

# Explicação Detalhada do Jogo Mario Jump

## 1. Estrutura Básica do Projeto

O projeto Mario Jump é um jogo simples feito em HTML, CSS e JavaScript. Ele utiliza uma estrutura básica de arquivos:

- index.html: arquivo principal que carrega o jogo.
- style.css: arquivo de estilo para deixar o jogo visualmente agradável.
- script.js (ou código JS embutido): onde está a lógica do jogo.

O jogo roda na janela do navegador e responde a eventos de teclado para controlar o personagem (Mario).

## 2. Funcionamento do Código HTML

O arquivo HTML define o layout básico, incluindo a tag `<canvas>` que é o espaço onde o jogo é desenhado.

Ele também inclui links para os arquivos de CSS e JS.

Exemplo básico:

```
<canvas id='gameCanvas' width='800' height='400'></canvas>
```

O canvas é a área onde o Mario e os elementos do jogo são renderizados dinamicamente pelo JavaScript.

## 3. Conceitos Principais do JavaScript no Jogo

- Variáveis para controlar a posição do Mario, velocidade, gravidade e estado do jogo.
- Função de loop principal que atualiza a posição do Mario e redesenha a tela a cada frame.
- Eventos de teclado para detectar quando o jogador aperta a tecla para pular.
- Detectar colisões para saber se o Mario 'morreu' ao bater em obstáculos.

O jogo funciona com um loop que chama a função 'update' repetidamente usando `requestAnimationFrame`, criando uma animação suave.

## 4. Lógica do Pulo do Mario

Quando a tecla de pulo é pressionada, o código altera a velocidade vertical do Mario para cima. Depois, a gravidade vai puxando o Mario para baixo, simulando o efeito do pulo e queda.

Exemplo de variáveis usadas:

- marioY: posição vertical.
- velocityY: velocidade vertical.
- gravity: força que puxa para baixo.

A cada frame, a posição vertical é atualizada com a velocidade, e a velocidade é atualizada com a gravidade.

## **5. Detectando a Morte e Reiniciando o Jogo**

Quando o Mario colide com um obstáculo, o jogo para, mostrando que ele morreu.

Normalmente, um botão aparece para permitir reiniciar o jogo, mas ainda estamos ajustando isso para aparecer corretamente.

O código monitora a posição do Mario e a posição dos obstáculos para detectar colisões.

## **6. Próximos Passos e Ajustes**

Ainda estamos finalizando detalhes visuais, como fazer o botão de reiniciar aparecer corretamente após a morte do Mario.

Também podemos implementar melhorias no design e adicionar sons ou níveis mais avançados.

*Documento gerado para ajudar no entendimento do código Mario Jump.*