Game Engine Development II

Week1

Hooman Salamat

Instructor

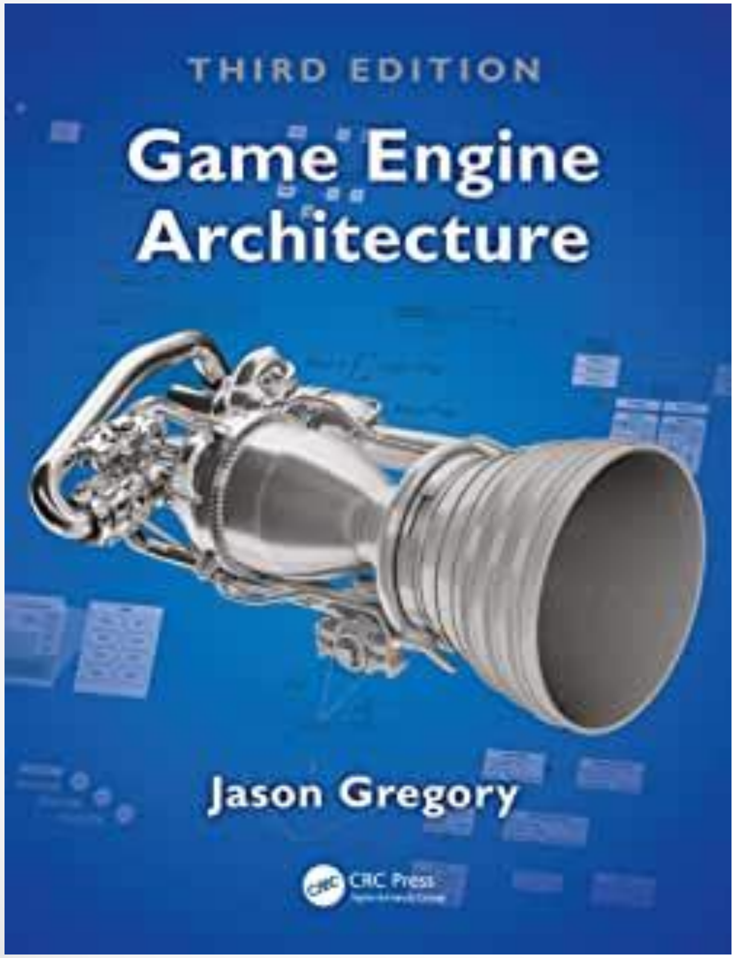
Hooman Salamat (Lectures & Labs)

- Hooman.Salamat@georgebrown.ca
- Discord: Hooman#2526

Assessment

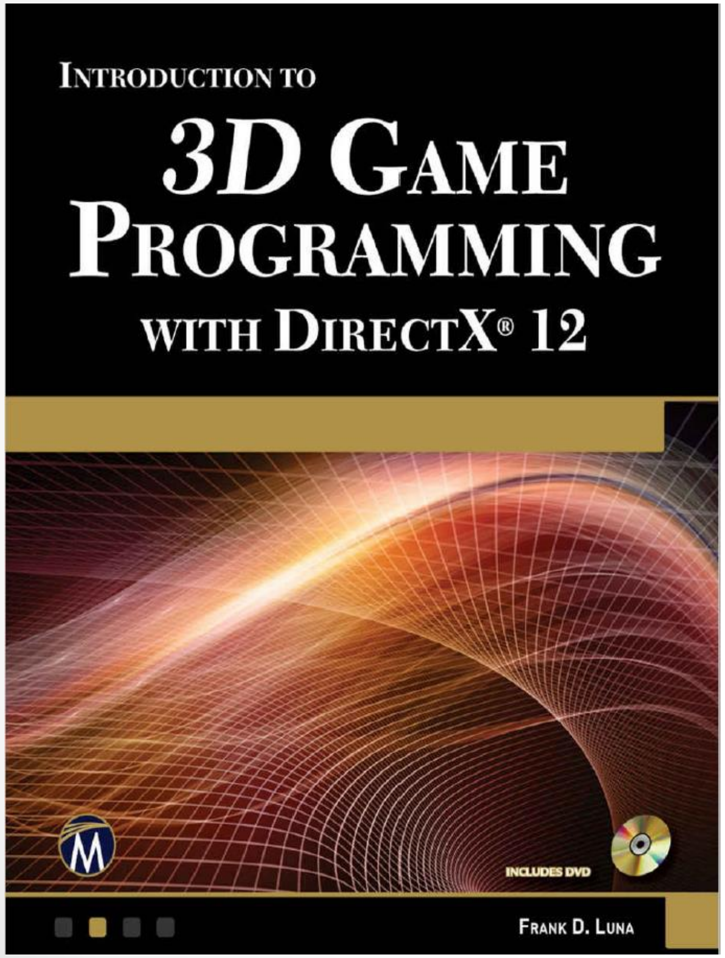
- 2x Assignments 20%
-
- 1x Final Exam 30%
-
- 1x Final Project 50%
-

Textbook 1



- Game Engine Architecture, Third Edition
- By: Jason Gregory
- ISBN-13: 978-1-1380-3545-4
- Publisher: CRC Press

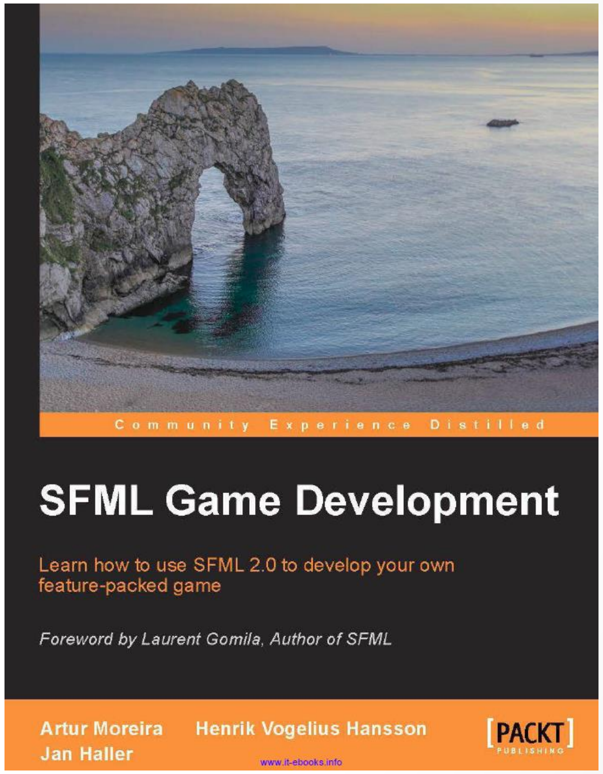
Textbook2



- Introduction to 3D Game Programming With DirectX12
- By: Frank D. Luna
- ISBN: 9781942270065
- Publisher: Mercury

Textbook3

- SFML Game Development
- Google it
- https://www.packtpub.com/game-development/sfml-game-development


Community Experience Distilled
SFML Game Development
Learn how to use SFML 2.0 to develop your own feature-packed game
Foreword by Laurent Gomila, Author of SFML
Artur Moreira
Henrik Vogelius Hansson
Jan Haller
[PACKT] PUBLISHING
www.it-ebooks.info

Course Materials

- https://github.com/hsalamat/GameEngineDevelopment2
- Our repository
  - https://github.com/hsalamat/GameEngineDevelopment2.git
  - Open the command line (cmd)
  - Cd "to a location that you want to clone"
  - Git clone https://github.com/hsalamat/GameEngineDevelopment2.git
  - Cd GameEngineDevelopment2
  - Git status

DoxyGen

- DoxyWizard
- DoxyGen
- All assignments must be documented by doxygen!
- https://www.doxygen.nl/index.html
  - **Select JAVADOC_AUTOBRIEF (in Project panel)**
  - **Select EXTRACT_ALL (in Build panel)**
  - **DeSelect SHOW_USED_FILES(in Build panel)**
  - **Select Recursive(in Input panel)**
  - **Select sourceBrowser(in SourceBrowser panel)**
  - **Select DISABLE_INDEX (in HTML panel)**
  - **Select GENERATE_TREEVIEW (in HTML panel)**
  - **Create a README.dox**

Tags

```
/** @file passbyDemo.cpp
 *  @brief difference in passing in a variable
 *  by ref/value/pointer to a function
 *  @author Hooman Salamat
 *  @bug No known bugs.
 */
```

Important: file name must match the actual file name!

Example

```
/** @file passbyDemo.cpp
 *  @brief difference in passing in a variable by
 *  ref/value/pointer to a function
 *  @author Hooman Salamat
 *  @bug No known bugs.
 */

#include <cstdio>
#include <string>
#include <stdio.h>
#include<iostream>
using  namespace std;
void passByVal(int val); //pass in a copy of the
variable
void passByRef(int& ref); //pass in the actual variable
void passByPtr(int* ptr); //pass in the address of the
variable

int main()
{
string bye;
int val = 5;
passByVal(val);
passByRef(val);
passByPtr(&val);

getline(cin, bye);
return 0;
}
```

```
void passByVal(int val)
{
val = 10;
printf("val = %i \n", val);
}

void passByRef(int& ref)
{
ref = 20;
printf("ref = %i \n", ref);
}

void passByPtr(int* ptr)
{
printf("*ptr = %i \n", *ptr);
*ptr = 30;
printf("*ptr = %i \n", *ptr);
}
```

README.dox example

```
/**

@mainpage assignment1

@author Hooman Salamat

@attention use WASD to move the paddle

@note this app doesn't work with mouse or arrows

this is my assignment 1. In this assignment, I am implementing ....

*/
```

Create your own repository for assignments and invite me and your partners as collaborators

1. Ask for the username of the person you're inviting as a collaborator. ...
2. On GitHub, navigate to the main page of the repository.
3. Under your repository name, click Settings.
4. In the left sidebar, click Manage access.
5. Click Invite a collaborator.

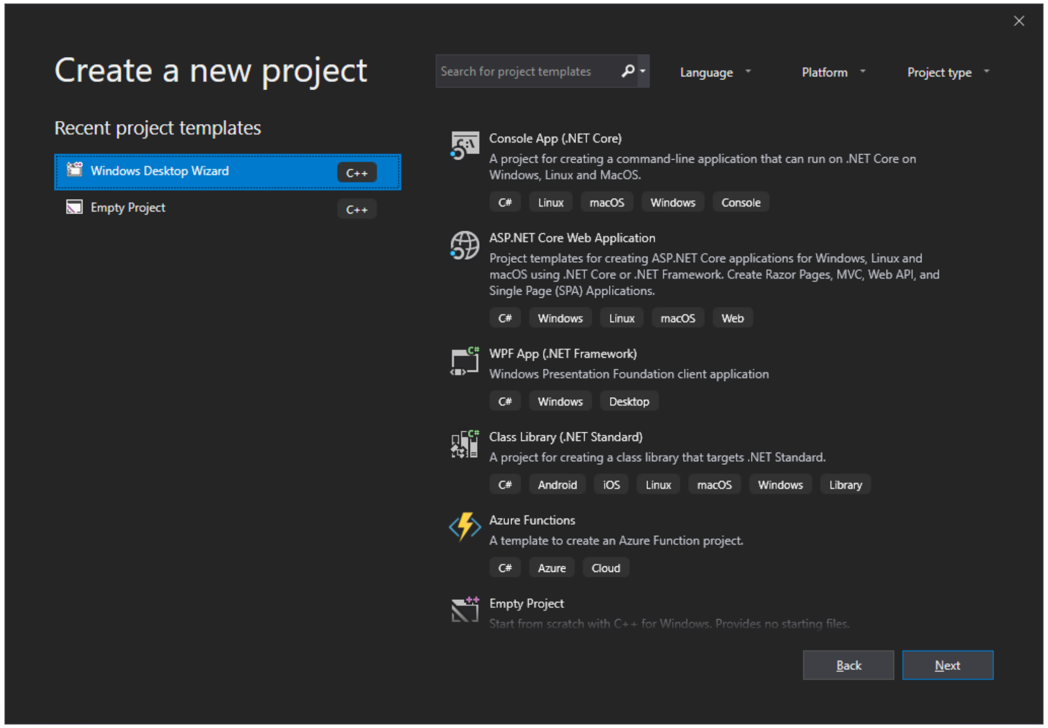
DirectX - Refresher

How to create a DirectX project

Demo

- 1. Download https://github.com/hsalamat/DirectX and unzip
- 2. Create an empty Window Desktop project under DirectX-master/Project1
- 3. Your project folder should be at the same level as “Common” and “Texture folder” under “DirectX-master”

Create a new project in Visual Studio 2019/2022


Create a new project

Search for project templates

Language

Platform

Project type

Recent project templates

Windows Desktop Wizard C++

Empty Project C++

Console App (.NET Core)
A project for creating a command-line application that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Console

ASP.NET Core Web Application
Project templates for creating ASP.NET Core applications for Windows, Linux and macOS using .NET Core or .NET Framework. Create Razor Pages, MVC, Web API, and Single Page (SPA) Applications.
C# Windows Linux macOS Web

WPF App (.NET Framework)
Windows Presentation Foundation client application
C# Windows Desktop

Class Library (.NET Standard)
A project for creating a class library that targets .NET Standard.
C# Android iOS Linux macOS Windows Library

