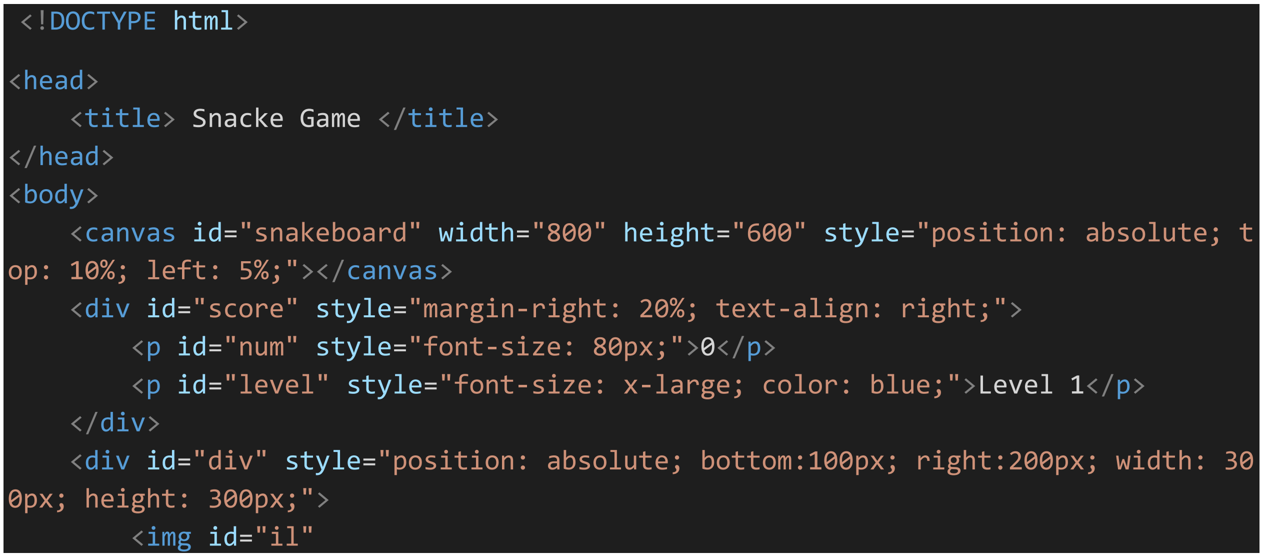
U18CO018
Shubham Shekhaliya
ITA (Assignment – 3)
Based on HTML, JavaScript, CSS and jQuery
Snake and Ball Game

