




# JOGO EM C

# NeoSnake



```
<p> Julian Campeão</p>
<p> Lucas Cruz</p>
<p> Vinicius Meurer</p>
```

# 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.

```
#####  
# ● ● ● NeoSnake #  
#####  
# #  
# 1. Modo Tradicional #  
# 2. Modo Tempo (1 minuto) #  
# 3. Modo Inverso #  
# 4. Modo Multiplayer Local #  
# 5. Sair #  
# #  
#####  
  
Escolha uma opcao:
```

```
#####  
# ● ● ● POWER-UPS #  
# ● ● ● #####  
# #  
# T - Turbo temporario #  
# S - Slow motion #  
# I - Imunidade temporaria #  
# D - Pontos em dobro #  
# #  
# Habilitar power-ups? #  
# 1. Sim #  
# 2. Nao #  
# #  
#####  
  
Escolha uma opcao:
```

# MODOS DE JOGO - TRADICIONAL



- 0 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



# 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.



# 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



## SLOW (S)

Velocidade reduzida  
para maior controle



## IMUNIDADE (I)

Invencibilidade  
temporária contra  
colisões



## DOUBLE POINT(D)

Frutas valem 20 pontos  
por 10 segundos



- 10 segundos na tela antes de desaparecer
- Aparição Aleatória
- 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 ATENÇÃO!



Caso tenha qualquer  
dúvida, sinta-se à  
vontade para  
perguntar :)