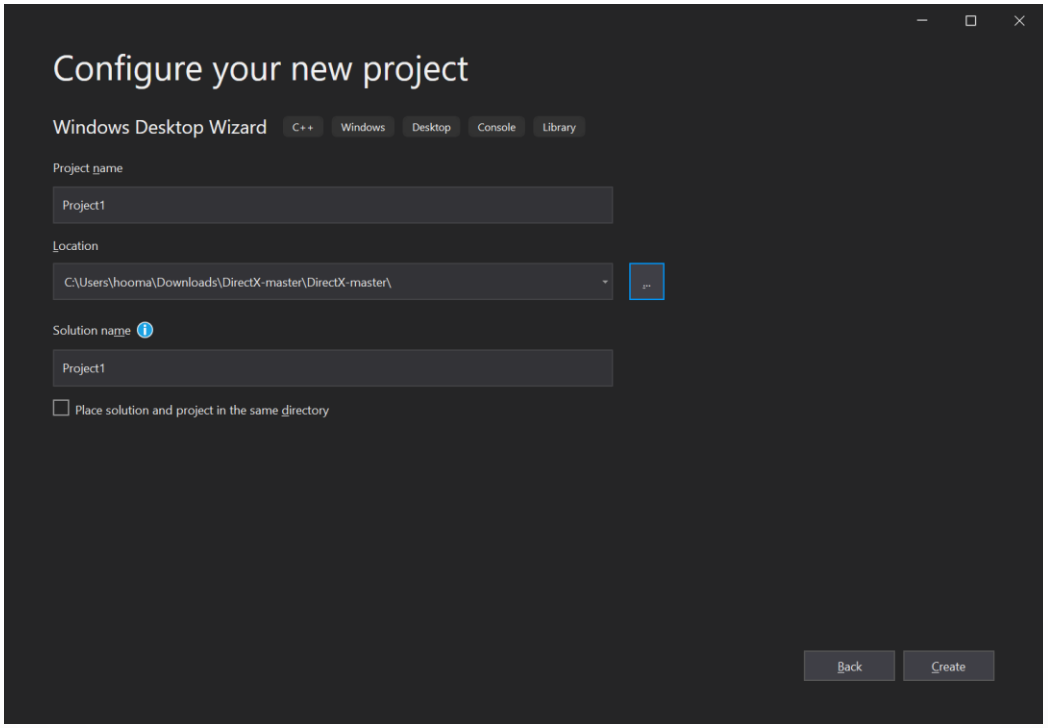
Azure Functions
A template to create an Azure Function project.
C# Azure Cloud

Empty Project
Start from scratch with C++ for Windows. Provides no starting files.

Back

Next

Configure your new project


Configure your new project

Windows Desktop Wizard

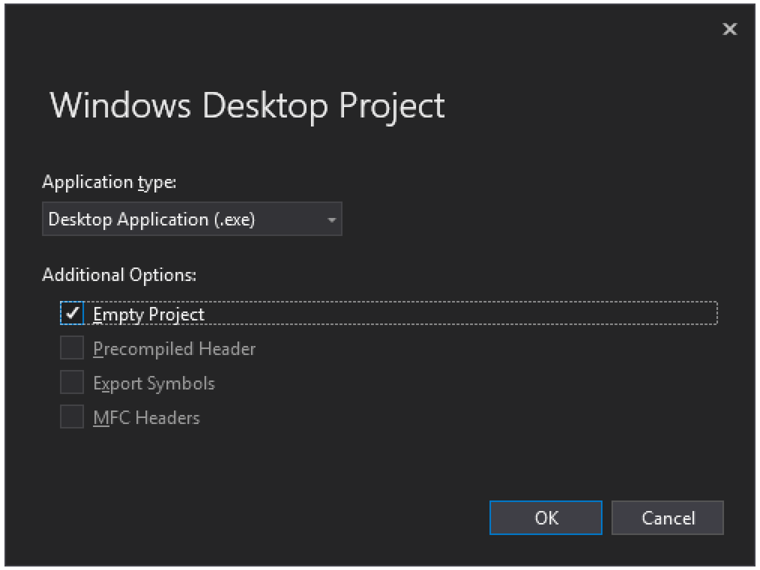
Project name: Project1

Location: C:\Users\hooma\Downloads\DirectX-master\DirectX-master\

Solution name: Project1

☐ Place solution and project in the same directory

Create an Empty Windows Desktop Project


Windows Desktop Project

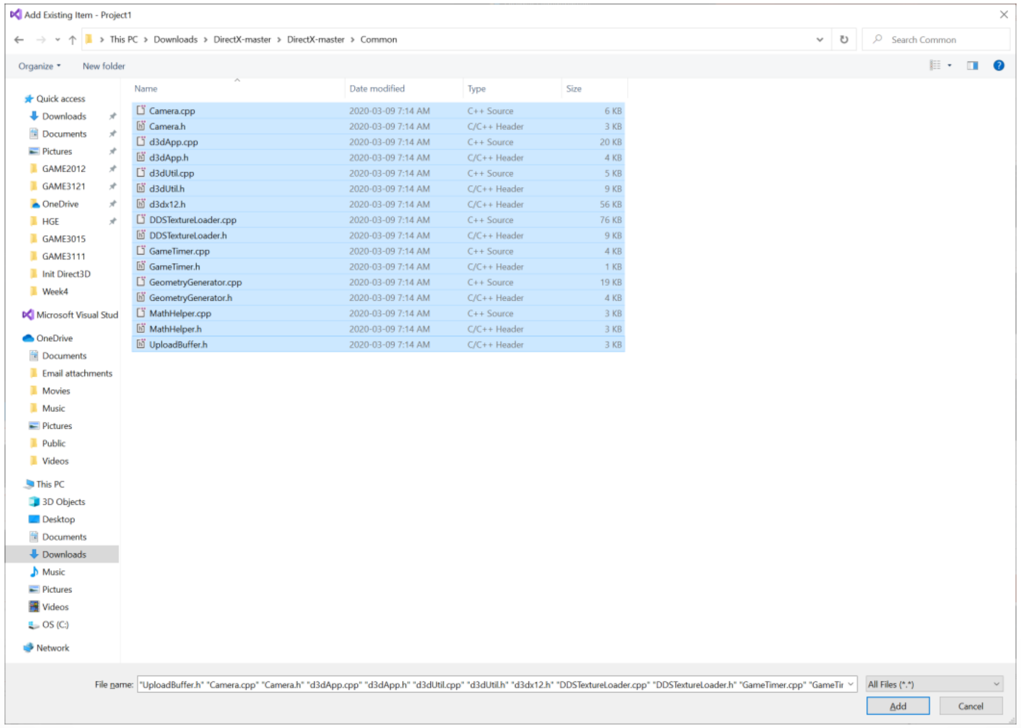
Application type:

Desktop Application (.exe)

Additional Options:

- [✓] Empty Project
- [ ] Precompiled Header
- [ ] Export Symbols
- [ ] MFC Headers

Add Existing Item

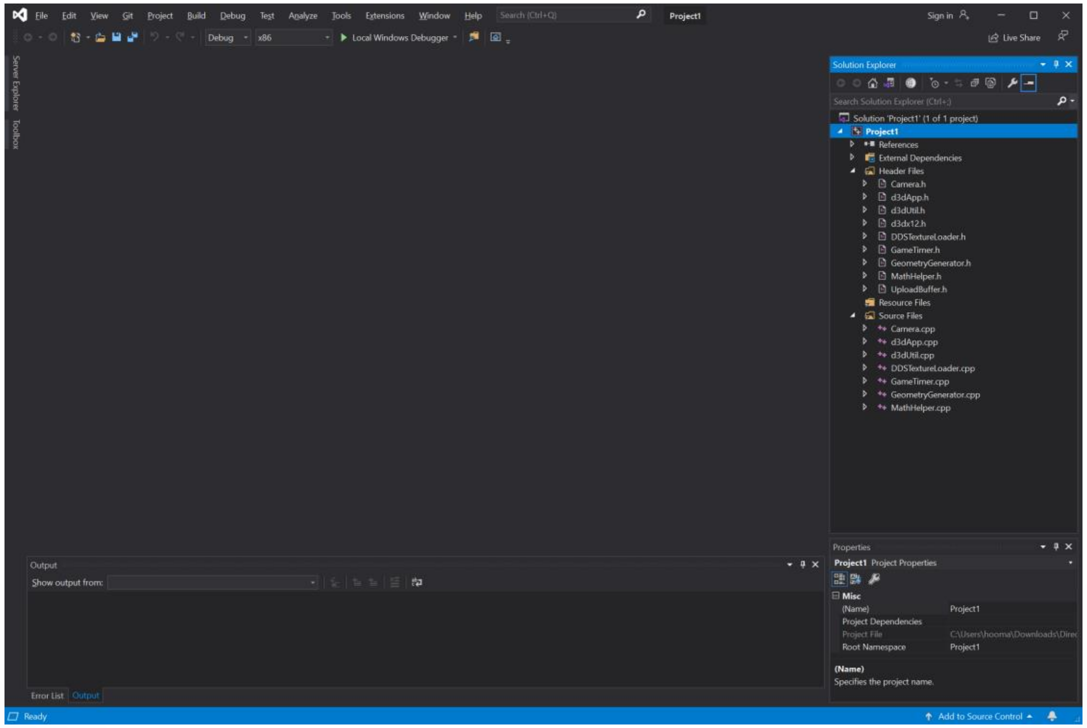


Name	Date modified	Type	Size
Camera.cpp	2020-03-09 7:14 AM	C++ Source	6 KB
Camera.h	2020-03-09 7:14 AM	C/C++ Header	3 KB
d3dApp.cpp	2020-03-09 7:14 AM	C++ Source	20 KB
d3dApp.h	2020-03-09 7:14 AM	C/C++ Header	4 KB
d3dUtil.cpp	2020-03-09 7:14 AM	C++ Source	5 KB
d3dUtil.h	2020-03-09 7:14 AM	C/C++ Header	9 KB
d3dx12.h	2020-03-09 7:14 AM	C/C++ Header	56 KB
DDSTextureLoader.cpp	2020-03-09 7:14 AM	C++ Source	76 KB
DDSTextureLoader.h	2020-03-09 7:14 AM	C/C++ Header	9 KB
GameTimer.cpp	2020-03-09 7:14 AM	C++ Source	4 KB
GameTimer.h	2020-03-09 7:14 AM	C/C++ Header	1 KB
GeometryGenerator.cpp	2020-03-09 7:14 AM	C++ Source	19 KB
GeometryGenerator.h	2020-03-09 7:14 AM	C/C++ Header	4 KB
MathHelper.cpp	2020-03-09 7:14 AM	C++ Source	3 KB
MathHelper.h	2020-03-09 7:14 AM	C/C++ Header	3 KB
UploadBuffer.h	2020-03-09 7:14 AM	C/C++ Header	3 KB

File name: "UploadBuffer.h" "Camera.cpp" "Camera.h" "d3dApp.cpp" "d3dApp.h" "d3dUtil.cpp" "d3dUtil.h" "d3dx12.h" "DDSTextureLoader.cpp" "DDSTextureLoader.h" "GameTimer.cpp" "GameTir

All Files (*.*)

Project1



Debug | x86

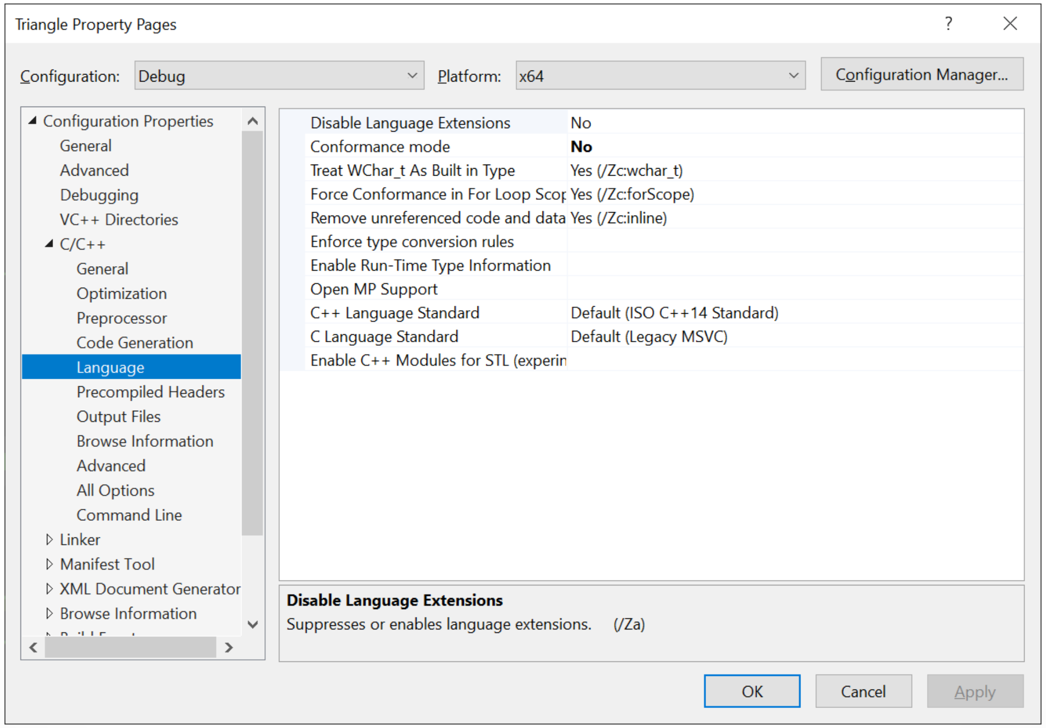
Project1 Project Properties

Misc

(Name)	Project1
Project Dependencies	
Project File	C:\Users\hooma\Downloads\Direc
Root Namespace	Project1

(Name)
Specifies the project name.