Specification of the game :-

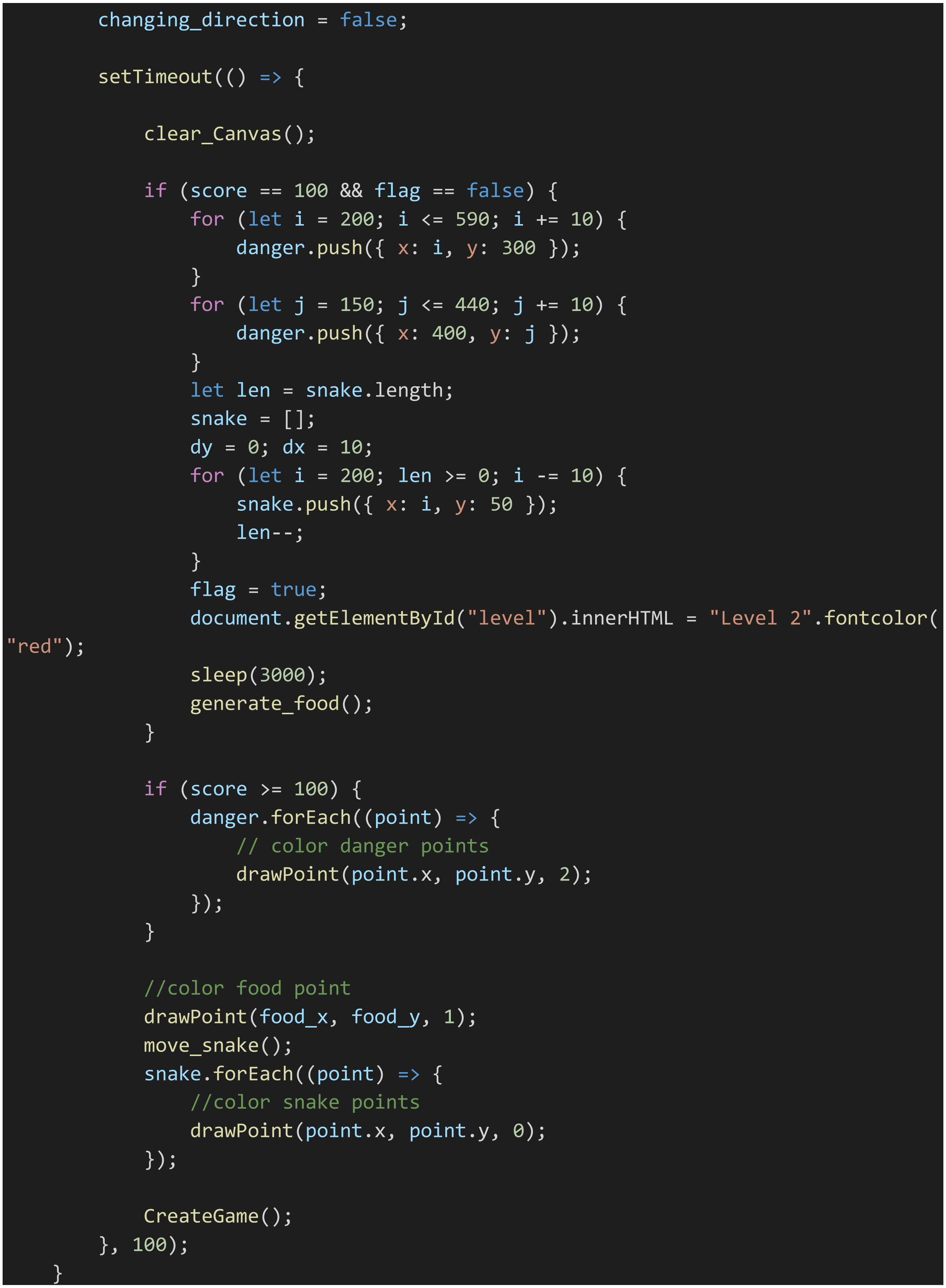
1. Layout must include snake with size four unit, ball and four buttons for directions. All
components must be clearly visible.
2. Ball should be placed at random position initially.
3. Once the ball is grabbed by the snake, the size of the snake should be incremented by
one unit and the score should increase by 10 units.
4. End of the Game must take place once the snake head touches the boundary wall.
5. Calculate game score continually. Once the score reaches 100 increase the level of
game. In the centre of the screen display “+” symbol with height maxy/2 and width
maxx/2. If the snake touches this “+” structure the game is over.

