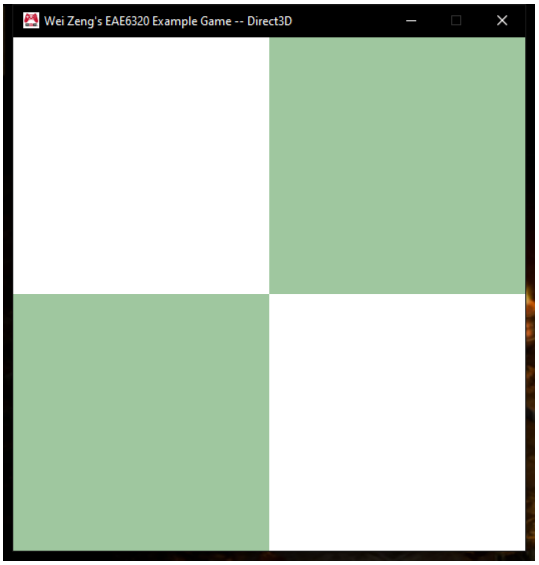
Assignment 3 Writeup

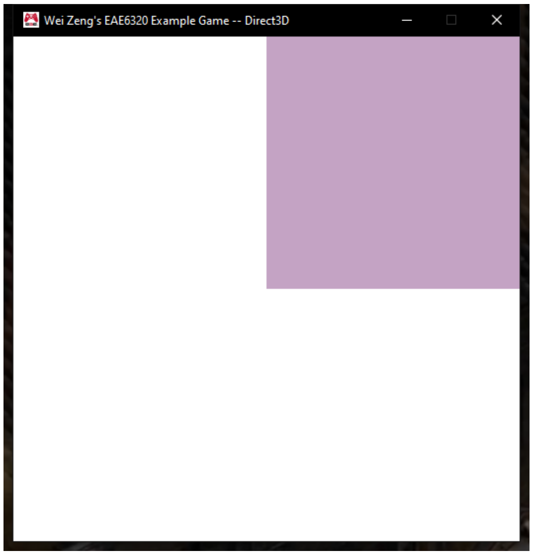
Game Executable:

https://github.com/WayGold/EAE6320_Assignments/blob/Assignment04/MyGame_.zip

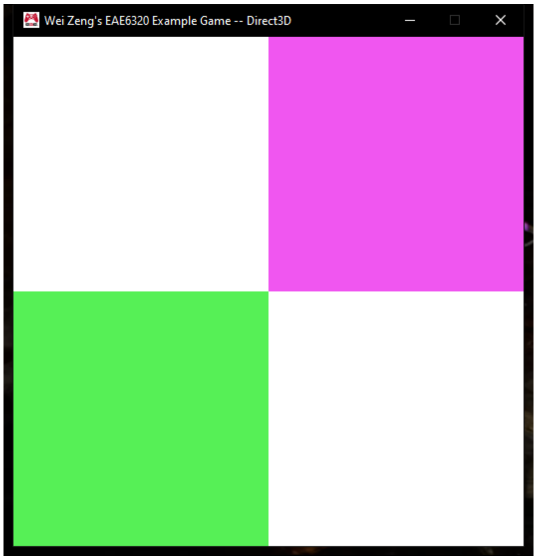
Game running:



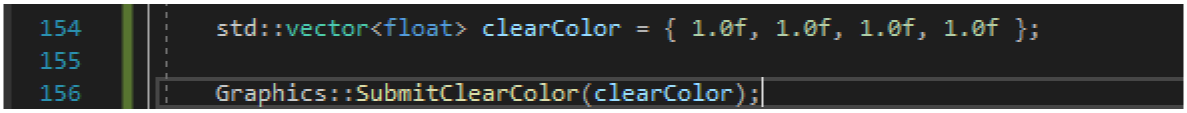
(Default State)



(Missing One Mesh - Press Space)



