ImGui + SDL + Boost.Signals2 Setup Guide

Prerequisites

- Visual Studio 2022 (recommended)
- vcpkg package manager
- Administrator privileges

1. Environment Setup

1.1 VCPKG Setup

1. Locate your vcpkg installation using Command Prompt:

```
where vcpkg
```

2. Set VCPKG_ROOT environment variable:
    - Open System Properties → Advanced → Environment Variables
    - Add new System Variable:
        - Name: VCPKG_ROOT
        - Value: Your vcpkg path (e.g., D:\vcpkg)

3. Verify setup:

```
echo %VCPKG_ROOT%
```

🚨 **IMPORTANT**: Restart Visual Studio after setting environment variables

2. Install Dependencies

2.1 Verify and Install Required Packages

First, check if ImGui is already installed:

```
vcpkg list | findstr imgui
```

If ImGui is already installed and you want to start fresh:

```
vcpkg remove imgui:x64-windows
vcpkg remove sdl2:x64-windows
```

Then install the required packages. Run in Command Prompt as administrator:

```
vcpkg install sdl2:x64-windows
vcpkg install boost-signals2:x64-windows
vcpkg install imgui[sdl2-binding,opengl3-binding]:x64-windows --recurse
```

After installation, verify that the ImGui SDL2 implementation files are present:

```
dir "installed\x64-windows\include\imgui\imgui_impl_sdl2.h"
```

If the implementation files are missing, you may need to reinstall with the recursive flag:

```
vcpkg remove imgui:x64-windows
vcpkg install imgui[sdl2-binding,opengl3-binding]:x64-windows --recurse --clean-after-build
```

2.2 Verify Libraries

Check these paths:

SDL2 Files:

```
%VCPKG_ROOT%\installed\x64-windows\include\SDL2
%VCPKG_ROOT%\installed\x64-windows\bin\SDL2.dll
%VCPKG_ROOT%\installed\x64-windows\lib\SDL2.lib
%VCPKG_ROOT%\installed\x64-windows\debug\bin\SDL2d.dll
%VCPKG_ROOT%\installed\x64-windows\debug\lib\SDL2d.lib
```

ImGui Files:

```
%VCPKG_ROOT%\installed\x64-windows\include\imgui\imgui.h
%VCPKG_ROOT%\installed\x64-windows\include\imgui\imgui_impl_sdl2.h
%VCPKG_ROOT%\installed\x64-windows\include\imgui\imgui_impl_opengl3.h
%VCPKG_ROOT%\installed\x64-windows\lib\imgui.lib
%VCPKG_ROOT%\installed\x64-windows\debug\lib\imguid.lib
```

3. Project Setup

3.1 Create New Project

1. Open Visual Studio
2. File → New → Project
3. Select "Empty Project" (C++)
4. Set Platform to x64
5. Choose project name/location

3.2 Project Properties Configuration

Right-click project → Properties, then set:

General Settings (All Configurations):

- Platform: x64
- C++ Language Standard: C++17
- Character Set: Use Multi-Byte Character Set
- Windows SDK Version: Latest
- Platform Toolset: Visual Studio 2022 (v143)

Include Directories (All Configurations): C/C++ → General → Additional Include Directories:

```
$(VCPKG_ROOT)\installed\x64-windows\include
```

Library Directories: Debug Configuration:

```
$(VCPKG_ROOT)\installed\x64-windows\debug\lib
```

Release Configuration:

```
$(VCPKG_ROOT)\installed\x64-windows\lib
```

Additional Dependencies: Debug Configuration:

```
SDL2d.lib
imguid.lib
opengl32.lib
```

Release Configuration:

```
SDL2.lib
imgui.lib
opengl32.lib
```

Runtime Library:

- Debug: Multi-threaded Debug DLL (/MDd)
- Release: Multi-threaded DLL (/MD)

Subsystem: Linker → System → SubSystem:

- Windows (/SUBSYSTEM:WINDOWS)

3.3 Environment PATH Setup

Debugging → Environment:

Debug Configuration:

```
PATH=$(VCPKG_ROOT)\installed\x64-windows\debug\bin;%PATH%
```

Release Configuration:

```
PATH=$(VCPKG_ROOT)\installed\x64-windows\bin;%PATH%
```

