II. Direct3D Foundations
6. Drawing in Direct3D

Game Graphic Programming
Kyung Hee University
Software Convergence
Prof. Daeho Lee

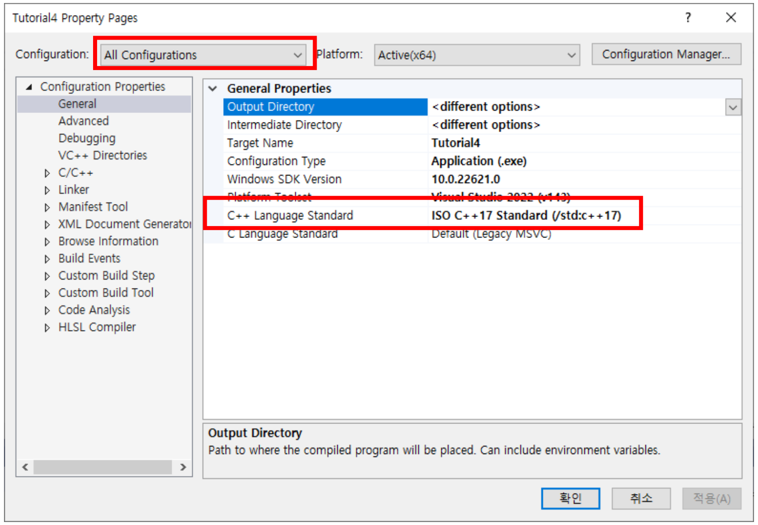
Useful Reference (1)

- Microsoft directX samples
  - https://github.com/microsoft/DirectX-Graphics-Samples/
- 3D game engine programming
  - DirectX12
    - https://www.3dgep.com/category/graphics-programming/directx/directx-12/
    - Source code
      - https://github.com/jpvanoosten/LearningDirectX12/tree/v0.0.4
        - cmake_minimum_required(VERSION 3.12.0)
          - https://cmake.org/download/ → Windows x64 Installer
        - Make solution: MakeVS.bat

          ```
          @ECHO OFF

          SET CMAKE="cmake.exe"
          SET CMAKE_GENERATOR="Visual Studio 17 2022"
          SET CMAKE_BINARY_DIR=build_vs2022

          MKDIR %CMAKE_BINARY_DIR% 2>NUL
          PUSHD %CMAKE_BINARY_DIR%

          %CMAKE% -G %CMAKE_GENERATOR% -Wno-dev %~dp0
          ```
        - ./build_vs2022/LearningDirectX12.sln

Useful Reference (2)

- C++ Language Standard: C++17 or C++ 20 (for all projects and all configurations)



Tutorial4 Property Pages

Configuration: All Configurations Platform: Active(x64) Configuration Manager...

Configuration Properties
- General
- Advanced
- Debugging
- VC++ Directories
- C/C++
- Linker
- Manifest Tool
- XML Document Generator
- Browse Information
- Build Events
- Custom Build Step
- Custom Build Tool
- Code Analysis
- HLSL Compiler

General Properties

Output Directory	<different options>
Intermediate Directory	<different options>
Target Name	Tutorial4
Configuration Type	Application (.exe)
Windows SDK Version	10.0.22621.0
Platform Toolset	Visual Studio 2022 (v143)
C++ Language Standard	ISO C++17 Standard (/std:c++17)
C Language Standard	Default (Legacy MSVC)

Output Directory
Path to where the compiled program will be placed. Can include environment variables.

- Dx12LibPCH.h (DX12Lib Project)

```
namespace fs = std::experimental::filesystem;
→ namespace fs = std::filesystem;
```

Useful Reference (3)

- Tutorial2

  ```
  // Tutorial2.cpp
  void Tutorial2::OnRender(RenderEventArgs& e){
     if (!m_ContentLoaded) return; // insertion
  ```

- Tutorial3 & Tutorial4

  -
    ```
    // Tutorial3.h & Tutorial4.h
    class Tutorialn : public Game{
       // …
       bool m_ContentLoaded;
    };
    ```

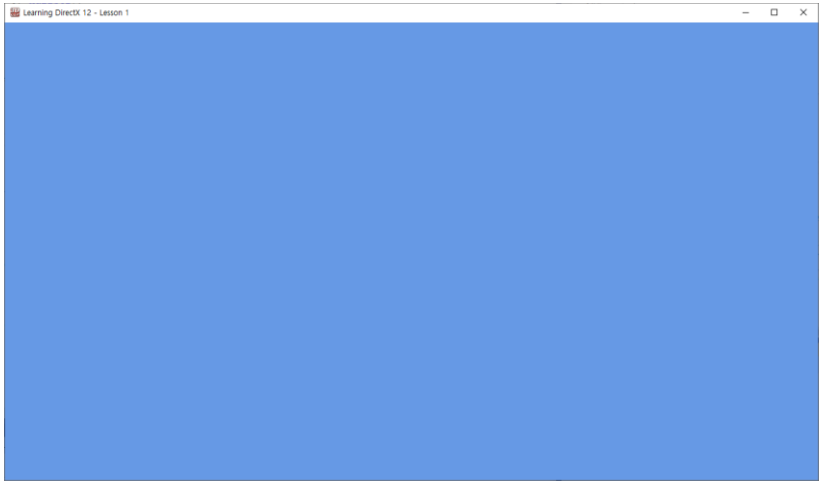
  -
    ```
    // Tutorial3.cpp & Tutorial4.cpp
    Tutorialn::Tutorialn(/*…*/):/*…*/,m_ContentLoaded(false)
    ```