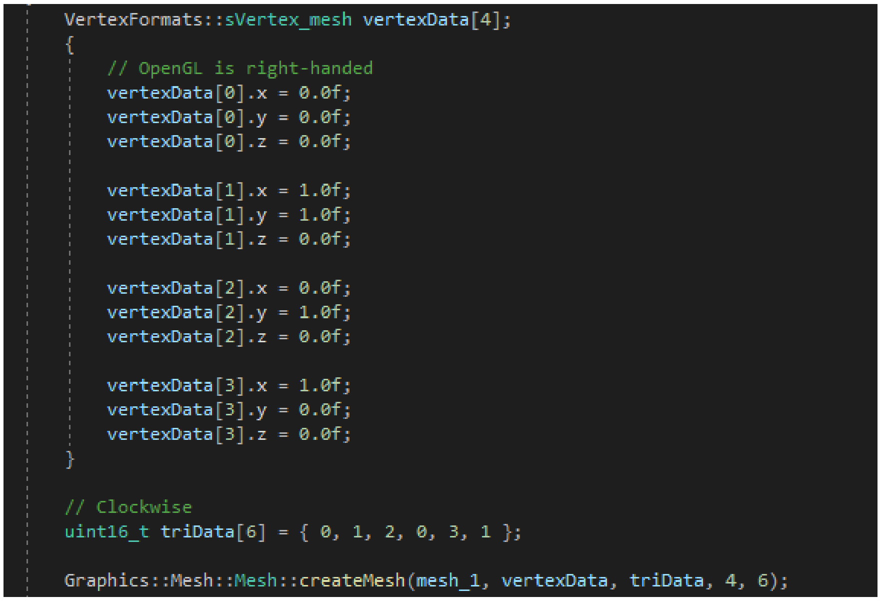
(Different Effect - Press Tab)



```
std::vector<float> clearColor = { 1.0f, 1.0f, 1.0f, 1.0f };

Graphics::SubmitClearColor(clearColor);
```

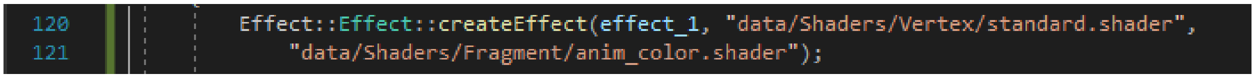
(Submitting Background Color)

In order to submit a mesh/effect pair to be drawn, you need to first create a mesh and an effect.



```
VertexFormats::sVertex_mesh vertexData[4];
{
    // OpenGL is right-handed
    vertexData[0].x = 0.0f;
    vertexData[0].y = 0.0f;
    vertexData[0].z = 0.0f;

    vertexData[1].x = 1.0f;
    vertexData[1].y = 1.0f;
    vertexData[1].z = 0.0f;

    vertexData[2].x = 0.0f;
    vertexData[2].y = 1.0f;
    vertexData[2].z = 0.0f;

    vertexData[3].x = 1.0f;
    vertexData[3].y = 0.0f;
    vertexData[3].z = 0.0f;
}

// Clockwise
uint16_t triData[6] = { 0, 1, 2, 0, 3, 1 };

Graphics::Mesh::Mesh::createMesh(mesh_1, vertexData, triData, 4, 6);
```

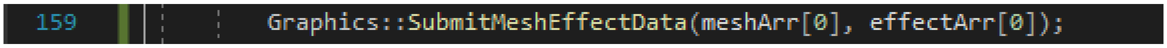
(Creating Mesh)



```
Effect::Effect::createEffect(effect_1, "data/Shaders/Vertex/standard.shader",
    "data/Shaders/Fragment/anim_color.shader");
```

(Creating Effect)

Once you have created a mesh and a corresponding effect, you could call the following function to submit.

