

## Objectives

- Identify and research free libraries that can replace Qt libraries.
  - Successfully install and configure wxWidgets, Boost.Signals2, and SFML libraries.
  - Transform existing Qt-based code to use the newly installed libraries while maintaining core functionality.
  - Debug and enhance the project code to address identified issues and meet key requirements.
- 

## Activities

### 1. Library Research and Installation

- Researched free libraries as alternatives to Qt.
- Found and reviewed a YouTube video tutorial on installing wxWidgets.
- Installed wxWidgets by following the YouTube video [Installation Tutorial](#).
- Documented the wxWidgets installation manual with citations for future reference.

### 2. Additional Library Installations

- Installed Boost.Signals2 and SFML libraries using vcpkg.

### 3. Qt Code Updates and Library Integration

- Updated Qt test code to align with three major points, addressing prior focus only on UI and signals.
- Transformed the Qt-based code to use wxWidgets, Boost.Signals2, and SFML libraries.
- Configured library paths and settings in Visual Studio for both Debug and Release modes:
  - **Include Paths:** \$(VCPKG\_ROOT)\installed\x64-windows\include

- **Debug Library Paths:** \$(VCPKG\_ROOT)\installed\x64-windows\debug\lib
- **Release Library Paths:** \$(VCPKG\_ROOT)\installed\x64-windows\lib
- **Debug Libraries:** sfml-graphics-d.lib, sfml-window-d.lib, sfml-system-d.lib
- **Release Libraries:** sfml-graphics.lib, sfml-window.lib, sfml-system.lib

#### 4. Debugging Project Properties

- Resolved configuration issues:
  - **Problem:** Cannot open file 'sfml-graphics-d.lib'
    - **Solution:** Reinstalled SFML in Debug mode using vcpkg install sfml:x64-windows.
  - **Problem:** Missing DLLs at runtime.
    - **Solution:** Copied DLLs from D:\vcpkg\installed\x64-windows\debug\bin to the output folder.

#### 5. Code Transformation and Focus Points

- Replaced Qt UI elements with wxWidgets (e.g., wxFrame, wxPanel).
- Used Boost.Signals2 for implementing the signal-slot system.
- Integrated SFML for scene management (e.g., Scene, GraphicsView, GraphicsItem).

#### 6. Code Debugging and Enhancement

- Addressed coding issues:
  - **Problem:** Cannot modify const sf::RectangleShape.
    - **Solution:** Used a temporary non-const sf::RectangleShape for drawing.
  - **Problem:** wxWindowBase::ProcessEvent access issue.

- **Solution:** Replaced `ProcessEvent` with `wxQueueEvent` for proper event queuing.
  - Added enhanced functionalities:
    - Mouse zoom, pan, and selection.
    - Scene-based rendering and item transformations.
    - Proper scaling and aspect ratio maintenance.
- 

## Achievements

- Successfully installed `wxWidgets`, `Boost.Signals2`, and `SFML` libraries.
  - Documented the `wxWidgets` installation process for easy reference.
  - Transformed Qt-based code to use `wxWidgets`, `Boost.Signals2`, and `SFML` libraries.
  - Debugged and resolved critical issues in the project properties and code.
  - Implemented advanced functionalities such as scene rendering, item transformations, and improved interactivity.
- 

## Problems & Solutions

1. **Problem:** Cannot open file '`sfml-graphics-d.lib`'
  - **Solution:** Reinstalled SFML in Debug mode using `vcpkg install sfml:x64-windows`.
2. **Problem:** Missing DLLs at runtime.
  - **Solution:** Copied DLLs from `D:\vcpkg\installed\x64-windows\debug\bin` to the output folder.
3. **Problem:** Cannot modify `const sf::RectangleShape`.
  - **Solution:** Used a temporary non-const `sf::RectangleShape` for drawing.
4. **Problem:** `wxWindowBase::ProcessEvent` access issue.

- **Solution:** Replaced ProcessEvent with wxQueueEvent for proper event queuing.