



GAME WORKSHOP



PINTIG LARO GAMES
(INDIE GAME STUDIO)

PREPARED BY: RODRIGO BONZERR LOPEZ (RODPROGRAMDEV)

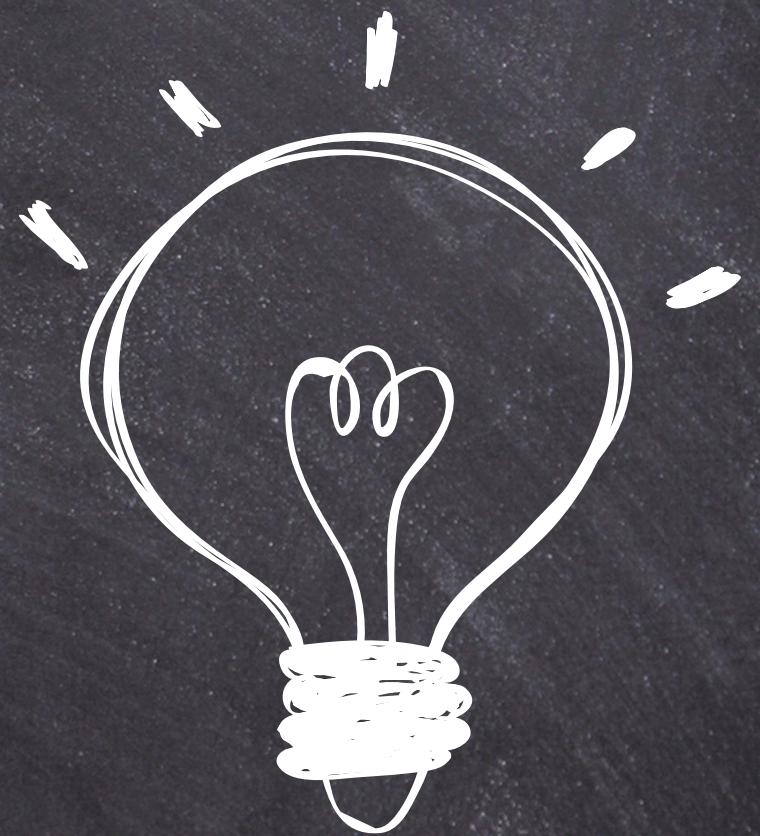
NOVEMBER 23

INTRODUCTION



INTRODUCTION TO GAME DEVELOPMENT USING SCRATCH AND JAVASCRIPT

Goal: By the end, students will create a simple interactive game in Scratch and a basic JavaScript-driven web game.



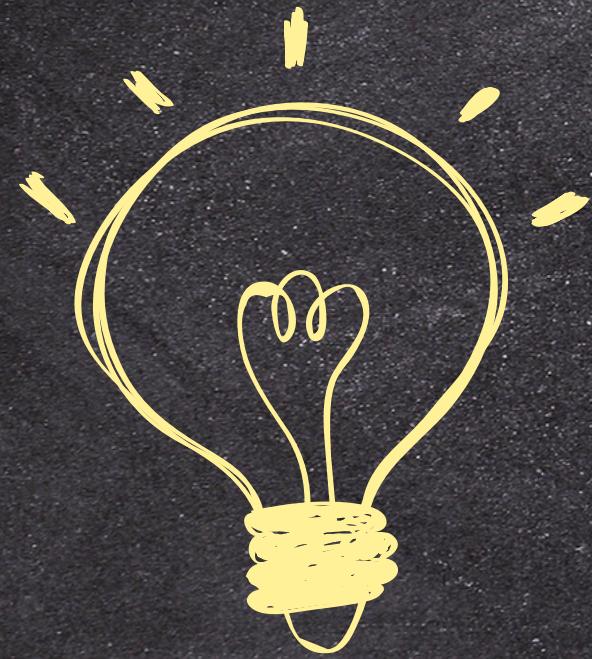
AGENDA



OBJECTIVE:

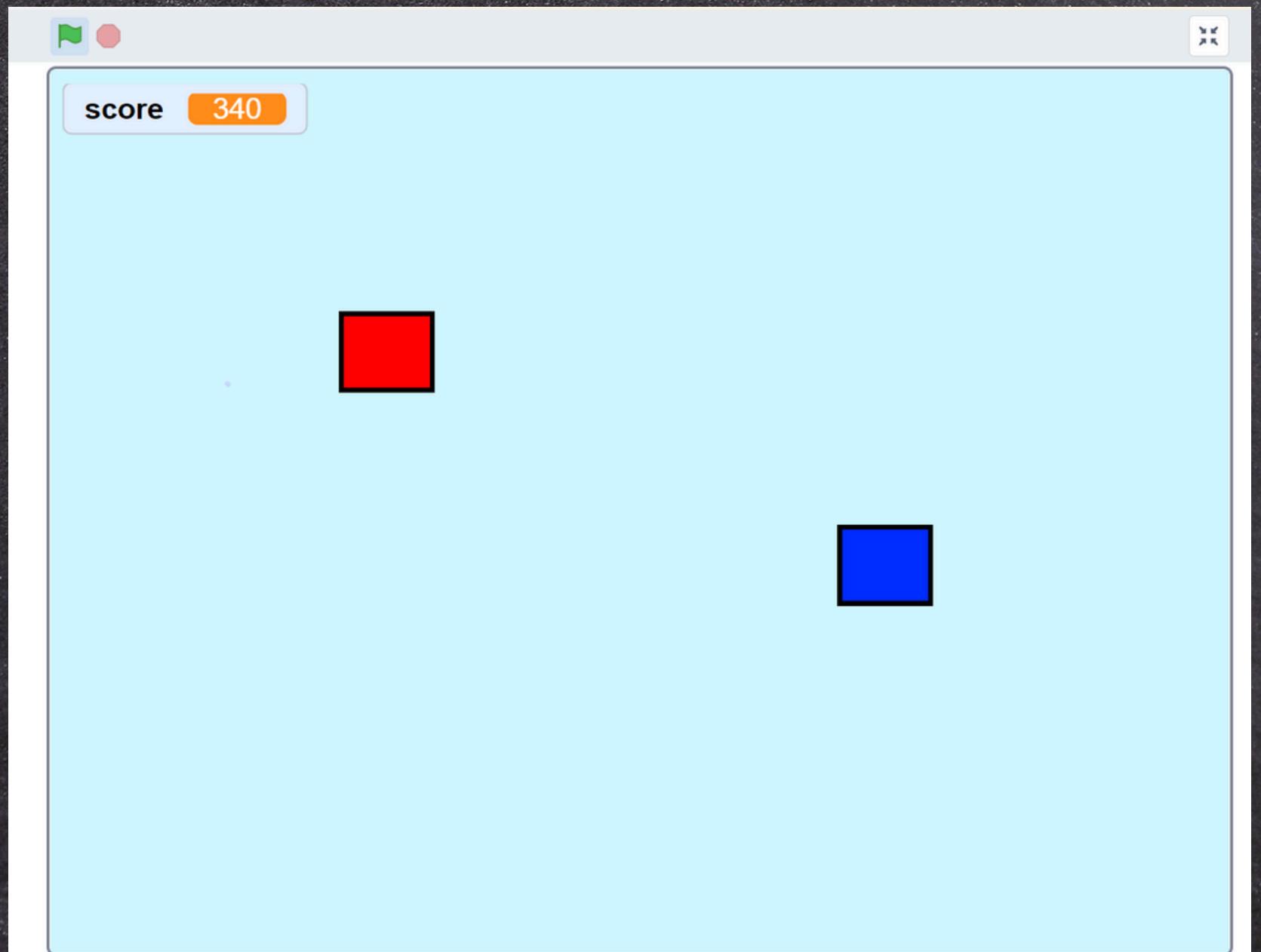
CREATE A SIMPLE GAME WHERE
PLAYER CONTROLS A SPRITE
TO COLLECT ITEMS.