Code:-

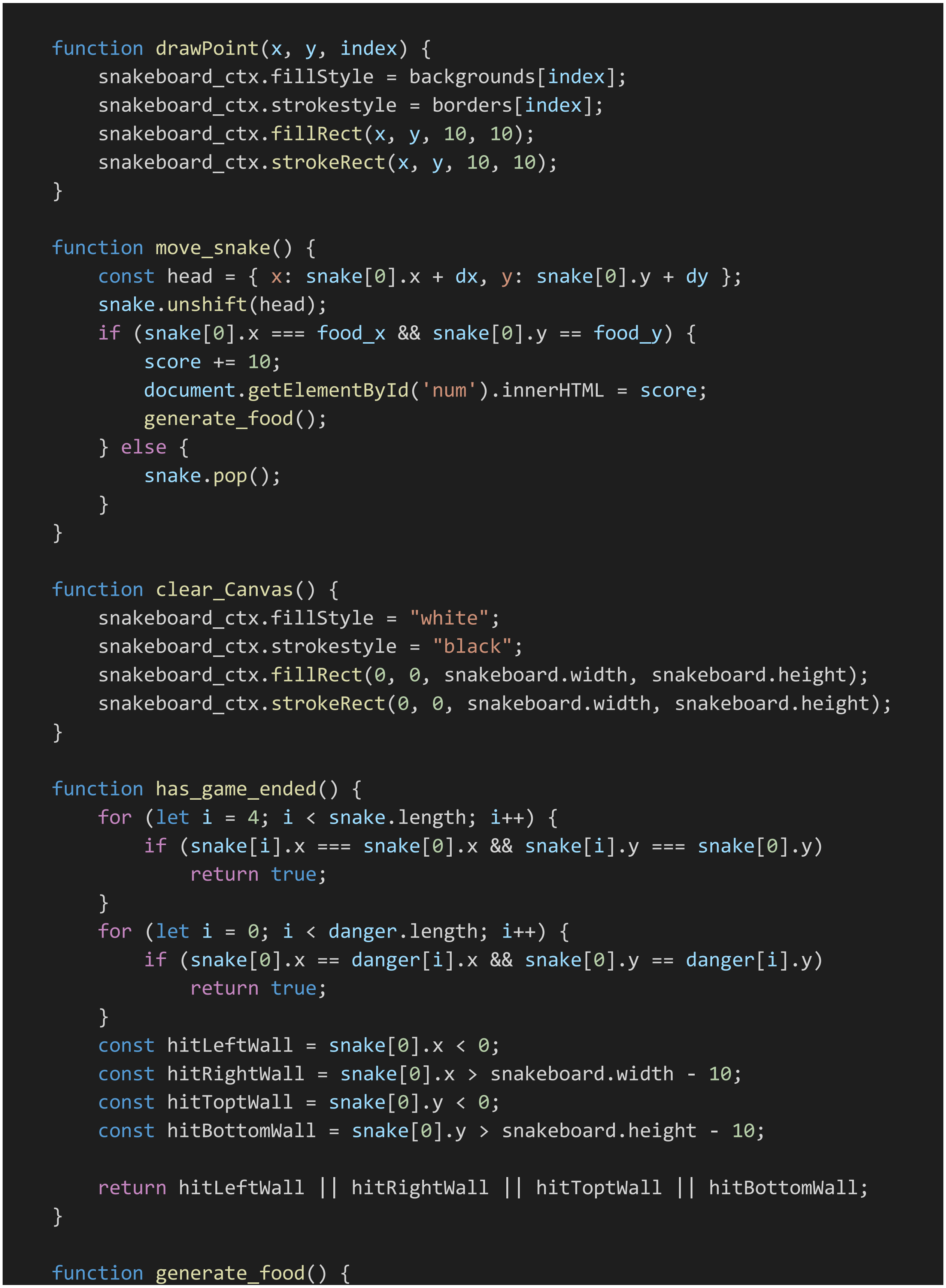


```
 <!DOCTYPE html>

<head>
    <title> Snacke Game </title>
</head>
<body>
    <canvas id="snakeboard" width="800" height="600" style="position: absolute; t
op: 10%; left: 5%;"></canvas>
    <div id="score" style="margin-right: 20%; text-align: right;">
        <p id="num" style="font-size: 80px;">0</p>
        <p id="level" style="font-size: x-large; color: blue;">Level 1</p>
    </div>
    <div id="div" style="position: absolute; bottom:100px; right:200px; width: 30
0px; height: 300px;">
        <img id="il"
```