Change the conformance mode to "No"

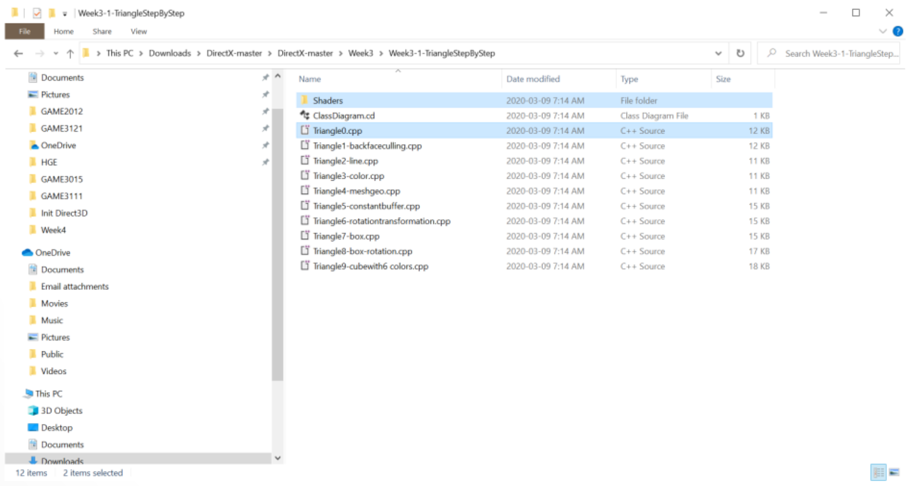

Triangle Property Pages

Disable Language Extensions	No
Conformance mode	**No**
Treat WChar_t As Built in Type	Yes (/Zc:wchar_t)
Force Conformance in For Loop Scop	Yes (/Zc:forScope)
Remove unreferenced code and data	Yes (/Zc:inline)
Enforce type conversion rules	
Enable Run-Time Type Information	
Open MP Support	
C++ Language Standard	Default (ISO C++14 Standard)
C Language Standard	Default (Legacy MSVC)
Enable C++ Modules for STL (experin	

Disable Language Extensions
Suppresses or enables language extensions. (/Za)

Copy the following files and place it under your project1

- C:\Users\(USERNAME)\Downloads\DirectX-master\DirectX-master\Week3\Week3-1-TriangleStepByStep

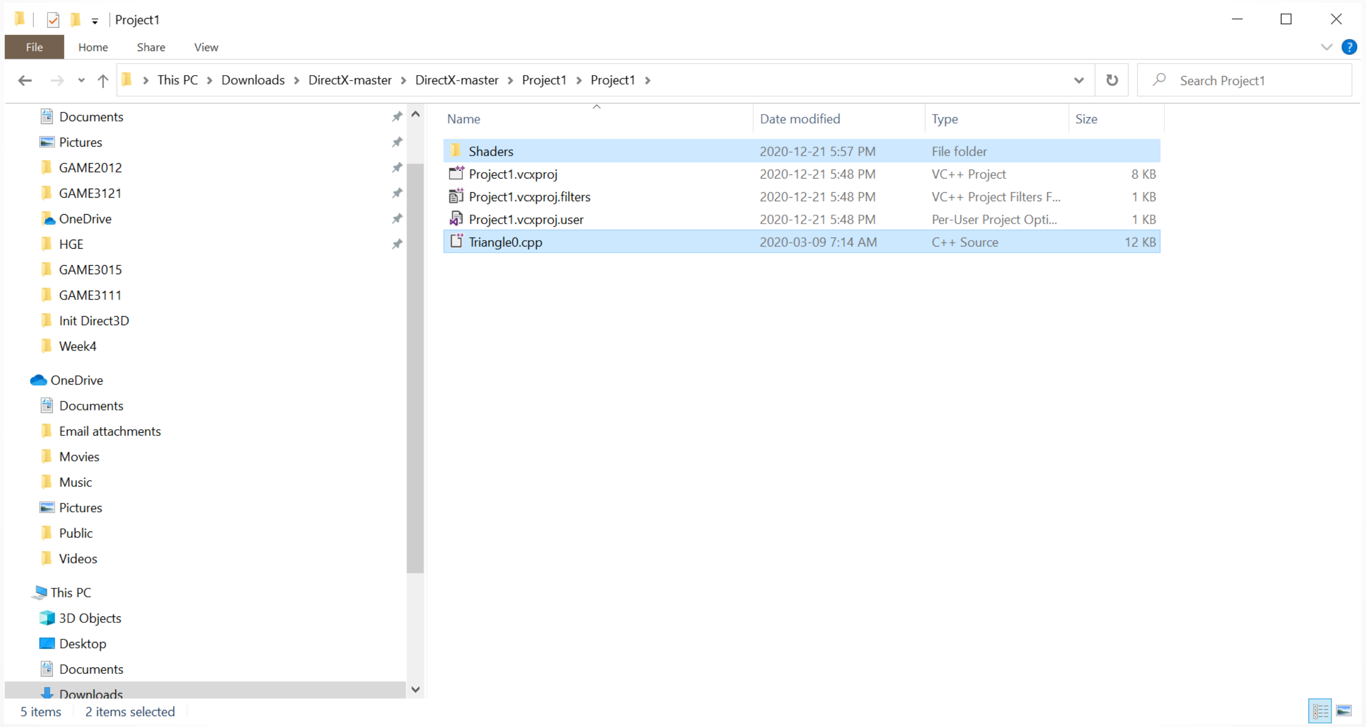


This PC > Downloads > DirectX-master > DirectX-master > Week3 > Week3-1-TriangleStepByStep

Name	Date modified	Type	Size
Shaders	2020-03-09 7:14 AM	File folder	
ClassDiagram.cd	2020-03-09 7:14 AM	Class Diagram File	1 KB
Triangle0.cpp	2020-03-09 7:14 AM	C++ Source	12 KB
Triangle1-backfaceculling.cpp	2020-03-09 7:14 AM	C++ Source	12 KB
Triangle2-line.cpp	2020-03-09 7:14 AM	C++ Source	11 KB
Triangle3-color.cpp	2020-03-09 7:14 AM	C++ Source	11 KB
Triangle4-meshgeo.cpp	2020-03-09 7:14 AM	C++ Source	11 KB
Triangle5-constantbuffer.cpp	2020-03-09 7:14 AM	C++ Source	15 KB
Triangle6-rotationtransformation.cpp	2020-03-09 7:14 AM	C++ Source	15 KB
Triangle7-box.cpp	2020-03-09 7:14 AM	C++ Source	15 KB
Triangle8-box-rotation.cpp	2020-03-09 7:14 AM	C++ Source	17 KB
Triangle9-cubewith6 colors.cpp	2020-03-09 7:14 AM	C++ Source	18 KB

12 items 2 items selected

Paste it here

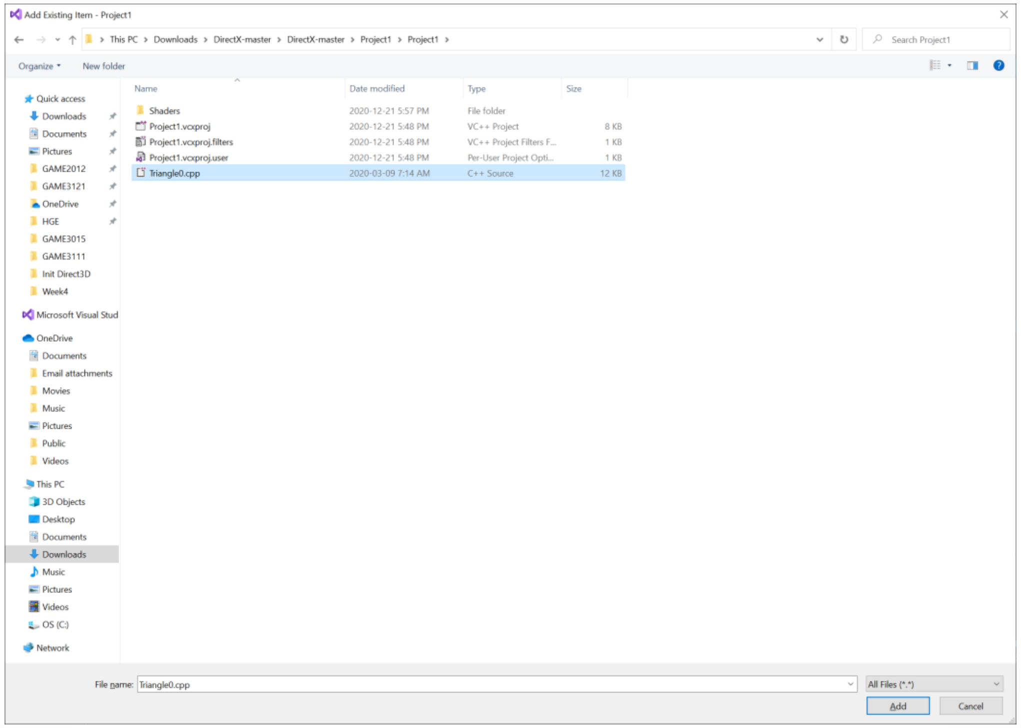

Project1

This PC > Downloads > DirectX-master > DirectX-master > Project1 > Project1

Name	Date modified	Type	Size
Shaders	2020-12-21 5:57 PM	File folder	
Project1.vcxproj	2020-12-21 5:48 PM	VC++ Project	8 KB
Project1.vcxproj.filters	2020-12-21 5:48 PM	VC++ Project Filters F...	1 KB
Project1.vcxproj.user	2020-12-21 5:48 PM	Per-User Project Opti...	1 KB
Triangle0.cpp	2020-03-09 7:14 AM	C++ Source	12 KB

5 items   2 items selected

Add Triangle0.cpp to your project1

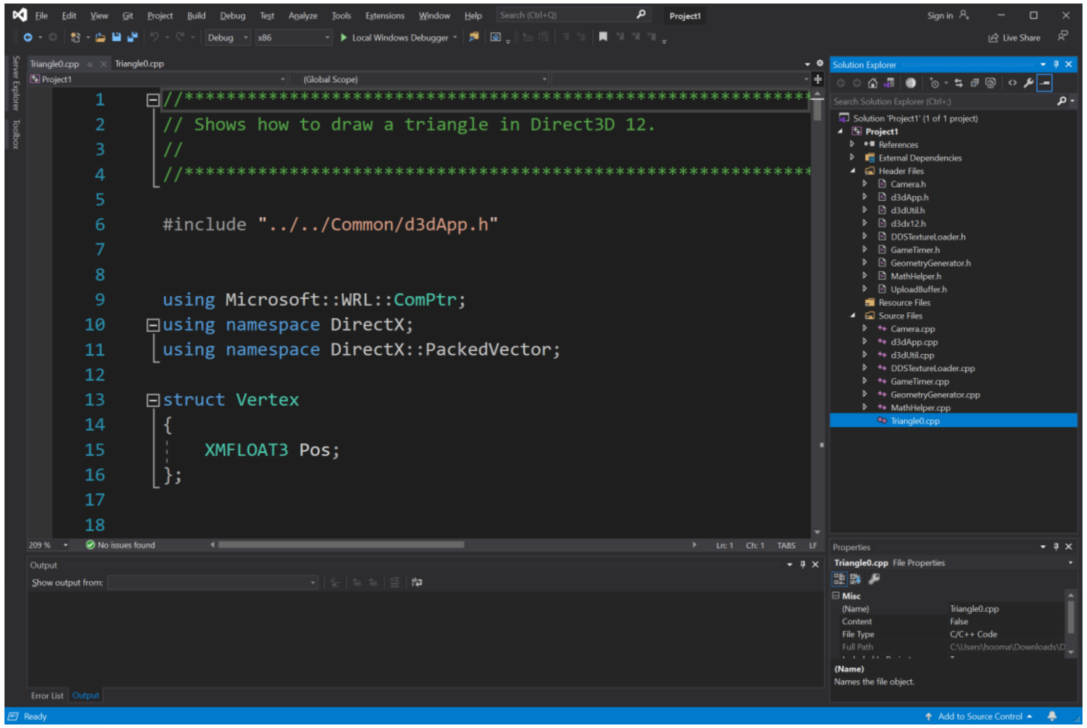


Name	Date modified	Type	Size
Shaders	2020-12-21 5:57 PM	File folder	
Project1.vcxproj	2020-12-21 5:48 PM	VC++ Project	8 KB
Project1.vcxproj.filters	2020-12-21 5:48 PM	VC++ Project Filters F...	1 KB
Project1.vcxproj.user	2020-12-21 5:48 PM	Per-User Project Opti...	1 KB
Triangle0.cpp	2020-03-09 7:14 AM	C++ Source	12 KB

File name: Triangle0.cpp

All Files (*.*)

