Atividade da disciplina POO

Aluno: Marcos Benner Diniz Moreira
Professor: Paulo Giovanni
Data: 06/10/2024

Realizar uma pesquisa sobre a biblioteca Raylib. Entregar um arquivo em formato PDF, contendo uma introdução sobre o assunto, onde ele é utilizado, quem desenvolveu, exemplos de códigos e de jogos criados com essa biblioteca. Não se esqueça de acrescentar as referências bibliográficas utilizadas para a realização do trabalho. ATENÇÃO: A entrega deverá ser feita pelo próprio Google Classroom, até o dia 06 de outubro de 2024.

1 - Introdução

A história da Raylib começa em agosto de 2013, quando Ramón Santamaría começou a desenvolver a biblioteca para apoiar um curso de desenvolvimento de videojogos. O objetivo era ensinar aos estudantes, mesmo àqueles que não fossem especialistas em programação ou design gráficos.

Raylib é uma biblioteca C/C++(Além destas, existem bindings para outras linguagens como Python, Lua e C#) de código aberto voltada para o desenvolvimento de jogos e aplicações gráficas. Foi projetada para ser simples e fácil de usar, tornando-a uma excelente escolha para iniciantes e para aqueles que desejam desenvolver jogos sem uma barreira de entrada muito alta quando comparada a outras bibliotecas gráficas. Raylib oferece uma grande quantidade de funcionalidades, como: renderização de gráficos 2D e 3D, detecção de colisão, manipulação de som, entrada de usuários (teclado, mouse e gamepad) e suporte a texturas e animações.

2 - Onde Raylib é Utilizada

A Raylib é uma biblioteca muito versátil que encontra aplicação em diversos setores do desenvolvimento de software e jogos, por causa da sua simplicidade, eficiência e flexibilidade. esta biblioteca é frequentemente utilizada em instituições de ensino, como universidades, escolas técnicas e cursos online, para ensinar programação gráfica e desenvolvimento de jogos. Ela também pode ser adotada em workshops, bootcamps e tutoriais destinados a iniciantes pois, esta foi desenvolvida visando justamente o ensino-aprendizagem sendo simples e bem documentada, o que facilita a compreensão dos conceitos básicos de programação gráfica e desenvolvimento de jogos para estudantes e iniciantes a rapidez permite que os alunos vejam resultados imediatos de seu código, mantendo a motivação e

facilitando a assimilação dos conceitos, outro ponto a se ressaltar é que existem exemplos, tutoriais e uma comunidade ativa a qual fornece um suporte adicional para o processo de ensino e aprendizagem.

Comumente usada também por Desenvolvedores Independentes que utilizam o Raylib para criar jogos, desde protótipos até lançamentos finais. É ideal para aplicações de pequena a média escala. Esses DEVs optam por esta ferramenta pois, não exige muitos recursos, o que é ideal para profissionais com equipes pequenas e orçamentos limitados. Por ser código aberto, permite a publicação dos jogos sem restrições significativas de licenciamento.

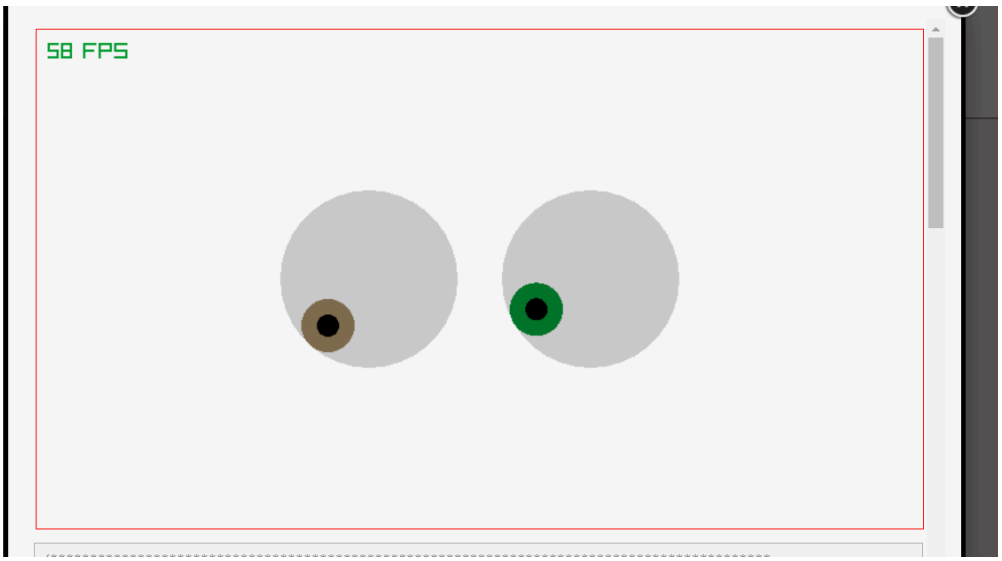
3 - Quem desenvolveu a Raylib?