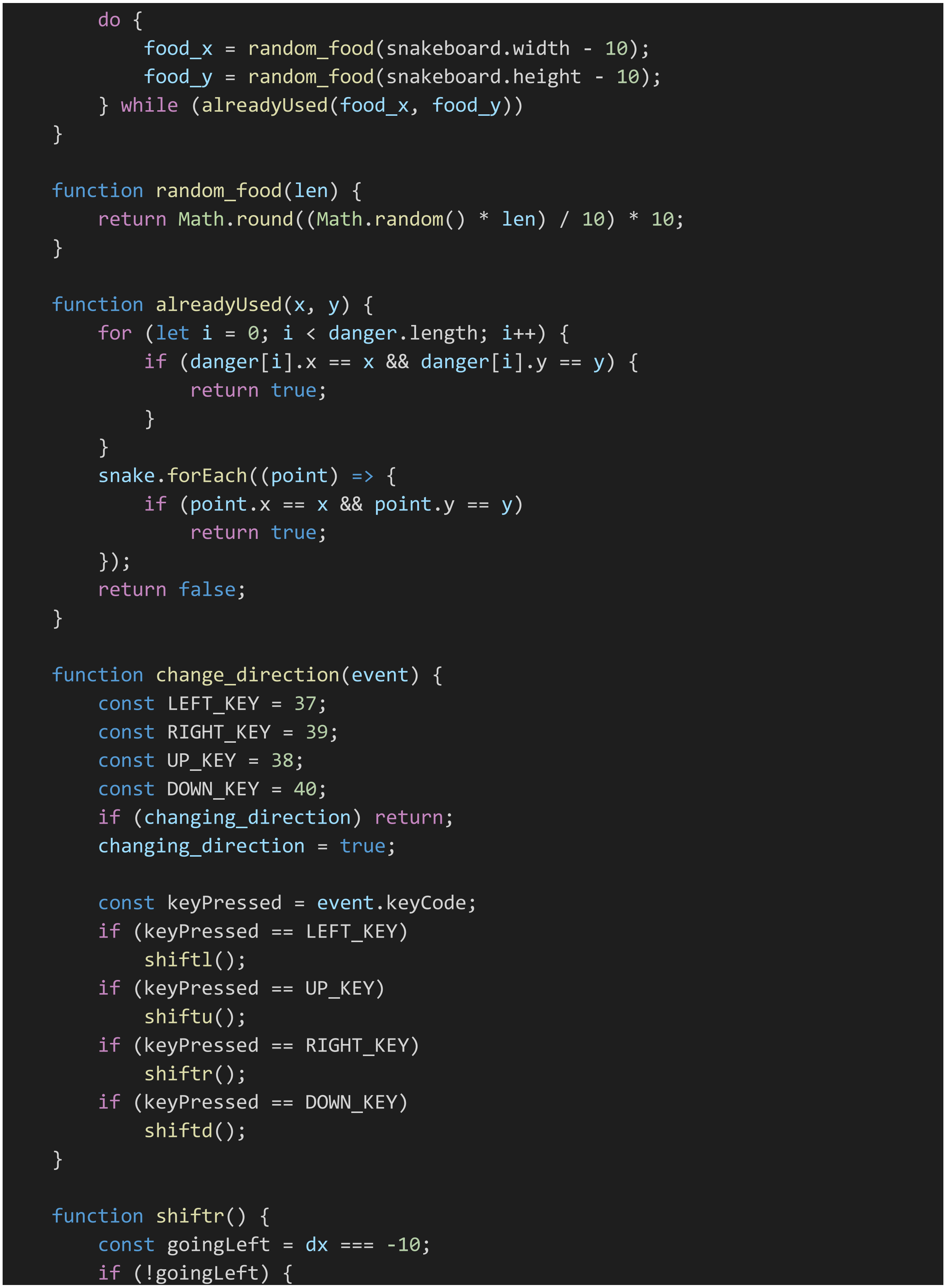
```
            style="position:relative; width: 120px; height: 60px; transform: rotate(180deg); left:0px; top: 120px; "
            src="arrow.png" onclick="shiftl()">
        <img id="iu"
            style="position:relative; width: 120px; height: 60px; transform: rotate(270deg); left: -33px; top: 30px; "
            src="arrow.png" onclick="shiftu()">
        <img id="ir" style="position:relative; width: 120px; height: 60px; left:180px; top: 57px; " src="arrow.png"
            onclick="shiftr()">
        <img id="id"
            style="position:relative; width: 120px; height: 60px; transform: rotate(90deg); left: -33px; top: 145px;"
            src="arrow.png" onclick="shiftd()">
    </div>
</body>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
    const snakeboard = document.getElementById("snakeboard");
    const snakeboard_ctx = snakeboard.getContext("2d");

    let snake = [{ x: 200, y: 200 }, { x: 190, y: 200 }, { x: 180, y: 200 }, { x: 170, y: 200 }];

    let danger = [];

    const borders = ['darkblue', 'darkgreen', 'darkred'];
    const backgrounds = ['lightblue', 'lightgreen', 'red'];

    let score = 0, changing_direction = false, dx = 10, dy = 0, flag = false, food_x, food_y;

    CreateGame();
    generate_food();

    document.addEventListener("keydown", change_direction);

    function CreateGame() {
        if (has_game_ended()) {
            document.getElementById("level").innerHTML = "Game Over".fontcolor("red");
            return;
        }
```