Change to 64 bit and Compile the code



```
//***************************************************************
// Shows how to draw a triangle in Direct3D 12.
//
//***************************************************************

#include "../../Common/d3dApp.h"


using Microsoft::WRL::ComPtr;
using namespace DirectX;
using namespace DirectX::PackedVector;

struct Vertex
{
    XMFLOAT3 Pos;
};

```

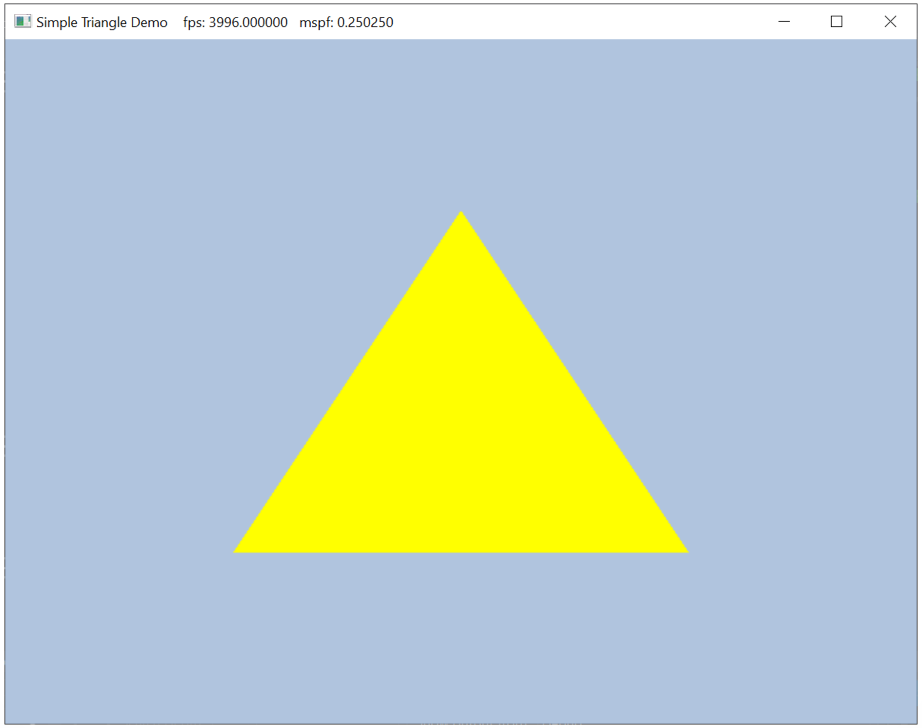
Solution 'Project1' (1 of 1 project)
- Project1
  - References
  - External Dependencies
  - Header Files
    - Camera.h
    - d3dApp.h
    - d3dUtil.h
    - d3dx12.h
    - DDSTextureLoader.h
    - GameTimer.h
    - GeometryGenerator.h
    - MathHelper.h
    - UploadBuffer.h
  - Resource Files
  - Source Files
    - Camera.cpp
    - d3dApp.cpp
    - d3dUtil.cpp
    - DDSTextureLoader.cpp
    - GameTimer.cpp
    - GeometryGenerator.cpp
    - MathHelper.cpp
    - Triangle0.cpp

Triangle0.cpp File Properties

Misc	
(Name)	Triangle0.cpp
Content	False
File Type	C/C++ Code
Full Path	C:\Users\hooma\Downloads\D

(Name)
Names the file object.

Simple Triangle



Objectives

- Be introduced to SFML or Simple and Fast Multimedia Library, which is a C++ framework
- Learn how to download and install SFML
- Explore an example and see the format of an SFML program
- Examine the Game class of an SFML program

SFML Tutorials

- Before we begin, here is a link to the main SFML tutorial site:
  - https://www.sfml-dev.org/tutorials/2.5/
  - https://www.sfml-dev.org/tutorials/2.5/start-vc.php

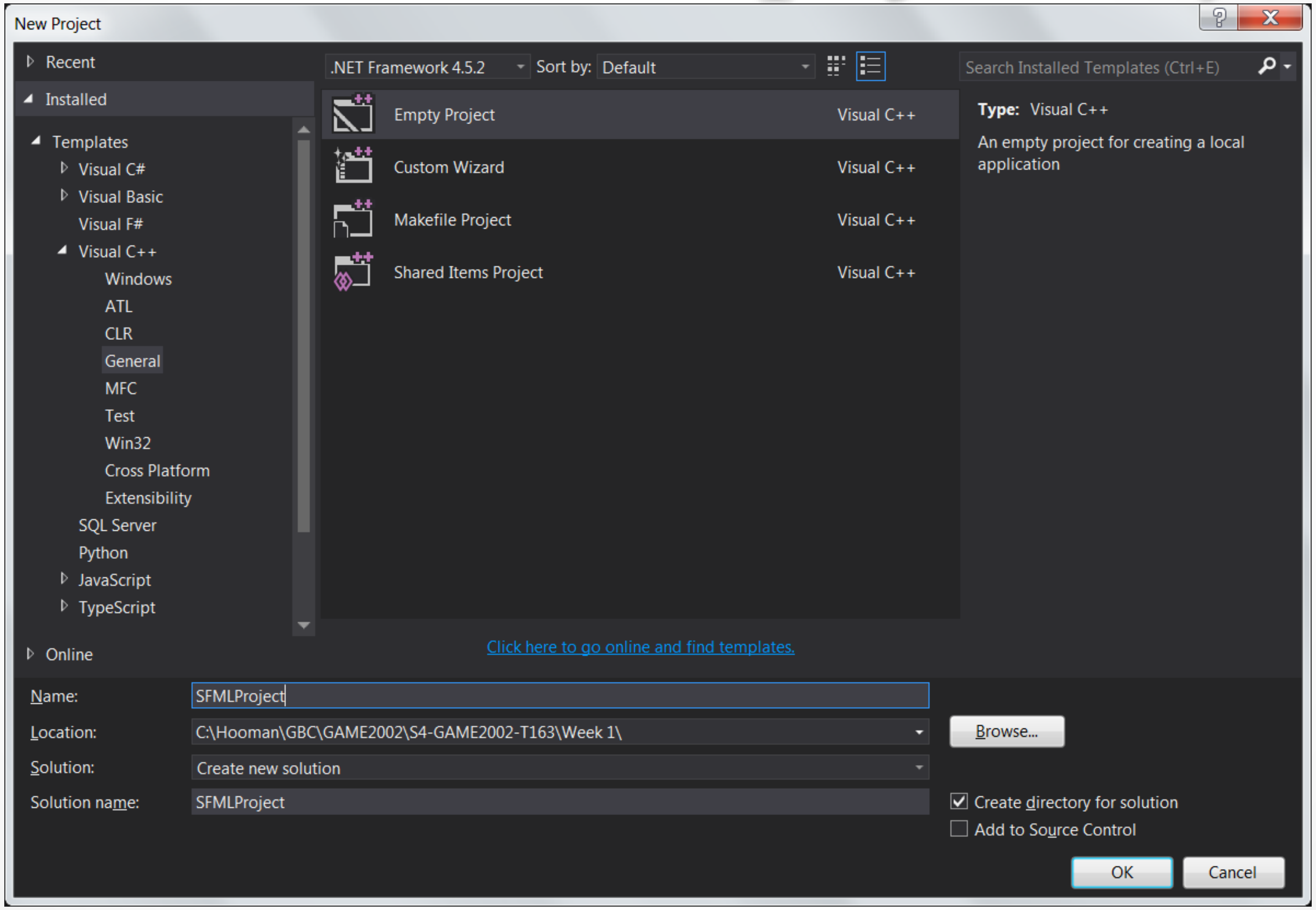
- Here you can also learn how to setup SFML for your version of Visual Studio – which we will go through in detail this week
- http://sfml-hooman.blogspot.ca/2017/12/setting-up-sfml-242-in-visual-studio.html

Setting up SFML with Visual Studio 2019/2022

Installing SFML

- First, you must download the SFML SDK from the download page.
- You must download the package that matches your version of Visual C++. Indeed, a library compiled with VC++ 10 (Visual Studio 2010) won't be compatible with VC++ 12 (Visual Studio 2013) for example. If there's no SFML package compiled for your version of Visual C++, you will have to build SFML yourself.

Create An Empty Project


New Project

- Recent
- Installed
  - Templates
    - Visual C#
    - Visual Basic
    - Visual F#
    - Visual C++
      - Windows
      - ATL
      - CLR
      - General
      - MFC
      - Test
      - Win32
      - Cross Platform
      - Extensibility
    - SQL Server
    - Python
    - JavaScript
    - TypeScript
- Online

.NET Framework 4.5.2 Sort by: Default

Search Installed Templates (Ctrl+E)

Empty Project	Visual C++
Custom Wizard	Visual C++
Makefile Project	Visual C++
Shared Items Project	Visual C++

Type: Visual C++

An empty project for creating a local application

Click here to go online and find templates.