Foi criada por Ramon Santamaria, um desenvolvedor de software espanhol. Ele desenvolveu a biblioteca com o intuito de criar uma ferramenta simples e eficiente para ensinar desenvolvimento de jogos e programação gráfica, além de servir como uma base prática para o desenvolvimento de seus próprios projetos.

Ramon começou a trabalhar na Raylib como um projeto pessoal. Ele queria uma solução que fosse fácil o suficiente para iniciantes, mas visava funcionalidades para criar projetos mais avançados. Em 2014, Raylib foi oficialmente lançada como uma biblioteca de código aberto, e rapidamente começou a atrair a atenção de desenvolvedores independentes e educadores, especialmente aqueles que buscavam uma solução simples para ensinar desenvolvimento de jogos e programação gráfica.

4 - Exemplo de códigos

Nesse exemplo é demonstrada a criação de olhos interativos o quais seguem a direção que o mouse aponta. código pode ser adaptado de acordo com a necessidade do desenvolvedor.




```c
#include "raylib.h"

#include <math.h>       // Required for: atan2f()

//------------------------------------------------------------------------------------
// Program main entry point
//------------------------------------------------------------------------------------
int main(void)
{
    // Initialization
    //--------------------------------------------------------------------------------------
    const int screenWidth = 800;
    const int screenHeight = 450;

    InitWindow(screenWidth, screenHeight, "raylib [shapes] example - following eyes");

    Vector2 scleraLeftPosition = { GetScreenWidth()/2.0f - 100.0f, GetScreenHeight()/2.0f };
    Vector2 scleraRightPosition = { GetScreenWidth()/2.0f + 100.0f, GetScreenHeight()/2.0f };
    float scleraRadius = 80;

    Vector2 irisLeftPosition = { GetScreenWidth()/2.0f - 100.0f, GetScreenHeight()/2.0f };
    Vector2 irisRightPosition = { GetScreenWidth()/2.0f + 100.0f, GetScreenHeight()/2.0f };
    float irisRadius = 24;

    float angle = 0.0f;
    float dx = 0.0f, dy = 0.0f, dxx = 0.0f, dyy = 0.0f;

    SetTargetFPS(60);               // Set our game to run at 60 frames-per-second
    //--------------------------------------------------------------------------------------
```

```
    // Main game loop
    while (!WindowShouldClose())    // Detect window close button or ESC key
    {
        // Update
        //----------------------------------------------------------------------------------
        irisLeftPosition = GetMousePosition();
        irisRightPosition = GetMousePosition();

        // Check not inside the left eye sclera
        if (!CheckCollisionPointCircle(irisLeftPosition, scleraLeftPosition, scleraRadius - irisRadius))
        {
            dx = irisLeftPosition.x - scleraLeftPosition.x;
            dy = irisLeftPosition.y - scleraLeftPosition.y;

            angle = atan2f(dy, dx);

            dxx = (scleraRadius - irisRadius)*cosf(angle);
            dyy = (scleraRadius - irisRadius)*sinf(angle);

            irisLeftPosition.x = scleraLeftPosition.x + dxx;
            irisLeftPosition.y = scleraLeftPosition.y + dyy;
        }

        // Check not inside the right eye sclera
        if (!CheckCollisionPointCircle(irisRightPosition, scleraRightPosition, scleraRadius - irisRadius))
        {
            dx = irisRightPosition.x - scleraRightPosition.x;
            dy = irisRightPosition.y - scleraRightPosition.y;

            angle = atan2f(dy, dx);

            dxx = (scleraRadius - irisRadius)*cosf(angle);
            dyy = (scleraRadius - irisRadius)*sinf(angle);

            irisRightPosition.x = scleraRightPosition.x + dxx;
            irisRightPosition.y = scleraRightPosition.y + dyy;
        }
        //----------------------------------------------------------------------------------
```

```
        // Draw
        //----------------------------------------------------------------------------------
        BeginDrawing();

            ClearBackground(RAYWHITE);

            DrawCircleV(scleraLeftPosition, scleraRadius, LIGHTGRAY);
            DrawCircleV(irisLeftPosition, irisRadius, BROWN);
            DrawCircleV(irisLeftPosition, 10, BLACK);

            DrawCircleV(scleraRightPosition, scleraRadius, LIGHTGRAY);
            DrawCircleV(irisRightPosition, irisRadius, DARKGREEN);
            DrawCircleV(irisRightPosition, 10, BLACK);

            DrawFPS(10, 10);

        EndDrawing();
        //----------------------------------------------------------------------------------
    }

    // De-Initialization
    //--------------------------------------------------------------------------------------
    CloseWindow();        // Close window and OpenGL context
    //--------------------------------------------------------------------------------------

    return 0;
}
```

5 - Jogos criados com a Biblioteca Reylib.

Ninja Twins

Ninja Twins é um jogo de puzzle desenvolvido pelo próprio criador da Raylib, Ramon Santamaria. O jogo desafia os jogadores a resolverem enigmas utilizando habilidades de movimentação e interação dos personagens. É um excelente exemplo de como Raylib pode ser utilizada para criar jogos com mecânicas envolventes e gráficos simples, mas eficazes.

Missile Command Clone

Clones de jogos clássicos como Missile Command são populares em comunidades de desenvolvedores que utilizam Raylib. Estes projetos servem tanto como exercícios de aprendizado quanto como demonstrações das capacidades gráficas e de controle de Raylib.
Gráficos 2D Retro: Emulação do estilo visual dos jogos arcade clássicos.
Mecânica de Defesa: Os jogadores devem defender áreas específicas lançando mísseis para interceptar ameaças.
Detecção de Colisões: Implementação de colisões entre mísseis e alvos, demonstrando o uso de funcionalidades de Raylib.

Raylib RPG Demo

Este é um projeto de demonstração que mostra como Raylib pode ser utilizada para criar elementos de RPG, incluindo movimentação de personagem, interação com NPCs e gráficos de tileset.

Gráficos de Tileset: Utilização de tiles para construir ambientes de jogo detalhados.Movimentação e Animação: Controle suave do personagem com animações básicas. Interação com Personagens: Implementação de diálogos e interações simples com NPCs.

6 - Conclusão

Portanto, mesmo que Raylib possa não ser a escolha principal para grandes estúdios de jogos devido à sua natureza mais orientada a educadores e desenvolvedores independentes, ela desempenha um papel crucial no ecossistema de criação de jogos. Projetos como Ninja Twins, clones de clássicos como Missile Command e Pong, e diversas criações em game jams demonstram a diversidade e a eficácia de Raylib para criar jogos atraentes para a comunidade gamer. A contínua expansão da comunidade e o suporte ativo garantem que esta biblioteca continuará a ser uma ferramenta valiosa para desenvolvedores de todos os níveis.

7 - Referências

Site oficial da biblioteca: https://www.raylib.com/ [acessado em 05/10/2024];
Link do jogo ninja twins: https://makrill.itch.io/ninja-twins [acessado em 05/10/2024];
Video de instalação: https://youtu.be/-F6THkPkF2I?si=COoi8ArTPiG-5HI6 [acessado em 05/06/2024].