getElementById("teams").style.color = 'black';
        board.removeEventListener("click", reveal);
        document.getElementById("teams").innerHTML = "Bheem Won, Nice Job!"
        bheem_audio.play();
        if (music_count == 1) background_audio.pause();
    }
    if (x == 2) { //draw
        document.getElementById("thegod").style.backgroundImage =
        "url('wp9141271-mighty-little-bheem-wallpapers.jpg')";
        board.removeEventListener("click", reveal);
        document.getElementById("teams").innerHTML = "Ahh!, So Close, Nvm Nice Try!";
        //document.getElementById("heading").style.color = 'rgb(51,79,131)';
    }
    showballaller_called=0;
}

```

8. Here I had add an eventlistener for click on the table with id board. Which call the reveal() function, whose going to take first get the element where the clicking had happened and checks whether it was baller or not and if its not then it changes the background, increase bheem's score and set the isballer to -1 so that if it is clicked again the background won't change. Then it compares the bheem's score and if bheem's score is greater than kalia's score then it calls showballaller() for the draw case by setting its parameter as 1. Also a audio is played when it reveal non-baller. In opposite case it first checks whether it was a draw or win and accordingly adjust the showballaller() call and audio's.

```

function reveal(event) {
let block_clicked = event.target;
let x = parseInt(block_clicked.getAttribute('data-isballer'));
//if(bhim_scor<kalia_score){
if (x === 0) {
    block_clicked.style.backgroundSize = "contain";
    block_clicked.style.backgroundImage = "url('./otherone.png')";
}
}

```

```

        bheem_score++;
        document.getElementById("score1").innerHTML = bheem_score;
        block_clicked.setAttribute('data-isballer', -1);
        reveal_audio.play();
        if (bheem_score > kalia_score) {
            showallballer(1);
        }
    }
//*/
//*else
else if (x != -1) {
    if (bheem_score == kalia_score) {
        showallballer(2);
        baller_audio.currentTime = 0.1;
        if (music_count == 1) background_audio.pause();
        baller_audio.play();
        setTimeout(function () {
            witch_audio.play();
        }, 1000);
    }
    else {
        showallballer(0);
        baller_audio.currentTime = 0.1;
        if (music_count == 1) background_audio.pause();
        baller_audio.play();
        setTimeout(function () {
            kalia_audio.play();
        }, 1000);
    }
}
board.addEventListener("click", reveal); //click calls reveal function

```

9. This is just the styling part hence its name. It first sets a background, and sets the clicked variable to true when the block it clicked and then it applies mouseout and mouseover functions, in which it checks where it was clicked or revealed(a baller), its worthy to note that a baller getting revealed when showallballer\_called is 0. And accordingly it adjust the background images and border colour to give a hovering effect. And end I just called it.

```

function just_for_style() {
    for (let i = 0; i < size; i++) {
        for (let j = 0; j < size; j++) {
            let style_block = board.querySelector(`[data-row="${i}"] [data-col="${j}"]`);

```

```

style_block.style.backgroundSize = "contain";
style_block.style.backgroundImage =
"url('./cricket-ground.png')");
let x = parseInt(style_block.getAttribute('data-isballer'));
/*baller fixing error
let clicked = false;
style_block.onclick = function () {
    clicked = true;
}
style_block.onmouseover = function () {
    hover_audio.play();
    if (!clicked && showallballer == 1) {
        style_block.style.backgroundImage = "url('./cricket-ground_hover.png')";
    }
    style_block.style.borderColor = 'rgb(83, 94, 65)';
}
style_block.onmouseout = function () {
    if (!clicked && showallballer == 1) {
        style_block.style.backgroundImage = "url('./cricket-ground.png')";
    }
    style_block.style.borderColor = 'rgb(99, 173, 75)';
}
}
}
}
just_for_style();

```

### 3 CSS Part

1. This part of simple, I had first reduced the margin to 0 for complete document. Then I had adjusted the table tag and made its position absolut. Then resized the cells in table tag. Then fixed the height of a container with id bg so that % value get applied to others, then I had added styles to headings, body tag(id=thegod which I used in js) then added styles to score board and all.