Name: SFMLProject

Location: C:\Hooman\GBC\GAME2002\S4-GAME2002-T163\Week 1\

Solution: Create new solution

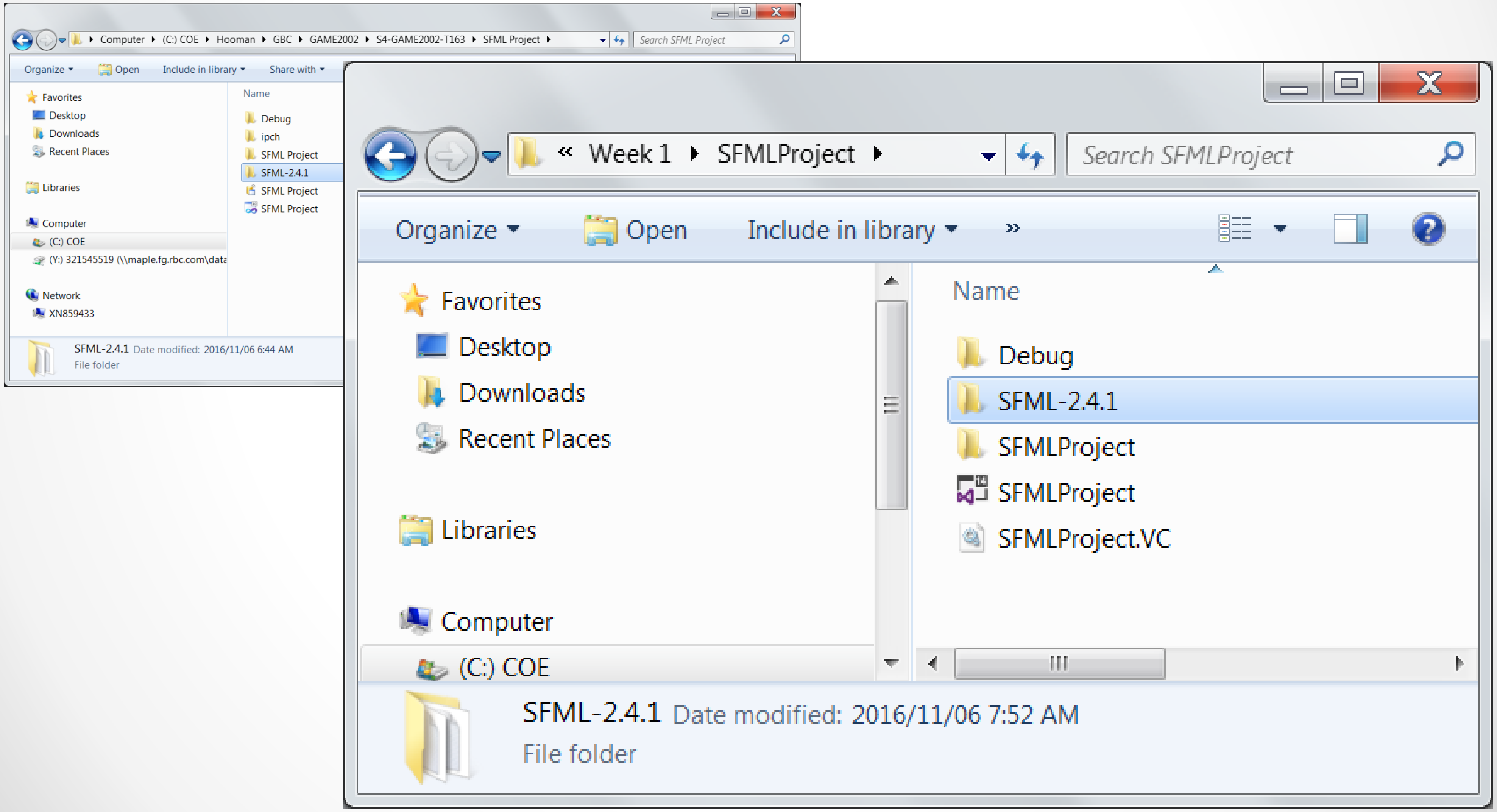
Solution name: SFMLProject

[✓] Create directory for solution

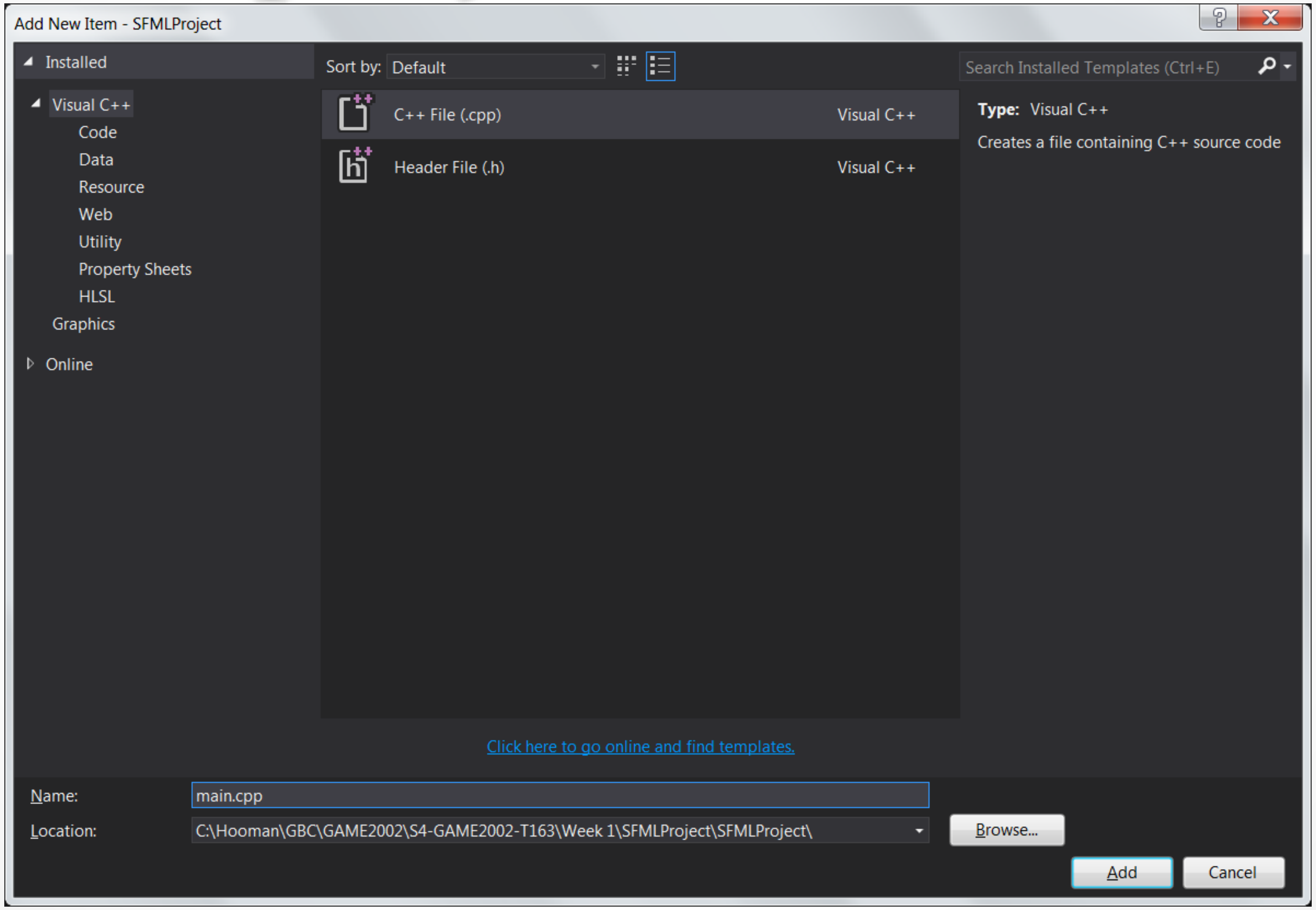
[ ] Add to Source Control

Browse... OK Cancel

Unzip and Copy SFML install folder under SFML Solution Directory



Add a C++ file to SFML project folder



Add New Item - SFMLProject

Sort by: Default

C++ File (.cpp) Visual C++

Header File (.h) Visual C++

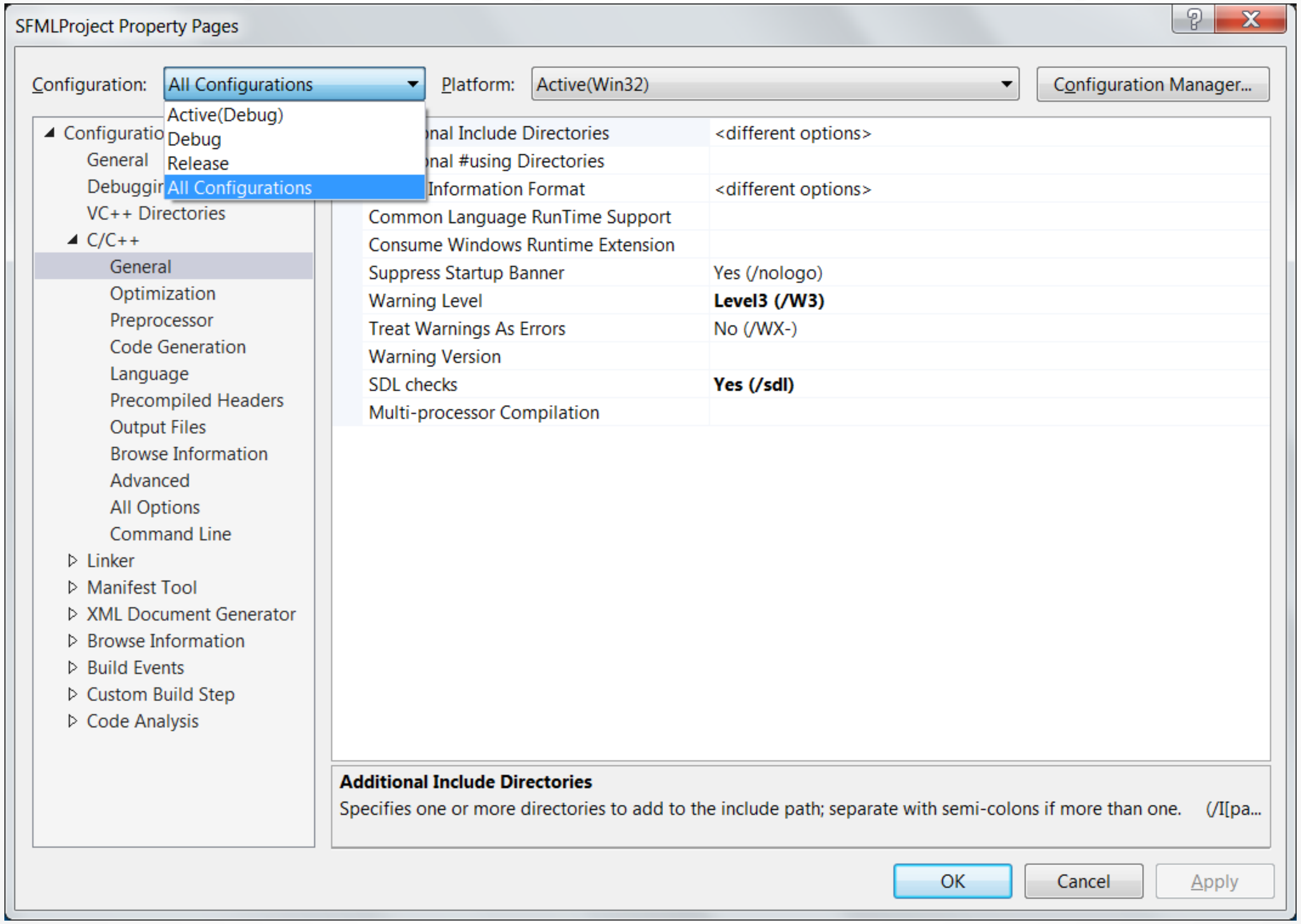
Type: Visual C++

Creates a file containing C++ source code

Name: main.cpp

Location: C:\Hooman\GBC\GAME2002\S4-GAME2002-T163\Week 1\SFMLProject\SFMLProject\

Go to Project Properties (Right Click on Project → Select Project) and Change the configuration to "All Configuration"


SFMLProject Property Pages

Configuration: All Configurations

Platform: Active(Win32)

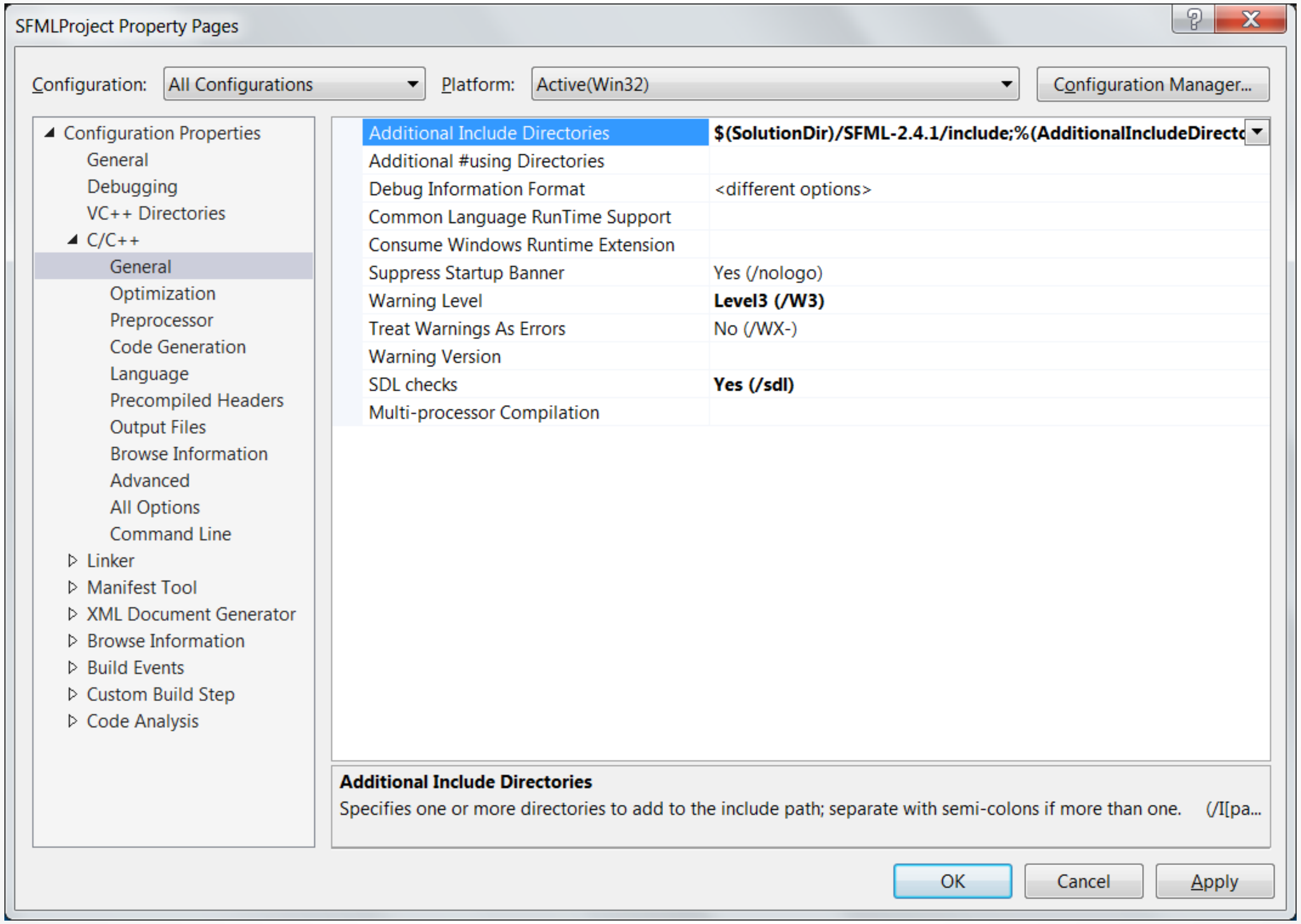
Configuration Manager...

nal Include Directories	<different options>
nal #using Directories	
Information Format	<different options>
Common Language RunTime Support	
Consume Windows Runtime Extension	
Suppress Startup Banner	Yes (/nologo)
Warning Level	**Level3 (/W3)**
Treat Warnings As Errors	No (/WX-)
Warning Version	
SDL checks	**Yes (/sdl)**
Multi-processor Compilation	

Additional Include Directories

Specifies one or more directories to add to the include path; separate with semi-colons if more than one. (/I[pa...

Add the SFML headers (*<sfml-install-path>/include*) to C/C++ » General » Additional Include Directories


SFMLProject Property Pages

Configuration: All Configurations   Platform: Active(Win32)   Configuration Manager...

