

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

JavaScript Game for beginners

Breakout HTML5 Game



Make a JavaScript Game - Create your first HTML5 Canvas
JavaScript web based game from scratch explore how make a
game

Free Course

<https://www.udemy.com/course/html5-game-canvas/>

Make your own games online - start with this amazing simple
game project creating a Breakout game from scratch

Explore how you can write code to create GAMES - Games that
run within any browser - Interactive Games that are fun to play
and even more fun to make yourself. Dynamic Games that can

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

be built upon to customize and **add your own game ideas**. See what you can build within just a few short hours.

Want to build online games - want to learn more about game creation - learn from experienced developer who has created over 100 online games. Source Code is included - try the code for yourself and customize it to make your own version of the game!!!! You will be amazed at how easy it can be.

Course covers all the basics of creating a fully functional game from start to finish!!!

- Setup your game within HTML
- Create page elements with JavaScript
- Explore how to set objects and values to prepare for game play
- Add interaction - keyboard arrow keys and mouse movement to move your character on the screen
- Add animations for smooth game movement
- Create conditions for ball behaviours, player behaviours
- Build multiple interactive items in the game
- Add collision detection to track interactions
- How to debug your game
- How to improve upon the game
- How to add bonus items,
- How to setup game start conditions and game win conditions.

Explore how you can create your first game using JavaScript and HTML5 canvas

Apply JavaScript to HTML5 canvas element - drawing on canvas lines and shapes.

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

How you can add animation to your Game with animation frames

How to make your game interactive with keyboard event listeners

Dynamic and interactive game project to practice and learn more about JavaScript and HTML5

Source Code is included

How to debug your HTML5 canvas game project

Making tweaks and adjustments adding game play , and scoring

Taught by an instructor with over 20 years of real world experience ready to help you learn more about HTML5 and JavaScript.

Start building your own version of the game TODAY.

This code is a simple implementation of a Brick Breaker game using JavaScript and HTML5 canvas element. The game consists of a canvas element and two main objects, the player and the ball. The player is controlled by the user using the keyboard or mouse, and the ball moves around the screen, bouncing off walls and bricks.

Let's break down the code step by step:

```
const canvas = document.createElement('canvas');  
canvas.style.background = 'black';  
const ctx = canvas.getContext('2d');  
document.body.prepend(canvas);
```

Here, a canvas element is created and added to the document body. The `getContext()` method is used to get a 2D drawing

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

context for the canvas, which allows us to draw graphics on it. The background color of the canvas is set to black.

```
const game = {  
  grid: 80  
  , ani: ''  
  , bricks: []  
  , num: 100  
  , gameover: true  
  , bonus: []  
};
```

An object named game is defined, which stores various game-related variables. The grid variable is used to set the size of each cell in the game, and num stores the number of bricks in the game. gameover is a boolean variable that indicates whether the game is over or not, and bricks is an array that stores the brick objects. bonus is an array that stores the bonus objects.

```
const ball = {  
  x: game.grid * 7  
  , y: game.grid * 5  
  , w: game.grid / 3  
  , h: game.grid / 3  
  , color: 'green'  
  , dx: 0  
  , dy: 0  
};
```

An object named ball is defined, which stores the position, size, and velocity of the ball. dx and dy represent the horizontal and vertical velocity of the ball, respectively.

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

```
const player = {  
  color: 'red'  
  , speed: 5  
};
```

An object named player is defined, which stores the color and speed of the player.

```
const keyz = {  
  ArrowLeft: false  
  , ArrowRight: false  
};
```

An object named keyz is defined, which stores the keyboard keys that control the player.

```
canvas.setAttribute('width', game.grid * 12);  
canvas.setAttribute('height', game.grid * 10);  
canvas.style.border = '1px solid black';  
canvas.addEventListener('click', (e) => {  
  if (game.gameover) {  
    game.gameover = false;  
    startGame();  
    game.ani = requestAnimationFrame(draw);  
  }  
  else if (!game.inplay) {  
    game.inplay = true;  
    ball.dx = 5;  
    ball.dy = -5;  
  }  
})
```