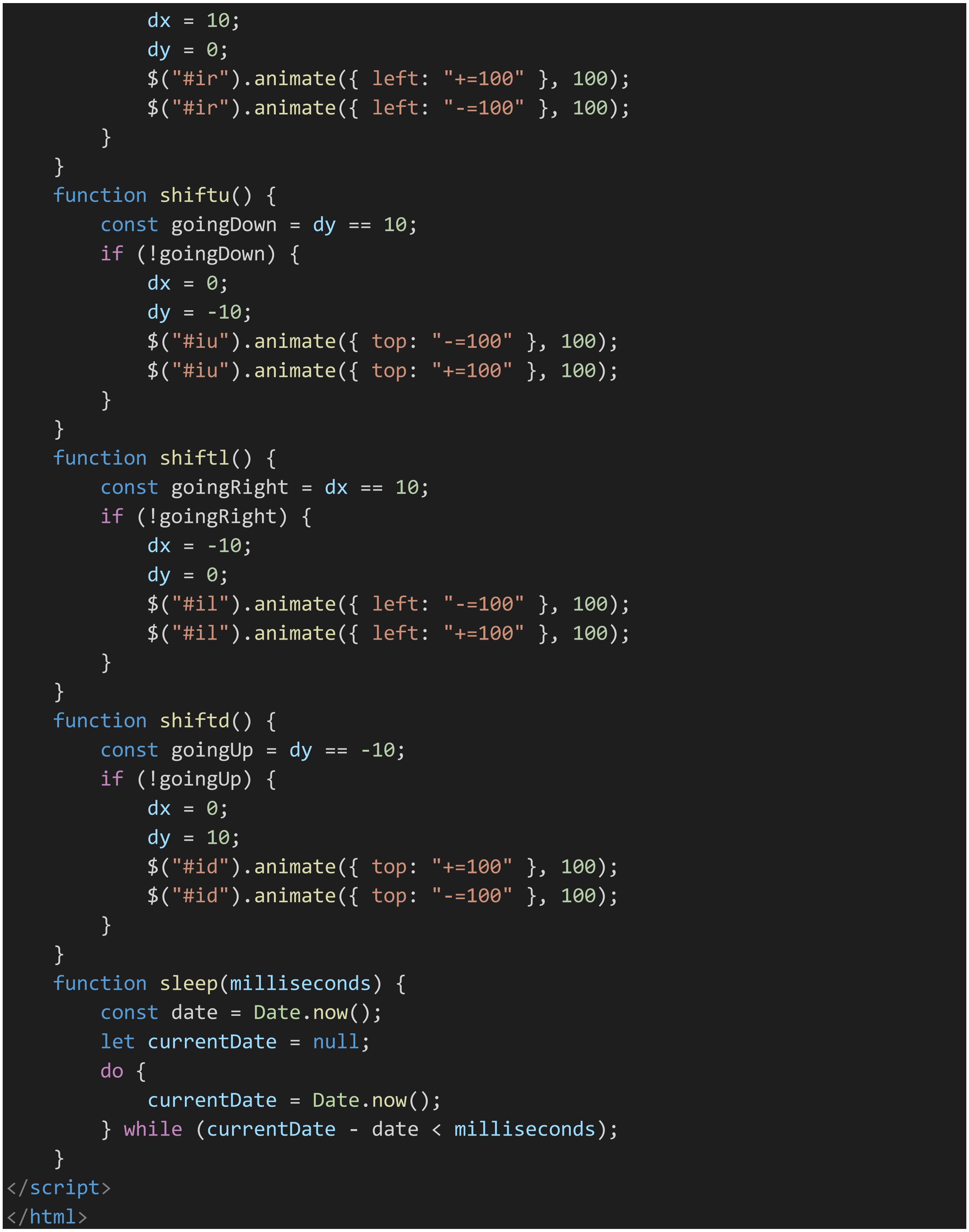
```
        changing_direction = false;

        setTimeout(() => {

            clear_Canvas();

            if (score == 100 && flag == false) {
                for (let i = 200; i <= 590; i += 10) {
                    danger.push({ x: i, y: 300 });
                }
                for (let j = 150; j <= 440; j += 10) {
                    danger.push({ x: 400, y: j });
                }
                let len = snake.length;
                snake = [];
                dy = 0; dx = 10;
                for (let i = 200; len >= 0; i -= 10) {
                    snake.push({ x: i, y: 50 });
                    len--;
                }
                flag = true;
                document.getElementById("level").innerHTML = "Level 2".fontcolor(
"red");
                sleep(3000);
                generate_food();
            }

            if (score >= 100) {
                danger.forEach((point) => {
                    // color danger points
                    drawPoint(point.x, point.y, 2);
                });
            }

            //color food point
            drawPoint(food_x, food_y, 1);
            move_snake();
            snake.forEach((point) => {
                //color snake points
                drawPoint(point.x, point.y, 0);
            });

            CreateGame();
        }, 100);
    }
```



```
function drawPoint(x, y, index) {
    snakeboard_ctx.fillStyle = backgrounds[index];
    snakeboard_ctx.strokestyle = borders[index];
    snakeboard_ctx.fillRect(x, y, 10, 10);
    snakeboard_ctx.strokeRect(x, y, 10, 10);
}

function move_snake() {
    const head = { x: snake[0].x + dx, y: snake[0].y + dy };
    snake.unshift(head);
    if (snake[0].x === food_x && snake[0].y == food_y) {
        score += 10;
        document.getElementById('num').innerHTML = score;
        generate_food();
    } else {
        snake.pop();
    }
}

function clear_Canvas() {
    snakeboard_ctx.fillStyle = "white";
    snakeboard_ctx.strokestyle = "black";
    snakeboard_ctx.fillRect(0, 0, snakeboard.width, snakeboard.height);
    snakeboard_ctx.strokeRect(0, 0, snakeboard.width, snakeboard.height);
}

function has_game_ended() {
    for (let i = 4; i < snake.length; i++) {
        if (snake[i].x === snake[0].x && snake[i].y === snake[0].y)
            return true;
    }
    for (let i = 0; i < danger.length; i++) {
        if (snake[0].x == danger[i].x && snake[0].y == danger[i].y)
            return true;
    }
    const hitLeftWall = snake[0].x < 0;
    const hitRightWall = snake[0].x > snakeboard.width - 10;
    const hitToptWall = snake[0].y < 0;
    const hitBottomWall = snake[0].y > snakeboard.height - 10;

    return hitLeftWall || hitRightWall || hitToptWall || hitBottomWall;
}

function generate_food() {
```

