Grupo: Murilo Santana, Jônatas Pereira, Edson Brenno, Endril Luiz

# Jogo da Cobra em C

## Contexto

Este documento descreve o desenvolvimento de um jogo da cobra implementado na linguagem C. O jogo foi desenvolvido utilizando manipulação de matrizes, controle de fluxo, e interação com o usuário. A versão atual do jogo inclui funcionalidades básicas como movimentação da cobra, geração de comida aleatória, crescimento da cobra ao comer a comida, e detecção de colisões que resultam no fim do jogo, além de alguns requisitos opcionais (ex: efeitos sonoros).

## Requisitos

**Requisitos Essenciais (Must Have)**

* O jogo deve iniciar com a cobra em uma posição central do campo.
* A cobra deve se mover na direção definida pelo jogador.
* A cobra deve crescer ao comer a comida que aparece no campo.
* O jogo deve terminar quando a cobra colidir com suas próprias caudas.
* A pontuação deve ser atualizada e exibida ao jogador em tempo real.

**Requisitos Desejáveis (Should Have)**

* A comida deve aparecer em posições aleatórias dentro do campo.
* O jogo deve ter uma borda que define os limites do campo.
* A velocidade da cobra deve ser ajustável.

**Requisitos Opcionais (Could Have)**

* Sons de fundo e efeitos sonoros.
* Diferentes níveis de dificuldade.
* Opções de personalização da cobra e do campo de jogo.

**Requisitos Não Necessários (Won't Have)**

* Funcionalidades multiplayer.
* Integração com redes sociais para compartilhamento de pontuações.

## Estrutura do Código

O código está estruturado em funções distintas que lidam com diferentes aspectos do jogo:

* **hideCursor**: Esconde o cursor no console para melhorar a experiência visual.
* **setCursorPosition**: Define a posição do cursor no console.
* **setup**: Inicializa as variáveis do jogo.
* **drawBorder**: Desenha a borda do campo de jogo.
* **updateScore**: Atualiza e exibe a pontuação.
* **drawSnakeAndFood**: Desenha a cobra e a comida no campo de jogo.
* **clearPrevious**: Limpa a posição anterior da cobra e da cauda.
* **input**: Captura a entrada de tecla do usuário para controlar a direção da cobra.
* **logic**: Contém a lógica principal do jogo, incluindo movimentação da cobra, verificação de colisões e crescimento da cobra.

## Implementação

**Passos para Implementação**

1. **Inicializar as variáveis e funções auxiliares.**
   * Função setup para definir as condições iniciais do jogo.
   * Função hideCursor para ocultar o cursor do console.
2. **Desenhar a borda do campo de jogo.**
   * Função drawBorder para desenhar as bordas ao redor do campo.
3. **Entrar no loop principal do jogo, onde:**
   * + Limpar a posição anterior da cobra e da cauda.
       1. Função clearPrevious para limpar as posições anteriores.
     + Atualizar a lógica do jogo (movimentação, colisões, crescimento).
       1. Função logic para mover a cobra, verificar colisões e atualizar o jogo.
     + Desenhar a cobra e a comida.
       1. Função drawSnakeAndFood para desenhar a cobra e a comida na nova posição.
     + Capturar a entrada do usuário.
       1. Função input para ler a entrada do teclado e ajustar a direção da cobra.
     + Pausar antes da próxima atualização da tela.
       1. Função Sleep para criar um intervalo entre as atualizações.
4. **Exibir mensagem de fim de jogo quando a cobra colidir com sua própria cauda.**
   * A mensagem de fim de jogo é exibida no console e o programa é fechado.

## Marcos

**Marco 1: Inicialização e Desenho da Borda**

* Implementação das funções de inicialização e desenho da borda.

**Marco 2: Movimentação e Lógica do Jogo**

* Implementação da lógica de movimentação e detecção de colisões.

**Marco 3: Entrada do Usuário e Atualização da Pontuação**

* Implementação das funções de captura de entrada e atualização da pontuação.

**Marco 4: Testes e Ajustes Finais**

* Testes do jogo e ajustes finais para garantir a funcionalidade correta.

## Coleta de Resultados

**Avaliação dos Requisitos**

Para avaliar se os requisitos foram atendidos, será verificado se todas as funcionalidades essenciais estão presentes e funcionando corretamente.

**Desempenho do Sistema**

O desempenho do jogo será avaliado com base na resposta do controle do usuário, na suavidade da movimentação da cobra e na precisão da detecção de colisões.