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

The size of the canvas is set using the `setAttribute()` method. A border is added to the canvas using the `style` property. An event listener is added to the canvas that listens for a click event. When the canvas is clicked, the `startGame()` function is called, and the `draw()` function is called using the `requestAnimationFrame()` method.

```
function resetBall() {
  ball.dx = 0;
  ball.dy = 0;
  ball.y = player.y - ball.h;
  ball.x = player.x + (player.w / 2);
  game.inplay = false;
}
```

```
function gameWinner() {
  game.gameover = true;
  game.inplay = false;
  cancelAnimationFrame(game.ani);
}
```

```
function gameOver() {
  game.gameover = true;
  game.in
```

The `ballmove` function checks for collisions with the walls, the player, and the bricks. If the ball collides with a brick, it removes the brick from the `game.bricks` array and adds a bonus object to the `game.bonus` array. If the ball collides with the player, it adjusts the ball's direction based on where it hits the player.

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

The drawPlayer function draws the player on the canvas using the ctx.fillRect method, with its position and dimensions based on the player object.

The drawBall function draws the ball on the canvas using the ctx.beginPath, ctx.arc, and ctx.fill methods, with its position and dimensions based on the ball object.

The drawBonus function draws a bonus object on the canvas using the ctx.fillRect method, with its position and dimensions based on the bonus object.

The movement function checks if the left or right arrow keys are pressed and adjusts the player's position accordingly. If the player reaches the edge of the canvas, it stops moving.

The checkWinner function checks if all the bricks have been removed from the game.bricks array, and if so, calls the gameWinner function.

The checkLoser function checks if the player has used up all their lives, and if so, calls the gameOver function.

Finally, the requestAnimationFrame method is called again to loop the draw function continuously, until the game is over or won.

Final Source Code Below

```
const canvas = document.createElement('canvas');  
canvas.style.background = 'black';
```

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

```
const ctx = canvas.getContext('2d');
document.body.prepend(canvas);
const game = {
  grid: 80
  , ani: ''
  , bricks: []
  , num: 100
  , gameover: true
  , bonus: []
};
const ball = {
  x: game.grid * 7
  , y: game.grid * 5
  , w: game.grid / 3
  , h: game.grid / 3
  , color: 'green'
  , dx: 0
  , dy: 0
};
const player = {
  color: 'red'
  , speed: 5
};
const keyz = {
  ArrowLeft: false
  , ArrowRight: false
};
canvas.setAttribute('width', game.grid * 12);
canvas.setAttribute('height', game.grid * 10);
canvas.style.border = '1px solid black';
canvas.addEventListener('click', (e) => {
  if (game.gameover) {
```


Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

```

    game.gameover = false;
    startGame();
    game.ani = requestAnimationFrame(draw);
  }
  else if (!game.inplay) {
    game.inplay = true;
    ball.dx = 5;
    ball.dy = -5;
  }
})
outputStartGame();
document.addEventListener('keydown', (e) => {
  if (e.code in keyz) {
    keyz[e.code] = true;
  }
  if (e.code === 'ArrowUp' && !game.inplay) {
    game.inplay = true;
    ball.dx = 5;
    ball.dy = -5;
  }
})
document.addEventListener('keyup', (e) => {
  if (e.code in keyz) {
    keyz[e.code] = false;
  }
})
document.addEventListener('mousemove', (e) => {
  const val = e.clientX - canvas.offsetLeft;
  if (val > player.w && val < canvas.width) {
    player.x = val - player.w;
    if (!game.inplay) {
      ball.x = val - (player.w / 2);
    }
  }
})

