Checklist – Raphael Lins / RGM: 27797660

Fase 1: Análise

- Criar um jogo simples onde o jogador controla um personagem que precisa evitar obstáculos.
- Animações, detecção de colisões e interação com o usuário.

```
// Posição do Metroid
let xMetroid = 100;
let yMetroid = 50;

// Velocidade do Metroid
let vxMetroid = 0;
let vyMetroid = 0;

// Gravidade
let gravidade = 1;

// Atualização da posição a cada frame:
xMetroid += vxMetroid;
yMetroid += vyMetroid;
vyMetroid += gravidade;

// Detecção de colisão com um obstáculo (xObstaculo, yObstaculo, largura, altura):
if (xMetroid + larguraMetroid > xObstaculo &&
   xMetroid < xObstaculo + larguraObstaculo &&
   yMetroid + alturaMetroid > yObstaculo &&
   yMetroid < yObstaculo + alturaObstaculo) {
   // Colisão detectada
```

}Fase 2: Planejamento

- Criar um jogo divertido e desafiador, com gráficos simples e jogabilidade intuitiva.
- Tempo de execução, consumo de memória, facilidade de entendimento do código.
- Utilizar JavaScript para manipular o DOM e criar a animação, utilizando um loop principal para atualizar o estado do jogo.
- Deteção de colisão, animação de pulo, atualização da pontuação.

- Uma função principal para iniciar o jogo e um loop principal para atualizar o estado do jogo.
- O que acontece quando o jogador atinge a pontuação máxima? O que acontece se o jogador não pressionar nenhuma tecla?

Fase 3: Desenho

- A complexidade do algoritmo é baixa (O(1) por iteração), o que é adequado para um jogo simples.
- A detecção de colisões poderia ser otimizada para jogos mais complexos com muitos elementos na tela.

Fase 4: Programação e Teste

- O código JavaScript implementa a lógica do jogo de forma clara.
- O código é relativamente bem-organizado, mas poderia ser melhor estruturado com funções mais específicas.
- Foram realizados testes básicos para verificar se o jogo funciona corretamente, mas testes mais rigorosos poderiam ser realizados para garantir a robustez do código.

Documentação e Avaliação

- A documentação está no README.md do repositório.
- A eficácia do algoritmo foi avaliada de forma informal, observando o desempenho do jogo. Uma avaliação mais rigorosa poderia ser feita utilizando ferramentas de profiling.

Apresentação e Conclusão

O código do Metroid Jump serve como um bom exemplo de um projeto de programação simples. Ele demonstra os conceitos básicos de programação e como aplicá-los para criar jogos. No entanto, há espaço para melhorias em termos de estrutura, testes e documentação. Ao seguir as melhores práticas de desenvolvimento de software, é possível criar jogos mais robustos, escaláveis e fáceis de manter.