4. Sample Code

Create main.cpp in your project and add this code:

```
#include <imgui/imgui.h>
#include <imgui/imgui_impl_sdl2.h>
#include <imgui/imgui_impl_opengl3.h>
#include <SDL2/SDL.h>
#include <SDL2/SDL_opengl.h>
#include <boost/signals2.hpp>
#include <iostream>
#include <string>
#include <Windows.h>

// Signal for window events
boost::signals2::signal<void(const std::string&)> onWindowEvent;

// Global variables for theming
float backgroundColor[3] = { 0.45f, 0.55f, 0.60f }; // RGB values for background color
bool isDarkTheme = true;

void printEvent(const std::string& msg) {
    std::cout << "Event: " << msg << std::endl;
}

// Windows Entry Point
```

```cpp
int WINAPI WinMain(HINSTANCE hInstance, HINSTANCE hPrevInstance, LPSTR lpCmdLine, int nCmdShow
{
    // Connect signal
    onWindowEvent.connect(&printEvent);

    // Initialize SDL
    if (SDL_Init(SDL_INIT_VIDEO | SDL_INIT_TIMER) != 0) {
        std::cerr << "Error: " << SDL_GetError() << std::endl;
        return -1;
    }

    // Setup OpenGL version
    SDL_GL_SetAttribute(SDL_GL_CONTEXT_FLAGS, 0);
    SDL_GL_SetAttribute(SDL_GL_CONTEXT_PROFILE_MASK, SDL_GL_CONTEXT_PROFILE_CORE);
    SDL_GL_SetAttribute(SDL_GL_CONTEXT_MAJOR_VERSION, 3);
    SDL_GL_SetAttribute(SDL_GL_CONTEXT_MINOR_VERSION, 0);

    // Create window with graphics context
    SDL_GL_SetAttribute(SDL_GL_DOUBLEBUFFER, 1);
    SDL_GL_SetAttribute(SDL_GL_DEPTH_SIZE, 24);
    SDL_GL_SetAttribute(SDL_GL_STENCIL_SIZE, 8);
    SDL_WindowFlags window_flags = (SDL_WindowFlags)(SDL_WINDOW_OPENGL | SDL_WINDOW_RESIZABLE
    SDL_Window* window = SDL_CreateWindow("ImGui + SDL Example",
        SDL_WINDOWPOS_CENTERED, SDL_WINDOWPOS_CENTERED,
        1280, 720,
        window_flags);
    SDL_GLContext gl_context = SDL_GL_CreateContext(window);
    SDL_GL_MakeCurrent(window, gl_context);
    SDL_GL_SetSwapInterval(1); // Enable vsync

    // Setup Dear ImGui
    IMGUI_CHECKVERSION();
    ImGui::CreateContext();
    ImGuiIO& io = ImGui::GetIO(); (void)io;
    io.ConfigFlags |= ImGuiConfigFlags_NavEnableKeyboard;

    ImGui::StyleColorsDark();

    // Setup Platform/Renderer backends
    ImGui_ImplSDL2_InitForOpenGL(window, gl_context);
    ImGui_ImplOpenGL3_Init("#version 130");

    // Emit window creation event
    onWindowEvent("Window created successfully");

    // Main loop
    bool done = false;
    while (!done)
    {
        SDL_Event event;
        while (SDL_PollEvent(&event))
        {
            ImGui_ImplSDL2_ProcessEvent(&event);
```

```
            if (event.type == SDL_QUIT)
                done = true;
            if (event.type == SDL_WINDOWEVENT && event.window.event == SDL_WINDOWEVENT_CLOSE)
                done = true;
        }

        // Start the Dear ImGui frame
        ImGui_ImplOpenGL3_NewFrame();
        ImGui_ImplSDL2_NewFrame();
        ImGui::NewFrame();

        // Create ImGui window
        {
            ImGui::Begin("Example Window");

            static float value = 0.0f;
            if (ImGui::SliderFloat("Slider", &value, 0.0f, 1.0f)) {
                onWindowEvent("Slider value changed: " + std::to_string(value));
            }

            // Color Preview
            ImGui::ColorEdit3("Background Color", backgroundColor);

            // Enhanced button functionality
            if (ImGui::Button("Click Me!")) {
                if (isDarkTheme) {
                    ImGui::StyleColorsLight();
                    backgroundColor[0] = 0.9f;
                    backgroundColor[1] = 0.9f;
                    backgroundColor[2] = 0.9f;
                }
                else {
                    ImGui::StyleColorsDark();
                    backgroundColor[0] = 0.45f;
                    backgroundColor[1] = 0.55f;
                    backgroundColor[2] = 0.60f;
                }
                isDarkTheme = !isDarkTheme;
                onWindowEvent("Theme changed to: " + std::string(isDarkTheme ? "Dark" : "Light
            }

            ImGui::Text("Click the button to toggle between dark and light theme!");

            // Display current theme status
            ImGui::TextColored(
                isDarkTheme ? ImVec4(1.0f, 1.0f, 0.0f, 1.0f) : ImVec4(0.0f, 0.0f, 1.0f, 1.0f),
                "Current Theme: %s",
                isDarkTheme ? "Dark" : "Light"
            );

            // Add Demo Window for reference
            static bool showDemo = false;  // Persistent state for demo window
            if (ImGui::Button("Toggle Demo Window")) {
```

```
                showDemo = !showDemo;
            }
            if (showDemo) {  // Render demo window if enabled
                ImGui::ShowDemoWindow(&showDemo);
            }

            ImGui::End();
        }

        // Rendering
        ImGui::Render();
        SDL_GL_MakeCurrent(window, gl_context);
        glViewport(0, 0, (int)io.DisplaySize.x, (int)io.DisplaySize.y);
        glClearColor(backgroundColor[0], backgroundColor[1], backgroundColor[2], 1.0f);
        glClear(GL_COLOR_BUFFER_BIT);
        ImGui_ImplOpenGL3_RenderDrawData(ImGui::GetDrawData());
        SDL_GL_SwapWindow(window);
    }

    // Cleanup
    ImGui_ImplOpenGL3_Shutdown();
    ImGui_ImplSDL2_Shutdown();
    ImGui::DestroyContext();

    SDL_GL_DeleteContext(gl_context);
    SDL_DestroyWindow(window);
    SDL_Quit();

    return 0;
}
```

