INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

RAISA PRISCILA DA SILVA

JOGO “COBRINHA” EM C++ COM RAYLIB

CAMPOS DO JORDÃO

2024

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

RAISA PRISCILA DA SILVA

Entrega Parcial de trabalho final apresentado ao Instituto Federal de São Paulo (IFSP), em cumprimento a exigência da disciplina de Programação Orientada a Objeto do curso de Analise e Desenvolvimento de Sistemas.

PROFESSOR: Paulo Giovani de Faria Zeferine.

NOVEMBRO

2024

RESUMO

O presente trabalho de conclusão de curso apresenta o desenvolvimento de um jogo digital chamado "COBRINHA!", uma releitura do clássico jogo Snake, utilizando a linguagem de programação C++ e a biblioteca gráfica Raylib. O projeto tem como objetivo aplicar conceitos de programação e design de jogos em um contexto prático, utilizando ferramentas modernas e acessíveis. O jogo "COBRINHA!" oferece uma experiência divertida e desafiadora, mantendo a simplicidade característica do original, enquanto explora aspectos de programação gráfica, lógica de jogo e manipulação de eventos em tempo real. O desenvolvimento foi realizado com foco em proporcionar uma interface intuitiva e responsiva, adequada tanto para iniciantes quanto para jogadores mais experientes. O resultado final é um jogo funcional, com potencial de expansão e melhorias futuras, evidenciando o aprendizado adquirido durante o curso.

Palavras-Chave: jogos; desenvolvimento; c++; raylib, skane, digital, programação.

ABSTRACT

This course conclusion work presents the development of a digital game called ‘COBRINHA!’, a reinterpretation of the classic Snake game, using the C++ programming language and the Raylib graphics library. The project aims to apply programming and game design concepts in a practical context, using modern and accessible tools. The game "COBRINHA!" offers a fun and challenging experience, maintaining the characteristic simplicity of the original, while exploring aspects of graphical programming, game logic and real-time event handling. The development was carried out with a focus on providing an intuitive and responsive interface, suitable for both beginners and more experienced players. The end result is a functional game, with potential for expansion and future improvements, highlighting the learning acquired during the course.

Keywords: games; development; c++; raylib, skane, digital, programming.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

SUMÁRIO

1 INTRODUÇÃO

O desenvolvimento de jogos eletrônicos tem se tornado uma área de interesse crescente tanto na indústria quanto na academia, devido ao seu impacto cultural e ao seu potencial de inovação tecnológica. Este projeto, intitulado "COBRINHA!", consiste em uma releitura do clássico jogo Snake, amplamente conhecido por sua simplicidade e jogabilidade envolvente.

1.1 Objetivos

O objetivo principal deste trabalho é aplicar os conhecimentos adquiridos ao longo do curso de Ciência da Computação, utilizando a linguagem C++ e a biblioteca Raylib para desenvolver um jogo completo e funcional.

1.2 Justificativa

A justificativa para a escolha deste projeto reside na popularidade atemporal do jogo Snake, que, apesar de ser simples em termos de mecânicas, oferece um excelente campo para a aplicação de conceitos fundamentais de programação, como manipulação de matrizes, controle de fluxo e gerenciamento de recursos gráficos. Além disso, a utilização da Raylib como biblioteca gráfica visa proporcionar uma experiência prática no desenvolvimento de jogos 2D, aproveitando sua facilidade de uso e compatibilidade com C++.

1.3 Aspectos Metodológicos

Metodologicamente, o projeto envolveu etapas de planejamento, desenvolvimento e testes, onde foram aplicados conceitos de programação modular, estrutura de dados e algoritmos. A implementação incluiu a criação de uma interface gráfica intuitiva, controle fluido da cobra por meio do teclado e um sistema de pontuação baseado no tamanho crescente da cobra à medida que ela coleta frutas na tela. O

trabalho também abordou aspectos teóricos relacionados ao design de jogos e à interação usuário-sistema, oferecendo uma visão abrangente sobre o processo de desenvolvimento de jogos.

1.4 Aporte Teórico

Os autores José Manzano (2010) e Harvey Deitel (2010) enriquecem o aporte teórico, contribuindo na lógica de programação utilizada no presente trabalho. Foi utilizado também a documentação da biblioteca gráfica Raylib, que foi extremamente necessário para a parte gráfica do jogo.

2 PROJETO PROPOSTO

O desenvolvimento do projeto "COBRINHA!" seguiu uma abordagem estruturada e dividida em fases, desde o planejamento inicial até a implementação e testes finais.

