

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="stylesheet" href="game.css">
  <title>Bounce Ball</title>
</head>

<body>

  <canvas id="game" width="650" height="450"></canvas>

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

</html>
-----
```

Java Script

```
let canvas = document.getElementById('game'),
    ctx = canvas.getContext('2d'),
    ballRadius = 9,
    x = canvas.width / (Math.floor(Math.random() * Math.random() * 10) +
3),
    y = canvas.height - 40,
    dx = 2,
    dy = -2;

let paddleHeight = 12,
    paddleWidth = 72;

// Paddle start position
let paddleX = (canvas.width - paddleWidth) / 2;

// Bricks
let rowCount = 5,
    columnCount = 9,
    brickWidth = 54,
    brickHeight = 18,
    brickPadding = 12,
    topOffset = 40,
    leftOffset = 33,
    score = 0;

// Bricks array
let bricks = [];
for (let c = 0; c < columnCount; c++) {
  bricks[c] = [];
  for (let r = 0; r < rowCount; r++) {
    // Set position of bricks
    bricks[c][r] = { x: 0, y: 0, status: 1 };
  }
}
```

```

}

// Mouse moving eventListener and function
document.addEventListener("mousemove", mouseMoveHandler, false);

// Move paddle with mouse
function mouseMoveHandler(e) {
    var relativeX = e.clientX - canvas.offsetLeft;
    if (relativeX > 0 && relativeX < canvas.width) {
        paddleX = relativeX - paddleWidth / 2;
    }
}

// Draw paddle
function drawPaddle() {
    ctx.beginPath();
    ctx.roundRect(paddleX, canvas.height - paddleHeight, paddleWidth,
paddleHeight, 30);
    ctx.fillStyle = '#333';
    ctx.fill();
    ctx.closePath();
}

// Draw ball
function drawBall() {
    ctx.beginPath();
    ctx.arc(x, y, ballRadius, 0, Math.PI * 2);
    ctx.fillStyle = '#333';
    ctx.fill();
    ctx.closePath();
}

// Draw Bricks
function drawBricks() {
    for (let c = 0; c < columnCount; c++) {
        for (let r = 0; r < rowCount; r++) {
            if (bricks[c][r].status === 1) {
                let brickX = (c * (brickWidth + brickPadding)) +
leftOffset;
                let brickY = (r * (brickHeight + brickPadding)) +
topOffset;
                bricks[c][r].x = brickX;
                bricks[c][r].y = brickY;
                ctx.beginPath();
                ctx.roundRect(brickX, brickY, brickWidth, brickHeight,
30);
                ctx.fillStyle = '#333';
                ctx.fill();
                ctx.closePath();
            }
        }
    }
}

// Track score
function trackScore() {
    ctx.font = 'bold 16px sans-serif';
    ctx.fillStyle = '#333';

```

```

    ctx.fillText('Score : ' + score, 8, 24);
}

// Check ball hit bricks
function hitDetection() {
    for (let c = 0; c < columnCount; c++) {
        for (let r = 0; r < rowCount; r++) {
            let b = bricks[c][r];
            if (b.status === 1) {
                if (x > b.x && x < b.x + brickWidth && y > b.y && y < b.y
+ brickHeight) {
                    dy = -dy;
                    b.status = 0;
                    score++;
                    // Check win
                    if (score === rowCount * columnCount) {
                        alert('You Win!');
                        document.location.reload();
                    }
                }
            }
        }
    }
}

// Main function
function init() {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    trackScore();
    drawBricks();
    drawBall();
    drawPaddle();
    hitDetection();

    // Detect left and right walls
    if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
        dx = -dx;
    }

    // Detect top wall
    if (y + dy < ballRadius) {
        dy = -dy;
    } else if (y + dy > canvas.height - ballRadius) {
        // Detect paddle hits
        if (x > paddleX && x < paddleX + paddleWidth) {
            dy = -dy;
        } else {
            // If ball don't hit paddle
            alert('Nice Try The Game Is Over!');
            document.location.reload();
        }
    }

    // Bottom wall
    if (y + dy > canvas.height - ballRadius || y + dy < ballRadius) {
        dy = -dy;
    }
}

```

```
        // Move Ball
        x += dx;
        y += dy;
    }
```

```
setInterval(init, 10);
```

```
-----
-----
```

```
styles css
```

```
body{
    height: 100%;
    display: flex;
    align-items: center;
    justify-content: center;
    background-color: #efefef;
}
```

```
#game{
    background-color: #ccc;
    margin-top: 10rem;
    border: 2px solid #333;
}
```

```
-----
-----
```

Project Code Copyright By Souraj Krishna