

Raylib: A C nyelvű játékfejlesztési könyvtár

A **raylib** egy egyszerű és könnyen használható C nyelvű könyvtár, amit játékfejlesztéshez és multimédiás alkalmazásokhoz készítettek.

Fő jellemzői

- **Platformfüggetlen:** Windows, Linux, MacOS és több más rendszeren is működik.
- **Könnyű Integrálni:** Nincs szükség külső függőségekre vagy bonyolult telepítési folyamatokra.
- **2D és 3D támogatás:** Támogatja mind a 2D, mind a 3D grafikai és audio funkcionalitást.

Alapvető funkciók

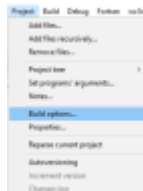
- **Ablakkezelés:** Könnyű létrehozni és kezelni az ablakokat, valamint kezelni az eseményeket.
- **Grafika:** Támogatja a 2D-s és 3D-s rajzolást, textúrák és sprite-ok kezelését.
- **Hangkezelés:** Képes audio fájlok lejátszására és hang effektek kezelésére.
- **Bemenet kezelése:** Egér, billentyűzet és más bemeneti eszközök támogatása.

Források

- **Hivatalos Weboldal:** raylib.com
- **Dokumentáció:** Teljes körű API dokumentáció elérhető a hivatalos weboldalon.
- **Példák és Tutorialok:** Sok példa és oktatóanyag található az oldalon

Telepítés és konfiguráció

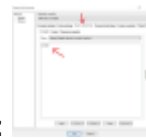
1. Van saját telepítője is, de egyszerűbb ha letöltjük az **include** és **lib** fájlokat tartalmazó **raylib.zip**-et.
2. Hozzunk létre egy konzol alkalmazást a codeblocks-ban
3. A létrejött **main.c** fájl könyvtárába tömörítsük ki a letöltött zip-et



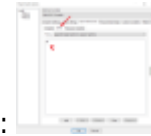
4. Nyissuk meg a **Projekt/Build options** menüt:



5. A linker settings fülön állítsuk be **/lib/libraylibdll.a**:



6. A search directories fülön adjuk hozzá az **include** könyvtárat:



7. Ugyanitt váltsunk a **linker** fülre és adjuk hozzá a **lib** könyvtárat:

Írjuk felül a jelenlegi main.c tartalmát az alábbi kóddal és futtassuk a programot.

```
#include <raylib.h>

int main() {
    // Window creation
    const int screenWidth = 640;
    const int screenHeight = 480;
    InitWindow(screenWidth, screenHeight, "raylib example");

    // Initialization
    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

        // Draw
        BeginDrawing();

            ClearBackground(LIGHTGRAY); // Clear the background with a color

            int MaxX = GetScreenWidth();
            int MaxY = GetScreenHeight();

            for (int i = 1; i <= 10; i++) {
                Color lineColor = GetColor(i + 5); // Choose color
                DrawLine(0, 0, (MaxX / 10) * i, MaxY, lineColor); // Draw
line

                int R = (MaxY - 10) / (2 * i); // Circle radius
                DrawCircle(MaxX - R, MaxY / 2, R, lineColor); // Draw circle

                // Draw ellipse
                DrawEllipse(MaxX / 2, MaxY / 8, MaxX / (4 * i), MaxY / 8,
lineColor);
            }

        EndDrawing();
    }

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

    return 0;
}
```

```
}

#include <raylib.h>
#include <math.h>

int main() {
    // Initialization
    const int screenWidth = 800;
    const int screenHeight = 450;
    InitWindow(screenWidth, screenHeight, "Sinus Function Plot - raylib");

    SetTargetFPS(60); // Set FPS

    while (!WindowShouldClose()) { // Main game loop
        // Update

        // Draw
        BeginDrawing();

        ClearBackground(RAYWHITE); // Clear the background

        // Draw the axes
        Vector2 origin = { (float)screenWidth/2, (float)screenHeight/2 };

        DrawLine(origin.x, 0, origin.x, screenHeight, BLACK); // Y-axis
        DrawLine(0, origin.y, screenWidth, origin.y, BLACK); // X-axis

        // Draw the sine function
        for(int i = -screenWidth/2; i < screenWidth/2; i++) {
            // Calculating points
            float x1 = (float)i;
            float y1 = sinf(x1 * DEG2RAD) * 100; // Scale the sine wave
            float x2 = x1 + 1;
            float y2 = sinf(x2 * DEG2RAD) * 100; // Scale the sine wave

            // Transform points to screen space
            x1 += origin.x;
            y1 = origin.y - y1; // Invert y1 to match screen coordinates
            x2 += origin.x;
            y2 = origin.y - y2; // Invert y2 to match screen coordinates

            // Draw line segment
            DrawLine(x1, y1, x2, y2, BLUE);
        }

        DrawText("Sinus Function Plot", 10, 10, 20, BLACK); // Title
        DrawText("X-Axis", screenWidth - 50, origin.y + 10, 10, BLACK);
        // X-axis label
    }
}
```

```
        DrawText("Y-Axis", origin.x + 10, 10, 10, BLACK); // Y-axis
label

        EndDrawing();
    }

    CloseWindow();

    return 0;
}
```

From:

<https://edu.iit.uni-miskolc.hu/> - Institute of Information Science - University of Miskolc

Permanent link:

https://edu.iit.uni-miskolc.hu/muszaki_informatika:raylib?rev=1706461198

Last update: **2024/01/28 16:59**

