

Código Java – script.js

```
let isJumping = false;

let gravityInterval;

let horizontalMovementInterval;

let moveDirection = null;

let currentPhase = 1;

// Detecta quando uma tecla é pressionada
document.addEventListener('keydown', (event) => {

    const mario = document.getElementById('mario');
    const copa = document.getElementById('copa');
    const princesa = document.getElementById('princesa');

    // Função para verificar colisão
    const isColliding = (rect1, rect2) => {
        return !(
            rect1.right < rect2.left ||
            rect1.left > rect2.right ||
            rect1.bottom < rect2.top ||
            rect1.top > rect2.bottom
        );
    };

    // Movimento lateral do Mario (direita/esquerda)
    if (event.key === 'ArrowRight' && !isJumping) {
        moveDirection = 'right';
        mario.style.left = `${mario.offsetLeft + 20}px`;
        mario.style.transform = 'scaleX(1)'; // Olha para a direita
    } else if (event.key === 'ArrowLeft' && !isJumping) {
        moveDirection = 'left';
        mario.style.left = `${mario.offsetLeft - 20}px`;
```

```
mario.style.transform = 'scaleX(-1)'; // Olha para a esquerda
}

// Quando pressionar espaço para pular
if (event.key === ' ' && !isJumping) {
  isJumping = true;
  mario.style.bottom = '120px'; // Altura do pulo inicial

  // Inicia o movimento horizontal se a seta estiver pressionada durante o pulo
  if (moveDirection) {
    aplicarMovimentoHorizontal();
  }

  // Volta a aplicar gravidade após o pulo
  setTimeout(() => {
    aplicarGravidade();
  }, 500);
}

// Verifica colisões com copa e princesa
setTimeout(() => {
  const marioRect = mario.getBoundingClientRect();
  const copaRect = copa.getBoundingClientRect();
  const princesaRect = princesa.getBoundingClientRect();

  // Colisão com o copa
  if (isColliding(marioRect, copaRect)) {
    alert('Você atingiu o copa, voltando para o início!');
    mario.style.left = '50px'; // Reinicia a fase
    mario.style.bottom = '0'; // No chão
  }
}
```

```

// Alcançou a princesa
if (isColliding(marioRect, princesaRect)) {
  if (currentPhase === 1) {
    alert('Você alcançou a princesa! Bem-vindo à Fase 2!');
    iniciarFaseDois();
  } else if (currentPhase === 2) {
    mostrarTelaComemoracao(); // Mostra a tela de comemoração
  }
}
}, 50);
});

// Aplica gravidade e faz o Mario cair no chão
function aplicarGravidade() {
  const mario = document.getElementById('mario');

  gravityInterval = setInterval(() => {
    // Verifica se o Mario ainda está no ar
    if (parseInt(mario.style.bottom) > 0) {
      mario.style.bottom = `${parseInt(mario.style.bottom) - 5}px`; // Desce gradualmente
    } else {
      mario.style.bottom = '0px'; // Mario no chão
      clearInterval(gravityInterval); // Para a gravidade
      clearInterval(horizontalMovementInterval); // Para o movimento lateral
      isJumping = false;
    }
  }, 50);
}

// Aplica movimento horizontal enquanto está no ar (movimento parabólico)