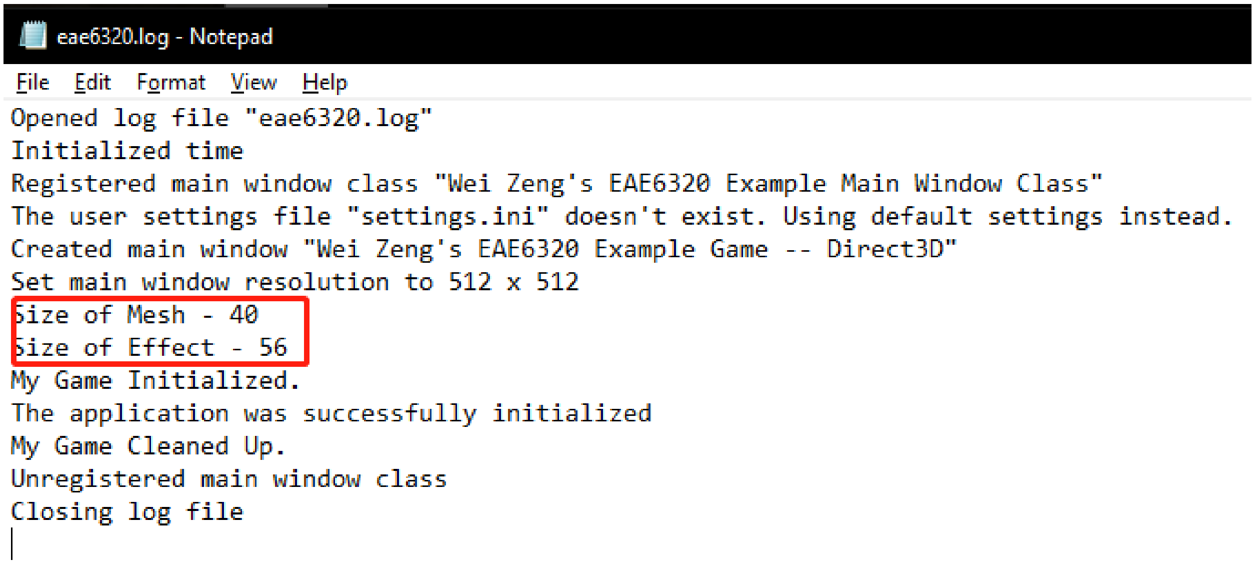


```
Graphics::SubmitMeshEffectData(meshArr[0], effectArr[0]);
```

(Submit with a mesh pointer and effect pointer)

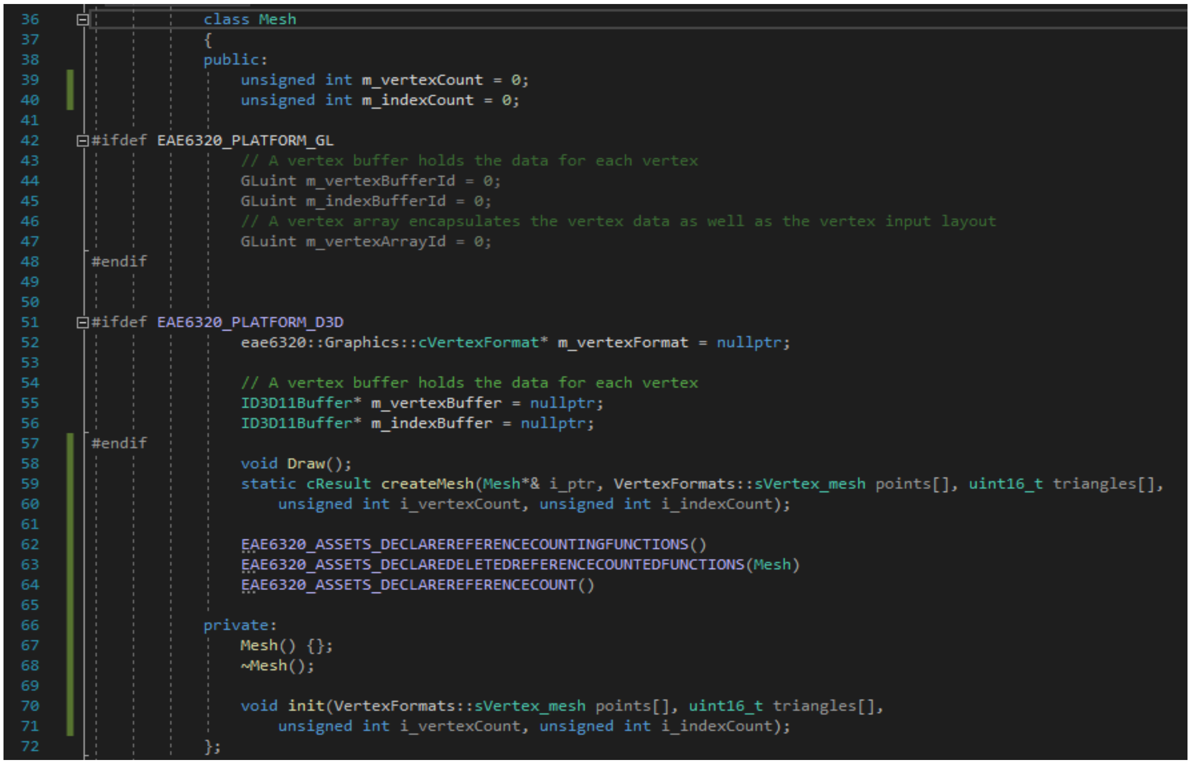
The reason why we need to cache the data is due to the fact that we are running multithreading and we don’t want unexpected behavior. If we don’t cache the render thread could potentially start rendering when the data is not submitted for example.

