James Moran

This report shows the development process for the <name-pending> DirectX application.

James Moran CGP600 AE2 – Individual Project Report

Contents

**No table of contents entries found.**

# Introduction

## Initial Development Actions

I started off by using an older Tutorial project (Tutorial 08 Exercise 01), as a basis for this assignment’s project. This is because I deemed this project the most advanced working version of the project (no discernible memory leaks), which is also stable. I went through the code, refactoring suitable functions/methods, as well as properties, into classes, with respects to which actions these methods/functions perform and which class would require the respective properties for such.

There is a global function (called in the scope of Main.cpp), that initialises the classes. When this function is called in the WinMain() function, assert() macro-calls are used to make sure the pointers to these classes, are valid. (cplusplus.com, 2000-2017)

For referring to pointers (not copying them), by certain classes, of which the pointer is that of another class,\*& is used for such (instead of \*\*), to call the functions of that pointer’s class. (Matthew Hoggan, 2017)

After refactoring all of the components of the project into their own classes, came the process of cleaning-up the project, by resolving as many warnings as possible.

First off, came the process of resolving macro-definition classes between winerror.h and dxgi.h. Although a process is listed on MSDN, to resolve this issue, I chose the answer that was not the accepted answer (with the second highest amount of up-votes), which simply involves adding $(WindowsSDK\_IncludePath) to the Include Directories section of VC++ Directories. This resolved the occurrence of these warning messages. (gradbot, 2012)

After resolving the issue noted in the above paragraph, I resolved an issue with DirectX Memory-Leaks, which I was made aware of, by D3D11 issuing warnings to me, via the output window. These appear after terminating the application, with messages similar to those listed in Appendix A: Fig. 1.

I resolved live objects not being cleaned-up (which is what these messages indicate), by following all of the steps of the tutorial, that is available from the same source as the quote that is noted, in Appendix A: Fig. 1. (Master Kenneth, 2014)

# Additions to Satisfy the Basic Requirements

After the initial development actions, came the process of adding to the project, in order to satisfy the basic requirements (listed on the assignment brief).

First off, came that of adding functionality to the project to load and draw .obj files (assets, exported from a 3D-Modeling package, such as Autodesk’s 3D Studio-Max).

# Appendix A: Figures

Figure 1: A quote from masterkenth.com, as an example of D3D11 Warning messages:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | D3D11 WARNING: Process is terminating. Using simple reporting. Please call ReportLiveObjects() at runtime for standard reporting. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING: Live Producer at 0x007464B0, Refcount: 4. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x007476F8, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A07580, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A07264, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A10ACC, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A11204, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A1295C, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A11944, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A11B60, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A120C4, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A133BC, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A1373C, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A14A2C, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A14CB4, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A153AC, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A16D7C, Refcount: 1. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A22B7C, Refcount: 1. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x0C17154C, Refcount: 1. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A26904, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A2A2CC, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x0C1687F4, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x0C16BE0C, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING:  Live Object at 0x01A1ADC4, Refcount: 0. [ STATE\_CREATION WARNING #0: UNKNOWN]  D3D11 WARNING: Live                         Object :     22 [ STATE\_CREATION WARNING #0: UNKNOWN]  DXGI WARNING: Live Producer at 0x006AE3A8, Refcount: 4. [ STATE\_CREATION WARNING #0: ]  DXGI WARNING:  Live Object at 0x006AEA38, Refcount: 2. [ STATE\_CREATION WARNING #0: ]  DXGI WARNING: Live                         Object :      1 [ STATE\_CREATION WARNING #0: ] |

(Master Kennth, 2014)

# References

1. © cplusplus.com, 2000-2017. *assert* [viewed on the 29/11/2017]. Available from: <http://www.cplusplus.com/reference/cassert/assert/>
2. Matthew Hoggan, 2012. *Reason to Pass a Pointer by Reference in C++?* [viewed on the 30/11/2017]. Available from: <https://stackoverflow.com/questions/10240161/reason-to-pass-a-pointer-by-reference-in-c>
3. gradbot, 2012, *Macro definition clash between directx headers and winerror.h* [viewed on the 07/12/2017]. Available from: https://stackoverflow.com/questions/12663091/macro-definition-clash-between-directx-headers-and-winerror-h
4. Master Kenneth, 2014, DirectX Leak Debugging [viewed on the 07/12/2017]. Available from: <http://masterkenth.com/directx-leak-debugging/>