```

```
function aplicarMovimentoHorizontal() {  
    const mario = document.getElementById('mario');  
  
    horizontalMovementInterval = setInterval(() => {  
        if (moveDirection === 'right') {  
            mario.style.left = `${mario.offsetLeft + 20}px`; // Move para a direita  
        } else if (moveDirection === 'left') {  
            mario.style.left = `${mario.offsetLeft - 20}px`; // Move para a esquerda  
        }  
    }, 50);  
}  
  
// Função para iniciar a fase 2  
function iniciarFaseDois() {  
    currentPhase = 2;  
  
    // Troca o cenário  
    const gameContainer = document.getElementById('game-container');  
    gameContainer.style.backgroundImage = "url('cenario2.jpg')"; // Troca o cenário  
  
    // Reposiciona elementos  
    const mario = document.getElementById('mario');  
    mario.style.left = '50px'; // Mario volta ao início  
    mario.style.bottom = '0'; // No chão  
  
    const copa = document.getElementById('copa');  
    copa.style.left = '500px'; // Reposiciona o copa  
  
    const princesa = document.getElementById('princesa');  
    princesa.style.left = '750px'; // Reposiciona a princesa  
}
```

```
// Função para mostrar a tela de comemoração
function mostrarTelaComemoracao() {
    const gameContainer = document.getElementById('game-container');
    const comemoracao = document.getElementById('comemoracao');

    gameContainer.style.display = 'none'; // Esconde o jogo
    comemoracao.style.display = 'flex'; // Mostra a tela de comemoração
}

// Função para reiniciar o jogo
function reiniciarJogo() {
    const gameContainer = document.getElementById('game-container');
    const comemoracao = document.getElementById('comemoracao');
    const mario = document.getElementById('mario');

    gameContainer.style.display = 'block'; // Mostra o jogo novamente
    comemoracao.style.display = 'none'; // Esconde a tela de comemoração
    mario.style.left = '50px'; // Mario volta ao início
    mario.style.bottom = '0px'; // No chão
    currentPhase = 1; // Volta para a fase 1
    gameContainer.style.backgroundImage = "url('cenario.jpg')"; // Troca o cenário para o da
    fase 1

    // Reinicia posição dos outros elementos
    const copa = document.getElementById('copa');
    copa.style.left = '500px'; // Reposiciona o copa

    const princesa = document.getElementById('princesa');
    princesa.style.left = '750px'; // Reposiciona a princesa
}
```

Explicação detalhada java

Explicação Detalhada do Código JavaScript

Vamos dividir o código em etapas e explicar cada parte detalhadamente.

Declaração de Variáveis Globais

javascript

Copiar código

```
let isJumping = false;
```

```
let gravityInterval;
```

```
let horizontalMovementInterval;
```

```
let moveDirection = null;
```

```
let currentPhase = 1;
```

- **isJumping**: Indica se o Mario está pulando ou não. Inicialmente definido como false.
- **gravityInterval**: Armazena o identificador do intervalo que controla a gravidade aplicada ao Mario.
- **horizontalMovementInterval**: Armazena o identificador do intervalo que controla o movimento horizontal do Mario durante o pulo.
- **moveDirection**: Armazena a direção atual de movimento do Mario ('right' para direita ou 'left' para esquerda). Inicialmente é null.
- **currentPhase**: Indica a fase atual do jogo. Começa em 1.

Evento de Tecla Pressionada

javascript

Copiar código

```
document.addEventListener('keydown', (event) => {
```

```
    // Código dentro do evento
```

```
});
```

- Adiciona um ouvinte de eventos que detecta quando uma tecla é pressionada. A função é executada sempre que uma tecla é pressionada.

Referências aos Elementos do Jogo

javascript

Copiar código

```
const mario = document.getElementById('mario');
```

```
const copa = document.getElementById('copa');
```

```
const princesa = document.getElementById('princesa');
```

- **mario**: Seleciona o elemento HTML que representa o Mario.
- **copa**: Seleciona o elemento que representa o inimigo Copa.
- **princesa**: Seleciona o elemento que representa a Princesa.

Função para Verificar Colisões

javascript

Copiar código

```
const isColliding = (rect1, rect2) => {  
  return !(  
    rect1.right < rect2.left ||  
    rect1.left > rect2.right ||  
    rect1.bottom < rect2.top ||  
    rect1.top > rect2.bottom  
  );  
};
```

- **isColliding**: Função que verifica se dois elementos estão colidindo, baseando-se nas posições deles na tela.