5. Build and Run

1. Make sure configuration is set to "Debug" and platform to "x64"
2. Build Solution (F7)
3. Run (F5)

Expected result:

- Window appears with ImGui interface
- Interactive UI elements working
- Theme toggle functioning
- Background color changeable
- Demo window available

6. Troubleshooting

Common Build Errors

LNK1168: Cannot open exe for writing

```
Error LNK1168: cannot open ... .exe for writing
```

Fix:

1. Close any running instances of your application
2. Close Visual Studio
3. Check Task Manager and end related processes
4. Delete the executable manually if needed
5. Clean and Rebuild solution

Library Not Found

```
Error LNK2019: unresolved external symbol ... referenced
```

Fix:

1. Check library paths in project properties
2. Verify debug/release library names match configuration
3. Ensure platform (x64) is consistent
4. Verify all required libraries are listed in Additional Dependencies

Include Files Not Found

```
Cannot open include file: 'imgui.h': No such file or directory
```

Fix:

1. Verify vcpkg installation is complete:

   ```
   vcpkg list
   dir "installed\x64-windows\include\imgui"
   ```

2. Check Additional Include Directories in project properties
3. Ensure include paths use correct folder structure (imgui/imgui.h)
4. If files are missing, try reinstalling:

   ```
   vcpkg remove imgui:x64-windows
   vcpkg remove sdl2:x64-windows
   ```

```
vcpkg install imgui[sdl2-binding,opengl3-binding]:x64-windows --recurse --clean-after-buil
```

Runtime DLL Issues

```
The application was unable to start correctly (0xc000007b)
```

Fix:

1. Verify PATH environment variable in project settings
2. Check debug/release DLL matching
3. Ensure all required DLLs are in the correct folders
4. Rebuild solution

Tips and Reminders

- Always build in x64 configuration
- Match debug/release libraries correctly
- Use proper include paths with imgui/ prefix
- Restart Visual Studio after environment changes
- Check Output window for detailed error messages

7. Features Overview

UI Elements

1. Main window with:
    - Slider control
    - Theme toggle button
    - Color picker
    - Demo window toggle
    - Status display

Functionality

1. Theme Switching:

    - Dark/Light theme toggle
    - Background color changes
    - Dynamic UI updates

2. Event System:

    - Slider value changes
    - Theme changes
    - Window creation events

3. Graphics:

    - OpenGL context
    - ImGui rendering
    - Vsync support

Additional Notes

- Uses Boost.Signals2 for event handling
- SDL2 for window management and OpenGL context
- Full ImGui demo window available for reference
- Proper folder structure maintained through vcpkg