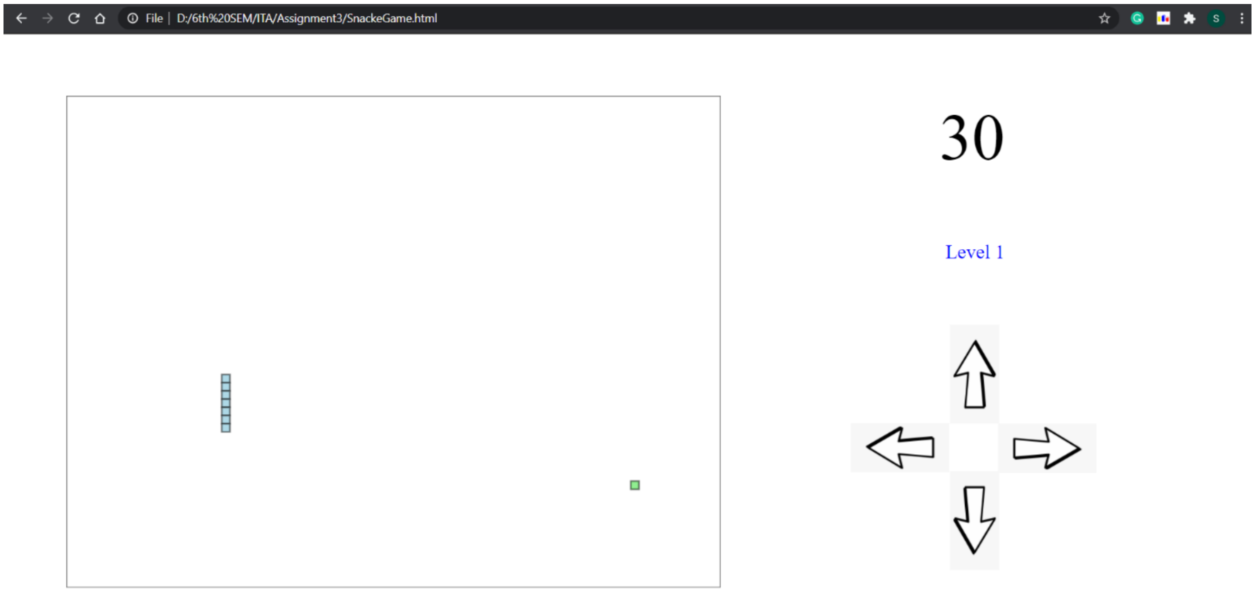


```
        do {
            food_x = random_food(snakeboard.width - 10);
            food_y = random_food(snakeboard.height - 10);
        } while (alreadyUsed(food_x, food_y))
    }

    function random_food(len) {
        return Math.round((Math.random() * len) / 10) * 10;
    }

    function alreadyUsed(x, y) {
        for (let i = 0; i < danger.length; i++) {
            if (danger[i].x == x && danger[i].y == y) {
                return true;
            }
        }
        snake.forEach((point) => {
            if (point.x == x && point.y == y)
                return true;
        });
        return false;
    }

    function change_direction(event) {
        const LEFT_KEY = 37;
        const RIGHT_KEY = 39;
        const UP_KEY = 38;
        const DOWN_KEY = 40;
        if (changing_direction) return;
        changing_direction = true;

        const keyPressed = event.keyCode;
        if (keyPressed == LEFT_KEY)
            shiftl();
        if (keyPressed == UP_KEY)
            shiftu();
        if (keyPressed == RIGHT_KEY)
            shiftr();
        if (keyPressed == DOWN_KEY)
            shiftd();
    }

    function shiftr() {
        const goingLeft = dx === -10;
        if (!goingLeft) {
```