```

* {
margin: 0px;
}

#board {
border-collapse: collapse;
background-color: rgb(61, 131, 40);
/*background-image: url("ffb57n.jpg");*/

```

```
border: 1px;
position: absolute;
top: 47%;
left: 35%;
}

td {
width: 30px;
height: 30px;
text-align: center;
border: 6px solid rgb(99, 173, 75);
}

#bg {
height: 60px;
}

#heading_bg {
/*background-color: rgba(2, 95, 110,0);*/
border: 4px solid #bf3030;
background-color: #bf3030;
border-top-left-radius: 40%;
border-top-right-radius: 40%;
border-bottom-left-radius: 15%;
border-bottom-right-radius: 15%;
display: inline-block;
height: 180%;
width: 100%;
text-align: center;
}

#heading {
/* font-family: 'League Script', cursive;*/
/*font-family: 'Abril Fatface', cursive;*/
/*font-family: 'Anton', sans-serif;*/
font-family: 'Playball', cursive;
/*font-family: 'Ultra', serif;*/
font-size: 300%;
font-weight: 900;
color: rgb(213, 154, 71);
}

#teams {
font-family: 'Clicker Script', cursive;
font-size: 250%;
```

```
color: rgb(254, 255, 253);
font-weight: 900;
padding-left: 4.5%;
}

body {
    background-size: cover;
    background-image: url("series_2018_03.jpg");
}

.dashboard {
    text-align: center;
    word-spacing: 100%;
    font-family: 'Freehand', cursive;
    font-size: 200%;
    margin-top: 4%;
}

#bheem_d,
#kalia_d {
    border: 5px solid rgb(107, 88, 60);
    /*here comes responsiveness*/
    background-color: rgba(107, 88, 60, 1);
    clip-path: polygon(1% 0%, 16% 0%, 53% 0%, 73% 0%, 84% 5%, 89% 9%, 93% 14%, 96% 19%, 99% 26%, 99% 34%, 100% 40%, 100% 48%, 100% 56%, 100% 63%, 100% 70%, 100% 75%, 91% 76%, 63% 75%, 52% 76%, 39% 75%, 24% 75%, 11% 75%, 3% 75%, 0% 74%);
    color: rgb(225, 215, 202);
    padding-bottom: 1%;
    padding-left: 0.2%;
}

#score1 {
    margin-right: 33%;
}

.score {
    background-color: rgb(111, 61, 25);
    border: 1px solid rgb(64, 120, 200);
    border-radius: 25%;
    text-decoration: none;
    color: #fff;
    padding-left: 1%;
}

#score2 {
```

```
padding-right: 2%;  
}
```

2. In this part I had styled the buttons, and selection tag. Styling these had a trick like the browser(safari) wasn't allowing my styles so I used webkitappearane none. Rest of part is just styling.

```
.aside_tools {  
    font-family: 'Ultra', serif;  
    font-size: 140%;  
    position: relative;  
    top: 350%;  
    left: 1%;  
    color: rgb(254, 254, 254);  
}  
  
#board_size,  
#level_setter {  
    -webkit-appearance: none;  
    width: 20%;  
    height: 10%;  
    border: 5px solid rgb(107, 88, 60);  
    /*here comes responsiveness*/  
    background-color: rgba(107, 88, 60, 1);  
    color: rgb(225, 215, 202);  
    font-family: 'League Script', cursive;  
    ;  
    font-size: 90%;  
    font-weight: 900;  
    border-top-right-radius: 50%;  
}  
  
#board_size:hover {  
    background-color: rgb(81, 66, 42);  
    cursor: pointer;  
}  
  
#newgame,  
#start {  
    -webkit-appearance: none;  
    padding: 0;  
    height: 400%;  
    position: relative;
```

```
    left: 1%;  
    width: 15%;  
    font-size: 30%;  
    border: 4px solid #bf3030;  
    background-color: #bf3030;  
    color: #fff;  
    border-top-left-radius: 40%;  
    border-top-right-radius: 40%;  
}  
  
#music {  
    -webkit-appearance: none;  
    padding: 0;  
    height: 400%;  
    position: relative;  
    left: 1%;  
    width: 9%;  
    font-size: 30%;  
    border: 4px solid #306ebf;  
    background-color: #305bbf;  
    color: #fff;  
    border-radius: 25%;  
}  
  
#music:hover {  
    background-color: #244491;  
    cursor: pointer;  
}  
  
#newgame:hover,  
#start:hover {  
    background-color: #902525;  
    cursor: pointer;  
}
```

## 4 Html Part

1. The head tag had some default thing we get use in VS code, then I had linked some fontstyle from google font and lastly I had also my js and css file.

