- Apk Ular Tangga Kuis Master
- Index Html

```
<!DOCTYPE html>
<html lang="en">
   <head>
      <meta charset="UTF-8">
      <link rel="stylesheet" href="./assets/styles/style.css">
      <link rel="stylesheet" href="./assets/styles/dice.css">
      <link rel="stylesheet" href="./assets/styles/modal.css">
      <title>UGALO</title>
      <script>
         window.console = window.console || function(t) {};
      </script>
      <script>
         if (document.location.search.match(/type=embed/gi)) {
            window.parent.postMessage("resize", "*");
         }
      </script>
   </head>
   <body translate="no">

      <div id="board">
      </div>

      <div class="player-wrapper">
         <div class="player-list">
         </div>
         <button type="button" class="button add-player yellow">
            Tambah Pemain
         </button>
         <button type="button" class="button start-game lightGreen">
            Mulai
         </button>
         <button type="button" class="button reset-game amber">
            Reset
         </button>
         <button data-action="modal" data-target="#peraturan-modal" type="button"
class="button lightBlue">
            Peraturan
         </button>
         <button type="button" id="random-dice" class="button blueGrey" style="color:
white;">
            Roll Dadu
         </button>

         <div class="dice-wrapper">
```

```
<div id="background"></div>
<div id="wrapper">
  <input id="secondroll" name="roll" type="checkbox">
  <input id="roll" name="roll" type="checkbox">
  <!-- <label for="roll">Roll it!</label>
  <label for="secondroll"><span>Stop!</span></label> -->
  <div id="platform">
    <div id="dice">
    <div class="side front">
      <div class="dot center"></div>
    </div>
    <div class="side front inner"></div>
    <div class="side top">
      <div class="dot dtop dleft"></div>
      <div class="dot dbottom dright"></div>
    </div>
    <div class="side top inner"></div>
    <div class="side right">
      <div class="dot dtop dleft"></div>
      <div class="dot center"></div>
      <div class="dot dbottom dright"></div>
    </div>
    <div class="side right inner"></div>
    <div class="side left">
      <div class="dot dtop dleft"></div>
      <div class="dot dtop dright"></div>
      <div class="dot dbottom dleft"></div>
      <div class="dot dbottom dright"></div>
    </div>
    <div class="side left inner"></div>
    <div class="side bottom">
      <div class="dot center"></div>
      <div class="dot dtop dleft"></div>
      <div class="dot dtop dright"></div>
      <div class="dot dbottom dleft"></div>
      <div class="dot dbottom dright"></div>
    </div>
    <div class="side bottom inner"></div>
    <div class="side back">
      <div class="dot dtop dleft"></div>
      <div class="dot dtop dright"></div>
      <div class="dot dbottom dleft"></div>
      <div class="dot dbottom dright"></div>
      <div class="dot center dleft"></div>
      <div class="dot center dright"></div>
    </div>
    <div class="side back inner"></div>
```

```
                <div class="side cover x"></div>
                <div class="side cover y"></div>
                <div class="side cover z"></div>
                </div>
            </div><!-- .platform -->
        </div><!--.wrapper -->

        <p id="dice-result">0</p>
      </div>
    </div>

    <div id="finish">
      <div></div>
    </div>

    <div id="peraturan-modal" class="modal-window">
      <span>
        <a data-action="close" title="Close" class="modal-close">Tutup</a>
        <h1>Peraturan</h1>
        <ol>
          <li>Permainan dimainkan oleh 2 orang atau lebih.</li>
          <li>Permain yang mendapatkan dadu 6 dapat mengeluarkan pion dan
dilanjutkan melempar dadu kembali.</li>
          <li>Setiap pemain akan mendapatkan 1 soal ketika menempati tiap
kotak.</li>
          <li>1 kotak terdapat 3 soal yang berbeda.</li>
          <li>Kotak berjumlah 50.</li>
          <li>Waktu untuk menyelesaikan soal yaitu 5 menit.</li>
          <li>Jika pemain dapat menyelesaikan soal, pemain tersebut berhak
mendapat 5 point dan jika pemain tidak dapat menyelesaikan soal, maka pemain tidak
mendapat point.</li>
          <li>Permainan dikatakan selesai jika salah satu pemain telah
menyelesaikan 50 kotak.</li>
        </ol>
      </span>
    </div>

    <div id="question-modal" class="modal-window">
      <span>
        <a data-action="close" title="Close" class="modal-close">Tutup</a>
      </span>
      <div class="question">

      </div>
    </div>

    <script src="./assets/js/jquery.js"></script>
    <script src="./assets/js/app.js"></script>
```

```
    <!-- https://www.1001fonts.com/digital-fonts.html -->
    <!-- <div>Icons made by <a href="https://www.flaticon.com/authors/nikita-golubev" title="Nikita Golubev">Nikita Golubev</a> from <a href="https://www.flaticon.com/" title="Flaticon">www.flaticon.com</a> is licensed by <a href="http://creativecommons.org/licenses/by/3.0/" title="Creative Commons BY 3.0" target="_blank">CC 3.0 BY</a></div> -->
    <!-- https://codepen.io/tameraydin/pen/CADvB -->
    <!-- http://soundbible.com/181-Roll-Dice-2.html -->
    <!-- https://www.genengnews.com/wp-content/uploads/2018/10/Feb1_2018_GEtty_489973431_SnakesAndLaddrs7098137751.jpg -->
  </body>
</html>
```