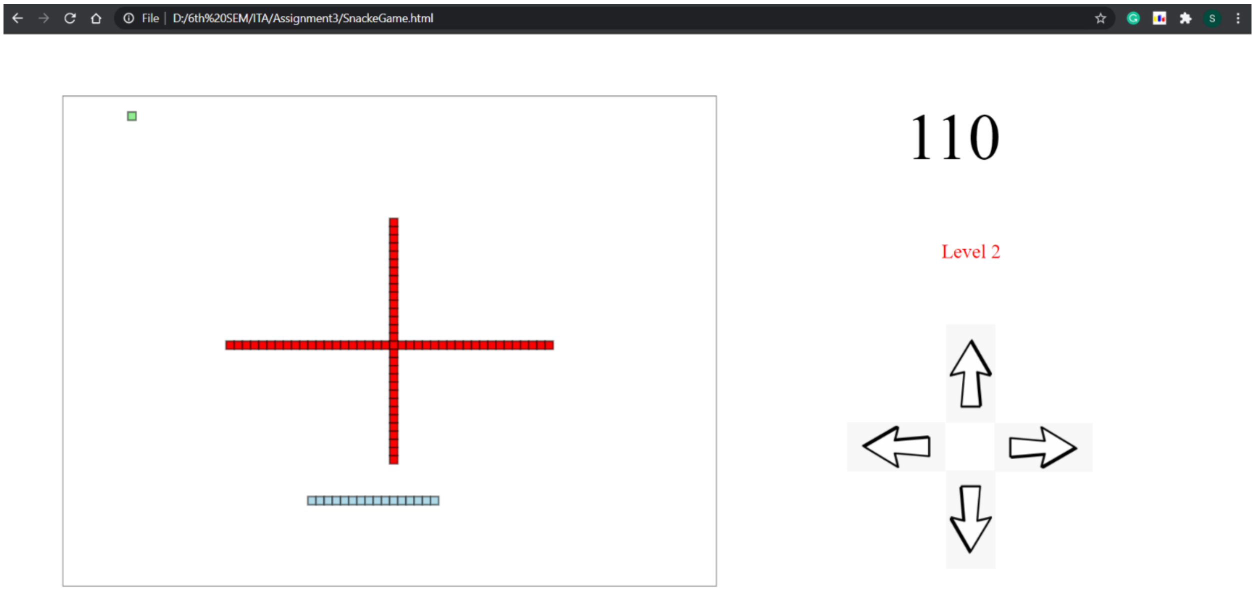
```
            dx = 10;
            dy = 0;
            $("#ir").animate({ left: "+=100" }, 100);
            $("#ir").animate({ left: "-=100" }, 100);
        }
    }
    function shiftu() {
        const goingDown = dy == 10;
        if (!goingDown) {
            dx = 0;
            dy = -10;
            $("#iu").animate({ top: "-=100" }, 100);
            $("#iu").animate({ top: "+=100" }, 100);
        }
    }
    function shiftl() {
        const goingRight = dx == 10;
        if (!goingRight) {
            dx = -10;
            dy = 0;
            $("#il").animate({ left: "-=100" }, 100);
            $("#il").animate({ left: "+=100" }, 100);
        }
    }
    function shiftd() {
        const goingUp = dy == -10;
        if (!goingUp) {
            dx = 0;
            dy = 10;
            $("#id").animate({ top: "+=100" }, 100);
            $("#id").animate({ top: "-=100" }, 100);
        }
    }
    function sleep(milliseconds) {
        const date = Date.now();
        let currentDate = null;
        do {
            currentDate = Date.now();
        } while (currentDate - date < milliseconds);
    }
</script>
</html>
```