```

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

```
    }  
    //console.log(player.x);  
  }  
})
```

```
function resetBall() {  
  ball.dx = 0;  
  ball.dy = 0;  
  ball.y = player.y - ball.h;  
  ball.x = player.x + (player.w / 2);  
  game.inplay = false;  
}
```

```
function gameWinner() {  
  game.gameover = true;  
  game.inplay = false;  
  //console.log('You WON');  
  cancelAnimationFrame(game.ani);  
}
```

```
function gameOver() {  
  game.gameover = true;  
  game.inplay = false;  
  //console.log('game over press to start again');  
  cancelAnimationFrame(game.ani);  
}
```

```
function outputStartGame() {  
  let output = "Click to Start Game";  
  ctx.font = Math.floor(game.grid * 0.7) + 'px serif';  
  if (canvas.width < 900) {  
    ctx.font = '20px serif';  
  }  
}
```

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

```

}
ctx.textAlign = 'center';
ctx.fillStyle = 'yellow';
ctx.fillText(output, canvas.width / 2, canvas.height / 2);
}

```

```

function startGame() {
  game.inplay = false;
  player.x = game.grid * 7;
  player.y = canvas.height - game.grid * 2;
  player.w = game.grid * 1.5;
  player.h = game.grid / 4;
  player.lives = 0;
  player.score = 0;
  game.bonus = [];
  resetBall();
  let buffer = 10;
  let width = game.grid;
  let height = game.grid / 2;
  let totalAcross = Math.floor((canvas.width - game.grid) / (width
+ buffer));
  let xPos = game.grid / 2;
  let yPos = 0;
  //console.log(totalAcross);
  let yy = 0;
  for (let i = 0; i < game.num; i++) {
    if (i % totalAcross == 0) {
      yy++;
      yPos += height + buffer;
      xPos = game.grid / 2;
    }
    if (yy < 5) {

```

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

```

    createBrick(xPos, yPos, width, height);
  }
  xPos += width + buffer;
}
}

```

```

function createBrick(xPos, yPos, width, height) {
  let ranCol = '#' + Math.random().toString(16).substr(-6);
  game.bricks.push({
    x: xPos
    , y: yPos
    , w: width
    , h: height
    , c: ranCol
    , v: Math.floor(Math.random() * 50)
    , bonus: Math.floor(Math.random() * 3)
  });
}

```

```

function collDetection(obj1, obj2) {
  const xAxis = (obj1.x < (obj2.x + obj2.w)) && ((obj1.x +
obj1.w) > obj2.x);
  const yAxis = (obj1.y < (obj2.y + obj2.h)) && ((obj1.y +
obj1.h) > obj2.y);
  const val = xAxis && yAxis;
  return val;
}

```

```

function draw() {
  ctx.clearRect(0, 0, canvas.width, canvas.height);
  movement();
  ballmove();
}

```

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

```

drawPlayer();
drawBall();
game.bonus.forEach((prize, index) => {
  prize.y += 5;
  drawBonus(prize);
  if (collDetection(prize, player)) {
    player.score += prize.points;
    let temp = game.bonus.splice(index, 1);
    ////console.log(temp);
  }
  if (prize.y > canvas.height) {
    let temp = game.bonus.splice(index, 1);
    ////console.log(temp);
  }
})
game.bricks.forEach((brick, index) => {
  ctx.beginPath();
  ctx.fillStyle = brick.c;
  ctx.strokeStyle = 'white';
  ////console.log(brick);
  ctx.rect(brick.x, brick.y, brick.w, brick.h);
  ctx.fill();
  ctx.stroke();
  ctx.closePath();
  if (collDetection(brick, ball)) {
    let rem = game.bricks.splice(index, 1);
    player.score += brick.v;
    console.log(ball.dy);
    //ball.dy++;
    if (ball.dy > -10 && ball.dy < 10) {
      ball.dy--;
    }
  }
})