The mesh class has a size of 40 bytes under x64, previously I had two vectors to store the vertex data and was using them mainly just for getting the size of the vertex array and index array. Vector took a lot of memory space so I swap them into two unsigned int just to store the array sizes and drastically reduce the space.



```
Opened log file "eae6320.log"
Initialized time
Registered main window class "Wei Zeng's EAE6320 Example Main Window Class"
The user settings file "settings.ini" doesn't exist. Using default settings instead.
Created main window "Wei Zeng's EAE6320 Example Game -- Direct3D"
Set main window resolution to 512 x 512
Size of Mesh - 40
Size of Effect - 56
My Game Initialized.
The application was successfully initialized
My Game Cleaned Up.
Unregistered main window class
Closing log file
```

(Log the size of mesh and effect)

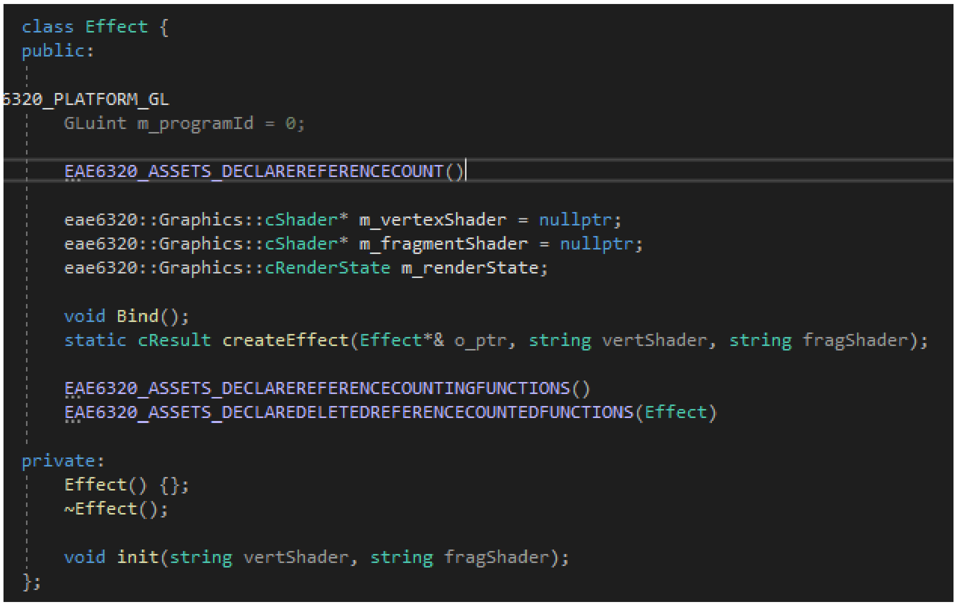


```
        class Mesh
        {
        public:
            unsigned int m_vertexCount = 0;
            unsigned int m_indexCount = 0;

#ifdef EAE6320_PLATFORM_GL
            // A vertex buffer holds the data for each vertex
            GLuint m_vertexBufferId = 0;
            GLuint m_indexBufferId = 0;
            // A vertex array encapsulates the vertex data as well as the vertex input layout
            GLuint m_vertexArrayId = 0;
#endif


#ifdef EAE6320_PLATFORM_D3D
            eae6320::Graphics::cVertexFormat* m_vertexFormat = nullptr;

            // A vertex buffer holds the data for each vertex
            ID3D11Buffer* m_vertexBuffer = nullptr;
