PROJECT NO 1

Mini Project using HTML, CSS and JavaScript

AIM: To develop a dynamic game using HTML, CSS, and JavaScript.

SOURCE CODE:

HTML

```
<!DOCTYPE html>
<html>

<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <script src="./scripts/js/game.js"></script>
    <style>
        canvas {
            border: 1px solid blue;
            background-color: lightsalmon;
        }
    </style>
</head>

<body onload="startGame()">
    <br><br>
    <button onmousedown="accelerate(-0.3)"
onmouseup="accelerate(0.05)">ACCELERATE</button>
    <p>Use the ACCELERATE button to stay in the air</p>
    <p>How long can you stay alive?</p>
</body>

</html>
```

JAVASCRIPT

```
var myGamePiece;
var myObstacles = [];
var myScore;

function startGame() {
    myGamePiece = new component(30, 30, "red", 10, 120);
    myGamePiece.gravity = 0.05;
    myScore = new component("30px", "Consolas", "black", 280, 40, "text");
    myGameArea.start();
}

var myGameArea = {
```

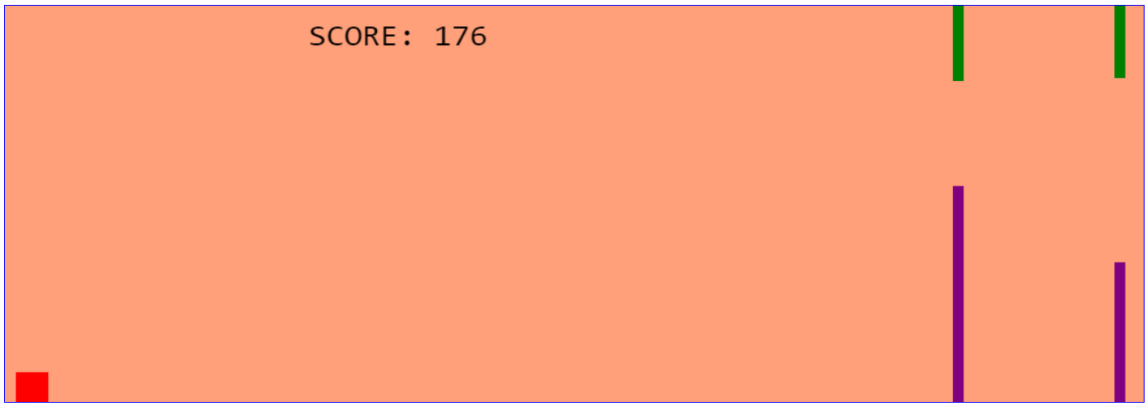
```
    canvas : document.createElement("canvas"),
    start : function() {
        this.canvas.width = 1050;
        this.canvas.height =400 ;
        this.context = this.canvas.getContext("2d");
        document.body.insertBefore(this.canvas, document.body.childNodes[0]);
        this.frameNo = 0;
        this.interval = setInterval(updateGameArea, 20);
        },
    clear : function() {
        this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);
    }
}

function component(width, height, color, x, y, type) {
    this.type = type;
    this.score = 0;
    this.width = width;
    this.height = height;
    this.speedX = 0;
    this.speedY = 0;
    this.x = x;
    this.y = y;
    this.gravity = 0;
    this.gravitySpeed = 0;
    this.update = function() {
        ctx = myGameArea.context;
        if (this.type == "text") {
            ctx.font = this.width + " " + this.height;
            ctx.fillStyle = color;
            ctx.fillText(this.text, this.x, this.y);
        } else {
            ctx.fillStyle = color;
            ctx.fillRect(this.x, this.y, this.width, this.height);
        }
    }
    this.newPos = function() {
        this.gravitySpeed += this.gravity;
        this.x += this.speedX;
        this.y += this.speedY + this.gravitySpeed;
        this.hitBottom();
    }
    this.hitBottom = function() {
        var rockbottom = myGameArea.canvas.height - this.height;
        if (this.y > rockbottom) {
            this.y = rockbottom;
            this.gravitySpeed = 0;
        }
    }
    this.crashWith = function(otherobj) {
        var myleft = this.x;
        var myright = this.x + (this.width);
        var mytop = this.y;
        var mybottom = this.y + (this.height);
        var otherleft = otherobj.x;
```

```
        var otherright = otherobj.x + (otherobj.width);
        var othertop = otherobj.y;
        var otherbottom = otherobj.y + (otherobj.height);
        var crash = true;
        if ((mybottom < othertop) || (mytop > otherbottom) || (myright < otherleft)
|| (myleft > otherright)) {
            crash = false;
        }
        return crash;
    }
}

function updateGameArea() {
    var x, height, gap, minHeight, maxHeight, minGap, maxGap;
    for (i = 0; i < myObstacles.length; i += 1) {
        if (myGamePiece.crashWith(myObstacles[i])) {
            return;
        }
    }
    myGameArea.clear();
    myGameArea.frameNo += 1;
    if (myGameArea.frameNo == 1 || everyinterval(150)) {
        x = myGameArea.canvas.width;
        minHeight = 20;
        maxHeight = 200;
        height = Math.floor(Math.random()*(maxHeight-minHeight+1)+minHeight);
        minGap = 50;
        maxGap = 200;
        gap = Math.floor(Math.random()*(maxGap-minGap+1)+minGap);
        myObstacles.push(new component(10, height, "green", x, 0));
        myObstacles.push(new component(10, x - height - gap, "purple", x, height +
gap));
    }
    for (i = 0; i < myObstacles.length; i += 1) {
        myObstacles[i].x += -1;
        myObstacles[i].update();
    }
    myScore.text="SCORE: " + myGameArea.frameNo;
    myScore.update();
    myGamePiece.newPos();
    myGamePiece.update();
}
function everyinterval(n) {
    if ((myGameArea.frameNo / n) % 1 == 0) {return true;}
    return false;
}
function accelerate(n) {
    myGamePiece.gravity = n;
}
```

OUTPUT SCREENSHOT:



ACCELERATE

Use the ACCELERATE button to stay in the air