```
<!DOCTYPE html>  
<html lang="en">
```

```

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Pooject</title>
  <link rel="stylesheet" href="style.css">
  <script src="script.js"></script>
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=League+Script&display=swap" rel="stylesheet">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Clicker+Script&display=swap" rel="stylesheet">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Playball&display=swap" rel="stylesheet">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Abrib+Fatface&family=Anton&family=Playball&family=Ultra&display=swap" rel="stylesheet">
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Freehand&display=swap" rel="stylesheet">
</head>

```

2. It had all the tags and element that I had used except those that are created using js.I had made a container with id bg which contains all others, it was because it to use % value in css I must have a container with fixed value but I can't fix the height of body tag since it was to have the background image so I just fixed the height of this guy.And same goes to making heading\_bg like I didn't want to the background to infere to the text, also it provided me more flexiblity like I had added two div inside, one of the being "teams", and that I used to print the result.Next part of code starightfoward like I just adjusted the html part accroding to css and js part.

```

<body id="thegod">
  <div id="bg">

```

```
<div id="heading_bg">
  <div id="heading">
    Dholakpur Cricket Championship
  </div>
  <div id="teams">
    Kalia v/s Chhota Bheem
  </div>
</div>
<div class="dashboard">
  <span id="bheem_d">
    Bheem's score:</span>
  <span id="score1" class="score"> 0
  </span><span id="kalia_d">
    Kalia's Score:</span>
  <span id="score2" class="score">0
  </span>
</div>
<aside class="aside_tools">
  <span>Boardsize:</span>
  <select id="board_size">
    <option value="select">
      Select size of board
    </option>
    <option value="10">
      10*10
    </option>
    <option value="9">
      9*9
    </option>
    <option value="8">
      8*8
    </option>
    <option value="7">
      7*7
    </option>
    <option value="6">
      6*6
    </option>
    <option value="5">
      5*5
    </option>
    <option value="4">
      4*4
    </option>
  </select>
  <br>
```