            ID3D11Buffer* m_indexBuffer = nullptr;
#endif
            void Draw();
            static cResult createMesh(Mesh*& i_ptr, VertexFormats::sVertex_mesh points[], uint16_t triangles[],
                unsigned int i_vertexCount, unsigned int i_indexCount);

            EAE6320_ASSETS_DECLAREREFERENCECOUNTINGFUNCTIONS()
            EAE6320_ASSETS_DECLAREDELETEDREFERENCECOUNTEDFUNCTIONS(Mesh)
            EAE6320_ASSETS_DECLAREREFERENCECOUNT()

        private:
            Mesh() {};
            ~Mesh();

            void init(VertexFormats::sVertex_mesh points[], uint16_t triangles[],
                unsigned int i_vertexCount, unsigned int i_indexCount);
        };
```

(Mesh class members)

The effect class is pretty much the same as the previous assignment with size 56 bytes under x64.

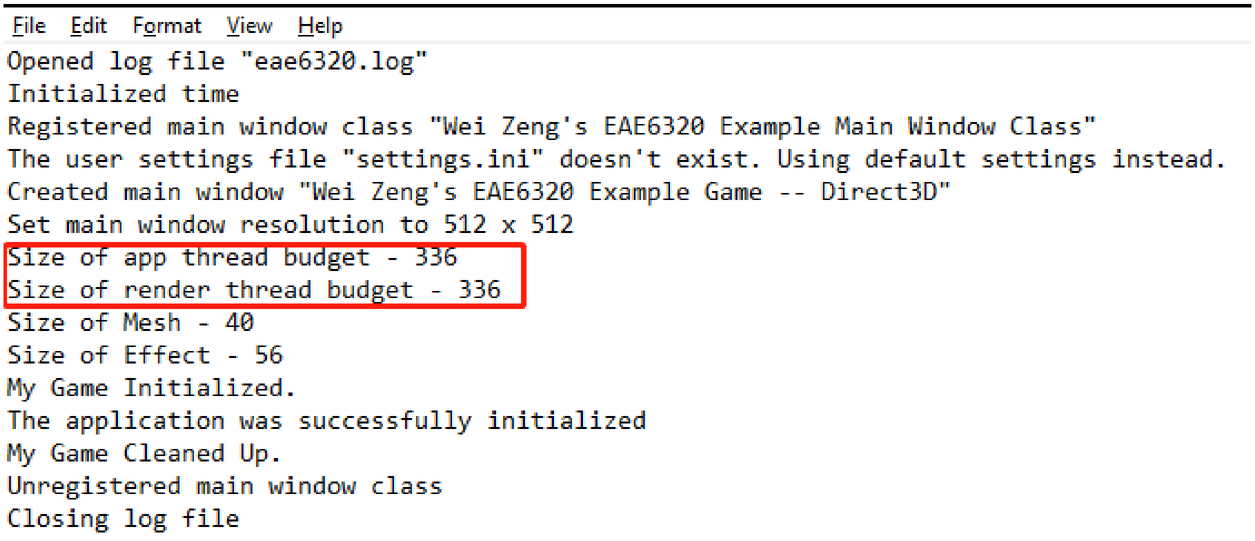
Adding 8 more bytes with the reference count variable anywhere would not make any difference.



```
class Effect {
public:

6320_PLATFORM_GL
    GLuint m_programId = 0;

    EAE6320_ASSETS_DECLAREREFERENCECOUNT()

    eae6320::Graphics::cShader* m_vertexShader = nullptr;
    eae6320::Graphics::cShader* m_fragmentShader = nullptr;
    eae6320::Graphics::cRenderState m_renderState;

    void Bind();
    static cResult createEffect(Effect*& o_ptr, string vertShader, string fragShader);

    EAE6320_ASSETS_DECLAREREFERENCECOUNTINGFUNCTIONS()
    EAE6320_ASSETS_DECLAREDELETEDREFERENCECOUNTEDFUNCTIONS(Effect)

private:
    Effect() {};
    ~Effect();

    void init(string vertShader, string fragShader);
};
```

