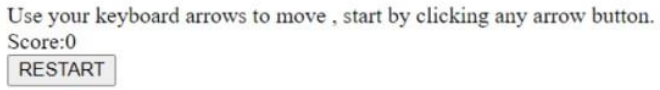
WEB APPLICATION OF SNAKE GAME

HTML CODE:

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

<head>
   <!-- Required meta tags -->
   <meta charset="utf-8">
   <meta name="viewport" content="width=device-width, initial-scale=1">

   <!-- CSS -->
   <link rel="stylesheet" href="snake.css">

   <title>Snake Game web application
   </title>
</head>

<body>
   <div class="container">
      <div class="instructions"> Use your keyboard arrows to move , start by clicking any arrow
button.
      </div>
      <div class="game">
      </div>
      <div class="game_controls">
         <div class="score_container">
            Score:<span class="score">0</span>
         </div>
         <button id="restart_game" type="button">RESTART</button>
      </div>
   </div>

   <script src="snake.js"></script>
</body>

</html>
```

First output srceen:



CSS CODE:

```
@import
url("https://fonts.googleapis.com/css2?family=Poppins&family=Potta+One&display=swap");

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  font-family: "Poppins", sans-serif;
}

body {
  background: #131212;
  color: rgb(194, 74, 74);
  display: flex;
  align-items: center;
  justify-content: center;
}
```

```
html,
body {
  margin: 0;
  height: 100%;
  display: flex;
  justify-content: center;
  align-items: center;
  font-family: sans-serif;
}

.container {
  position: relative;
  padding: 20px;
}

.score_container {
  font-size: 16px;
  text-align: center;
  margin: 10px 0;
}

#restart_game {
  margin: 0 auto;
  display: block;
  padding: 9px 5px;
  background: #212121;
  color: #00ff80;
  font-weight: bold;
  border: none;
}

.title {
  font-size: 16px;
}

.game {
  height: 480px;
  width: 480px;
  background: #e0e0e0;
  margin: 0 auto;
}

.tile {
  float: left;
```

```
  margin: 0;
  content: "";
  background: #e0e0e0;
  box-sizing: border-box;
  position: relative;
  display: block;
  box-shadow: inset 0px 0px 0px 1px rgba(0, 134, 255, 0.05);
}

.body {
  background: #997b7b;
  border: 2px solid #0e0d0d;
  border-radius: 3px;
}

.head {
  background: rgb(126, 3, 3);
}

.fruit {
  background: #00b35a;
  border-radius: 5px;
}

.instructions {
  font-size: 13px;
  color: #f1f0f0;
  padding-bottom: 20px;
}
```

JAVASCRIPT CODE:

```
var direction;
var tilesNum = 225;
var tilesPerRow = Math.sqrt(tilesNum);
var rowStartLeft = new Array();
var rowStartTop = new Array();
var rowEndBottom = new Array();
var rowEndRight = new Array();
var emptyTiles = new Array();
```

```
var body = [3, 2, 1];
var moving;
var fruitGenerator;
var powerGenerator;
var gameDiv = document.getElementsByClassName('game')[0];
var boxDimensions = (100 / tilesPerRow).toFixed(3);
var restartButton = document.getElementById('restart_game');
var scoreSpan = document.getElementsByClassName('score')[0];
var score = 0;
var speed = 0.1;

restartButton.addEventListener("click", function () {
   restartGame();
}, false);

function createGrid() {
   for (var i = 1; i <= tilesNum; i++) {
      gameDiv.innerHTML = gameDiv.innerHTML + '<div class="tile" data-tile="' + i + '"
style="width:' + boxDimensions + '%; height:' + boxDimensions + '%"></div>';
   }
}

function createBody() {
   for (var i = 1; i <= body.length; i++) {
      if (i == 3) {
         document.querySelector('[data-tile="' + i + '"]').classList.add("head", "body");
      } else if (i == 1 || i == 2) {
         document.querySelector('[data-tile="' + i + '"]').classList.add("body");
      }
   }
}

// Array consisting of upmost left boxes
for (var i = 1; i <= tilesNum; i += tilesPerRow) {
   rowStartLeft.push(i);
}