- Main.js

```
'use strict';

(function($) {
  $.fn.removeClassWild = function(mask) {
    return this.removeClass(function(index, cls) {
      var re = mask.replace(/\*/g, '\\S+');
      return (cls.match(new RegExp('\\b' + re + '', 'g')) || []).join(' ');
    });
  };
})(jQuery);

const QUESTION_PER_BLOCK = 3;
var isQuestionModalOpen = false;
var player_should_point = 0;
var dice_audio = new Audio('./assets/sfx/roll_dice.mp3');
var dice_a = 0;
var dice_b = 0;
var dice = 0;
const color = [
  'red','blue','purple','deepPurple','indigo','pink','lightBlue','cyan','teal','green','lightGreen','lime',
  'yellow','amber','orange','deepOrange','brow','grey','blueGrey'
];
const ficha = [
  {
    position: -1,
    point: 0,
    isFinish: false,
  }
```

```
];
var turn = 0;
var gameIsFinish = false;
const snakes = {
   snake_16: 4,
   snake_29: 10,
   snake_39: 20,
   snake_45: 34,
};
const laders = {
   lader_6: 14,
   lader_17: 23,
   lader_27: 33,
   lader_38: 43,
}

// loadQuestion();
updatePlayer();
$('#random-dice').attr('disabled', true);
$('.reset-game').attr('disabled', true);

dice_audio.onended = function() {
   $('#roll').attr('checked', false);
   $('#dice-result').text(dice_a + dice_b);

   if(ficha[turn].position === -1) {
      if((dice_a + dice_b) === 6) {
         ficha[turn].position = 0;
         updatePosition(ficha[turn]);
         $('#random-dice').attr('disabled', false);
      } else {
         updateTurn(1);
         $('#random-dice').attr('disabled', false);
      }
   } else {
      randomDiceAndMoveFiche(dice_a, dice_b);
   }
}

$('#random-dice').click(function() {
   $('#roll').attr('checked', true);
   $(this).attr('disabled', true);
   dice_audio.play();
   dice_a = Math.ceil(Math.random() * 6);
```

```
    dice_b = 0;//Math.ceil(Math.random() * 6);
});

$('.add-player').click(function() {
    var player = {
        position: -1,
        point: 0,
        isFinish: false,
    }

    ficha.push(player);
    updatePlayer();

    if(ficha.length >= 3) {
        $('button.add-player').attr('disabled', true);
    }
});

$('.start-game').click(function() {
    $('#random-dice').attr('disabled', false);
    $('.reset-game').attr('disabled', false);
    $('.add-player').attr('disabled', true);
    $(this).attr('disabled', true);
});

$('.reset-game').click(function() {
    window.localStorage.clear();
    document.location.reload();
});

function randomDiceAndMoveFiche(a = 0, b = 0) {
    dice = a + b;
    var move = 1;

    var anim = setInterval(function() {
        ficha[turn].position += 1;
        updatePosition(ficha[turn]);

        if(ficha[turn].position >= 50 && move !== dice) {
            clearInterval(anim);
            rewindPosition(dice - move);
            return;
        }
```

```
        if(move === dice) {
            $('#random-dice').attr('disabled', false);
            var isSnake = snakes['snake_'+ficha[turn].position];
            if (isSnake) {
                ficha[turn].position = isSnake;
            }

            var isLaders = laders['lader_'+ficha[turn].position];
            if (isLaders) {
                ficha[turn].position = isLaders;
            }

            updatePosition(ficha[turn]);

            if(ficha[turn].position === 50) {
                ficha[turn].isFinish = true;
                alert('Player-'+ turn +' winner with point: '+ ficha[turn].point);
                updateTurn(1);
                clearInterval(anim);
                return;
            }

            player_should_point = turn;
            openQuestionModal(ficha[turn]);

            if(dice !== 6) {
                updateTurn(1);
            }

            clearInterval(anim);
        }

        move++;
    }, 250);
}

$('[data-action=modal]').click(function() {
    var target = $(this).data('target');
    window.localStorage.setItem('opened_modal', target);
    $(target).addClass('open');
});

$('[data-action=close]').click(function() {
    var target = window.localStorage.getItem('opened_modal');
```

```
    $(target).removeClass('open');
    window.localStorage.removeItem('opened_modal');
    if(isQuestionModalOpen) {
        var point = prompt('Point yang didapat player:');
        point = parseInt(point);
        ficha[player_should_point].point += point;
        updatePlayer(false);
        isQuestionModalOpen = false;
    }
});

function updatePlayer(isUpdatePosition = true) {
    $('.player-list').empty();
    $('#board').empty();
    ficha.map(function(player, index) {
        var wrapper =   '<div class="player '+ color[index] +' '+ (index == turn ? 'active' :
'') +'">'+
                        '<p>'+
                            'Player-'+ (index + 1) +':'+
                            '<span class="point">'+ player.point +'</span>'+
                        '</p>'+
                    '</div>';
        $('#board').append('<div id="player-'+index+'" class="ficha '+ color[index] +'
position-'+player.position+'"></div>');
        $('.player-list').append(wrapper);

        if(isUpdatePosition) {
            updatePosition(player);
        }
    });
}

function updatePosition(ficha_player = null) {
    $('#player-'+turn).removeClassWild('position-*');
    $('#player-'+turn).addClass('position-'+ (ficha_player.position));
}

function rewindPosition(block = 0) {
    var anim_rewind = setInterval(function() {
        ficha[turn].position -= 1;
        updatePosition(ficha[turn]);

        if(block === 1) {
            $('#random-dice').attr('disabled', false);
```

```
            var isSnake = snakes['snake_'+ficha[turn].position];
            if (isSnake) {
               ficha[turn].position = isSnake;
            }

            var isLaders = laders['lader_'+ficha[turn].position];
            if (isLaders) {
               ficha[turn].position = isLaders;
            }

            updatePosition(ficha[turn]);

            if(dice !== 6) {
               updateTurn(1);
            }

            clearInterval(anim_rewind);
         }

         block--;
      }, 250);
}

function updateTurn(number, max = 0) {
   turn = number === 0 ? 0 : (turn + number);

   if(max === 3) {
      gameIsFinish = true;
      $('#random-dice').attr('disabled', true);
      alert('game finished');
      return;
   }

   if(number === 0)
      return;

   if (turn === ficha.length) {
      updateTurn(0, max++);
   }

   if(ficha[turn].isFinish) {
      updateTurn(1, max++);
   }
```

```
    $('.player.active').removeClass('active');
    var player_list = document.getElementsByClassName('player');
    $(player_list[turn]).addClass('active');

}

function loadQuestion(callback) {
    var xobj = new XMLHttpRequest();
    xobj.overrideMimeType("application/json");
    xobj.open('GET', './assets/question/question.json', true); // Replace 'my_data' with
the path to your file
    xobj.onreadystatechange = function () {
          if (xobj.readyState == 4 && xobj.status == "200") {
            // Required use of an anonymous callback as .open will NOT return a value but
simply returns undefined in asynchronous mode
            window.localStorage.setItem('question', xobj.responseText);
          }
    };
    xobj.send(null);
}

function openQuestionModal(player) {
    if(player.position === 0) {
        return;
    }
    window.localStorage.setItem('opened_modal', '#question-modal');
    var number_question = window.localStorage.getItem('question_'+ player.position);
    if(!number_question) {
        window.localStorage.setItem('question_'+ player.position, '1');
        number_question = 1;
    } else {
        number_question = parseInt(number_question);
        if (number_question >= QUESTION_PER_BLOCK) {
            window.localStorage.setItem('question_'+ player.position, '1');
        } else {
            window.localStorage.setItem('question_'+ player.position, (number_question +
1));
            number_question += 1;
        }
    }

    $('#question-modal').addClass('open');
    $('.question').css('background-image', 'url(./assets/question/'+ player.position +'_'+
number_question +'.png)');
```

```
    isQuestionModalOpen = true;
}
```

- Style css

```
@charset "UTF-8";
body {
  overflow-x: hidden;
  overflow-y: hidden;
  font: 500 14px/1.5 -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto,
Oxygen-Sans, Ubuntu, Cantarell, "Helvetica Neue", sans-serif;
}
@font-face {
  font-family: digital7;
  src: url(../font/digital-7/digital-7.ttf);
}
/***** Player */
.player-wrapper {
    height: auto;
    width: 225px;
    float: right;
    border: 1px solid #212121;
    padding: 5px;
}

.player-list {
    width: 97%;
    float: left;
}

.player {
    width: 99%;
    height: 30px;
    padding: 5px;
    margin-bottom: 7px;
    border-radius: 3px;
}

.player > p {
    margin: 0px;
    padding-top: 6px;
}

.player.active {
```

```
  background-image: url(../img/turn_dice.png);
  background-repeat: no-repeat;
  background-position: center;
  background-position-x: 190px;
}

span.point {
    margin-left: 15px;
}

/**** Dice */
#dice-result {
  padding-top: 100px;
  text-align: center;
  font-family: digital7;
  font-size: 80px;
}
.dice-wrapper {
  height: 180px;
  width: 212px;
  /* float: right; */
  border: 1px solid #212121;
  padding: 5px;
}

/**** Player color */
.red, #board .ficha.red::before, #board .ficha.red::after {
    background-color: #f44336;
}

.pink, #board .ficha.pink::before, #board .ficha.pink::after {
    background-color: #e91e63;
}

.purple, #board .ficha.purple::before, #board .ficha.purple::after {
    background-color: #9c27b0;
}

.deepPurple, #board .ficha.deepPurple::before, #board .ficha.deepPurple::after {
    background-color: #673ab7;
}

.indigo, #board .ficha.indigo::before, #board .ficha.indigo::after {
    background-color: #3f51b5;
```

```
}

.blue, #board .ficha.blue::before, #board .ficha.blue::after {
  background-color: #2196f3;
}

.lightBlue, #board .ficha.lightBlue::before, #board .ficha.lightBlue::after {
  background-color: #03a9f4;
}

.cyan, #board .ficha.cyan::before, #board .ficha.cyan::after {
  background-color: #00bcd4;
}

.teal, #board .ficha.teal::before, #board .ficha.teal::after {
  background-color: #009688;
}

.green, #board .ficha.green::before, #board .ficha.green::after {
  background-color: #4caf50;
}

.lightGreen, #board .ficha.lightGreen::before, #board .ficha.lightGreen::after {
  background-color: #8bc34a;
}

.lime, #board .ficha.lime::before, #board .ficha.lime::after {
  background-color: #cddc39;
}

.yellow, #board .ficha.yellow::before, #board .ficha.yellow::after {
  background-color: #ffeb3b;
}

.amber, #board .ficha.amber::before, #board .ficha.amber::after {
  background-color: #ffc107;
}

.orange, #board .ficha.orange::before, #board .ficha.orange::after {
  background-color: #ff9800;
}

.deepOrange, #board .ficha.deepOrange::before, #board .ficha.deepOrange::after {
  background-color: #ff5722;
```

```
}

.brown, #board .ficha.brown::before, #board .ficha.brown::after {
   background-color: #795548;
}

.grey, #board .ficha.grey::before, #board .ficha.grey::after {
   background-color: #9e9e9e;
}

.blueGrey, #board .ficha.blueGrey::before, #board .ficha.blueGrey::after {
   background-color: #607d8b;
}

/***** Control of labels and inputs */
.button {
  margin: 3px auto;
  width: 100%;
  padding: 7px;
  border-radius: 3px;
}

input, label {
  display: none;
  user-select: none;
}

#board {
  position: absolute;
  overflow: visible;
  width: 950px;
  height: 600px;
  background: url(../img/game_bg2.jpg);
  background-size: cover;
  border: 3px solid black;
  left: 70px;
  top: 10px;
}
#board .ficha {
  position: absolute;
  width: 6px;
  height: 40px;
  overflow: visible;
  bottom: 0;
```

```
  left: -30px;
  transition: all 0.5s;
  transform: translate(25px, -11px);
  -webkit-transform: translate(25px, -11px);
  border: 2px solid black;
  z-index: 3;
}
#board .ficha::before, #board .ficha::after {
  content: "";
  display: block;
  width: 20px;
  height: 20px;
  position: absolute;
  top: -4px;
  left: -7px;
  border-radius: 100%;
  border: 2px solid black;
  box-sizing: border-box;
  border-bottom: 0;
}
#board .ficha::before {
  top: 30px;
  width: 40px;
  left: -17px;
  border-bottom: 2px solid black;
  border-top: 1px solid black;
}

/***** Piece position for each numbered tile. TODO: reduce using SCSS :P */
#board > .ficha.position-0 {
    left: 035px;
    bottom: 062px;
  }

#board > .ficha.position-1 {
  left: 170px;
  bottom: 047px;
}

#board > .ficha.position-2 {
  left: 247px;
  bottom: 040px;
}
```

```
#board > .ficha.position-3 {
  left: 327px;
  bottom: 030px;
}

#board > .ficha.position-4 {
  left: 410px;
  bottom: 020px;
}

#board > .ficha.position-5 {
  left: 485px;
  bottom: 015px;
}

#board > .ficha.position-6 {
  left: 570px;
  bottom: 015px;
}

#board > .ficha.position-7 {
  left: 647px;
  bottom: 018px;
}

#board > .ficha.position-8 {
  left: 727px;
  bottom: 027px;
}

#board > .ficha.position-9 {
  left: 800px;
  bottom: 047px;
}

#board > .ficha.position-10 {
  left: 845px;
  bottom: 115px;
}

#board > .ficha.position-11 {
  left: 778px;
  bottom: 160px;
}
```

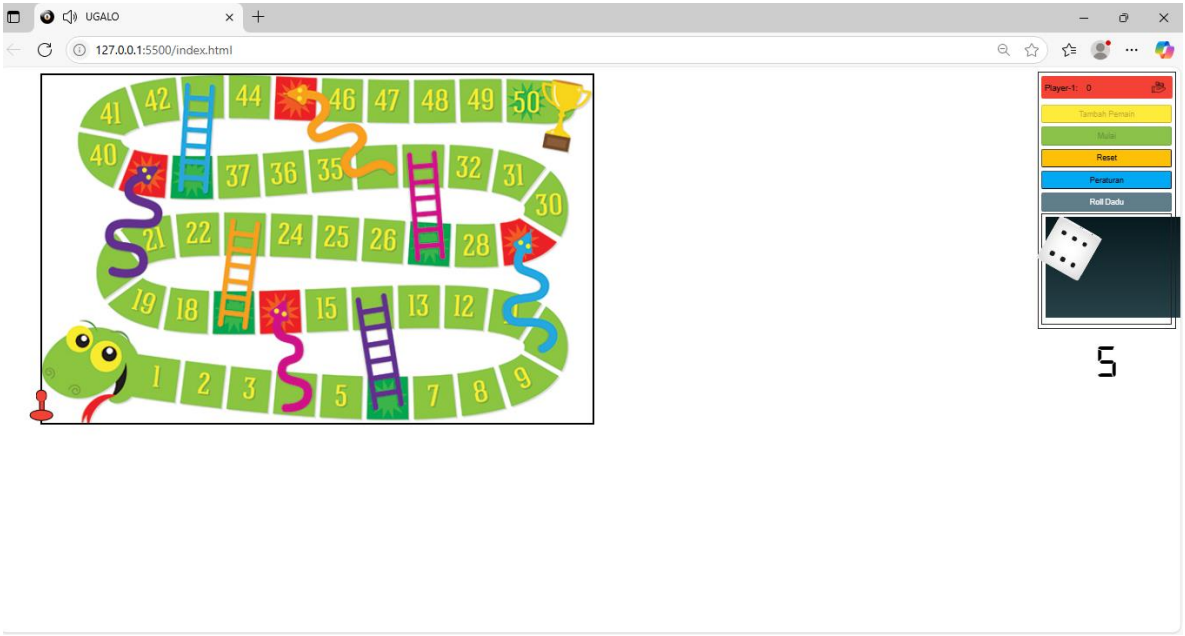
```
#board > .ficha.position-12 {
  left: 698px;
  bottom: 170px;
}

#board > .ficha.position-13 {
  left: 620px;
  bottom: 170px;
}

#board > .ficha.position-14 {
  left: 540px;
  bottom: 168px;
}

#board > .ficha.position-15 {
  left: 462px;
  bottom: 162px;
}

#board > .ficha.position-16 {
  left: 382px;
  bottom: 161px;
}

#board > .ficha.position-17 {
  left: 300px;
  bottom: 158px;
}

#board > .ficha.position-18 {
  left: 220px;
  bottom: 165px;
}

#board > .ficha.position-19 {
  left: 145px;
  bottom: 178px;
}

#board > .ficha.position-20 {
  left: 090px;
  bottom: 240px;
```

```
}

#board > .ficha.position-21 {
  left: 168px;
  bottom: 287px;
}

#board > .ficha.position-22 {
  left: 240px;
  bottom: 297px;
}

#board > .ficha.position-23 {
  left: 320px;
  bottom: 300px;
}

#board > .ficha.position-24 {
  left: 399px;
  bottom: 295px;
}

#board > .ficha.position-25 {
  left: 477px;
  bottom: 289px;
}

#board > .ficha.position-26 {
  left: 558px;
  bottom: 285px;
}

#board > .ficha.position-27 {
  left: 640px;
  bottom: 275px;
}

#board > .ficha.position-28 {
  left: 720px;
  bottom: 275px;
}

#board > .ficha.position-29 {
  left: 800px;
```

```
  bottom: 285px;
}

#board > .ficha.position-30 {
  left: 845px;
  bottom: 345px;
}

#board > .ficha.position-31 {
  left: 785px;
  bottom: 397px;
}

#board > .ficha.position-32 {
  left: 708px;
  bottom: 413px;
}

#board > .ficha.position-33 {
  left: 628px;
  bottom: 415px;
}

#board > .ficha.position-34 {
  left: 550px;
  bottom: 410px;
}

#board > .ficha.position-35 {
  left: 468px;
  bottom: 407px;
}

#board > .ficha.position-36 {
  left: 387px;
  bottom: 400px;
}

#board > .ficha.position-37 {
  left: 309px;
  bottom: 395px;
}

#board > .ficha.position-38 {
```

```
  left: 232px;
  bottom: 389px;
}

#board > .ficha.position-39 {
  left: 152px;
  bottom: 400px;
}

#board > .ficha.position-40 {
  left: 072px;
  bottom: 430px;
}

#board > .ficha.position-41 {
  left: 092px;
  bottom: 505px;
}

#board > .ficha.position-42 {
  left: 172px;
  bottom: 525px;
}

#board > .ficha.position-43 {
  left: 242px;
  bottom: 535px;
}

#board > .ficha.position-44 {
  left: 328px;
  bottom: 535px;
}

#board > .ficha.position-45 {
  left: 410px;
  bottom: 530px;
}

#board > .ficha.position-46 {
  left: 490px;
  bottom: 527px;
}
```

```
#board > .ficha.position-47 {
  left: 565px;
  bottom: 525px;
}

#board > .ficha.position-48 {
  left: 649px;
  bottom: 525px;
}

#board > .ficha.position-49 {
  left: 728px;
  bottom: 527px;
}

#board > .ficha.position-50 {
  left: 808px;
  bottom: 525px;
}
```



➢ Apk Bouncing Game

- Index Html

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>Bouncing game</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<div id="instruction-panel">
   <h1>Game Instructions</h1>
   <p>Here are the instructions for the game:</p>
   <ul>
      <li>Click on any ball to make it bounce.</li>
      <li>The ball will bounce up to a maximum height of 350px.</li>
      <li>The ball's color will change in middle of the animation.</li>
   </ul>
</div>

<div id="game-container">
   <div class="wall"></div> <!-- Wall added -->
   <div id="ball1" class="ball"></div>
   <div id="ball2" class="ball"></div>
   <div id="ball3" class="ball"></div>
</div>
<script src="script.js"></script>
<div id="game-description">
<p style="font-size: 24px;">This is a fun bouncing game where you can make the balls bounce by clicking on them. This game is generated for fidget purpose.</p>
<p style="font-size: 30px;">Developer: Prasuk Jain(96)</p>
</div>
</body>
</html>
```

- Main.Js

```
var balls = document.getElementsByClassName("ball");

for (var i = 0; i < balls.length; i++) {
   balls[i].addEventListener("click", bounceBall);
}

function bounceBall() {
   var ball = this;

   var posY = parseInt(ball.style.bottom) || 0;
```

```
    var maxHeight = 350; // Maximum height for the ball to bounce

    if (posY < maxHeight) {
        ball.style.animation = "bounce 2s linear";
        ball.style.bottom = (posY + maxHeight) + "px";
    }

    setTimeout(function() {
        ball.style.animation = "";
        ball.style.bottom = posY + "px";
    }, 2000);

    setTimeout(function() {
        var randomColor = generateRandomColor();
        ball.style.backgroundColor = randomColor;
    }, 1000); /*change ball colour after it mid of animation*/
}

function generateRandomColor() {
    var letters = "0123456789ABCDEF";
    var color = "#";

    for (var i = 0; i < 6; i++) {
        color += letters[Math.floor(Math.random() * 16)];
    }