```
<br>
<span>Difficulty:</span>
<select id="level.setter">
    <option value="select">
        Select Challenge Level
    </option>
    <option value="0">
        Low
    </option>
    <option value="1">
        Normative
    </option>
    <option value="2">
        High
    </option>
</select>
<br>
<div style="height: 10px;font-size: 70px;">
    <button id="start">Start</button>
    <button id="music">Music</button>
    <p id="test"></p>
    <form>
        <input type="submit" id="newgame" value="RESET">
    </form>
</div>
</aside>
<table id="board"></table>
</div>
</body>
</html>
```

## 5 Images

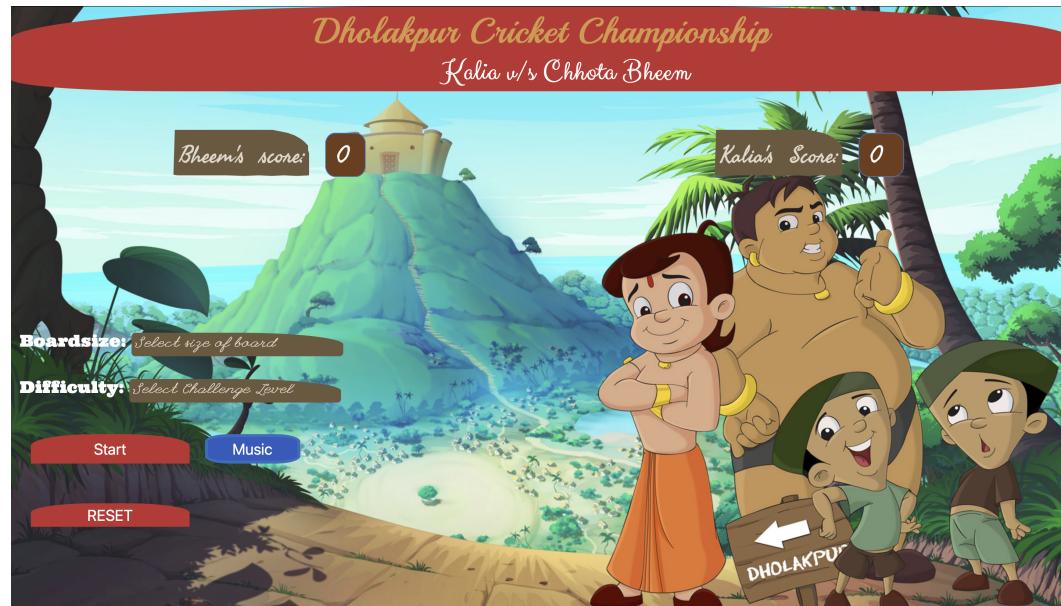


Figure 1: When User Open the webpage

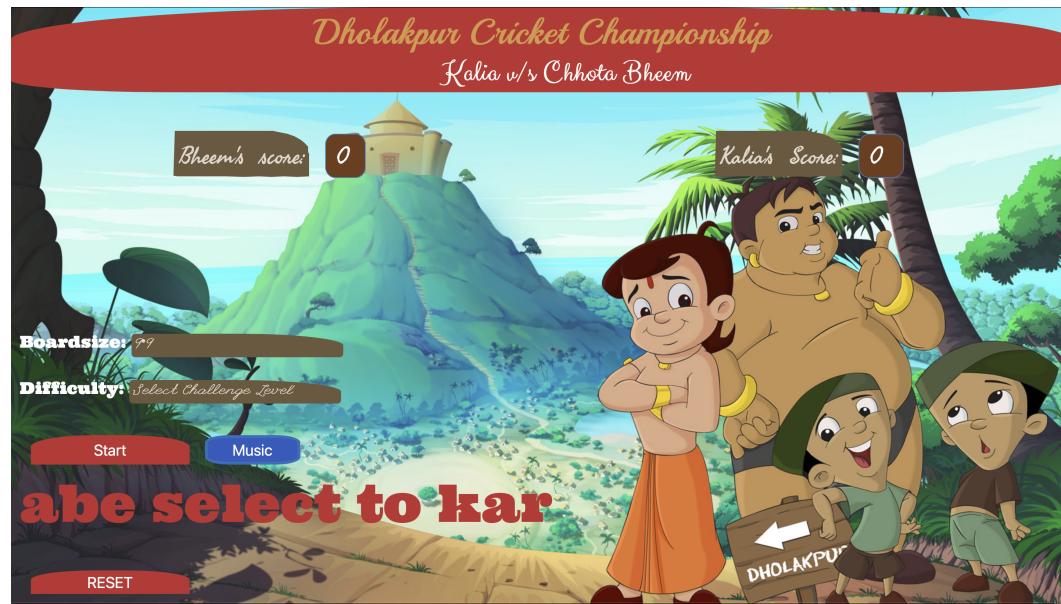


Figure 2: Validation of clicking



Figure 3: When User clicks start button with valid selection

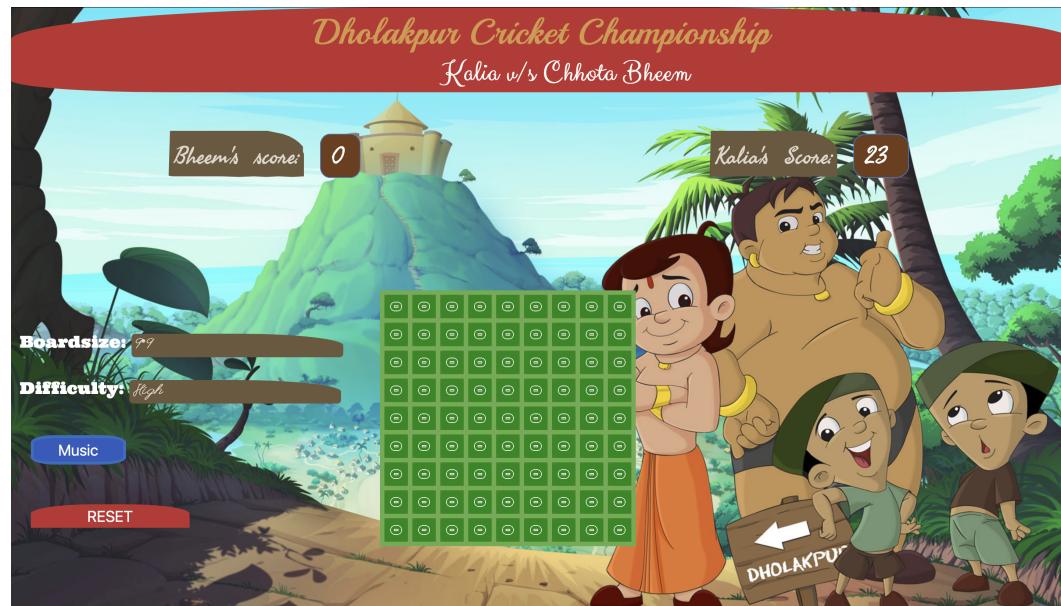


Figure 4: A high setting

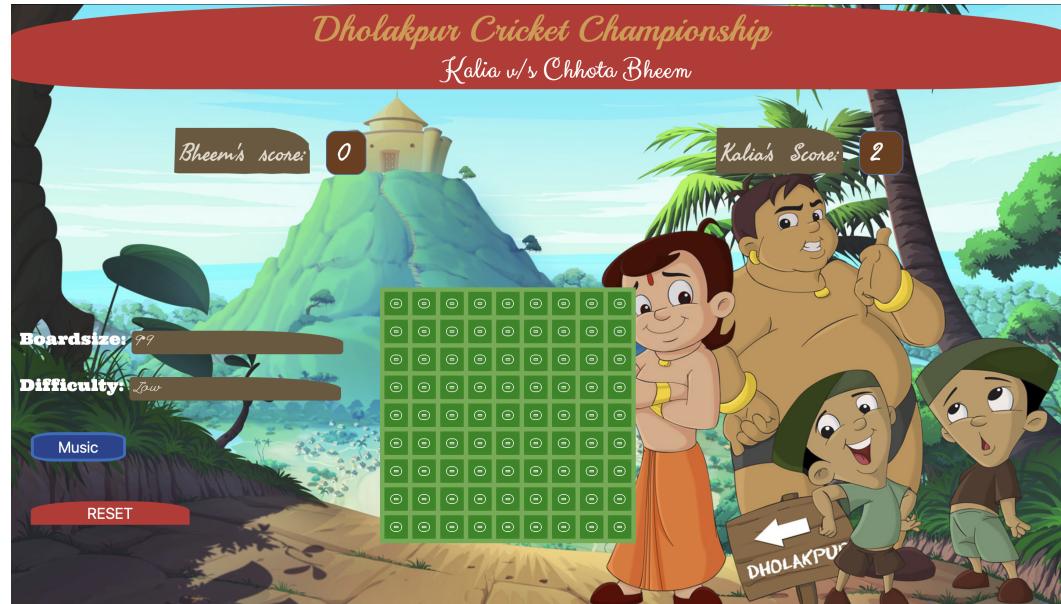


Figure 5: A low setting



Figure 6: A win

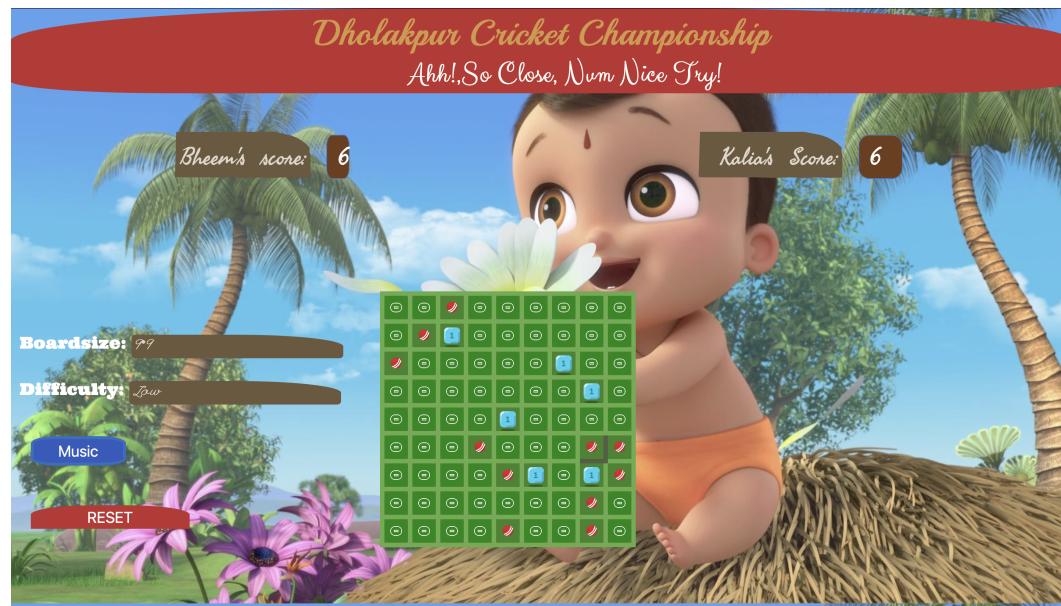


Figure 7: A draw



Figure 8: A loss