2.1 Planejamento Inicial

No início do projeto, foram definidos os requisitos principais para o jogo, com ênfase em simplicidade e jogabilidade intuitiva. O escopo do projeto foi delimitado para incluir as funcionalidades básicas do jogo “Snake”, tais como movimento contínuo da cobra, detecção de colisão com paredes e com o próprio corpo, e um sistema de pontuação baseado na coleta de frutas. Também foi proposto um atalho para poder pausar o jogo, congelando a cobra na tela.

2.2 Ferramentas Utilizadas

Para o projeto, foram utilizadas diversas ferramentas gratuitas para o desenvolvimento do jogo. A linguagem de programação C++ foi escolhida pela sua eficiência e flexibilidade, permitindo controle preciso sobre memória e execução. Junto com a linguagem, foi utilizado como biblioteca gráfica a Raylib, criado por Ramon Santamaria e contribuidores, por ser simples e de fácil integração com C++, além de ser uma biblioteca *open-sourse* amplamente utilizada para desenvolvimento de jogos 2D.

Foi utilizado para escrever e depurar o código o Visual Studio Code como ambiente de desenvolvimento, por ser um editor de código leve e extensível. E para controle de versão, Git e repositório no GitHub. Também foi aproveitado um *template* do Raylib para o Visual Studio Code, para que fosse possível rodar o jogo pelo ambiente de desenvolvimento sem precisar de outras ferramentas.

2.3 Descrição do Projeto

O jogo "COBRINHA!" consiste em uma tela onde o jogador controla uma cobra que cresce ao consumir frutas espalhadas pelo mapa. A cobra se move continuamente em uma direção e o jogador pode alterar o movimento usando as teclas direcionais. O jogo termina quando a cobra colide com as bordas da tela ou com o próprio corpo. O placar é exibido na tela, indicando o número de frutas coletadas.

Durante o desenvolvimento, foram utilizados conceitos de programação estrutural para organizar o código. A Raylib foi utilizada para desenhar os elementos gráficos na tela e lidar com entradas do teclado, proporcionando uma experiência de usuário fluida e responsiva.

A seguir, um quadro mostrando as entradas que o jogador utilizará durante o jogo:

Entradas	Descrição
Direcional para cima	Move a cobra para cima, caso não esteja nessa direção.
Direcional para baixo	Move a cobra para baixo, caso não esteja nessa direção.
Direcional para a direita	Move a cobra para a direita, caso não esteja nessa direção.
Direcional para a esquerda	Move a cobra para a esquerda, caso não esteja nessa direção.
Tecla “P”	Pausa o jogo, caso já esteja pausado, irá retomar o jogo de onde parou.
Tecla “Enter”	Inicia um novo jogo.

Quadro – Entradas do Jogador

3 RESULTADOS OBTIDOS

O jogo "COBRINHA!" foi implementado com sucesso, atendendo aos requisitos estabelecidos no planejamento inicial. A seguir, serão apresentadas algumas capturas de tela do jogo desenvolvido, incluindo a tela inicial do jogo, a jogabilidade com a pontuação atual, a tela de pause e a tela de “game over”.

3.1 Capturas de tela

As imagens a seguir ilustram a interface gráfica simples e funcional e suas respectivas descrições, que preserva a estética minimalista do jogo original, ao mesmo tempo em que adiciona elementos visuais modernos e responsivos.

Figura 1 – Tela inicial

A Tela inicial tem um fundo preto, com 2 textos centralizados, sendo o primeiro o título do jogo “COBRINHA!” em maiúsculo e na cor verde, e o segundo em branco “Iniciar? <ENTER>”. O jogo só irá começar após o usuário apertar a tecla solicitada.