    return color;
}
```

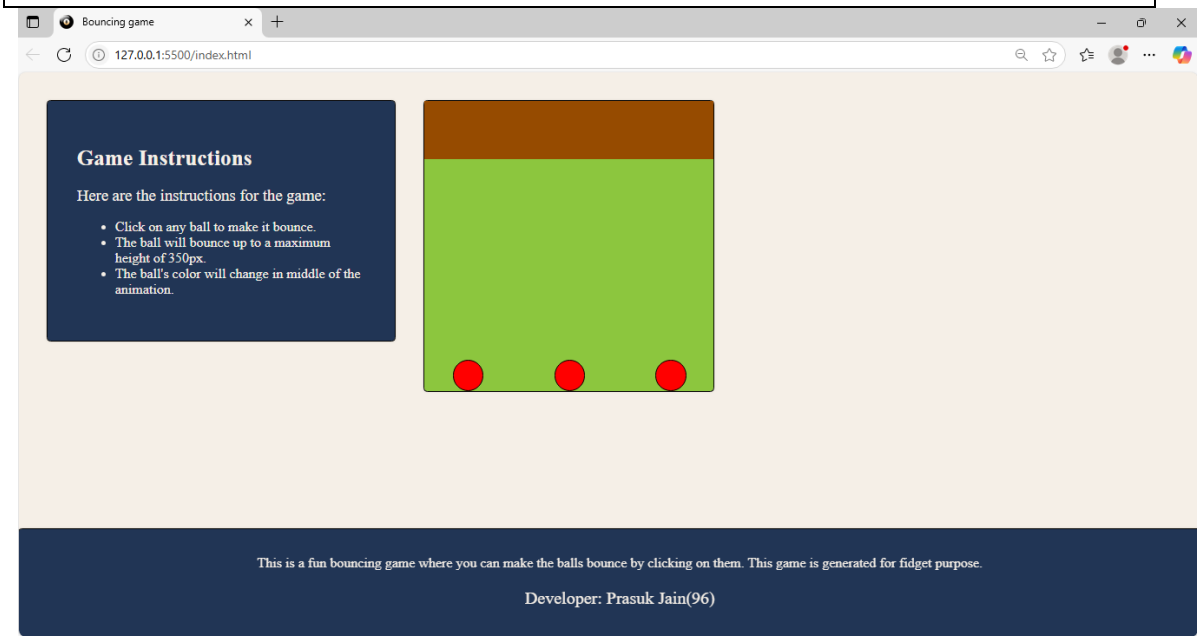
- Style css

```
body {
    margin: 0;
    padding: 0;
    background-color: #F5EFE7; /* Skin color */
}

#game-container {
    width: 500px;
    height: 500px;
    position: relative;
    margin: 50px 0px 0px 700px;
    border-radius: 5px;
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);
```

```
    outline: 2px solid black; /* Add black outline */
    background-color: #8CC63E; /* Grass green */
    display:inline-block;
}

.wall {
    width: 100%;
    height: 100px;
    position: absolute;
    top: 0;
    background-color: #964B00; /* Fox Brown */
}

.ball {
    width: 50px;
    height: 50px;
    border-radius: 50%;
    position: absolute;
    bottom: 0;
    background-color: red;
    animation: none;
    border: 2px solid black; /* Add black outline */
}

#instruction-panel {
    position: absolute;
    top: 50px;
    left: 50px;
    width: 500px;
    padding: 50px;
    background-color: #213555;
    border-radius: 5px;
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);
    outline: 2px solid black; /* Add black outline */
    display: inline-block;
    color : #F5EFE7
}

#instruction-panel h1 {
    font-size: 38px;
    margin-bottom: 10px;
}

#instruction-panel p {
```

```
    font-size: 28px;
    margin-bottom: 10px;
}

#instruction-panel ul {
    font-size: 24px;
    margin-left: 26px;
}

#game-description {
    position: fixed;
    bottom: 0;
    left: 0;
    width: 100%;
    padding: 20px;
    background-color: #213555;
 border-radius: 5px;
    box-shadow: 0 0 5px rgba(0, 0, 0, 0.2);
    outline: 2px solid black; /* Add black outline */
    color: #F5EFE7;
    font-size: 24px;
    text-align: center;
}

#ball1 {
    left: 50px;
}

#ball2 {
    left: 225px;
}

#ball3 {
    left: 400px;
}

@keyframes bounce {
    0% {
        bottom: 0;
    }
    50% {
        bottom: 350px;
    }
    100% {
```

```
        bottom: 0;
    }
}
```



- Apk Biliard

- Inderx.Html

```
<!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">
  <link rel="icon" type="image/x-icon" href="./src/assets/images/ball_8.png">

    <meta property="og:title" content="JS Billiards" />
   <meta property="og:image" content="jsbilliards.png" />
   <meta
    property="og:description"
    content="8-ball pool with 2 players built with Javascript, HTML, Canvas."
   />
   <meta
    property="og:url"
    content="https://cptleo92.github.io/JSBilliards/"
   />

  <link rel="stylesheet" href="./dist/styles.css">
```

```
  <link href="https://fonts.googleapis.com/css2?family=PT+Sans&display=swap"
rel="stylesheet">
  <link
    rel="stylesheet"
    href="https://cdnjs.cloudflare.com/ajax/libs/animate.css/4.1.1/animate.min.css"
  />
  <script type="text/javascript" src="./dist/main.js"></script>
  <title>JS Billiards!</title>
</head>
<body>
  <div class="instructions animate__animated animate__fadeInDown animate__slow">
    <h1>Welcome to JS Billiards!</h1>
    <p>This is your classic game of 8-ball pool. The controls are simple:</p>
    <ul>
      <li>If you have ball in hand, click to place the ball.</li>
      <li>Click to set the angle. You will see the stick rotate along with your mouse
position.</li>
      <li>Click, <strong>HOLD</strong> and <strong>RELEASE</strong> to shoot!
The power of the shot corresponds with the distance from the stick to the cue ball.</li>
    </ul>

      <div class="animate-flicker hidden">Loading...</div>

  </div>
  <canvas id="table" class="hidden"></canvas>
  <section class="info hidden unselectable">
    <section class="info-top"><p class="tracker"></p></section>
    <section class="info-bottom">
      <section class="info-bottom-left"></section>
      <section class="info-bottom-middle">
        <button class="reset">Reset Table</button>
        <div class="socials">
          <a href="http://www.linkedin.com/in/sirleoc"><img
src="src/assets/images/linkedin_icon_white.png" alt="LinkedIn"></a>
          <a href="http://www.github.com/cptleo92"><img
src="src/assets/images/github_icon_white.png" alt="Github"></a>
        </div>
      </section>
      <section class="info-bottom-right"></section>
    </section>
  </section>
</body>
</html>
```

- Main js

```
/*
 * ATTENTION: The "eval" devtool has been used (maybe by default in mode: "development").
 * This devtool is neither made for production nor for readable output files.
 * It uses "eval()" calls to create a separate source file in the browser devtools.
 * If you are trying to read the output file, select a different devtool (https://webpack.js.org/configuration/devtool/)
 * or disable the default devtool with "devtool: false".
 * If you are looking for production-ready output files, see mode: "production" (https://webpack.js.org/configuration/mode/).
 */
/******/ (() => { // webpackBootstrap
/******/  var __webpack_modules__ = ({

/***/ "./src/ball.js":
/*!*********************!*\
  !*** ./src/ball.js ***!
  \*********************/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

eval("const Util = __webpack_require__(/*! ./util */ \"./src/util.js\");\n\nconst RADIUS = 15;\n\nclass Ball {\n  constructor (num) {\n    this.num = num;\n    this.type = this.getType(num);\n    this.radius = RADIUS;\n    this.size = this.radius * 2;\n    this.pos = [0,0];\n    this.vel = [0,0];\n    this.wallCollided = false;\n    this.onTable = true; \n    this.sinking = false;\n\n    this.img = new Image();    \n    this.img.src = `src/assets/images/ball_${this.num}.png`\n\n  }\n\n  isStationary() {\n    return (this.vel[0] === 0 && this.vel[1] === 0);\n  }\n\n  resetBall() {\n    this.vel[0] = 0;\n    this.vel[1] = 0;\n    this.size = this.radius * 2;\n    this.onTable = true;\n    this.sinking = false;\n  }\n\n  getType(num) {\n    if (num === 0) {\n      return \"white\"\n    } else if (num < 8) {\n      return \"solid\";\n    } else if (num > 8) {\n      return \"stripe\";\n    } else if (num === 8) {\n      return \"eight\"\n    }\n  }\n\n  draw(ctx) {      \n    if (this.onTable) {     \n      ctx.drawImage(this.img, \n        Math.round(this.pos[0] - this.radius), \n        Math.round(this.pos[1] - this.radius),\n        this.size,\n        this.size\n      );\n    }\n\n    if (this.sinking && this.size >= 0) {\n      ctx.drawImage(this.img, this.pos[0] - this.radius, this.pos[1] - this.radius, this.size--, this.size--);\n    }\n  }\n\n  move(timeDelta) {\n    const velScale = timeDelta / (1000 / 60);\n    let x = this.pos[0];\n    let y = this.pos[1];\n    let dx = this.vel[0] * velScale;\n    let dy = this.vel[1] * velScale;\n\n    this.pos = [x + dx, y + dy]; \n    \n    if (dx !== 0) {\n      Math.abs(dx) < .05 ? this.vel[0] = 0 : this.vel[0] *= .994;\n    }\n\n    if (dy !== 0) {\n      Math.abs(dy) < .05 ? this.vel[1] = 0 : this.vel[1]
```

