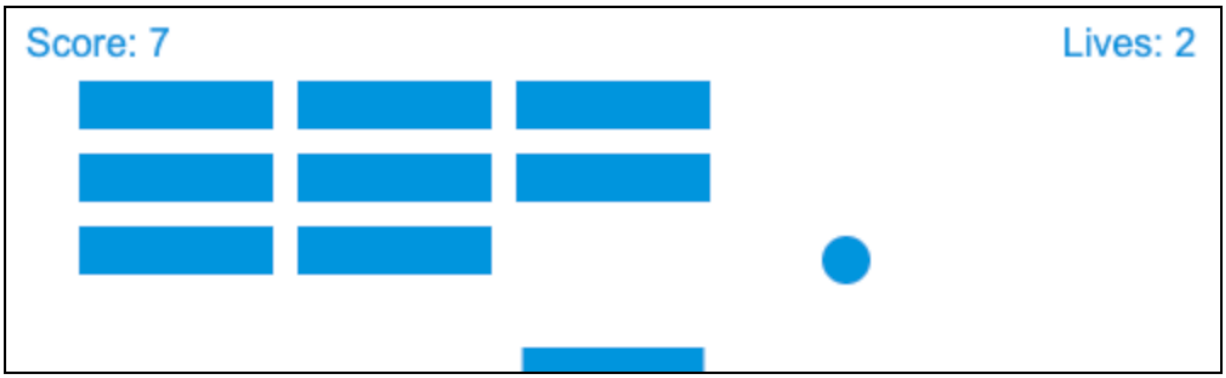
JavaScript BreakOut Game



This code is an HTML file that contains a canvas element with an id of "canvas" and some JavaScript code to create a game of Brick Breaker. The canvas element is used to display the game graphics and the JavaScript code is used to handle user input, move the game elements, and detect collisions between the ball and the bricks.

The JavaScript code initializes variables for the game elements, including the ball, paddle, and bricks, and adds event listeners to handle user input from the keyboard and mouse. There are functions to draw the game elements on the canvas, including the ball, paddle, bricks, score, and lives. There is also a function to detect collisions between the ball and the bricks.