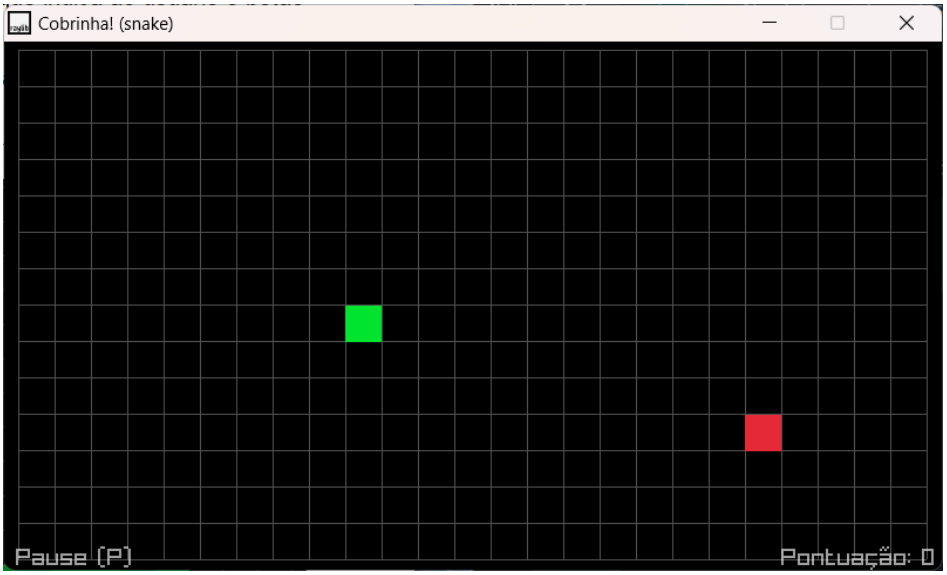

Figura 2 – Jogabilidade (início)

Após a tela inicial, o jogo inicia, com o tamanho da cobra sendo 1, e a primeira fruta na cor RED aparece na tela para ser comida pelo jogador. O fundo continua preto, mas dessa vez com linhas em cinza escuro desenhando a área jogável. Note-se que nos cantos inferiores da tela, contém um atalho que indica ao usuário o botão para pausar o jogo (nesse caso, a letra “P” do teclado), caso necessário, e também mostra a pontuação atual do jogador, indicando quantas frutas ele conseguir comer.

Figura 3 – Jogabilidade (avançado)

Ainda na jogabilidade, dessa vez em um estado mais avançado do jogo, mostrando uma pontuação de 32 frutas comidas. Note-se que a cobra tem a cabeça em um tom mais claro (cor GREEN), para diferenciar do restante do corpo, que tem a cor LIME. Também é possível observar que, a cada 3 frutas comidas, a cobra ganha uma “linha” (uma tonalidade de verde mais escura do que a do corpo, nesse caso utilizando a cor DARKGREEN), para deixar o jogo um pouco mais agradável visualmente.

Figura 4 – Pause

Nessa imagem, é possível ver a mensagem “Jogo Pausado!” centralizada e em branco. Para continuar o jogo, é necessário que o jogador aperte P novamente. Vale ressaltar que, durante essa tela, o jogo fica completamente congelado, sendo impossível perder até que saia do estado de pause.

Figura 5 – Game Over

Após alguma colição contra a parede ou até mesmo com o próprio corpo da cobra, o jogador recebe a mensagem “Você Perdeu! Pontuação: X” que, no caso da imagem, é 32, mas pode mudar de partida a partida. Em seguida, uma nova mensagem “Tentar Novamente? <ENTER>”, indicando ao jogador que a partida será iniciada novamente caso ele aperte o tecla “Enter” no teclado.

4 CONCLUSÃO

O projeto "COBRINHA!" permitiu explorar uma variedade de conceitos e técnicas de programação, desde manipulação de gráficos 2D até controle de fluxo e lógica de jogo. O uso da linguagem C++ em conjunto com a biblioteca Raylib proporcionou uma experiência rica em aprendizado, destacando a importância de uma estrutura de código bem planejada e eficiente.

Como conclusão, o jogo desenvolvido atendeu às expectativas, oferecendo uma experiência de jogabilidade fluida e intuitiva. No entanto, foram identificadas áreas para possíveis melhorias futuras, tais como:

- **Adição de níveis de dificuldade**: Introduzindo obstáculos e variação na velocidade da cobra para aumentar o desafio;
- **Implementação de uma tabela de recordes**: Deixar o jogo mais competitivo, incentivando o jogador a superar o recorde anterior;
- **Aprimoramento visual**: Inclusão de novos gráficos e efeitos visuais para tornar o jogo mais atrativo.

O trabalho contribuiu para o desenvolvimento de habilidades técnicas e práticas no campo de programação de jogos, proporcionando uma base sólida para projetos futuros e para o aprofundamento em áreas relacionadas ao desenvolvimento de software e design de jogos.

REFERÊNCIAS

DEITEL, Harvey. C++ Como Programar. 5. ed. São Paulo: Pearson, 2006.