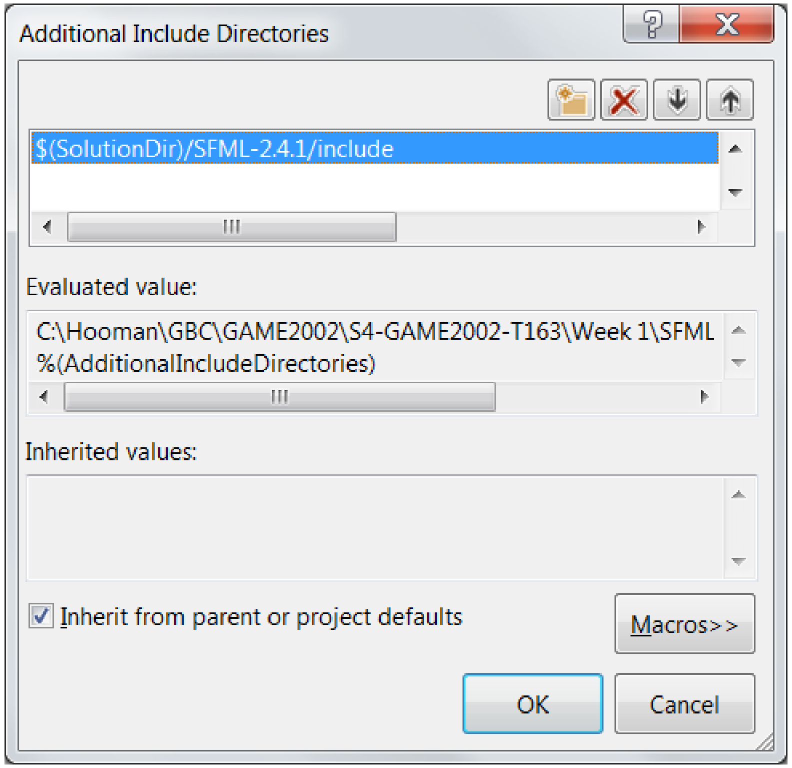
- Configuration Properties
  - General
  - Debugging
  - VC++ Directories
  - C/C++
    - General
    - Optimization
    - Preprocessor
    - Code Generation
    - Language
    - Precompiled Headers
    - Output Files
    - Browse Information
    - Advanced
    - All Options
    - Command Line
  - Linker
  - Manifest Tool
  - XML Document Generator
  - Browse Information
  - Build Events
  - Custom Build Step
  - Code Analysis

Additional Include Directories	**$(SolutionDir)/SFML-2.4.1/include;%(AdditionalIncludeDirecto**
Additional #using Directories	
Debug Information Format	<different options>
Common Language RunTime Support	
Consume Windows Runtime Extension	
Suppress Startup Banner	Yes (/nologo)
Warning Level	**Level3 (/W3)**
Treat Warnings As Errors	No (/WX-)
Warning Version	
SDL checks	**Yes (/sdl)**
Multi-processor Compilation	

Additional Include Directories
Specifies one or more directories to add to the include path; separate with semi-colons if more than one. (/I[pa...

OK Cancel Apply

Additional Include Directories



Additional Include Directories

```
$(SolutionDir)/SFML-2.4.1/include
```

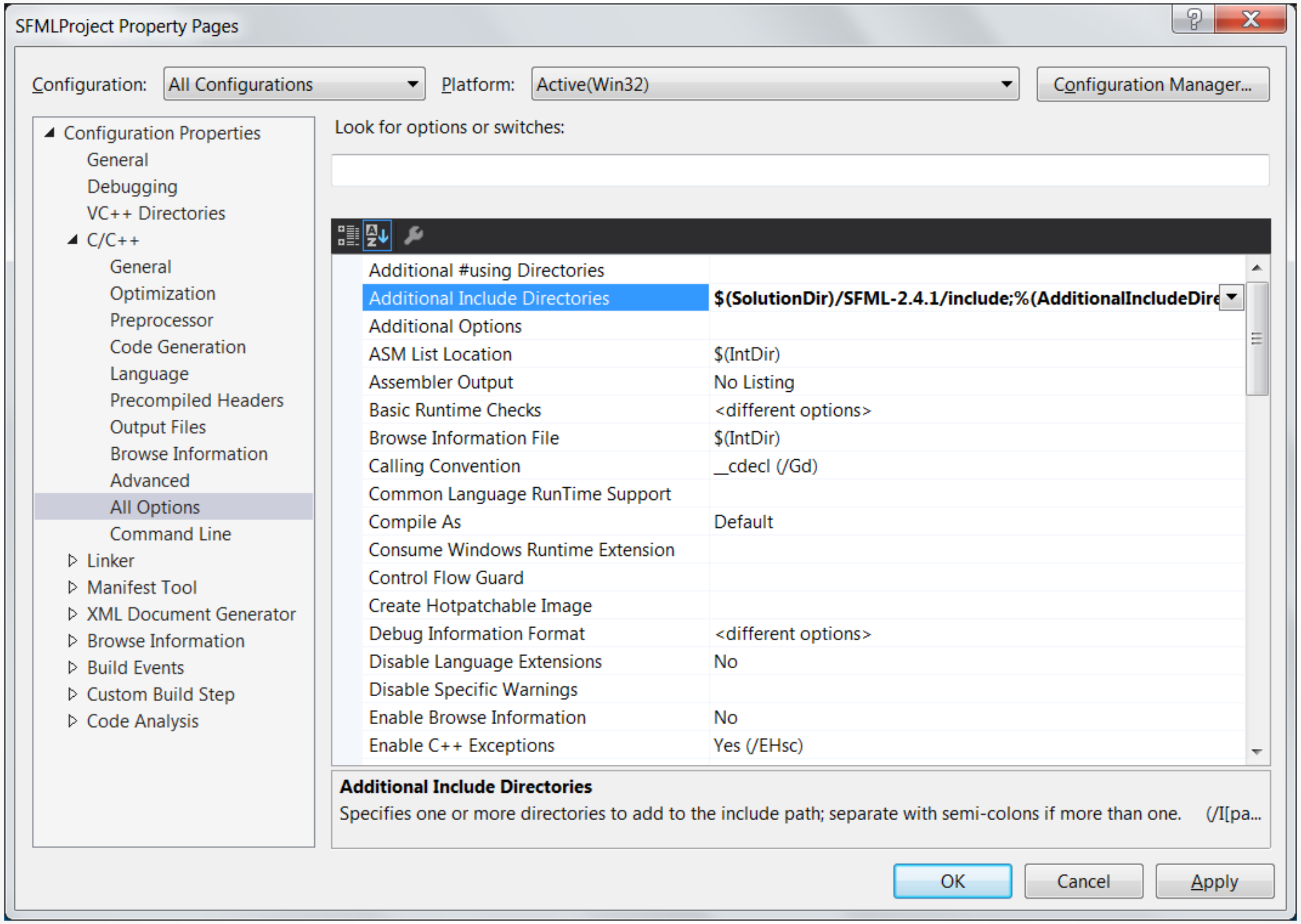
Evaluated value:

```
C:\Hooman\GBC\GAME2002\S4-GAME2002-T163\Week 1\SFML
%(AdditionalIncludeDirectories)
```

Inherited values:

☑ Inherit from parent or project defaults

Add the SFML headers (*<sfml-install-path>/include*) to C/C++ » All Options » Additional Include Directories


SFMLProject Property Pages

Configuration: All Configurations
Platform: Active(Win32)
Configuration Manager...

- Configuration Properties
  - General
  - Debugging
  - VC++ Directories
  - C/C++
    - General
    - Optimization
    - Preprocessor
    - Code Generation
    - Language
    - Precompiled Headers
    - Output Files
    - Browse Information
    - Advanced
    - All Options
    - Command Line
  - Linker
  - Manifest Tool
  - XML Document Generator
  - Browse Information
  - Build Events
  - Custom Build Step
  - Code Analysis

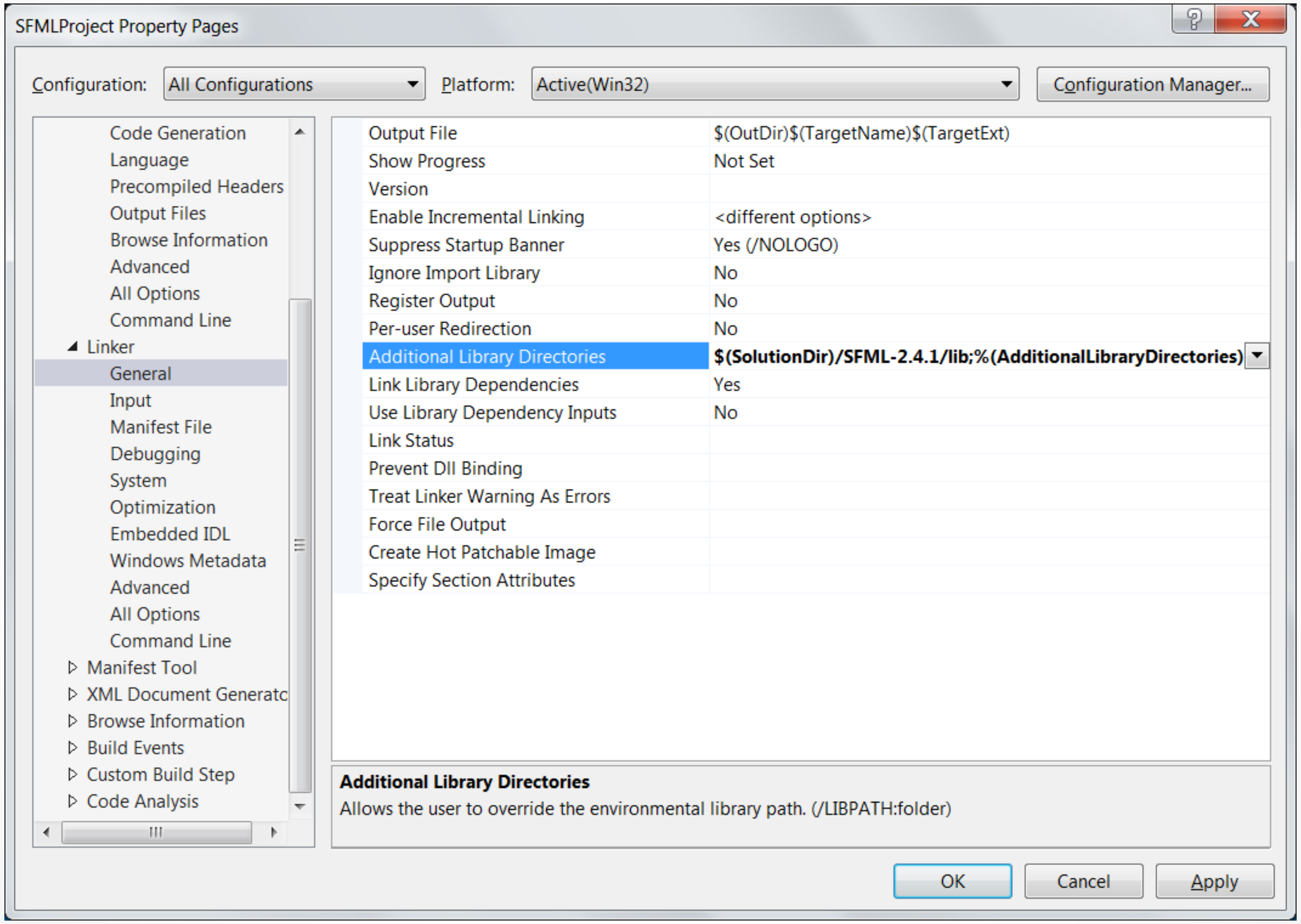
Look for options or switches:

Additional #using Directories	
Additional Include Directories	**$(SolutionDir)/SFML-2.4.1/include;%(AdditionalIncludeDire**
Additional Options	
ASM List Location	$(IntDir)
Assembler Output	No Listing
Basic Runtime Checks	<different options>
Browse Information File	$(IntDir)
Calling Convention	__cdecl (/Gd)
Common Language RunTime Support	
Compile As	Default
Consume Windows Runtime Extension	
Control Flow Guard	
Create Hotpatchable Image	
Debug Information Format	<different options>
Disable Language Extensions	No
Disable Specific Warnings	
Enable Browse Information	No
Enable C++ Exceptions	Yes (/EHsc)

Additional Include Directories
Specifies one or more directories to add to the include path; separate with semi-colons if more than one. (/I[pa...

OK Cancel Apply

Add the SFML libraries (*<sfml-install-path>/lib*) to Linker » General » Additional Library Directories

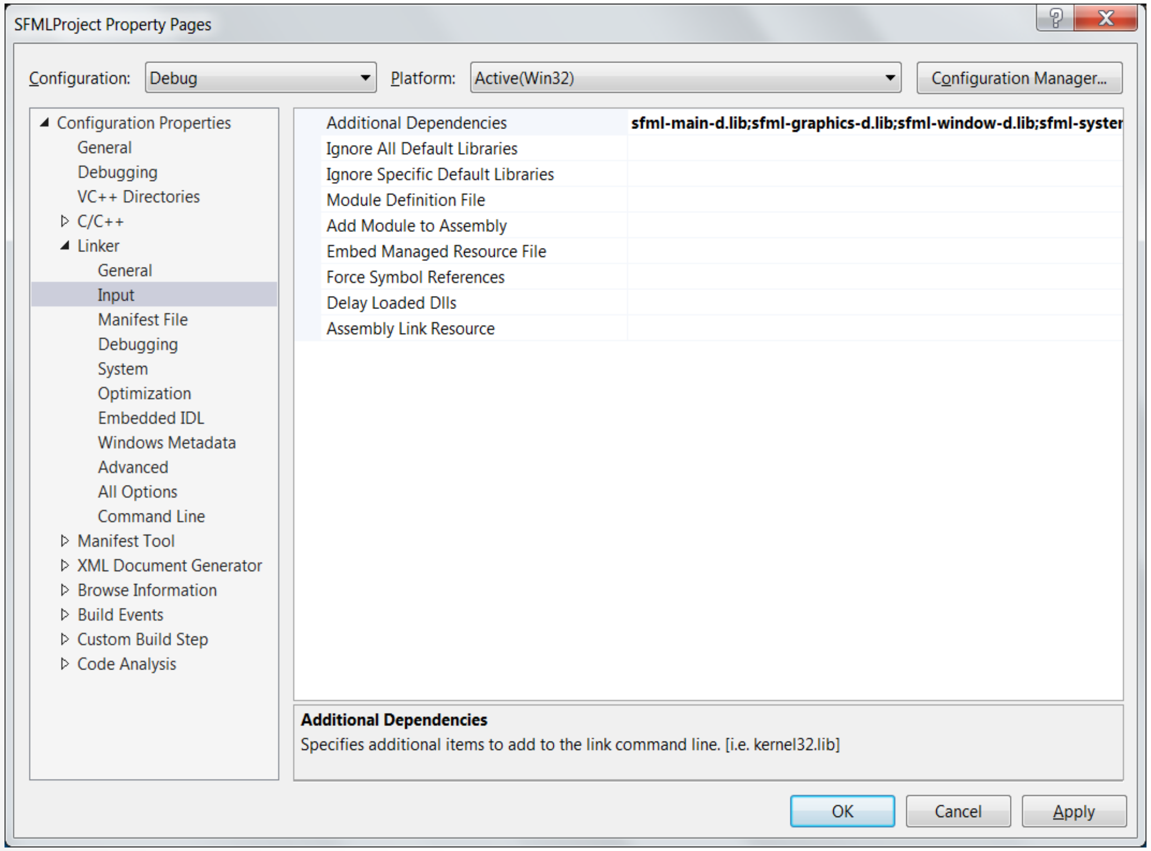

SFMLProject Property Pages

Configuration: All Configurations Platform: Active(Win32) Configuration Manager...