The game loop is handled by the draw function, which is called repeatedly using requestAnimationFrame. This function updates the position of the ball, checks for collisions, and redraws the game elements on the canvas. If the player breaks all the bricks, they win the game. If the player loses all their lives, they lose the game. The game can be restarted by reloading the page.

```
<!DOCTYPE html>
```

```
<html>

<head>
  <title>Nav Menu</title>
  <style>
    canvas {
      border: 1px solid black;
    }
  </style>
</head>

<body>
  <canvas id="canvas" width="500px"></canvas>
  <script>
    // Initialize variables
    var canvas = document.getElementById("canvas");
    var ctx = canvas.getContext("2d");
    var x = canvas.width / 2;
    var y = canvas.height - 30;
    var dx = 2;
    var dy = -2;
    var ballRadius = 10;
    var paddleHeight = 10;
    var paddleWidth = 75;
    var paddleX = (canvas.width - paddleWidth) / 2;
    var rightPressed = false;
    var leftPressed = false;
    var brickRowCount = 3;
    var brickColumnCount = 5;
```

```
var brickWidth = 80;
var brickHeight = 20;
var brickPadding = 10;
var brickOffsetTop = 30;
var brickOffsetLeft = 30;
var score = 0;
var lives = 3;
var bricks = [];
for (var c = 0; c < brickColumnCount; c++) {
    bricks[c] = [];
    for (var r = 0; r < brickRowCount; r++) {
        bricks[c][r] = { x: 0, y: 0, status: 1 };
    }
}

// Add event listeners
document.addEventListener("keydown", keyDownHandler, false);
document.addEventListener("keyup", keyUpHandler, false);
document.addEventListener("mousemove", mouseMoveHandler, false);

// Functions to handle events
function keyDownHandler(e) {
    if (e.keyCode == 39) {
        rightPressed = true;
    }
    else if (e.keyCode == 37) {
        leftPressed = true;
    }
}
```

```
    function keyUpHandler(e) {
        if (e.keyCode == 39) {
            rightPressed = false;
        }
        else if (e.keyCode == 37) {
            leftPressed = false;
        }
    }

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

    function collisionDetection() {
        for (var c = 0; c < brickColumnCount; c++) {
            for (var r = 0; r < brickRowCount; r++) {
                var 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++;
                        if (score == brickRowCount * brickColumnCount) {
                            alert("You win!");
```

```
                    document.location.reload();
                }
            }
        }
    }
}

// Draw functions
function drawBall() {
    ctx.beginPath();
    ctx.arc(x, y, ballRadius, 0, Math.PI * 2);
    ctx.fillStyle = "#0095DD";
    ctx.fill();
    ctx.closePath();
}

function drawPaddle() {
    ctx.beginPath();
    ctx.rect(paddleX, canvas.height - paddleHeight, paddleWidth,
paddleHeight);
    ctx.fillStyle = "#0095DD";
    ctx.fill();
    ctx.closePath();
}
// Draw functions (continued)
function drawBricks() {
    for (var c = 0; c < brickColumnCount; c++) {
        for (var r = 0; r < brickRowCount; r++) {
```

```
                if (bricks[c][r].status == 1) {
                    var brickX = (c * (brickWidth + brickPadding)) +
brickOffsetLeft;
                    var brickY = (r * (brickHeight + brickPadding)) +
brickOffsetTop;
                    bricks[c][r].x = brickX;
                    bricks[c][r].y = brickY;
                    ctx.beginPath();
                    ctx.rect(brickX, brickY, brickWidth, brickHeight);
                    ctx.fillStyle = "#0095DD";
                    ctx.fill();
                    ctx.closePath();
                }
            }
        }
    }

    function drawScore() {
        ctx.font = "16px Arial";
        ctx.fillStyle = "#0095DD";
        ctx.fillText("Score: " + score, 8, 20);
    }

    function drawLives() {
        ctx.font = "16px Arial";
        ctx.fillStyle = "#0095DD";
        ctx.fillText("Lives: " + lives, canvas.width - 65, 20);
    }
```

```
function draw() {
    ctx.clearRect(0, 0, canvas.width, canvas.height);
    drawBricks();
    drawBall();
    drawPaddle();
    drawScore();
    drawLives();
    collisionDetection();

    // Bounce ball off walls and paddle
    if (x + dx > canvas.width - ballRadius || x + dx < ballRadius) {
        dx = -dx;
    }
    if (y + dy < ballRadius) {
        dy = -dy;
    }
    else if (y + dy > canvas.height - ballRadius) {
        if (x > paddleX && x < paddleX + paddleWidth) {
            dy = -dy;
        }
        else {
            lives--;
            if (!lives) {
                alert("Game over!");
                document.location.reload();
            }
            else {
                x = canvas.width / 2;
                y = canvas.height - 30;
```

```
                dx = 2;
                dy = -2;
                paddleX = (canvas.width - paddleWidth) / 2;
            }
        }
    }

    // Move paddle
    if (rightPressed && paddleX < canvas.width - paddleWidth) {
        paddleX += 7;
    }
    else if (leftPressed && paddleX > 0) {
        paddleX -= 7;
    }

    x += dx;
    y += dy;
    requestAnimationFrame(draw);
}

draw();
</script>
</body>

</html>
```

Explanation:

The code creates a simple game where the player has to use the paddle to bounce the ball and break the bricks. The game

consists of an HTML canvas element where the graphics are drawn using JavaScript.

The code initializes all the variables needed for the game, including the canvas, the ball, the paddle, the bricks, the score, and the lives. It also adds event listeners for keyboard and mouse input to move the paddle.

The code defines several functions to draw the game elements on the canvas, including the ball, the paddle, the bricks, the score, and the lives. It also defines a function to handle collisions between the ball and the bricks.

The main game loop is implemented in the `draw()` function, which is called repeatedly using the `requestAnimationFrame()` method. The function clears the canvas, draws the game elements, checks for collisions, and updates the position of the ball and the paddle.

The game ends when the player runs out of lives or breaks all the bricks. In either case, an alert is shown to notify the player, and the game is reset by reloading the page.