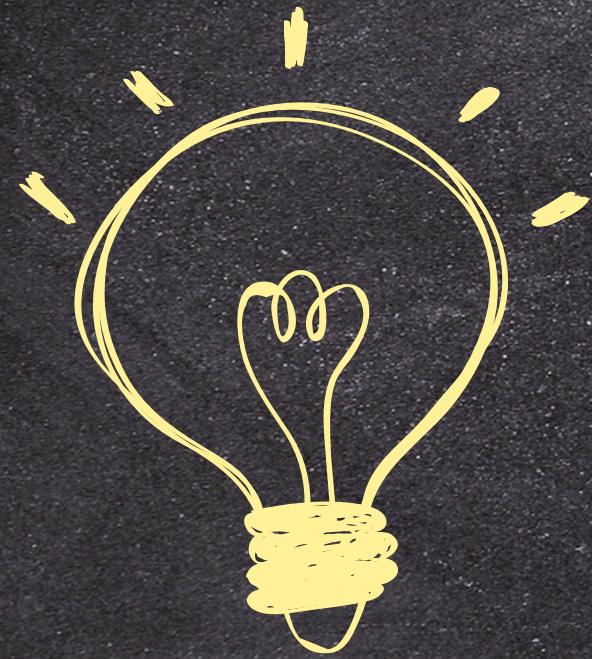




PROJECT I

SCRATCH MINI-GAME





OBJECTIVE: Create a simple game where the player controls a sprite to collect items.

Step by Step Guide:
Set-up the Stage:

Choosing a background and a main character

Basic Controls:
Show how to use “when key pressed” blocks for movement

Game Logic:
Adding collectibles and using “if” statements for collision detection.

Scoring System:
Adding points or ending the game based on player actions.



The Scratch interface shows a game in progress. The stage features a red square at the top right and a blue square at the bottom right. A red square is positioned in the center. A score of 210 is displayed in the top left corner. The script on the left defines the player's movement:

```
when green flag clicked
set score to 0
forever
  if up arrow pressed then
    change y by 10
  end
  if down arrow pressed then
    change y by -10
  end
  if left arrow pressed then
    change x by -10
  end
  if right arrow pressed then
    change x by 10
  end
```

The sprite settings on the right show:

- Sprite: Sprite1 (x: -106, y: 29)
- Show: On
- Size: 100
- Direction: 90

The stage settings on the far right show:

- Backdrops: 1



Variables

Make a Variable

my variable

score

set [my variable] to 0

change [my variable] by 1

show variable [my variable]

hide variable [my variable]

Make a List

My Blocks

Make a Block

score 0

when green flag clicked

set score to 0

forever

if key up arrow pressed? then

 change y by 10

if key down arrow pressed? then

 change y by -10

if key left arrow pressed? then

 change x by -10

if key right arrow pressed? then

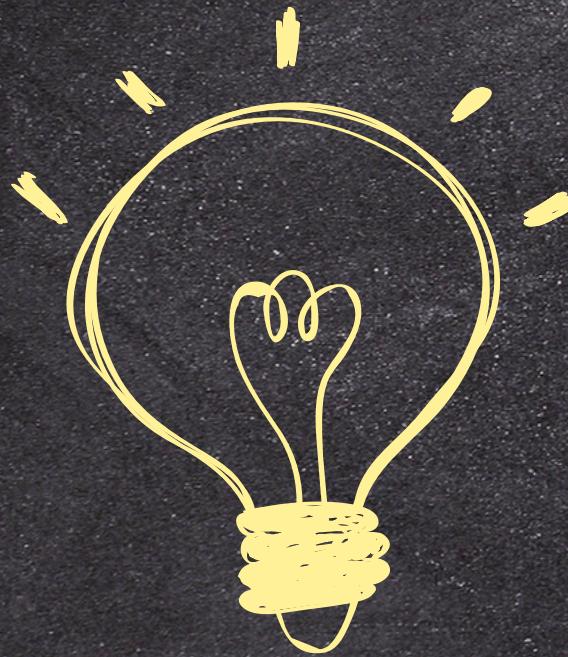
 change x by 10

Stage

Sprite Sprite1 x 166 y -42

Show [] Hide [] Size 100 Direction 90

Sprite1



PROJECT 2

JS MINI-GAME

Score: 410

```
<!DOCTYPE html>
<html lang="en">
  <head> ... </head>
  <body> flex
    <canvas id="gameCanvas" width="600" height="400">
  ... <script>... </script> == $0
</body>
</html>
```