```

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

```

ball.dy *= -1;
if (brick.bonus === 1) {
  game.bonus.push({
    x: brick.x
    , y: brick.y
    , h: brick.h
    , w: brick.w
    , points: Math.ceil(Math.random() * 100) + 50
    , color: 'white'
    , alt: 'black'
    , counter: 10
  })
}
if (game.bricks.length === 0) {
  gameWinner();
}
}
})
if (collDetection(player, ball)) {
  ball.dy *= -1;
  let val1 = ball.x + (ball.w / 2) - player.x;
  let val2 = val1 - player.w / 2;
  let val3 = Math.ceil(val2 / (player.w / 10));
  ball.dx = val3;
  //console.log(val1, val2, val3);
};
let output1 = player.lives === 1 ? 'Life Left' : 'Lives Left';
let output = `${output1} : ${player.lives} Score :
${player.score}`;
ctx.font = Math.floor(game.grid * 0.7) + 'px serif';
if (game.gameover) {
  ctx.font = '24px serif';

```

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

```

    output = `Score ${player.score} GAME OVER Click to Start
Again`;
}
if (canvas.width < 900) {
    ctx.font = '20px serif';
}
ctx.textAlign = 'center';
ctx.fillStyle = 'white';
ctx.fillText(output, canvas.width / 2, canvas.height - 20);
if (!game.gameover) {
    game.ani = requestAnimationFrame(draw);
}
}

```

```

function movement() {
    if (keyz.ArrowLeft) {
        player.x -= player.speed;
    }
    if (keyz.ArrowRight) {
        player.x += player.speed;
    }
    if (!game.inplay) {
        ball.x = player.x + player.w / 2;
    }
}

```

```

function ballmove() {
    if (ball.x > canvas.width || ball.x < 0) {
        ball.dx *= -1;
    }
    if (ball.y < 0) {
        ball.dy *= -1;
    }
}

```

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

```
}
if (ball.y > canvas.height) {
  player.lives--;
  resetBall();
  if (player.lives < 0) {
    gameOver();
  }
}
if (ball.dy > -2 && ball.dy < 0) {
  ball.dy = -3;
}
if (ball.dy > 0 && ball.dy < 2) {
  ball.dy = 3;
}
ball.x += ball.dx;
ball.y += ball.dy;
}

function drawBall() {
  ctx.beginPath();
  ctx.strokeStyle = 'white';
  ctx.rect(ball.x, ball.y, ball.w, ball.h);
  //ctx.stroke();
  ctx.closePath();
  ctx.beginPath();
  ctx.fillStyle = ball.color;
  let adj = ball.w / 2;
  ctx.arc(ball.x + adj, ball.y + adj, ball.w / 2, 0, Math.PI * 2);
  ctx.fill();
  ctx.closePath();
}
```


Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

```
function drawBonus(obj) {
  if (obj.counter < 0) {
    if (obj.color === 'black') {
      obj.color = 'white';
      obj.alt = 'black';
      obj.counter = 10;
    }
    else {
      obj.color = 'black';
      obj.alt = 'white';
      obj.counter = 10;
    }
  }
  //console.log(obj.counter);
  obj.counter--;
  ctx.beginPath();
  ctx.strokeStyle = obj.color;
  ctx.rect(obj.x, obj.y, obj.w, obj.h);
  ctx.strokeRect(obj.x, obj.y, obj.w, obj.h);
  ctx.fillStyle = obj.alt;
  ctx.fill();
  ctx.closePath();
  ctx.beginPath();
  ctx.fillStyle = obj.color;
  ctx.font = '14px serif';
  ctx.textAlign = 'center';
  ctx.fillText(obj.points, obj.x + obj.w / 2, obj.y + obj.h / 2);
  ctx.closePath();
}

function drawPlayer() {
  ctx.beginPath();
```

Get the Course FREE <https://www.udemy.com/course/html5-game-canvas/>

```
ctx.rect(player.x, player.y, player.w, player.h);  
ctx.fillStyle = player.color;  
ctx.fill();  
ctx.closePath();  
}
```