Movimento Lateral do Mario

javascript

Copiar código

```
if (event.key === 'ArrowRight' && !isJumping) {  
  moveDirection = 'right';  
  mario.style.left = `${mario.offsetLeft + 20}px`;  
  mario.style.transform = 'scaleX(1)'; // Olha para a direita  
} else if (event.key === 'ArrowLeft' && !isJumping) {  
  moveDirection = 'left';  
  mario.style.left = `${mario.offsetLeft - 20}px`;  
  mario.style.transform = 'scaleX(-1)'; // Olha para a esquerda  
}
```

- **Movimento para a Direita (ArrowRight):**
 - Verifica se a tecla pressionada é a seta para a direita e se o Mario não está pulando.

- Define moveDirection como 'right'.
- Atualiza a posição horizontal do Mario, movendo-o 20 pixels para a direita.
- Usa transform: scaleX(1) para garantir que o Mario esteja olhando para a direita.
- **Movimento para a Esquerda (ArrowLeft):**
 - Similar ao movimento para a direita, mas para a esquerda.
 - Define moveDirection como 'left'.
 - Move o Mario 20 pixels para a esquerda.
 - Usa transform: scaleX(-1) para virar o Mario para a esquerda.

Pulo do Mario

javascript

Copiar código

```
if (event.key === ' ' && !isJumping) {
  isJumping = true;
  mario.style.bottom = '120px'; // Altura do pulo inicial

  // Inicia o movimento horizontal se a seta estiver pressionada durante o pulo
  if (moveDirection) {
    aplicarMovimentoHorizontal();
  }

  // Volta a aplicar gravidade após o pulo
  setTimeout(() => {
    aplicarGravidade();
  }, 500);
}
```

- Verifica se a tecla de espaço foi pressionada e se o Mario não está pulando.
- Define isJumping como true para indicar que o Mario está no ar.
- Ajusta a posição vertical do Mario para 120 pixels, fazendo-o "subir" na tela.
- **Movimento Parabólico:** Se uma direção de movimento (moveDirection) estiver definida, chama a função aplicarMovimentoHorizontal() para mover o Mario horizontalmente enquanto ele está no ar.

- **Gravidade:** Após 500 milissegundos (tempo do pulo), chama aplicarGravidade() para fazer o Mario descer.

Verificação de Colisões Após o Movimento

javascript

Copiar código

```
setTimeout(() => {
    const marioRect = mario.getBoundingClientRect();
    const copaRect = copa.getBoundingClientRect();
    const princesaRect = princesa.getBoundingClientRect();

    // Colisão com o copa
    if (isColliding(marioRect, copaRect)) {
        alert('Você atingiu o copa, voltando para o início!');
        mario.style.left = '50px'; // Reinicia a fase
        mario.style.bottom = '0'; // No chão
    }

    // Alcançou a princesa
    if (isColliding(marioRect, princesaRect)) {
        if (currentPhase === 1) {
            alert('Você alcançou a princesa! Bem-vindo à Fase 2!');
            iniciarFaseDois();
        } else if (currentPhase === 2) {
            mostrarTelaComemoracao(); // Mostra a tela de comemoração
        }
    }
}, 50);
```

- Após um pequeno atraso de 50 milissegundos, para garantir que a posição do Mario foi atualizada, verifica colisões.
- **Colisão com o Copa:**
 - Se o Mario colidir com o Copa, exibe um alerta.

- Reposiciona o Mario no início da fase, ajustando sua posição horizontal e colocando-o no chão.
- **Colisão com a Princesa:**
 - Se o Mario colidir com a Princesa:
 - **Fase 1:** Se estiver na fase 1, exibe uma mensagem de sucesso e chama `iniciarFaseDois()` para passar para a fase 2.
 - **Fase 2:** Se estiver na fase 2, chama `mostrarTelaComemoracao()` para exibir a tela de vitória.