html body script

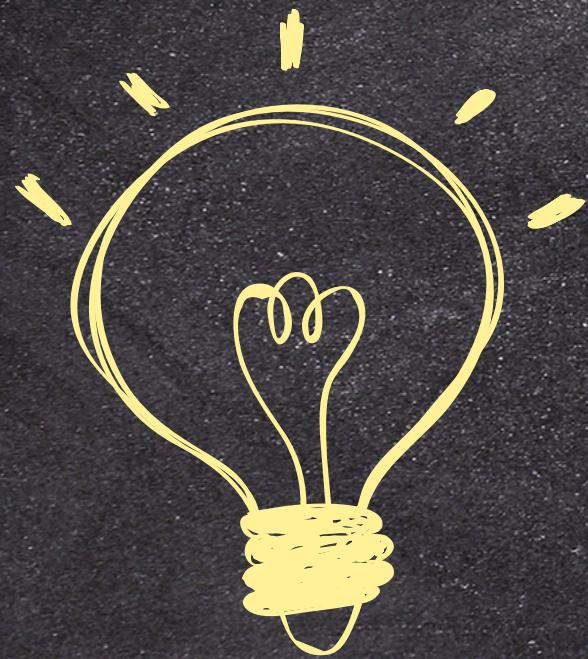
element.style { }

script { user agent stylesheet display: none; }

Inherited from body

body { game1.html:16 margin: 0; padding: 0; display: flex; justify-content: center; align-items: center; height: 100vh; background: linear-gradient(45deg, #89f7fe,

NOV 2024



OBJECTIVE: To Implement the described functionality in Scratch and will mirror the drag-and-drop programming style using the concepts below:

Step by Step Guide:
Set-up the Stage:

Choosing a background and a main character

Basic Controls:

Show how to use “when key pressed” blocks for movement

Game Logic:

Adding collectibles and using “if” statements for collision detection.

Scoring System:

Adding points or ending the game based on player actions.

SETTING UP STAGE: HTML AND CSS CODE

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Simple JS Game</title>
  <style>
    canvas {
      border: 2px solid black;
      display: block;
      margin: 0 auto;
    }
  </style>
</head>
```

SETTING UP STAGE: HTML AND CSS CODE



```
<body>
  <canvas id="gameCanvas" width="600" height="400"></canvas>
</body>
```

JAVASCRIPT CODE: Game Logic

Draw stage and Player



```
<script>
    //Initialize Canvas
    const canvas = document.getElementById("gameCanvas");
    const ctx = canvas.getContext("2d");

    //Game Variables
    let player = {x: 50, y:50, size: 20, speed: 5};
    let item = { x: randomPosition(canvas.width), y: randomPosition(canvas.height), size: 20 };

    // Random Position Function
    function randomPosition(max) {
        return Math.floor(Math.random() * (max - 20)); // Prevent item from spawning outside the canvas
    }
```

JAVASCRIPT CODE: Game Logic

Draw stage and Player



```
40  
41 //Game Loop  
42 function gameLoop(){  
43     // Clear the Canvas  
44     ctx.clearRect(0,0, canvas.width, canvas.height);  
45  
46     // Draw Player  
47     ctx.fillStyle = 'red';  
48     ctx.fillRect(player.x, player.y, player.size, player.size);  
49  
50     // Draw Item  
51     ctx.fillStyle = 'blue';  
52     ctx.fillRect(item.x, item.y, item.size, item.size);  
53  
54 }  
55  
56 gameLoop();  
57  
58 </script>
```

JAVASCRIPT CODE: Game Logic

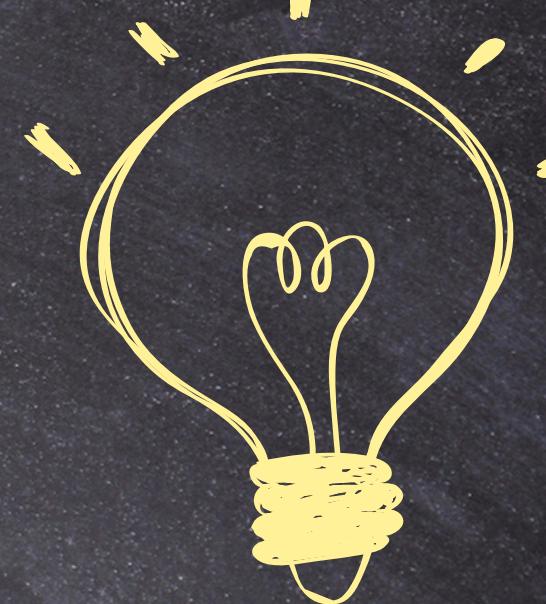
Add Basic Controls



