# JOGO EM C NeoSnake



Julian Campeão

Lucas Cruz

Vinicius Meurer

## CONTEXTUALIZANDO



- Versão moderna e aprimorada do tradicional jogo da cobrinha.
- Desenvolvido em puramente com linguagem C e bibliotecas nativas.
- Combina a nostalgia do jogo clássico com novas funcionalidades.
- Interface de console com gráficos em ASCII para experiência retro (não usa bibliotecas gráficas



# INOVAÇÕES



- Múltiplos modos de jogo para diferentes experiências.
- Sistema de power-ups opcionais onde o jogador escolhe se quer ativar ou não.
- A cada 100 pontos = novo nível.
- Velocidade aumenta conforme progride de nível.

# **MODOS DE JOGO - TRADICIONAL**

• O modo tradicional preserva a essência original do jogo.

• Partida sem limite de tempo.

Progressão infinita até a primeira colisão



# **MODOS DE JOGO - TEMPO**

• o jogador tem exatamente 60 segundos para alcançar a maior pontuação possível

- Tempo fixo de 1 minuto
- Contador regressivo visível na interface
- Fim automático quando o tempo esgota

```
Pontos: 10
                                Nivel: 1
                                Tempo: 51
0000
                                ESC: Menu
                                P: Pausar
```

## **MODOS DE JOGO - INVERSO**

 Todos os controles são invertidos, testando a capacidade de adaptação do jogador.

 Mesma mecânica do jogo tradicional, mas dificuldade psicológica adicional da mudança de controles.

```
Pontos: 0
Nivel: 1
```

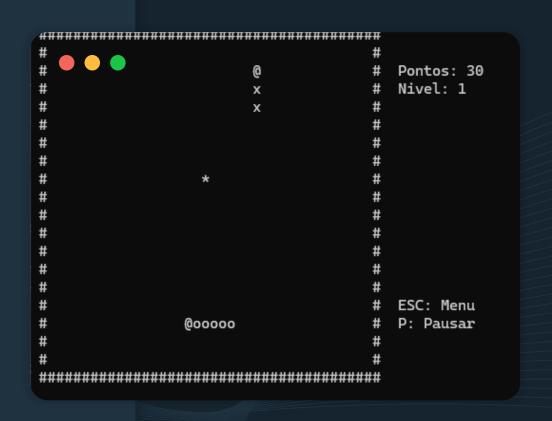
# MODOS DE JOGO - MULTIPLAYER COLAB.

 Cobras diferentes na mesma tela, trabalhando em equipe para maximizar a pontuação.

• Jogador 1 controla com setas do teclado e Jogador 2 controla com teclas WASD

Ambas as cobras coletam pontos

 Qualquer cobra que colidir termina o jogo automaticamente



# **POWER-UPS**



#### TURBO (T)

Velocidade duplicada por 5 segundos



#### IMUNIDADE (I)

Invencibilidade temporária contra colisões



#### SLOW (S)

Velocidade reduzida para maior controle



#### DOUBLE POINT(D)

Frutas valem 20 pontos por 10 segundos



- 10 segundos na tela antes de desaparecer
- ApariçãoAleatória
- Simbolos únicos para identificação

# **CONCEITOS TRABALHADOS - STRUCTS**



#### Posicao

- Coordenadas x,y no campo de jogo
- Utilizada para localizar elementos na tela
- Base para todas as posições de objetos



#### Cobra

- Array de posições do corpo
- Tamanho atual e direção de movimento
- Gerencia estado completo da cobra

# **CONCEITOS TRABALHADOS - STRUCTS**



#### PowerUp

- Características dos itens especiais
- Posição, tipo, estado ativo e tempo restante
- Controla comportamento e exibição dos power-ups



#### EstadoJogo

- Informações globais da partida
- Pontuação, nível, tempo e configurações
- Centraliza dados importantes do jogo

# CONCEITOS TRABALHADOS - FUNÇÕES



#### Utilitárias

Servem para manipular interface de console

#### **Exemplos**

- limpar\_tela()
- posicionar\_cursor()
- ocultar\_cursor()



#### Inicialização

Servem para realizar a configuração inicial dos elementos

#### **Exemplos**

- inicializar\_cobra()
- gerar\_fruta()
- inicializar\_jogo()

# CONCEITOS TRABALHADOS - FUNÇÕES



#### Renderização

Funções responsáveis pela exibição visual

#### **Exemplos**

- desenhar\_borda()
- desenhar\_cobra()
- desenhar\_hud()



#### Lógicas

Implementam regras e mecânicas do jogo

#### **Exemplos**

- mover\_cobra()
- verificar\_colisao\_parede()
- processar\_entrada()

## **CONCEITOS TRABALHADOS - PONTEIROS**



#### Uso de ponteiros

Estruturas complexas (Cobra) sempre passadas por ponteiro. A estrutura Cobra contém:

- Array corpo[MAX\_COBRA] = 100 posições × 8 bytes = 800 bytes
- Variáveis tamanho e direcao = 8 bytes adicionais
- Total: ~808 bytes por struct

#### **Exemplos**

inicializar\_cobra(Cobra\* cobra) - Configura uma nova cobra
mover\_cobra(Cobra\* cobra) - Move a cobra pelo campo
desenhar\_cobra(Cobra\* cobra) - Desenha a cobra na tela
verificar\_colisao\_parede(Cobra\* cobra) - Verifica se bateu na parede
verificar\_colisao\_corpo(Cobra\* cobra) - Verifica se bateu no próprio corpo

# OBRIGADO PELA ATENÇAO! vontade para perguntar:)

Caso tenha qualquer dúvida, sinta-se à