Função para Aplicar Gravidade

javascript

Copiar código

```
function aplicarGravidade() {
    const mario = document.getElementById('mario');

    gravityInterval = setInterval(() => {
        // Verifica se o Mario ainda está no ar
        if (parseInt(mario.style.bottom) > 0) {
            mario.style.bottom = `${parseInt(mario.style.bottom) - 5}px`; // Desce gradualmente
        } else {
            mario.style.bottom = '0px'; // Mario no chão
            clearInterval(gravityInterval); // Para a gravidade
            clearInterval(horizontalMovementInterval); // Para o movimento lateral
            isJumping = false;
        }
    }, 50);
}
```

- **aplicarGravidade():** Simula o efeito da gravidade, fazendo o Mario descer gradualmente até o chão.
- Usa um intervalo que a cada 50 milissegundos diminui a posição vertical do Mario em 5 pixels.
- Quando o Mario atinge o chão (`bottom <= 0`), os intervalos de gravidade e movimento horizontal são limpos, e `isJumping` é definido como `false`, permitindo que o Mario possa pular novamente.

Função para Aplicar Movimento Horizontal no Ar

javascript

Copiar código

```
function aplicarMovimentoHorizontal() {  
    const mario = document.getElementById('mario');  
  
    horizontalMovementInterval = setInterval(() => {  
        if (moveDirection === 'right') {  
            mario.style.left = `${mario.offsetLeft + 20}px`; // Move para a direita  
        } else if (moveDirection === 'left') {  
            mario.style.left = `${mario.offsetLeft - 20}px`; // Move para a esquerda  
        }  
    }, 50);  
}
```

- **aplicarMovimentoHorizontal()**: Move o Mario horizontalmente enquanto ele está no ar, criando um movimento parabólico.
- Dependendo de moveDirection, o Mario se move 20 pixels para a direita ou esquerda a cada 50 milissegundos.

Função para Iniciar a Fase 2

javascript

Copiar código

```
function iniciarFaseDois() {  
    currentPhase = 2;  
  
    // Troca o cenário  
    const gameContainer = document.getElementById('game-container');  
    gameContainer.style.backgroundImage = "url('cenario2.jpg')"; // Troca o cenário  
  
    // Reposiciona elementos  
    const mario = document.getElementById('mario');  
    mario.style.left = '50px'; // Mario volta ao início  
    mario.style.bottom = '0'; // No chão
```

```
const copa = document.getElementById('copa');  
copa.style.left = '500px'; // Reposiciona o copa
```

```
const princesa = document.getElementById('princesa');  
princesa.style.left = '750px'; // Reposiciona a princesa  
}
```

- **iniciarFaseDois()**: Configura o jogo para a fase 2.
- Atualiza currentPhase para 2.
- Troca a imagem de fundo do contêiner do jogo para cenario2.jpg.
- Reposiciona o Mario, o Copa e a Princesa para suas posições iniciais na fase 2.

Função para Mostrar a Tela de Comemoração

javascript

Copiar código

```
function mostrarTelaComemoracao() {  
    const gameContainer = document.getElementById('game-container');  
    const comemoracao = document.getElementById('comemoracao');  
  
    gameContainer.style.display = 'none'; // Esconde o jogo  
    comemoracao.style.display = 'flex'; // Mostra a tela de comemoração  
}
```

- **mostrarTelaComemoracao()**: Exibe a tela de vitória ao final da fase 2.
- Esconde o contêiner do jogo e mostra a <div> de comemoração, que contém mensagens de parabéns e um botão para reiniciar o jogo.