EDUC8S. Raylib CPP Starter Template for VSCODE. Disponível em: < https://github.com/educ8s/Raylib-CPP-Starter-Template-for-VSCODE>. Acesso em 28 de Outubro. 2024.

MANZANO, José Augusto Navarro Garcia. Programação de Computadores com C++: guia prático de orientação e desenvolvimento. 1. ed. São Paulo: Érica, 2010. 302 p.

RAYLIB. Documentação Raylib. Disponível em: <https://www.raylib.com>. Acesso em: 02 de Novembro. 2024.

RAYLIB. Cheatsheet. Disponível em: <https://www.raylib.com/cheatsheet/cheatsheet.html>. Acesso em: 11 de Novembro. 2024.

RAYSAN. Repositório Raylib. Disponível em: <https://github.com/raysan5/raylib>. Acesso em: 12 de Novembro. 2024.

APÊNDICE A: CÓDIGO COMPLETO DO JOGO “COBRINHA!”

```
// cobrinha.cpp
// baseado no jogo clássico snake

#include "raylib.h"

// ---------- Definindo estruturas ----------

typedef struct Cobra {
    Vector2 pos;    // Posição da cobrinha na tela
    Vector2 tam;    // Tamanho da cobrinha
    Vector2 vel;    // Velocidade da cobrinha
    Color cor;      // Cor da cobrinha
} Cobra;

typedef struct Fruta {
    Vector2 pos;    // Posição da fruta
    Vector2 tam;    // Tamanho da fruta
    bool ativo;     // Estado da fruta: ativa (não foi comida) ou não
    Color cor;      // Cor da fruta
} Fruta;

// ---------- Definindo tamanhos ----------

#define TAM_COBRA      256
#define TAM_QUADRADO    31
#define TAM_LARGURA    800
#define TAM_ALTURA     450

// ---------- Variáveis Globais ----------

int framesCounter = 0;
int contTamCobra = 0;
bool telaInicial = true;
bool primeiraVez = true;
bool gameOver = false;
bool pause = false;
bool permitirMovimento = false;

Fruta fruta = { 0 };
Cobra cobra[TAM_COBRA] = { 0 };
Vector2 cobraPos[TAM_COBRA] = { 0 }; // posição da cobra
Vector2 offset = { 0 }; // deslocamento

Color transWhite = Color{255, 255, 255, 160}; // Cor personalizada (branco, mas com
opacidade)

//------------------------------------------------------------------------
// Controla a lógica do jogo a cada quadro
//------------------------------------------------------------------------
void InitGame(){
    framesCounter = 0;
    gameOver = false;
    pause = false;
```

```
    contTamCobra = 1;
    permitirMovimento = false;

    // calcula o deslocamento da tela, para centralizar a área jogavel, a fruta e a
cobrinha
    offset.x = TAM_LARGURA%TAM_QUADRADO;
    offset.y = TAM_ALTURA%TAM_QUADRADO;

    // definindo a posição, tamanho e cor da cobrinha
    for (int i = 0; i < TAM_COBRA; i++){
        cobra[i].pos = (Vector2){ offset.x/2, offset.y/2 }; // posição inicial no
canto da tela
        cobra[i].tam = (Vector2){ TAM_QUADRADO, TAM_QUADRADO }; // tamanho inicial
é 1x1 (ou 31x31 pixels)
        cobra[i].vel = (Vector2){ TAM_QUADRADO, 0 }; // define a direção inicial da
cobrinha (direita)

        if (i == 0) // Define a cabeça da cobra como VERDE
            cobra[i].cor = GREEN;
        else{ // o resto do corpo ira alternar entre VERDE ESCURO e VERDE LIMÃO
            if (i % 3 == 0)
                cobra[i].cor = DARKGREEN;
            else
                cobra[i].cor = LIME;
        }
    }

    for (int i = 0; i < TAM_COBRA; i++){
        cobraPos[i] = (Vector2){ 0.0f, 0.0f };
    }

    fruta.tam = (Vector2){ TAM_QUADRADO, TAM_QUADRADO }; // tamanho da fruta é 1x1
(ou 31x31 pixels)
    fruta.cor = RED; // Define a cor da fruta como VERMELHO
    fruta.ativo = false;
}

//------------------------------------------------------------------------------
// Controla a lógica do jogo a cada quadro
//------------------------------------------------------------------------------
void UpdateGame(){
    if (!gameOver){ // Se o player não perdeu o jogo...
        if (IsKeyPressed('P')) // Se a tecla 'P' for apertada
            pause = !pause; // muda o estado de pause

        if (!pause){ // Se o jogo não estiver pausado
            // ----- Direção da cobrinha -----