```
*= .994;\n    }\n\n    if ((x > 1180 || x < 20) || (y < 20 || y > 580))
{\n      this.resetBall();\n      this.pos = [\n        Math.floor(Math.random() * 1000) +
200, \n        Math.floor(Math.random() * 600) + 50, \n        ];\n    }
\n    \n  }\n\n  collideEdge(wall) {\n    let vx = this.vel[0];\n    let vy =
this.vel[1];\n    let notCollided = true;\n\n    if (notCollided) {\n      if (wall.type ===
'horizontal') {\n        this.vel = [vx, -vy];\n      } else if (wall.type === 'vertical')
{\n        this.vel = [-vx, vy];\n      } else if (wall.type === '1-diag') {\n        this.vel = [-
vy, -vx];\n      } else if (wall.type === '2-diag') {\n        this.vel = [vy, vx];\n      }
\n      notCollided = false;\n    }\n\n    setTimeout( () => {\n      notCollided =
true;\n    }, 100);\n      \n    //fix pos to avoid sticking onto walls\n    let buffer = 3;\n    if
(wall.location === 'top') {\n      this.pos[1] += buffer;\n    } else if (wall.location ===
'bottom') {\n      this.pos[1] -= buffer;\n    } else if (wall.location === 'left')
{\n      this.pos[0] += buffer;\n    } else if (wall.location === 'right') {\n      this.pos[0] -
= buffer;\n    }\n  }  \n\n  sink() {\n    this.onTable = false;\n    this.vel[0] =
0;\n    this.vel[1] = 0;\n    this.sinking = true;    \n  }\n}\n\nmodule.exports =
Ball;\n\n//# sourceURL=webpack://Billiards/./src/ball.js?");

/***/ }),

/***/ "./src/cue_ball.js":
/*!*************************!*\
  !*** ./src/cue_ball.js ***!
  \*************************/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

eval("const Util = __webpack_require__(/*! ./util */ \"./src/util.js\");\nconst Ball =
__webpack_require__ (/*! ./ball.js */ \"./src/ball.js\");\nconst Power =
__webpack_require__(/*! ./power.js */ \"./src/power.js\");\n\nclass CueBall extends
Ball {\n  constructor() {\n    super(0);\n\n    this.ballInHand =
true;\n    this.behindTheLine = true;      \n    this.canBeHit = false;\n    this.canvas =
document.getElementById(\"table\");\n    this.ctx =
this.canvas.getContext('2d');\n    this.power =
1;\n\n    this.handleBallInHand();   \n  }\n\n  handleBallInHand() {\n    this.canBeHit =
false;\n    this.ballInHand = true;\n\n    const placeBall = function(e) {\n      let [x, y] =
Util.getCursorPos(e);\n      this.vel[0] = 0;\n      this.vel[1] = 0;\n\n      if
(this.behindTheLine) {\n        this.pos[0] = Util.clamp(x, 890, 1125);\n        this.pos[1]
= Util.clamp(y, 80, 530);   \n      } else {\n        this.pos[0] = Util.clamp(x, 65,
1125);\n        this.pos[1] = Util.clamp(y, 80,
530);  \n    }  \n    \n    this.canvas.addEventListener(\"click\", () =>
{    \n      this.ballInHand = false;      \n      this.behindTheLine =
false;\n      this.canBeHit =
true;\n      this.canvas.removeEventListener(\"mousemove\",
placeBall);          \n    }, {once:
true})          \n  }.bind(this);\n\n    this.canvas.addEventListener(\"mousemove\",
```

```
placeBall)  \n  }    \n\n  calcHit(e, callback) { \n    let [x, y] =
Util.getCursorPos(e);                \n    let cx = this.pos[0];\n    let cy =
this.pos[1];\n    let dist = Util.getPointDistance(x, y, cx, cy);  \n    let vec = [(x - cx) /
dist, (y - cy) / dist]    \n    // console.log(power);  \n    this.holdMouseForPower(
(power) => {\n      let vel = [vec[0] * power, vec[1] * power]    \n      this.hitCue(vel,
callback);        \n    });\n    \n  }\n\n  holdMouseForPower(callback) {  \n    let
increasing = true; \n    const minPower = 1;\n    const maxPower = 40;\n    const
increment = 2;  \n    \n    const powerCounter = () => {\n      const interval =
setInterval(() => {\n        if (this.power <= maxPower && increasing)
{\n          this.power += increment;\n        } else {\n          increasing =
false;\n        }\n        \n        if (this.power >= minPower && !increasing)
{\n          this.power -= increment;\n        } else {\n          increasing =
true;  \n        }\n      }, 50);    \n\n      const clearCounter = () =>
{\n        clearInterval(interval);\n        this.canvas.removeEventListener(\"mousedown\"
, powerCounter);\n        callback(this.power);\n        this.power =
1;\n      }\n  \n      this.canvas.addEventListener(\"mouseup\", clearCounter, {once:
true});    \n    }\n    this.canvas.addEventListener(\"mousedown\",
powerCounter);\n  }\n\n  hitCue(vel, callback) {    \n    // console.log('hit
cue');\n    this.vel = vel.map( num => {\n      if (num < -51) {\n        return -50;\n      }
else if (num > 51) {\n        return 50;\n      } else {\n        callback();\n        return
num;\n      }\n    })  \n  }\n\n  handleScratch()
{   \n    this.resetBall();\n    this.handleBallInHand();  \n  }\n}\n\nmodule.exports =
CueBall;\n\n//# sourceURL=webpack://Billiards/./src/cue_ball.js?");

/***/ }),

/***/ "./src/game-view.js":
/*!**************************!*\
  !*** ./src/game-view.js ***!
  \**************************/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

eval("const Stick = __webpack_require__(/*! ./stick.js */ \"./src/stick.js\")\nconst Util
= __webpack_require__(/*! ./util.js */ \"./src/util.js\");\nconst Power =
__webpack_require__(/*! ./power.js */ \"./src/power.js\");\n\nclass GameView
{\n  constructor(game, ctx) {\n    this.game = game;\n    this.table =
this.game.table;\n    this.ctx = ctx;\n    this.lastTime = 0;      \n\n    this.info =
document.querySelector(\".info\");    \n    this.init();  \n  }\n\n  init()
{\n    this.resetButton();\n\n    let table = document.getElementById(\"table\");\n    let
info = document.querySelector(\".info\");\n    let instructions =
document.querySelector(\".instructions\");\n    let text =
document.querySelector(\".animate-
flicker\");\n\n    requestAnimationFrame(this.animate.bind(this))  \n\n    setTimeout(()
=> {\n      text.innerHTML = \"Click anywhere to
```

```
continue!\"\n      window.addEventListener(\"click\", () =>
{    \n        table.classList.remove(\"hidden\");        \n        info.classList.remove(\"hidde
n\");\n        instructions.classList.add(\"hidden\");    \n      }, {once: true})    \n    },
3500);      \n  }\n\n  animate(time) {\n    const timeDelta = time -
this.lastTime;\n    this.game.update(timeDelta);\n    this.draw();\n    this.lastTime =
time;\n    requestAnimationFrame(this.animate.bind(this));\n  }    \n\n  draw ()
{  \n    this.ctx.clearRect(0, 0, this.table.width,
this.table.height);\n    this.game.table.balls.forEach( (ball) => ball.draw(this.ctx));\n    //
this.game.table.pockets.forEach( (pocket) =>
pocket.draw(this.ctx))\n    this.game.table.drawPocketed();\n    \n    if
(!this.game.cue.ballInHand) {\n      this.game.stick.draw(this.ctx,
this.game.cue);\n    }  \n  }\n\n  resetButton() {    \n    const reset =
document.querySelector(\"button\");   \n\n    reset.addEventListener(\"click\", () =>
{\n      this.table.resetTable();\n      this.game.reset();\n    })\n  }\n\n}\n\nmodule.exports
= GameView;\n\n//# sourceURL=webpack://Billiards/./src/game-view.js?");