```
40
41 // Keyboard Input
42 const keys = {};
43 window.addEventListener('keydown', (e) => keys[e.key] = true);
44 window.addEventListener('keyup', (e) => keys[e.key] = false);
45
46
47
48
49 function gameLoop(){
50
51     // Move Player
52     if (keys['ArrowUp']) player.y -= player.speed;
53     if (keys['ArrowDown']) player.y += player.speed;
54     if (keys['ArrowLeft']) player.x -= player.speed;
55     if (keys['ArrowRight']) player.x += player.speed;
56
57
58
59     // Draw Player
60     ctx.fillStyle = 'red';
61     ctx.fillRect(player.x, player.y, player.size, player.size);
62
63
64     // Draw Item
65     ctx.fillStyle = 'blue';
66     ctx.fillRect(item.x, item.y, item.size, item.size);
67
68     // Repeat Game Loop
69     requestAnimationFrame(gameLoop);
70 }
```

JAVASCRIPT CODE: Game Logic

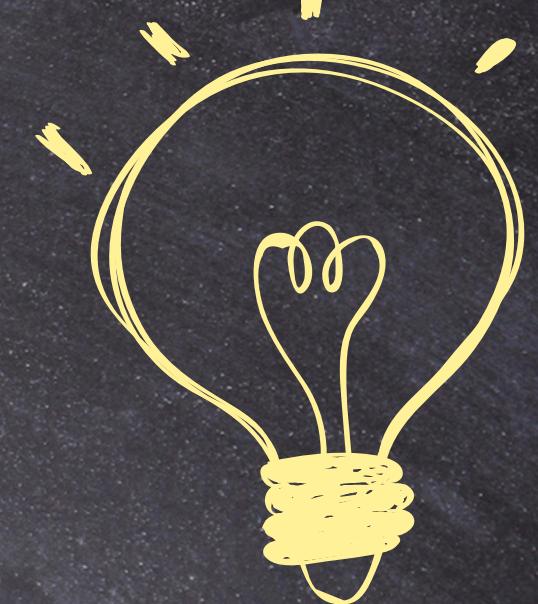
Add Collision Wall Check



```
59     // Add Wall Collision Check and Keep Player Within Bounds
60     player.x = Math.max(0, Math.min(canvas.width - player.size, player.x));
61     player.y = Math.max(0, Math.min(canvas.height - player.size, player.y));
62 }
```

JAVASCRIPT CODE: Game Logic

Add Score variable and Collision with Item Function

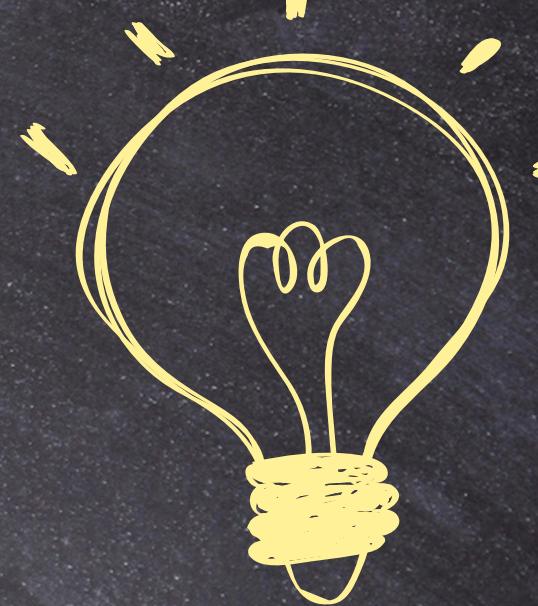


```
45  
46 // Collision Detection  
47 function checkCollision(a, b) {  
48     return (  
49         a.x < b.x + b.size &&  
50         a.x + a.size > b.x &&  
51         a.y < b.y + b.size &&  
52         a.y + a.size > b.y  
53     );  
54 }  
55
```

```
75  
76     // Check Collision with Item  
77     if (checkCollision(player, item)) {  
78         score += 10;  
79         item.x = randomPosition(canvas.width);  
80         item.y = randomPosition(canvas.height);  
81     }  
82
```

JAVASCRIPT CODE: Game Logic

Draw Score on Screen



```
102  
103     // Draw Score  
104     ctx.fillStyle = 'black';  
105     ctx.font = '20px Arial';  
106     ctx.fillText(`Score: ${score}`, 10, 20);  
107  
108
```