Output :-



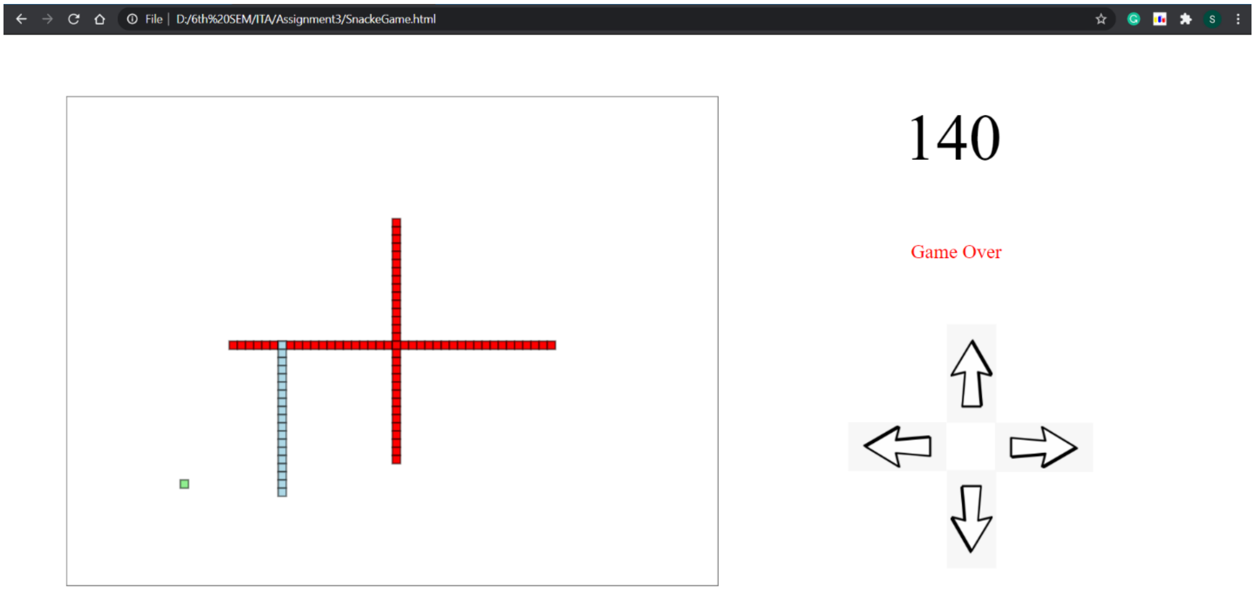
30

Level 1



110

Level 2



140

Game Over