Output File	$(OutDir)$(TargetName)$(TargetExt)
Show Progress	Not Set
Version	
Enable Incremental Linking	<different options>
Suppress Startup Banner	Yes (/NOLOGO)
Ignore Import Library	No
Register Output	No
Per-user Redirection	No
Additional Library Directories	**$(SolutionDir)/SFML-2.4.1/lib;%(AdditionalLibraryDirectories)**
Link Library Dependencies	Yes
Use Library Dependency Inputs	No
Link Status	
Prevent Dll Binding	
Treat Linker Warning As Errors	
Force File Output	
Create Hot Patchable Image	
Specify Section Attributes	

Additional Library Directories
Allows the user to override the environmental library path. (/LIBPATH:folder)

Add all the SFML libraries "sfml-graphics-d.lib", "sfml-window-d.lib" and "sfml-system-d.lib"in the project's properties, in Linker » Input » Additional Dependencies


SFMLProject Property Pages

Configuration: Debug   Platform: Active(Win32)   Configuration Manager...

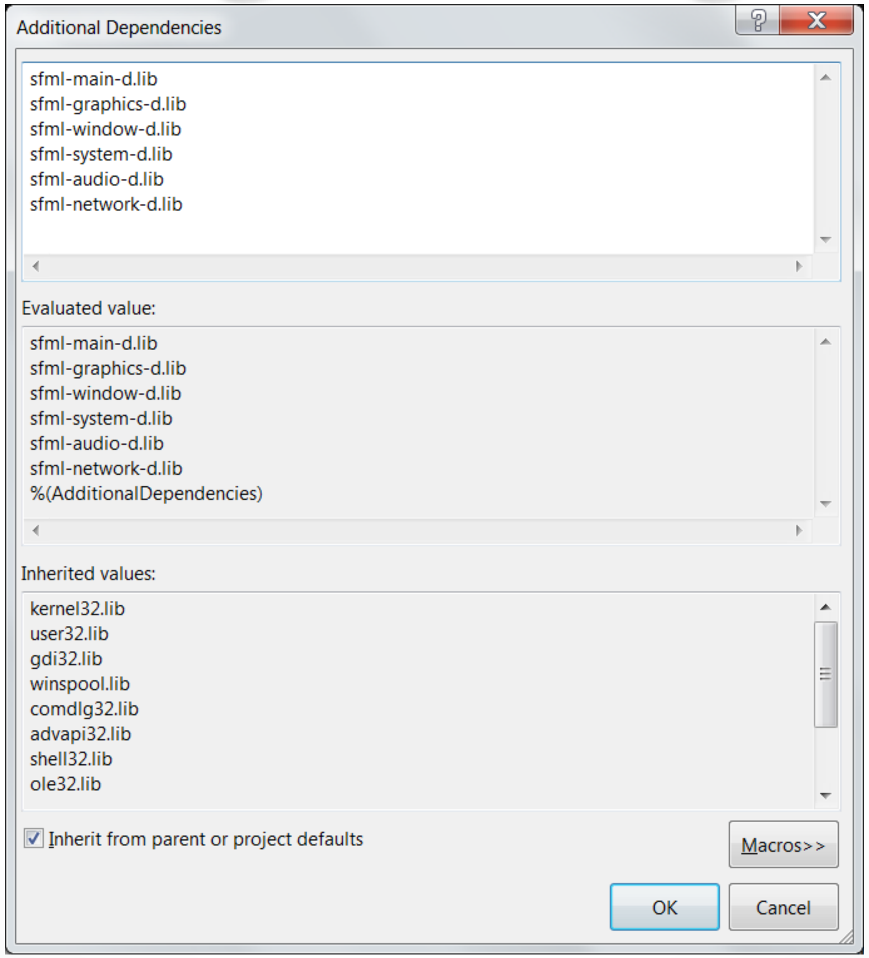
- Configuration Properties
  - General
  - Debugging
  - VC++ Directories
  - C/C++
  - Linker
    - General
    - Input
    - Manifest File
    - Debugging
    - System
    - Optimization
    - Embedded IDL
    - Windows Metadata
    - Advanced
    - All Options
    - Command Line
  - Manifest Tool
  - XML Document Generator
  - Browse Information
  - Build Events
  - Custom Build Step
  - Code Analysis

Additional Dependencies	**sfml-main-d.lib;sfml-graphics-d.lib;sfml-window-d.lib;sfml-syster**
Ignore All Default Libraries	
Ignore Specific Default Libraries	
Module Definition File	
Add Module to Assembly	
Embed Managed Resource File	
Force Symbol References	
Delay Loaded Dlls	
Assembly Link Resource	

Additional Dependencies
Specifies additional items to add to the link command line. [i.e. kernel32.lib]

- It is important to link to the libraries that match the configuration: "sfml-xxx-d.lib" for Debug, and "sfml-xxx.lib" for Release.

Additional Dependencies for Debug Configuration



```
sfml-main-d.lib
sfml-graphics-d.lib
sfml-window-d.lib
sfml-system-d.lib
sfml-audio-d.lib
sfml-network-d.lib
```

Evaluated value:

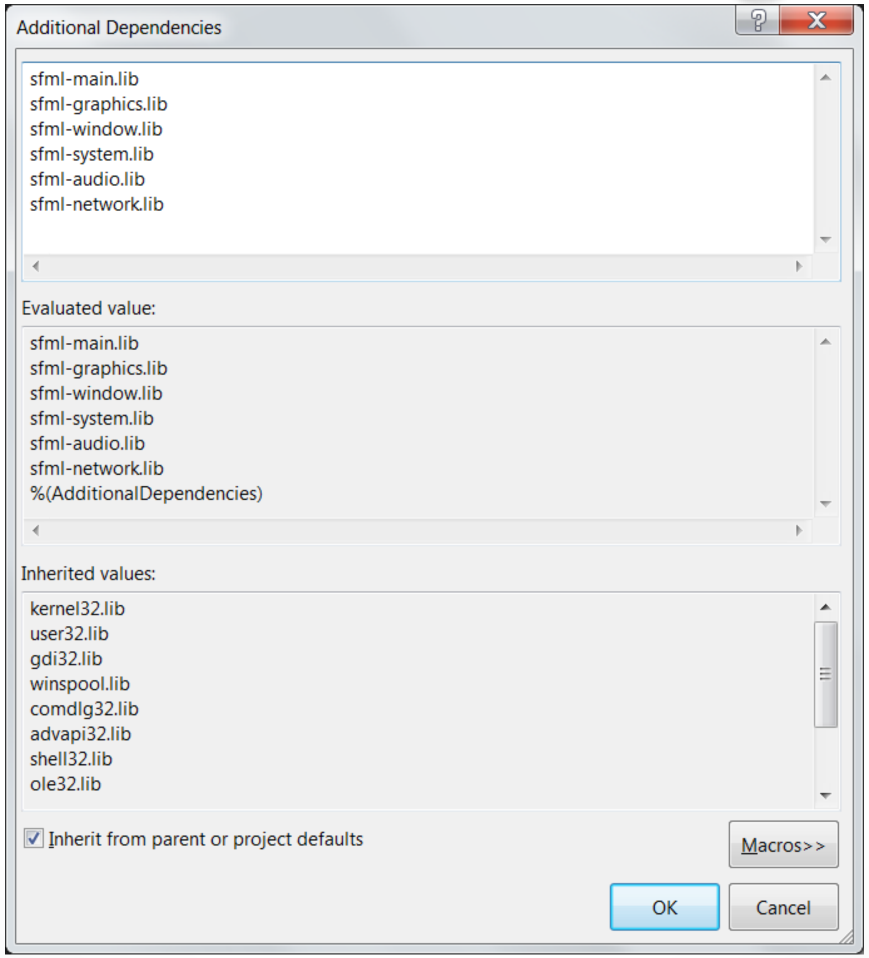
```
sfml-main-d.lib
sfml-graphics-d.lib
sfml-window-d.lib
sfml-system-d.lib
sfml-audio-d.lib
sfml-network-d.lib
%(AdditionalDependencies)
```

Inherited values:

```
kernel32.lib
user32.lib
gdi32.lib
winspool.lib
comdlg32.lib
advapi32.lib
shell32.lib
ole32.lib
```

☑ Inherit from parent or project defaults

Additional Dependencies for Release Configuration


Additional Dependencies

```
sfml-main.lib
sfml-graphics.lib
sfml-window.lib
sfml-system.lib
sfml-audio.lib
sfml-network.lib
```

Evaluated value:

```
sfml-main.lib
sfml-graphics.lib
sfml-window.lib
sfml-system.lib
sfml-audio.lib
sfml-network.lib
%(AdditionalDependencies)
```

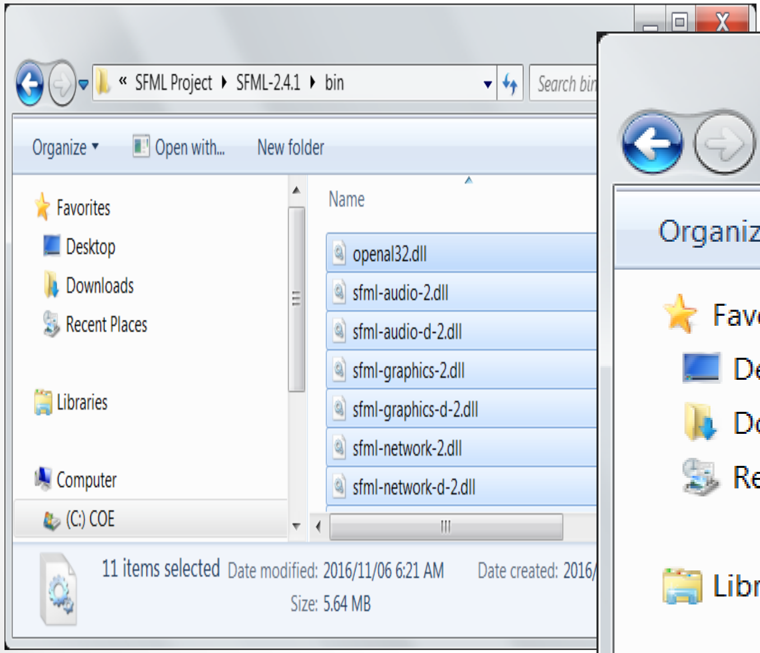
Inherited values:

```
kernel32.lib
user32.lib
gdi32.lib
winspool.lib
comdlg32.lib
advapi32.lib
shell32.lib
ole32.lib
```