Função para Reiniciar o Jogo

javascript

Copiar código

```
function reiniciarJogo() {  
    const gameContainer = document.getElementById('game-container');  
    const comemoracao = document.getElementById('comemoracao');  
    const mario = document.getElementById('mario');
```

```

gameContainer.style.display = 'block'; // Mostra o jogo novamente
comemoracao.style.display = 'none'; // Esconde a tela de comemoração
mario.style.left = '50px'; // Mario volta ao início
mario.style.bottom = '0px'; // No chão
currentPhase = 1; // Volta para a fase 1

gameContainer.style.backgroundImage = "url('cenario.jpg')"; // Troca o cenário para o da
fase 1

// Reinicia posição dos outros elementos
const copa = document.getElementById('copa');
copa.style.left = '500px'; // Reposiciona o copa

const princesa = document.getElementById('princesa');
princesa.style.left = '750px'; // Reposiciona a princesa
}

```

- **reiniciarJogo()**: Reseta o jogo para o estado inicial.
- Mostra novamente o contêiner do jogo e esconde a tela de comemoração.
- Reposiciona o Mario, o Copa e a Princesa para suas posições iniciais.
- Reseta currentPhase para 1 e troca o cenário para o da fase 1 (cenario.jpg).

7 Questões de Múltipla Escolha com Respostas

1. **Qual é a finalidade da variável isJumping no código?**
 - A) Indicar se o Mario está se movendo para a esquerda.
 - B) Verificar se o Mario está pulando, para evitar múltiplos pulos simultâneos.
 - C) Controlar a fase atual do jogo.
 - D) Definir a direção do movimento horizontal.
 - **Resposta Correta:** B) Verificar se o Mario está pulando, para evitar múltiplos pulos simultâneos.
2. **O que a função isColliding(rect1, rect2) verifica?**
 - A) Se dois elementos estão colidindo na tela.
 - B) Se o Mario está pulando.
 - C) Se o jogador pressionou a tecla de espaço.
 - D) Se o cenário precisa ser trocado.

- **Resposta Correta:** A) Se dois elementos estão colidindo na tela.

3. **O que acontece quando o jogador pressiona a tecla de espaço e !isJumping é verdadeiro?**

- A) O Mario começa a pular e a variável isJumping é definida como true.
- B) O Mario se move para a esquerda.
- C) O jogo reinicia.
- D) A fase atual é incrementada.
- **Resposta Correta:** A) O Mario começa a pular e a variável isJumping é definida como true.

4. **Qual é o propósito da função aplicarGravidade()?**

- A) Fazer o Mario se mover horizontalmente enquanto está no ar.
- B) Simular o efeito da gravidade, fazendo o Mario descer gradualmente até o chão.
- C) Verificar colisões entre o Mario e outros elementos.
- D) Reiniciar o jogo após uma colisão.
- **Resposta Correta:** B) Simular o efeito da gravidade, fazendo o Mario descer gradualmente até o chão.

5. **Quando a função iniciarFaseDois() é chamada, o que acontece?**

- A) O jogo é reiniciado para a fase 1.
- B) O Mario volta ao início, e o cenário é trocado para a fase 2.
- C) A tela de comemoração é exibida.
- D) O Copa e a Princesa são removidos do jogo.
- **Resposta Correta:** B) O Mario volta ao início, e o cenário é trocado para a fase 2.

6. **Qual é o efeito de mario.style.transform = 'scaleX(-1)' no código?**

- A) Vira o Mario de cabeça para baixo.
- B) Faz o Mario olhar para a esquerda, invertendo-o horizontalmente.
- C) Aumenta o tamanho do Mario.
- D) Deixa o Mario transparente.
- **Resposta Correta:** B) Faz o Mario olhar para a esquerda, invertendo-o horizontalmente.

7. **O que a função reiniciarJogo() faz quando chamada após vencer o jogo?**

- A) Finaliza o jogo e fecha a janela.

- B) Reinicia o jogo, retornando à fase 1 e reposicionando os elementos iniciais.
- C) Pausa o jogo temporariamente.
- D) Salva o progresso atual do jogo.
- **Resposta Correta:** B) Reinicia o jogo, retornando à fase 1 e reposicionando os elementos iniciais.