/***/ }),

/***/ "./src/game.js":
/*!*********************!*\
  !*** ./src/game.js ***!
  \*********************/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

eval("const Player = __webpack_require__(/*! ./player.js */ \"./src/player.js\");\nconst
Table = __webpack_require__(/*! ./table.js */ \"./src/table.js\");\nconst Util =
__webpack_require__(/*! ./util.js */ \"./src/util.js\")\nconst CueBall =
__webpack_require__(/*! ./cue_ball.js */ \"./src/cue_ball.js\");\nconst Stick =
__webpack_require__(/*! ./stick.js */ \"./src/stick.js\")\n\nclass Game {\n  constructor
(canvas, ctx) {\n    this.canvas = canvas;\n    this.table = new Table(canvas,
ctx);\n    this.pockets = this.table.pockets;\n    this.walls = this.table.walls;\n    this.balls
= this.table.balls;\n    this.cue = this.table.balls[0];\n    this.ctx = ctx;\n\n    this.players
= [new Player(1), new Player(2)];\n    this.waitForHit = true;\n    this.openBreak =
true;    \n    this.currentPlayer = this.players[0];    \n    this.otherPlayer =
this.players[1];\n    this.pocketed = null;\n    this.firstBallHit = null;\n    this.scratched =
false;\n    this.over = false;\n\n    this.stick = new
Stick(this.canvas);\n    this.updateTracker();\n    this.play();\n    \n\n    // setInterval(()
=> {\n    //   console.log('cue pos: ' + this.cue.pos)\n    //   console.log('cue vel: ' +
this.cue.vel)\n    // }, 500);\n  }\n\n  play() {  \n    const clickToHit = (e) => {  \n      if
(this.cue.canBeHit && this.waitForHit) {\n        this.stick.rotating =
false;\n        this.cue.calcHit(e, () => {        \n          this.waitForHit =
false;        \n          this.stick.visible =
false;\n        });\n      }\n    }      \n    this.canvas.addEventListener(\"click\",
clickToHit);    \n  }\n\n  reset() {\n    this.waitForHit = true;\n    this.openBreak = true;
```

```
\n    this.stick.visible = true;\n    this.stick.rotating = true;\n    this.pocketed =
null;\n    this.firstBallHit = null;\n    this.scratched = false;\n    this.over =
false;\n\n    this.players.forEach( player => {\n      player.ballType =
null;\n      player.lastBall =
false;\n    })\n\n    this.updateTracker();\n  }\n\n  update(timeDelta)
{  \n    this.moveBalls(timeDelta);\n    this.detectCollisions();\n    this.detectWallColli
sions();\n    this.detectPocketed();\n    \n    if (!this.waitForHit)
{\n      this.checkStopped();\n    }\n  }\n\n  checkStopped() {\n    if (this.balls.every(
ball => ball.isStationary())) {\n      this.waitForHit =
true;\n      this.endOfTurn();\n    }\n  }\n\n  endOfTurn() {\n    let checkBalls =
this.table.pocketed.filter(ball => ball.type === this.currentPlayer.ballType)\n    if
(checkBalls.length === 7) {\n      this.currentPlayer.lastBall = true\n    };\n\n    if
(!this.balls[8].onTable && !this.over) {\n      this.gameOver();\n      this.over =
true;\n    }\n\n    this.checkScratch();\n    if (this.scratched)
{\n      this.cue.handleScratch();    \n      this.resolveTurn(true);\n    } else if
(this.pocketed !== null) {   \n        let type = this.pocketed.type;       \n        if
(this.openBreak && type !== 'white')
{\n          this.assignType(type);\n          this.openBreak =
false;          \n          this.resolveTurn(false);    \n        } else if (type !==
this.currentPlayer.ballType) {          \n          this.resolveTurn(true);\n        } else
{\n          this.resolveTurn(false);\n        }\n    } else
{\n      this.resolveTurn(true);\n    }   \n  }\n\n  resolveTurn(switchPlayer)
{\n    this.pocketed = null; \n    this.waitForHit = true; \n    this.firstBallHit =
null;\n    this.scratched = false\n    if (switchPlayer)
{this.switchTurn()};\n    this.updateTracker();\n\n    this.stick.visible =
true;\n    this.stick.rotating = true;\n  }\n\n  switchTurn()
{\n    this.players.reverse();\n    this.currentPlayer =
this.players[0];    \n    this.otherPlayer = this.players[1];\n  }\n\n  updateTracker()
{\n    const p = document.querySelector(\".tracker\");\n    if (!this.over) {\n      const
player = this.currentPlayer.num;\n      const turn =
this.currentPlayer.ballType;      \n\n      if (turn === null) {\n        p.innerHTML = `It is
Player ${player}'s turn! Open table!`\n      } else if (this.currentPlayer.lastBall)
{\n        p.innerHTML = `It is Player ${player}'s turn! Sink the 8 to win!`\n      } else
{\n        p.innerHTML = `It is Player ${player}'s turn! You are
${turn}.`\n      }\n    }\n  }\n\n  assignType(type) {\n    if (type === 'solid')
{\n      this.currentPlayer.ballType = 'solid';\n      this.otherPlayer.ballType =
'stripe';\n    } else if (type === 'stripe') {\n      this.currentPlayer.ballType =
'stripe';\n      this.otherPlayer.ballType = 'solid';\n    }\n  }\n\n  moveBalls(timeDelta)
{\n    this.balls.forEach( ball1 =>
{\n      ball1.move(timeDelta);        \n    })\n  }\n  \n  \n\n  detectCollisions()
{      \n    let obj1;\n    let obj2;    \n    let colDist = this.cue.radius * 2.1;    \n    \n    for
(let i = 0; i < 16; i++) {\n      obj1 = this.balls[i];        \n      if (!obj1.onTable)
{continue};\n      for (let j = i + 1; j < 16; j++)\n      {\n        obj2 =
this.balls[j];    \n        if (!obj2.onTable) {continue};\n        if (Util.getDistance(obj1,
```

```
obj2) <= colDist) {  \n  Util.ballCollisionMath(obj1, obj2);\n\n  if (obj1 instanceof CueBall && !this.firstBallHit && !obj1.ballInHand)
{\n  this.firstBallHit = obj2;  \n  }\n\n  }
\n  \n  }\n  }  \n  }  \n\n  detectWallCollisions() {\n  for (let i = 0; i < 16; i++)
{\n  let ball = this.balls[i];  \n  if (!ball.onTable && ball.isStationary)
{continue}\n\n  let bx = ball.pos[0];\n  let by = ball.pos[1];\n\n  for (let j = 0; j <
18; j++) {\n  let wall = this.walls[j];\n  let wx1 = wall.x1;\n  let wx2 =
wall.x2;\n  let wy1 = wall.y1;\n  let wy2 = wall.y2;\n\n  //find closest point
on wall\n  let wallLen = Util.getPointDistance(wx1, wy1, wx2, wy2);\n  let dot
= ( ((bx - wx1) * (wx2 - wx1)) + ((by - wy1) * (wy2 - wy1)) ) / Math.pow(wallLen,
2);\n  let closestX = wx1 + (dot * (wx2 - wx1));  \n  let closestY = wy1 +
(dot * (wy2 - wy1));  \n\n  //make sure closest point is on the line\n  if
(!wall.isPointCollide(closestX, closestY)) {\n  continue;\n  }\n\n  let
distance = Util.getPointDistance(bx, by, closestX, closestY);\n\n  if (distance <=
ball.radius)
{  \n  ball.collideEdge(wall);\n  }  \n  }\n  }\n  }\n\n  detectPocke
ted() {\n  for (let i = 0; i < 16; i++) {\n  let ball = this.balls[i];  \n  if
(!ball.onTable && ball.isStationary) {continue}\n  if (ball instanceof CueBall &&
ball.ballInHand) {continue}\n\n  for (let j = 0; j < 6; j++) {\n  let pocket =
this.pockets[j];\n  let r = pocket.radius;\n\n  let dist =
Util.getPointDistance(ball.pos[0], ball.pos[1], pocket.x, pocket.y);  \n\n  if (dist
<= r) {  \n  ball.sink();\n  if (ball.type === 'solid' || ball.type === 'stripe')
{\n  this.table.pocketed.push(ball);\n  }\n  if (!this.pocketed &&
ball.type !== 'white') {  \n  this.pocketed = ball;  \n  }\n  if
(ball instanceof CueBall && !ball.ballInHand) {\n  this.scratched =
true;  \n  }\n  }\n  }\n  }\n  }  \n\n  checkScratch() {\n  if (this.firstBallHit
=== null) {\n  this.scratched = true;\n  } else if (this.firstBallHit.type === 'eight')
{\n  if (!this.currentPlayer.lastBall) {\n  this.scratched = true;\n  }\n  } else
{\n  if (this.currentPlayer.ballType !== this.firstBallHit.type && !this.openBreak)
{  \n  this.scratched = true;  \n  }  \n  }\n  }\n\n  gameOver() {\n  let num =
this.currentPlayer.num;\n  const p = document.querySelector(\".tracker\");\n\n  if
(this.openBreak) {\n  p.innerHTML = `Player ${num} wins by sinking the 8 on the
break!`\n  return;\n  }\n\n  let type = this.currentPlayer.ballType;\n  let
checkBalls = this.table.pocketed.filter( ball => ball.type === type );\n  \n  if
(checkBalls.length === 7 && !this.scratched) {\n  p.innerHTML = `Player ${num}
wins!`\n  } else {\n  p.innerHTML = `Player ${num}
loses!`\n  }  \n  }\n}\n\nmodule.exports = Game;\n\n//#
sourceURL=webpack://Billiards/./src/game.js?");

/***/ }),

/***/ "./src/index.js":
/*!**********************!*\
  !*** ./src/index.js ***!
```