            // Tratamento com entrada de dado feita pelo player

            // Se o player apertar a SETA_DIREITA
            if (IsKeyPressed(KEY_RIGHT) && (cobra[0].vel.x == 0) && permitirMovi-
mento){
                cobra[0].vel = (Vector2){ TAM_QUADRADO, 0 };
                permitirMovimento = false; // não permite novo movimento até que
seja atualizado na tela
```

```
            }
            // Se o player apertar a SETA_ESQUERDA
            if (IsKeyPressed(KEY_LEFT) && (cobra[0].vel.x == 0) && permitirMovimen-
to){
                cobra[0].vel = (Vector2){ -TAM_QUADRADO, 0 };
                permitirMovimento = false; // não permite novo movimento até que
seja atualizado na tela
            }
            // Se o player apertar a SETA_CIMA
            if (IsKeyPressed(KEY_UP) && (cobra[0].vel.y == 0) && permitirMovimen-
to){
                cobra[0].vel = (Vector2){ 0, -TAM_QUADRADO };
                permitirMovimento = false; // não permite novo movimento até que
seja atualizado na tela
            }
            // Se o player apertar a SETA_BAIXO
            if (IsKeyPressed(KEY_DOWN) && (cobra[0].vel.y == 0) && permitirMovimen-
to){
                cobra[0].vel = (Vector2){ 0, TAM_QUADRADO };
                permitirMovimento = false; // não permite novo movimento até que
seja atualizado na tela
            }

            // ----- Movimento da cobrinha -----
            for (int i = 0; i < contTamCobra; i++)
                cobraPos[i] = cobra[i].pos;

            if ((framesCounter % 5) == 0){
                for (int i = 0; i < contTamCobra; i++){
                    if (i == 0){ // se for a cabeça, atualiza a a posição de acordo
com a direção atual
                        cobra[0].pos.x += cobra[0].vel.x;
                        cobra[0].pos.y += cobra[0].vel.y;
                        permitirMovimento = true; // após a atualização do
movimento da cabeça, permite nova direção
                    }
                    else // para o resto do corpo, a posição atual vai para a
posição anterior
                        cobra[i].pos = cobraPos[i-1];
                }
            }

            // ----- Se colidir com a parede -----
            if (((cobra[0].pos.x) > (TAM_LARGURA - offset.x)) ||
                ((cobra[0].pos.y) > (TAM_ALTURA - offset.y)) ||
                (cobra[0].pos.x < 0) || (cobra[0].pos.y < 0))
            {
                gameOver = true;
            }

            // ----- Se colidir com a cobrinha -----
            for (int i = 1; i < contTamCobra; i++){
                if ((cobra[0].pos.x == cobra[i].pos.x) && (cobra[0].pos.y ==
cobra[i].pos.y))
                    gameOver = true;
            }
```

```
            // ----- Calcula a posição da fruta -----
            if (!fruta.ativo){ // Se fruta for comida
                fruta.ativo = true; // Informa que ela está ativa novamente

                // pega uma posição aleatória
                fruta.pos = (Vector2){ GetRandomValue(0, (TAM_LARGURA/TAM_QUADRADO)
- 1)*TAM_QUADRADO + offset.x/2, GetRandomValue(0, (TAM_ALTURA/TAM_QUADRADO) -
1)*TAM_QUADRADO + offset.y/2 };

                // Confere se a posição escolhida não esteja ocupada pela cobra
                for (int i = 0; i < contTamCobra; i++){
                    while ((fruta.pos.x == cobra[i].pos.x) && (fruta.pos.y ==
cobra[i].pos.y)){
                        fruta.pos = (Vector2){ GetRandomValue(0,
(TAM_LARGURA/TAM_QUADRADO) - 1)*TAM_QUADRADO + offset.x/2, GetRandomValue(0,
(TAM_ALTURA/TAM_QUADRADO) - 1)*TAM_QUADRADO + offset.y/2 };
                        i = 0;
                    }
                }
            }

            // ----- Se a cobra colidir (comer) a fruta -----
            if ((cobra[0].pos.x < (fruta.pos.x + fruta.tam.x) && (cobra[0].pos.x +
cobra[0].tam.x) > fruta.pos.x) &&
                (cobra[0].pos.y < (fruta.pos.y + fruta.tam.y) && (cobra[0].pos.y +