// Array consisting of upmost right boxes
for (var i = tilesPerRow; i <= tilesNum; i += tilesPerRow) {
   rowEndRight.push(i);
}

// Array consisting of upmost top boxes
for (var i = 1; i <= tilesPerRow; i += 1) {
   rowStartTop.push(i);
```

```
}

// Array consisting of upmost bottom boxes
for (var i = (tilesNum - tilesPerRow) + 1; i <= tilesNum; i += 1) {
   rowEndBottom.push(i);
}

window.addEventListener("keydown", control, false);

function control(e) {
   // RIGHT ARROW
   if (e.keyCode == "39") {
      if (direction != 'r' && direction != 'l') {
         changeDirection('r');
      }
   }

   // LEFT ARROW
   if (e.keyCode == "37") {
      if (direction != 'l' && direction != 'r') {
         changeDirection('l');
      }
   }

   // DOWN ARROW
   if (e.keyCode == "40") {
      if (direction != 'd' && direction != 'u') {
         changeDirection('d');
      }
   }

   // UP ARROW
   if (e.keyCode == "38") {
      if (direction != 'u' && direction != 'd') {
         changeDirection('u');
      }
   }
}

function changeDirection(d) {
   var directionDeciderNum,
      directionArrayInit,
      directionArrayOf;
   switch (d) {
      case "r":
```

```
            directionDeciderNum = 1;
            directionArrayInit = rowEndRight;
            directionArrayOf = rowStartLeft;
            break;
        case "l":
            directionDeciderNum = -1;
            directionArrayInit = rowStartLeft;
            directionArrayOf = rowEndRight;
            break;
        case "d":
            directionDeciderNum = tilesPerRow;
            directionArrayInit = rowEndBottom;
            directionArrayOf = rowStartTop;
            break;
        case "u":
            directionDeciderNum = -tilesPerRow;
            directionArrayInit = rowStartTop;
            directionArrayOf = rowEndBottom;
            break;
    }

    clearInterval(moving);

    moving = setInterval(function () {
        direction = d;
        var head = document.getElementsByClassName('head')[0];
        var nextTileNum = directionArrayInit.indexOf(parseInt(head.dataset.tile, 10)) > -1 ?
directionArrayOf[directionArrayInit.indexOf(parseInt(head.dataset.tile, 10))] :
parseInt(head.dataset.tile, 10) + directionDeciderNum;
        if (body.indexOf(nextTileNum) > -1) {
            scoreSpan.innerHTML = +score + ". GAME OVER";
            restartGame();
        } else {
            var nextTile = document.querySelector('[data-tile ="' + nextTileNum + '"]');

            var lastTile = document.querySelector('[data-tile ="' + body[body.length - 1] + '"]');
            body.unshift(nextTileNum);

            nextTile.classList.add("head", "body");

            // IF EATEN FRUIT
            if (nextTile.classList.contains('fruit')) {
                score += 1;
                scoreSpan.innerHTML = score;
                speed = score % 2 == 0 ? speed += 0.01 : speed;
```

```
            nextTile.classList.remove('fruit');
            clearInterval(fruitGenerator);
            generateFruit();
            fruitGen();
        }

        // IF JUST MOVING
        else {
            lastTile.classList.remove("body");
            body.pop();
        };
        head.classList.remove("head");
    }
  }, 10 / speed);
}

function generateFruit() {
  var rand;
  var fruit = document.getElementsByClassName('fruit')[0];
  if (fruit) {
    fruit.classList.remove('fruit');
  }
  do {
    rand = Math.floor(Math.random() * tilesNum);
  } while (body.indexOf(rand) > -1);
  document.querySelector('[data-tile ="' + rand + '"]').classList.add('fruit');
}

function fruitGen() {
  fruitGenerator = setInterval(function () {
    generateFruit();
  }, 3000)
};

function startGame() {
  createGrid();
  createBody();
  generateFruit();
}

function restartGame() {
  scoreSpan.innerHTML = +score + ". GAME OVER";
  clearInterval(fruitGenerator);
  clearInterval(moving);
  body = [3, 2, 1];
```

```
    speed = 0.08;
    score = 0;
    document.querySelector('.game').innerHTML = "";
    direction = '';
    startGame();
}

startGame();
```

OVERALL OUTPUT SCREEN: