**Documentación del Proyecto: Gravity\_Ball (con SFML 3.0.0)**

Este documento resume la configuración completa del entorno de desarrollo para un proyecto en C++ con SFML 3.0.0 usando Visual Studio Code en Windows.

**Objetivo del proyecto**

Crear una simulación simple de física 2D con una bola que rebota, usando SFML para gráficos.

**Estructura del proyecto**

Gravity\_Ball/

│ ── .vscode/

│ ── tasks.json

│ ── launch.json

│ ── cpp\_properties.json

│ ── main.cpp

└── output/

└── main.exe (generado tras la compilación)

**Requisitos del sistema**

* Sistema operativo: Windows 10/11
* Compilador: g++ (de MinGW-w64)
* Ruta del compilador: C:/msys64/ucrt64/bin/g++
* Versión de MinGW: 13.2.0
* Entorno de desarrollo: Visual Studio Code
* Extensión de C/C++: Microsoft C/C++ Extension (ms-vscode.cpptools)
* SFML versión: 3.0.0
* Ruta de SFML: C:/SFML/SFML-3.0.0

**Configuración de tasks.json**

"version": "2.0.0",

"tasks": [

{

"label": "Compilar SFML",

"type": "shell",

"command": "g++",

"args": [

"-IC:/SFML/SFML-3.0.0/include",

"-DSFML\_STATIC",

"-LC:/SFML/SFML-3.0.0/lib",

"${workspaceFolder}/.vscode/main.cpp",

"-lsfml-graphics-s",

"-lsfml-window-s",

"-lsfml-system-s",

"-lopengl32",

"-lfreetype",

"-lwinmm",

"-lgdi32",

"-mconsole",

"-o",

"${workspaceFolder}/.vscode/main.exe"

],

"group": {

"kind": "build",

"isDefault": true

},

"problemMatcher": ["$gcc"]

}

]

**Configuración de launch.json**

{

"version": "0.2.0",

"configurations": [

{

"name": "Ejecutar SFML",

"type": "cppdbg",

"request": "launch",

"program": "${workspaceFolder}/.vscode/main.exe",

"args": [],

"stopAtEntry": false,

"cwd": "${workspaceFolder}",

"environment": [],

"externalConsole": true,

"MIMode": "gdb",

"miDebuggerPath": "gdb.exe",

"setupCommands": [

{

"description": "Habilitar formato pretty-printing",

"text": "-enable-pretty-printing",

"ignoreFailures": true

}

]

}

]

**Configuración de cpp\_properties .json**

{

    "configurations": [

        {

            "name": "Win32",

            "includePath": [

                "${workspaceFolder}/\*\*",

                "C:/SFML/SFML-3.0.0/include"

            ],

            "defines": [

                "\_DEBUG",

                "UNICODE",

                "\_UNICODE"

            ],

            "cStandard": "c17",

            "cppStandard": "gnu++17",

            "intelliSenseMode": "windows-gcc-x64"

        }

    ],

    "version": 4

}

**Detalles de configuración**

* **SFML estático:** Se usa -DSFML\_STATIC para enlazar contra las versiones estáticas de las bibliotecas.
* **Dependencias explícitas:** -lopengl32 -lfreetype -lwinmm -lgdi32 necesarias al usar SFML en modo estático.
* **Ruta de inclusión y bibliotecas:** Especificadas con -I y -L.
* **Consola externa:** Se activa con "externalConsole": true

**Comando manual de compilación (terminal)**

g++ -IC:/SFML/SFML-3.0.0/include -DSFML\_STATIC -LC:/SFML/SFML-3.0.0/lib \

.vscode/main.cpp -lsfml-graphics-s -lsfml-window-s -lsfml-system-s \

-lopengl32 -lfreetype -lwinmm -lgdi32 -mconsole -o .vscode/main.exe

**Notas adicionales**

* Asegúrate de tener g++ y gdb disponibles en el PATH o especifica su ruta completa si no los reconoce.
* Si usas OneDrive, ten cuidado con rutas largas o problemas de sincronización.
* Los errores más comunes fueron:
  + SFML/Graphics.hpp: No such file or directory → falta de -I
  + cannot find lopengl32 → faltaba -l antes del nombre: debía ser -lopengl32
  + undefined reference to 'WinMain' → usar -mconsole al compilar