cobra[0].tam.y) > fruta.pos.y))
            {
                cobra[contTamCobra].pos = cobraPos[contTamCobra - 1];
                contTamCobra += 1; // Aumenta tamanho da cobra
                fruta.ativo = false; // Registra que a fruta foi comida
            }

            framesCounter++;
        }
    }
    else{
        if (IsKeyPressed(KEY_ENTER)){ // Player inicia o jogo novamente
            InitGame();
            gameOver = false;
        }
    }
}

//------------------------------------------------------------------------------
// Renderização que desenha o estado do jogo na tela
//------------------------------------------------------------------------------
void DrawGame(){
    BeginDrawing();

        ClearBackground(BLACK);

        while(telaInicial){
            BeginDrawing();

                // Mensagem tela inicial
```

```
                DrawText("COBRINHA!", TAM_LARGURA / 2 - MeasureText("COBRINHA!",
80)/2, TAM_ALTURA / 2 - 80, 80, LIME);
                DrawText("Iniciar? <ENTER>", TAM_LARGURA / 2 - Measure-
Text("Iniciar? <ENTER>", 40)/2, TAM_ALTURA / 2, 40, RAYWHITE);

                if(IsKeyPressed(KEY_ENTER))
                    telaInicial = false;

            EndDrawing();
        }

        if (!gameOver){
            // Desenha o grid
            for (int i = 0; i < TAM_LARGURA/TAM_QUADRADO + 1; i++){
                DrawLineV((Vector2){TAM_QUADRADO*i + offset.x/2, offset.y/2},
(Vector2){TAM_QUADRADO*i + offset.x/2, TAM_ALTURA - offset.y/2}, DARKGRAY);
            }

            for (int i = 0; i < TAM_ALTURA/TAM_QUADRADO + 1; i++){
                DrawLineV((Vector2){offset.x/2, TAM_QUADRADO*i + offset.y/2},
(Vector2){TAM_LARGURA - offset.x/2, TAM_QUADRADO*i + offset.y/2}, DARKGRAY);
            }

            // Desenha a cobrinha na tela
            for (int i = 0; i < contTamCobra; i++) DrawRectangleV(cobra[i].pos,
cobra[i].tam, cobra[i].cor);

            // Desenha a fruta na tela
            DrawRectangleV(fruta.pos, fruta.tam, fruta.cor);

            // Escreve os atalhos do jogo no canto inferior esquerdo
            DrawText("Pause (P)", 10, TAM_ALTURA - 20, 20, transWhite);

            // Escreve a pontuação do jogador (quantas frutas comeu) no canto
superior direito
            DrawText(TextFormat("Pontuação: %i", (int)contTamCobra-1), (TAM_LARGURA
- 7) - MeasureText(TextFormat("Pontuação: %i", (int)contTamCobra-1), 20),
TAM_ALTURA - 20, 20, transWhite);

            if (pause)
                DrawText("Jogo Pausado!", TAM_LARGURA/2 - MeasureText("Jogo
Pausado!", 40)/2, TAM_ALTURA/2 - 40, 40, RAYWHITE);
        }
        else{
            DrawText(TextFormat("Você Perdeu! Pontuação: %i", (int)contTamCobra-1),
GetScreenWidth()/2 - MeasureText(TextFormat("Você Perdeu! Pontuação: %.0i",
(int)contTamCobra-1), 40)/2, GetScreenHeight()/2 - 50, 40, RAYWHITE);
            DrawText("Tentar Novamente? <ENTER>", TAM_LARGURA / 2 - Measure-
Text("Tentar Novamente? <ENTER>", 30)/2, TAM_ALTURA / 2, 30, RAYWHITE);
        }

    EndDrawing();
}

//----------------------------------------------------------------------------
// Chama as funções UpdateGame e DrawGame para atualizar o jogo
```

```c
//----------------------------------------------------------------------------------
void UpdateDrawFrame(){
    UpdateGame();
    DrawGame();
}

//----------------------------------------------------------------------------------
// Função principal
//----------------------------------------------------------------------------------
int main(){
    // Inicia a janela
    InitWindow(TAM_LARGURA, TAM_ALTURA, "Cobrinha! (snake)");

    InitGame();

    // Framerate
    SetTargetFPS(60);

    // Loop do jogo
    while (!WindowShouldClose()){ // Enquanto janela não for fechada...
        // Atualiza a tela
        UpdateDrawFrame();
    }

    CloseWindow(); // Fecha a janela e limpa recursos do raylib

    return 0;
}
```