```
  \**********************/
/***/ ((__unused_webpack_module, __unused_webpack_exports,
__webpack_require__) => {

eval("// const Ball = require(\"./ball.js\");\n// const Table = require(\"./table.js\");\nconst
Util = __webpack_require__(/*! ./util.js */ \"./src/util.js\");\nconst GameView =
__webpack_require__(/*! ./game-view.js */ \"./src/game-view.js\")\nconst Game =
__webpack_require__(/*! ./game.js */
\"./src/game.js\")\n\ndocument.addEventListener(\"DOMContentLoaded\", event =>
{\n  const canvas = document.getElementById(\"table\");\n  const ctx =
canvas.getContext('2d');\n  canvas.width = 1200;\n  canvas.height = 600; \n\n  let game
= new Game(canvas, ctx);\n  let gameView = new GameView(game, ctx);\n
\n  document.addEventListener(\"click\", e =>
{\n    console.log(Util.getCursorPos(e));\n  })\n  \n})\n\n\n//#
sourceURL=webpack://Billiards/./src/index.js?");

/***/ }),

/***/ "./src/player.js":
/*!***********************!*\
  !*** ./src/player.js ***!
  \***********************/
/***/ ((module) => {

eval("class Player {\n  constructor(num) {\n    this.num = num;\n    this.ballType =
null;    \n    this.lastBall = false;\n  }\n}\n\nmodule.exports = Player;\n\n//#
sourceURL=webpack://Billiards/./src/player.js?");

/***/ }),

/***/ "./src/pocket.js":
/*!***********************!*\
  !*** ./src/pocket.js ***!
  \***********************/
/***/ ((module) => {

eval("const RADIUS = 28;\n\nclass Pocket {\n  constructor(x, y) {\n    this.radius =
RADIUS;\n    this.x = x;\n    this.y = y;\n  }\n  \n  draw(ctx)
{\n    ctx.beginPath();\n    ctx.lineWidth = 3;\n    ctx.strokeStyle =
\"red\";\n    ctx.arc(this.x, this.y, this.radius, 0, Math.PI *
2);\n    ctx.stroke();\n  }\n}\n\nmodule.exports = Pocket;\n\n//#
sourceURL=webpack://Billiards/./src/pocket.js?");

/***/ }),
```

```
/***/ "./src/power.js":
/*!**********************!*\
  !*** ./src/power.js ***!
  \**********************/
/***/ ((module) => {

eval("// not currently implemented\n\nclass Power {\n  constructor() {\n    this.img = new Image();\n    this.img.src = 'src/assets/images/BlueBar.png'  \n    this.visible = false;\n  }\n\n  draw(power, cue, ctx) {\n    if (this.visible) {\n      let [x, y] = cue.pos;\n      ctx.drawImage(this.img,x - 50, y + 20, this.getWidth(power), 200);  \n    }\n    // 60, 350, 900, 200 src info\n  }\n\n  getWidth(power) {\n    let coef = 900 / 7;\n    return power * coef;\n  }\n}\n\nmodule.exports = Power;\n\n//# sourceURL=webpack://Billiards/./src/power.js?");

/***/ }),

/***/ "./src/stick.js":
/*!**********************!*\
  !*** ./src/stick.js ***!
  \**********************/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

eval("const Util = __webpack_require__(/*! ./util.js */ \"./src/util.js\")\n\nclass Stick {\n  constructor() {  \n    this.img = new Image();\n    this.img.src = 'src/assets/images/cue.png';\n    this.visible = true;  \n    this.rotating = true;\n    this.shooting = false;\n\n    this.canvas = document.getElementById(\"table\");\n    this.canvas.addEventListener(\"mousemove\", e => {\n      if (this.rotating) {  \n        [this.mouseX, this.mouseY] = Util.getCursorPos(e);\n      }\n    })\n  }\n\n  draw(ctx, cue) {\n    let dist = 20 + (cue.power * 2);\n\n    if (this.visible) {  \n      let x = cue.pos[0];\n      let y = cue.pos[1];\n      let offset = cue.radius / 2 + 3;\n      let opposite = this.mouseY - y;\n      let adjacent = this.mouseX - x;  \n      ctx.save();\n      ctx.translate(x, y);  \n      ctx.rotate(Math.atan2((opposite) * -1, adjacent * -1));  \n      ctx.translate(-x, -y);  \n      ctx.drawImage(this.img, x + dist, y - offset);\n      ctx.restore(); \n    }  \n  }\n}\n\nmodule.exports = Stick;\n\n//# sourceURL=webpack://Billiards/./src/stick.js?");

/***/ }),

/***/ "./src/table.js":
/*!**********************!*\
  !*** ./src/table.js ***!
  \**********************/
```

```
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

eval("const Pocket = __webpack_require__(/*! ./pocket.js */ \"./src/pocket.js\");\nconst Wall = __webpack_require__(/*! ./wall.js */ \"./src/wall.js\");\nconst Ball = __webpack_require__ (/*! ./ball.js */ \"./src/ball.js\");\nconst CueBall = __webpack_require__(/*! ./cue_ball.js */ \"./src/cue_ball.js\");\nconst Util = __webpack_require__(/*! ./util.js */ \"./src/util.js\");\n\nclass Table {\n  constructor (canvas, ctx) {\n    this.ctx = ctx;\n\n    this.balls = this.generateBalls();\n    this.positionBalls();\n\n    this.walls = this.generateWalls();\n    \n    this.pockets = this.generatePockets();\n    this.pocketed = [];\n\n    this.width = canvas.width;\n    this.height = this.width / 2;\n\n    this.solidSection = document.querySelector(\".info-bottom-left\");\n    this.stripeSection = document.querySelector(\".info-bottom-right\");\n\n    // this.test();\n  }\n\n  test() {\n    for (let i = 1; i <= 5; i++) {\n      this.balls[i].sink();\n      this.pocketed.push(this.balls[i]);\n    }\n  }\n\n  generateBalls () {\n    const balls = [];\n    for(let i = 0; i <= 15; i++) {\n      let ball;\n      if (i === 0) {\n        ball = new CueBall();\n        balls.push(ball);\n      } else {\n        ball = new Ball(i);\n        balls.push(ball);\n      }\n    }\n    return balls;\n  }\n\n  drawPocketed() {\n    this.pocketed.forEach( ball => {\n      if (ball.type === 'solid') {\n        this.solidSection.appendChild(ball.img);\n      } else {\n        this.stripeSection.appendChild(ball.img);\n      }\n    })\n  }\n\n  resetTable() {\n    this.resetInfo();\n\n    for (const ball of this.balls) {\n      \n      ball.resetBall();\n      if (ball instanceof CueBall) {\n        ball.behindTheLine = true;\n        ball.handleBallInHand()\n      }\n    }\n    this.positionBalls();\n  }\n\n  resetInfo() {\n    this.pocketed = [];\n    this.solidSection.innerHTML = \"\";\n    this.stripeSection.innerHTML = \"\";\n  }\n\n  positionBalls () { \n    let x = 325;\n    let y = 300;\n\n    let r = this.balls[0].radius;\n    let d = (r * 2);\n    \n    this.balls[1].pos = [x, y];\n    this.balls[8].pos = [x - (d * 2), y];\n\n    const POSITIONS = [\n      [x - d, y - r], \n      [x - d, y + r],\n      [x - (d * 2), y - d],\n      [x - (d * 2), y + d],\n      [x - (d * 3), y + r],\n      [x - (d * 3), y + (d + r)],\n      [x - (d * 3), y - (d + r)],\n      [x - (d * 3), y - r],\n      [x - (d * 4), y - d],\n      [x - (d * 4), y],\n      [x - (d * 4), y + d],\n      [x - (d * 4), y + (d * 2)],\n      [x - (d * 4), y - (d * 2)]\n    ]\n\n    const REMAINING = [2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15];\n\n    for(let i = 0; i < 13; i++) {\n      let idx = Math.floor(Math.random() * REMAINING.length);\n      this.balls[REMAINING[idx]].pos = POSITIONS[i];\n      REMAINING.splice(idx, 1); \n    }\n  }\n\n  generateWalls() {\n    return [\n      new Wall(109, 67, 555, 67, 'horizontal', 'top'), //top-left\n      new Wall(631, 67, 1080, 67, 'horizontal', 'top'), //top-right\n      new Wall(109, 530, 555, 530, 'horizontal', 'bottom'),//bottom-left\n      new Wall(631, 530, 1080, 530, 'horizontal', 'bottom'), //bottom-right\n      new Wall(76, 104, 76, 495, 'vertical', 'left'), //left\n      new Wall(1120, 104, 1120, 495, 'vertical', 'right'), //right\n      // corners, clockwise starting from top-left\n      new Wall(109, 67, 88, 50, '2-diag'),\n      \n      new Wall(555, 67, 567, 52, '1-diag'),\n      new Wall(631, 67, 622, 52, '2-diag'),\n      new
```