```
<style>  
    body {  
        margin: 0;  
        padding: 0;  
        display: flex;  
        justify-content: center;  
        align-items: center;  
        height: 100vh;  
        background: linear-gradient(45deg, #89f7fe, #66a6ff);  
        font-family: 'Arial', sans-serif;  
    }  
    canvas {  
        border: 5px solid #ffffff;  
        box-shadow: 0px 4px 10px rgba(0, 0, 0, 0.25);  
        border-radius: 15px;  
        background: #f0f8ff;  
    }  
</style>
```

CONCLUSION

SCRATCH CONCEPTS AND THEIR JAVASCRIPT EQUIVALENT



Sprite

HTML Element with JS Manipulation

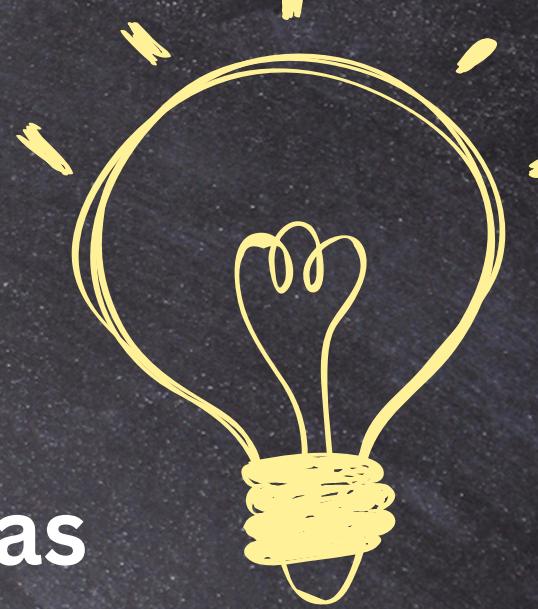
Scratch sprites can be created in JavaScript as HTML elements (like `<div>`) styled with CSS, which we manipulate using JavaScript.

Costumes

CSS Classes or Images

In Scratch, sprites have costumes for different appearances. In JS, you can swap background-image or class to simulate costume changes.

SCRATCH CONCEPTS AND THEIR JAVASCRIPT EQUIVALENT



Backdrop

Background CSS or HTML Canvas

In Scratch, the backdrop sets the scene. In JS, you set the background-color or image of the game area using CSS, or draw on HTML CANVAS

Movement Blocks (move 10 steps)

Position Manipulation in JS

Scratch moves sprites with blocks. In JS, we modify the left and top CSS properties of elements to simulate movement.

SCRATCH CONCEPTS AND THEIR JAVASCRIPT EQUIVALENT



Event Blocks(when space key pressed)

Event Listeners (keydown)

Scratch uses blocks to respond to events. JS uses addEventListener() to detect events like key presses or mouse clicks.

If statements(If left arrow pressed)

Conditional Statements

Scratch's "if" blocks are like if statements in JS, used to check conditions and control game logic.

SCRATCH CONCEPTS AND THEIR JAVASCRIPT EQUIVALENT



Broadcast message

Function Calls

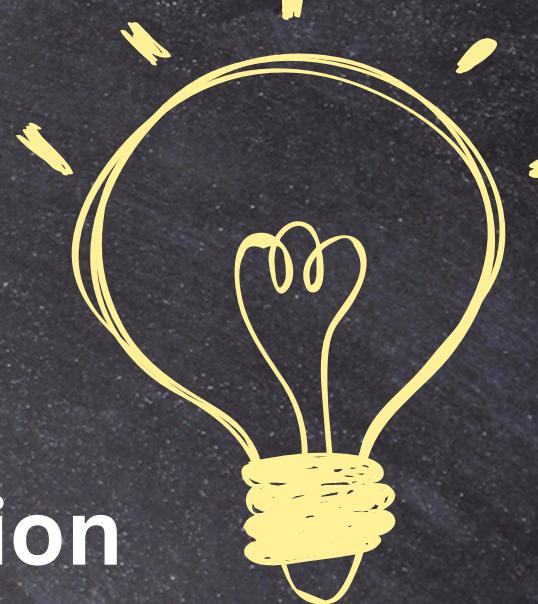
Scratch broadcasts messages to trigger code. In JS, you can call functions directly to trigger actions.

Score Variable

JS Variables

In Scratch, we create a variable for score; In JS, we declare and initialize it for example let score = 0; and it updates on events.

SCRATCH CONCEPTS AND THEIR JAVASCRIPT EQUIVALENT



Forever Loop

setInterval or Game Loop Function

Scratch's "forever" loop keep checking conditions. In JS, we use `setInterval()` or `requestAnimationFrame()` for continuous updates.

Score Variable

JS Variables

In Scratch, we create a variable for score; In JS, we declare and initialize it for example `let score = 0;` and it updates on events.

THANK YOU!!