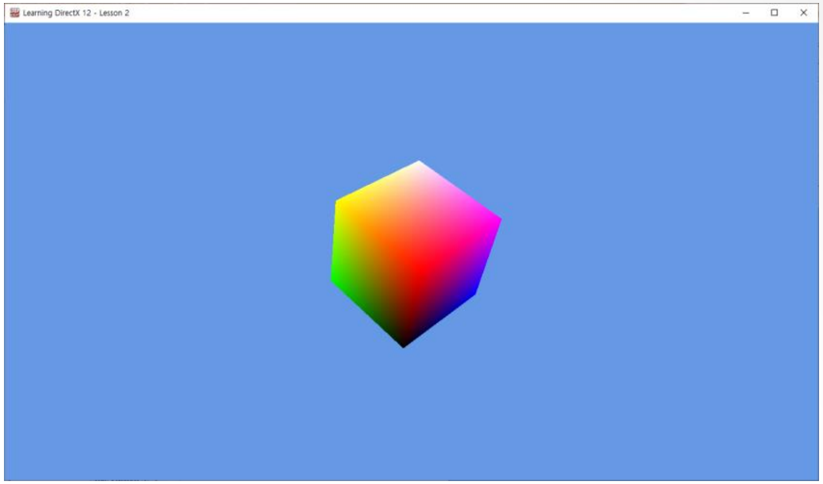
  -
    ```
    bool Tutorialn::LoadContent(){
       // …
       m_ContentLoaded = true;
       return true;
    }
    ```

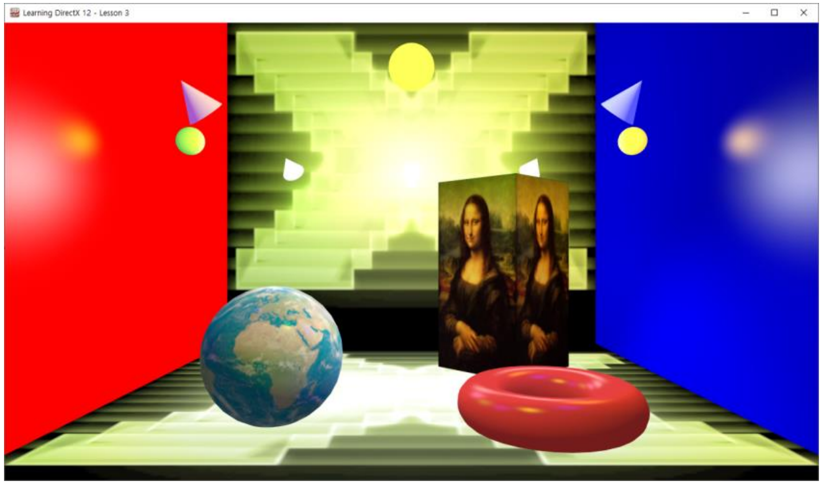
  -
    ```
    void Tutorialn::UnloadContent(){m_ContentLoaded=false;}
    ```

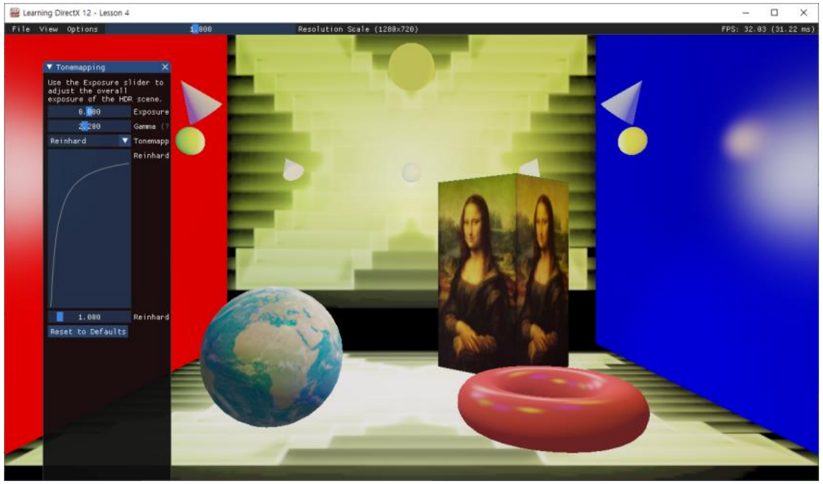
  -
    ```
    void Tutorialn::OnRender(RenderEventArgs& e){
       if (!m_ContentLoaded) return;
       // …
    }
    ```

Useful Reference (4)









1.000 Resolution Scale (1280x720)

FPS: 32.03 (31.22 ms)

Use the Exposure slider to adjust the overall exposure of the HDR scene.

0.000 Exposure

2.200 Gamma (?)

Reinhard Tonemapp

Reinhard

1.000 Reinhard