```
Wall(1080, 67, 1105, 52, '1-diag'),\n      new Wall(1120, 104, 1150, 80, '1-
diag'),\n      new Wall(1120, 495, 1150, 520, '2-diag'),\n      new Wall(1080, 530, 1105,
555, '2-diag'),\n      new Wall(631, 530, 622, 555, '1-diag'),\n      new Wall(555, 530,
564, 555, '2-diag'),\n      new Wall(109, 530, 88, 555, '1-diag'),\n      new Wall(76, 495,
55, 520, '1-diag'),\n      new Wall(76, 104, 55, 85, '2-
diag')\n    ];\n\n  }\n\n  generatePockets() {\n    return [\n      new Pocket(58,
58),\n      new Pocket(592, 43),\n      new Pocket(1132, 58),\n      new Pocket(1132,
545),\n      new Pocket(592, 558),\n      new Pocket(58,
545),\n    ]\n  }\n  \n}\n\nmodule.exports = Table;\n\n//#
sourceURL=webpack://Billiards/./src/table.js?");

/***/ }),

/***/ "./src/util.js":
/*!*********************!*\
  !*** ./src/util.js ***!
  \*********************/
/***/ ((module) => {

eval("const Util = {\n  getDistance: function (b1, b2) {\n    let x1 = b1.pos[0];\n    let
x2 = b2.pos[0];\n    let y1 = b1.pos[1];\n    let y2 = b2.pos[1];\n    return Math.sqrt(((x1
- x2) ** 2) + ((y1 - y2) ** 2));\n  },\n  \n  getPointDistance: function (x1, y1, x2, y2)
{\n    return Math.sqrt(((x1 - x2) ** 2) + ((y1 - y2) ** 2));\n  },\n\n  ballCollisionMath:
function (obj1, obj2) {\n    let vCollision = {x: obj2.pos[0] - obj1.pos[0], y: obj2.pos[1]
- obj1.pos[1]};\n    let distance = Math.sqrt((obj2.pos[0]-obj1.pos[0])*(obj2.pos[0]-
obj1.pos[0]) + (obj2.pos[1]-obj1.pos[1])*(obj2.pos[1]-obj1.pos[1]));\n    let
vCollisionNorm = {x: vCollision.x / distance, y: vCollision.y / distance};\n    let
vRelativeVelocity = {x: obj1.vel[0] - obj2.vel[0], y: obj1.vel[1] - obj2.vel[1]};\n    let
speed = vRelativeVelocity.x * vCollisionNorm.x + vRelativeVelocity.y *
vCollisionNorm.y;\n\n    if (speed < 0) {\n      return;\n    }\n\n    obj1.vel[0] -= (speed
* vCollisionNorm.x);\n    obj1.vel[1] -= (speed * vCollisionNorm.y);\n    obj2.vel[0]
+= (speed * vCollisionNorm.x);\n    obj2.vel[1] += (speed *
vCollisionNorm.y);\n  },\n\n  getCursorPos: function (e) {\n    const canvas =
document.getElementById(\"table\");\n    const rect =
canvas.getBoundingClientRect();\n    const x = (e.clientX - rect.left) / (rect.right -
rect.left) * canvas.width\n    const y = (e.clientY - rect.top) / (rect.bottom - rect.top) *
canvas.height\n    return [x, y];\n  },\n\n  clamp: function (val, min, max) {\n    return
val > max ? max : val < min ? min : val;\n  }\n}\n\nmodule.exports = Util;\n\n//#
sourceURL=webpack://Billiards/./src/util.js?");

/***/ }),

/***/ "./src/wall.js":
/*!*********************!*\
```

```
  !*** ./src/wall.js ***!
  \*********************/
/***/ ((module, __unused_webpack_exports, __webpack_require__) => {

eval("const Util = __webpack_require__(/*! ./util.js */ \"./src/util.js\");\n\nclass Wall
{\n  constructor(x1, y1, x2, y2, type, location) {  \n    this.color = \"red\";\n\n    this.x1
= x1;\n    this.x2 = x2;\n    this.y1 = y1;\n    this.y2 = y2;  \n\n    this.type =
type;\n    this.location = location;\n  }\n\n  draw(ctx) {\n    ctx.moveTo(this.x1,
this.y1);\n    ctx.lineTo(this.x2, this.y2);\n    ctx.strokeStyle =
this.color;\n    ctx.lineWidth = 3;\n    ctx.stroke();    \n  }\n\n  isPointCollide(x, y)
{\n    let dist1 = Util.getPointDistance(this.x1, this.y1, x, y);\n    let dist2 =
Util.getPointDistance(this.x2, this.y2, x, y);\n    let wallLength =
Util.getPointDistance(this.x1, this.y1, this.x2, this.y2);\n\n    let buffer = 0.2;\n\n    if
(dist1 + dist2 >= wallLength - buffer && dist1 + dist2 <= wallLength + buffer)
{\n      return true;\n    }\n    return false;\n  }\n  \n}\n\nmodule.exports = Wall;\n\n//#
sourceURL=webpack://Billiards/./src/wall.js?");

/***/ })

/******/  });
/************************************************************************
***/
/******/  // The module cache
/******/  var __webpack_module_cache__ = {};
/******/
/******/  // The require function
/******/  function __webpack_require__(moduleId) {
/******/    // Check if module is in cache
/******/    var cachedModule = __webpack_module_cache__[moduleId];
/******/    if (cachedModule !== undefined) {
/******/      return cachedModule.exports;
/******/    }
/******/    // Create a new module (and put it into the cache)
/******/    var module = __webpack_module_cache__[moduleId] = {
/******/      // no module.id needed
/******/      // no module.loaded needed
/******/      exports: {}
/******/    };
/******/
/******/    // Execute the module function
/******/    __webpack_modules__[moduleId](module, module.exports,
__webpack_require__);
/******/
/******/    // Return the exports of the module
```

```
/******/    return module.exports;
/******/  }
/******/
/************************************************************************
***/
/******/
/******/  // startup
/******/  // Load entry module and return exports
/******/  // This entry module can't be inlined because the eval devtool is used.
/******/  var __webpack_exports__ = __webpack_require__("./src/index.js");
/******/
/******/ })()
;
```

- Style css

```
body {
  font-family: 'PT Sans', sans-serif;
  color: rgb(223, 219, 219);
  background-color: rgb(67, 39, 83);
}

@keyframes flickerAnimation {
  0%   { opacity:1; }
  50%  { opacity:0; }
  100% { opacity:1; }
}

@-o-keyframes flickerAnimation{
  0%   { opacity:1; }
  50%  { opacity:0; }
  100% { opacity:1; }
}

@-moz-keyframes flickerAnimation{
  0%   { opacity:1; }
  50%  { opacity:0; }
  100% { opacity:1; }
}

@-webkit-keyframes flickerAnimation{
  0%   { opacity:1; }
  50%  { opacity:0; }
  100% { opacity:1; }
}
```

```
.animate-flicker {
  -webkit-animation: flickerAnimation 2s infinite;
  -moz-animation: flickerAnimation 2s infinite;
  -o-animation: flickerAnimation 2s infinite;
  animation: flickerAnimation 2s infinite;

  margin: 80px auto;
}

#table {
  background-image: url("../src/assets/images/table.png");
  background-size: 1200px 600px;
  display: block;
  margin: 40px auto;
  /* border: 5px solid white; */
}

.instructions {
  padding: 30px;
  text-align: center;
  font-size: 35px;
  background-color: black;
  width: 60%;
  margin: 50px auto;
  height: 720;
  border-radius: 10px;
  box-shadow: 0px 20px 10px black;
}

ul {
  font-size: 25px;
  margin: auto;
  width: 75%;
  text-align: left;
  list-style: circle
}

li {
  padding: 5px;
}

#table.hidden, .info.hidden, .instructions.hidden{
  display: none;
```

```
}

.info {
  width: 1200px;
  height: 250px;
  display: flex;
  flex-direction: column;
  margin: auto;
  border: 10px solid rgb(201, 195, 195);
}

.info-top {
  border-bottom: 3px solid white;
}

.info-bottom {
  display: flex;
  width: auto;
  height: 250px;
  justify-content: space-between;
  /* background-color: red; */
}

.info-bottom img {
  width: 40px;
  height: 40px;
}

.info-bottom-left {
  display: flex;
  width: 500px;
  padding: 20px;
  border-right: 3px solid white;
}

.info-bottom-right {
  display: flex;
  width: 500px;
  padding: 20px;
  border-left: 3px solid white;
}

.info button {
  background-color: #964175;
```

```
  border-radius: 5px;
  color: white;
  padding: 15px 5px;
  text-align: center;
  text-decoration: none;
  display: block;
  font-size: 15px;
  cursor: pointer;
  margin: 10px;
  margin-top: 15px;
}

.info button:hover {
  background-color: #75305b;
  transition: 200ms;
}

.info p {
  text-align: center;
  font-size: 35px;
  font-family: serif;
  font-weight: bold;
}

.socials {
  display: flex;
  opacity: 50%;
}

.socials img:hover {
  opacity: 50%;
  transition: 200ms;
}

.socials img {
  padding-left: 12px;
}

.unselectable {
  -webkit-touch-callout: none;
  -webkit-user-select: none;
  -khtml-user-select: none;
  -moz-user-select: none;
  -ms-user-select: none;
```

```
  user-select: none;
}
```