[✓] Inherit from parent or project defaults

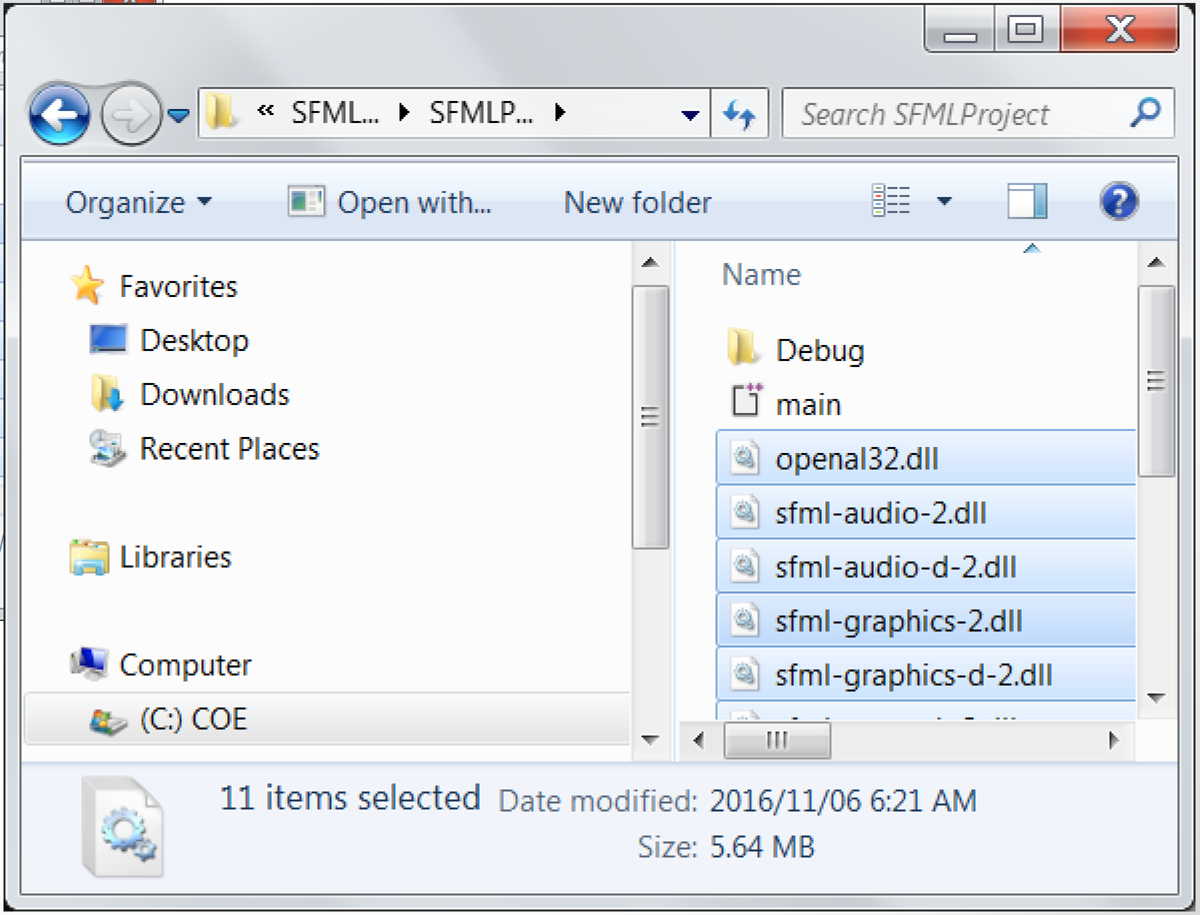
Copy all the dlls under (*<sfml-install-path>/bin)* to SFML project folder

where main.cpp is!



openal32.dll
sfml-audio-2.dll
sfml-audio-d-2.dll
sfml-graphics-2.dll
sfml-graphics-d-2.dll
sfml-network-2.dll
sfml-network-d-2.dll

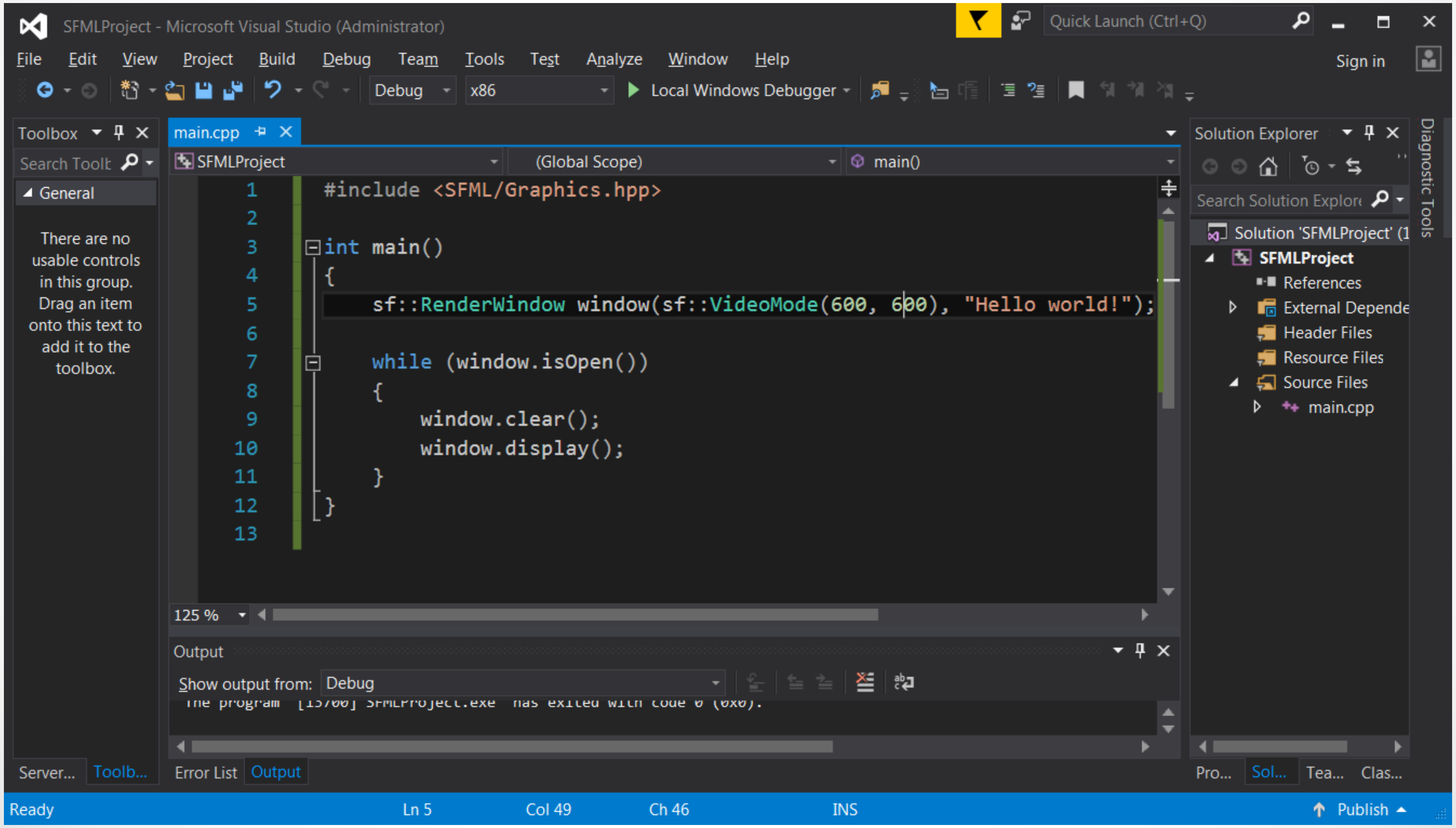
11 items selected Date modified: 2016/11/06 6:21 AM Date created: 2016/
Size: 5.64 MB



Debug
main
openal32.dll
sfml-audio-2.dll
sfml-audio-d-2.dll
sfml-graphics-2.dll
sfml-graphics-d-2.dll

11 items selected Date modified: 2016/11/06 6:21 AM
Size: 5.64 MB

Put the following code inside the main.cpp file and Run



```
#include <SFML/Graphics.hpp>

int main()
{
    sf::RenderWindow window(sf::VideoMode(600, 600), "Hello world!");

    while (window.isOpen())
    {
        window.clear();
        window.display();
    }
}
```

```
The program '[13700] SFMLProject.exe' has exited with code 0 (0x0).
```

Replace the main.cpp with the following code

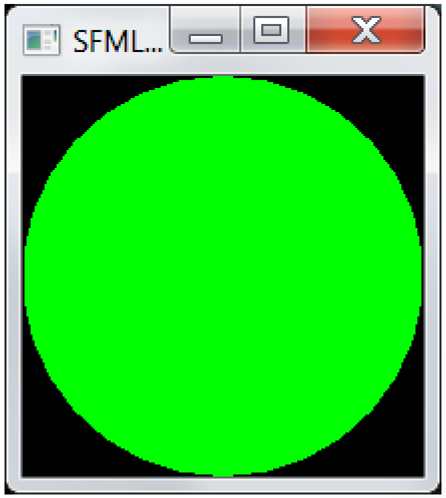
```
#include <SFML/Graphics.hpp>

int main()
{
    sf::RenderWindow window(sf::VideoMode(200, 200), "SFML works!");
    sf::CircleShape shape(100.f);
    shape.setFillColor(sf::Color::Green);

    while (window.isOpen())
    {
        sf::Event event;
        while (window.pollEvent(event))
        {
            if (event.type == sf::Event::Closed)
                window.close();
        }

        window.clear();
        window.draw(shape);
        window.display();
    }

    return 0;
}
```



Visual Studio Tips!

- If you choose to link the dynamic libraries, i.e.: *sfml-graphics.lib*, *sfml-window.lib* and *sfml-system.lib*, for **Release** or... *sfml-graphics-d.lib*, *sfml-window-d.lib* and *sfml-system-d.lib* for **Debug**...
  - **Do NOT add SFML_STATIC to the Preprocessor section**
  - **Remember to copy and paste the appropriate DLLs from bin to the same folder as your new .exe!**
- You can use main() instead of WinMain() even after choosing a Windows Application by including the appropriate sfml-main.lib or sfml-main-d.lib in the Linker->Input

Intro to SFML

- SFML is a library which adds multimedia content to your programs built in C++
- Five modules:
  - System
  - Window
  - Graphics
  - Audio
  - Network

- We'll start the course off by working with the first three for a few weeks

System Module

- The system is the core module
  - All other modules are built upon it
- It provides vector classes (2D and 3D), clocks, threads, Unicode strings and other things

- To use in your program:
  - Include sfml-system.lib in your Linker->Input
  - Or sfml-system-d.lib for Debug configuration

Window Module

- This module allows you to create application windows as well as collecting user input, such as mouse movement or key presses
  - You've seen Windows Application in Visual Studio before, but thus far your programs have been exclusively Console Applications

- To use in your program:
  - Include sfml-window.lib in your Linker->Input
  - Or sfml-window-d.lib for Debug configuration

Graphics Module

- The Graphics module allows you to include all functionality related to 2D rendering
  - Using images, texts, shapes and colors

- To use in your program:
  - Include sfml-graphics.lib in your Linker->Input
  - Or sfml-graphics-d.lib for Debug configuration

Audio Module

- The Audio module is, of course, provided so that you can add sounds to your game
  - Covers sound effects and music tracks

- To use in your program:
  - Include sfml-audio.lib in your Linker->Input
  - Or sfml-audio-d.lib for Debug configuration

Network Module

- Yes! SFML has a Network module that will allow you to setup multiplayer games
  - Includes everything you need to communicate over a LAN or the Internet using protocols such as HTTP and FTP

- And yes, we will be covering that in this course!

- To use in your program:
  - Include sfml-network.lib in your Linker->Input
  - Or sfml-network-d.lib for Debug configuration

SFML "Hello World"

```cpp
#include <SFML/Graphics.hpp>

int main()
{
    sf::RenderWindow window(sf::VideoMode(200, 200), "Hello World!");
    sf::CircleShape shape(100.f);
    shape.setFillColor(sf::Color::Green);

    while (window.isOpen())
    {
        sf::Event event;
        while (window.pollEvent(event))
        {
            if (event.type == sf::Event::Closed)
                window.close();
        }

        window.clear();
        window.draw(shape);
        window.display();
    }

    return 0;
}
```

Tips for Good Coding

- By this point, you should know how to code efficiently and use object-oriented features
- But let's reiterate some good concepts:

- Modularity
  - Keep your code separated into small pieces that perform a particular function
    - Separated into headers and implementation files
    - This will allow you to reuse that code easily, not only in the current program but other programs as well

Tips for Good Coding (cont'd.)

- Abstraction
  - Encapsulate functionality into classes and functions
  - This will prevent code duplications
  - Functions go way back to first term

- Consistency
  - Choose your coding style and stick to it so that it can be read easily and is more professional
  - Usually, this refers to how you use whitespace
  - Also how you use body braces, i.e.: { }

Abstraction into Practice

- To get you more familiar with SFML, we're going to take a minimal example on the next slide and convert the code into a class
- Through this, you should be able to see how we can break down the functionality into pieces and demonstrate how those pieces work together
- So let's get started!

Minimal Example

```
#include <SFML/Graphics.hpp>

int main()
{
    sf::RenderWindow window(sf::VideoMode(640,
    480), "SFML Application");
    sf::CircleShape shape;
     shape.setRadius(40.f);
     shape.setPosition(100.f, 100.f);
     shape.setFillColor(sf::Color::Cyan);
     while (window.isOpen())
     {
            sf::Event event;
            while (window.pollEvent(event))
            {
                if (event.type == sf::Event::Closed)
                window.close();
            }
            window.clear();
            window.draw(shape);
            window.display();
     }
}
```

Game Class

```
class Game
{
    public:
         Game();
         void run();
      private:
         void processEvents();
         void update();
         void render();
      private:
         sf::RenderWindow mWindow;
         sf::CircleShape mPlayer;
};

int main()
{
      Game game;
      game.run();
}
```

Game Implementation

```
Game::Game()
: mWindow(sf::VideoMode(640, 480), "SFML Application‟), mPlayer()
{
    mPlayer.setRadius(40.f);
    mPlayer.setPosition(100.f, 100.f);
    mPlayer.setFillColor(sf::Color::Cyan);
}

void Game::run()
{
    while (mWindow.isOpen())
    {
          processEvents();
          update();
          render();
    }
}
```

Game Implementation (cont'd.)

```
void Game::processEvents()
{
    sf::Event event;
    while (mWindow.pollEvent(event))
    {
            if (event.type == sf::Event::Closed)
                    mWindow.close();
    }
}

void Game::update()
{
}
```