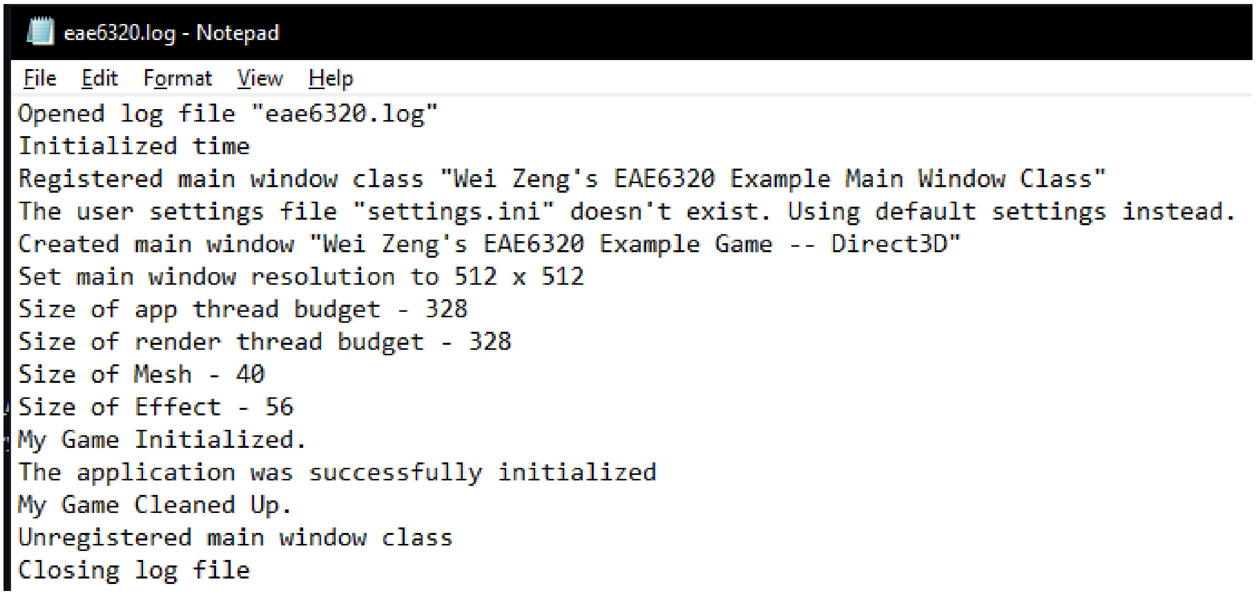
(Effect class members)

The total memory I budget for the render data was 336 and then reduced to 328. Originally I used a vector to store background color so I swapped it into a float array of 4 elements. I used two arrays each to store the submitted pointers for meshes and effects and a gameObject counter to keep track of the relationships between meshes and effects.



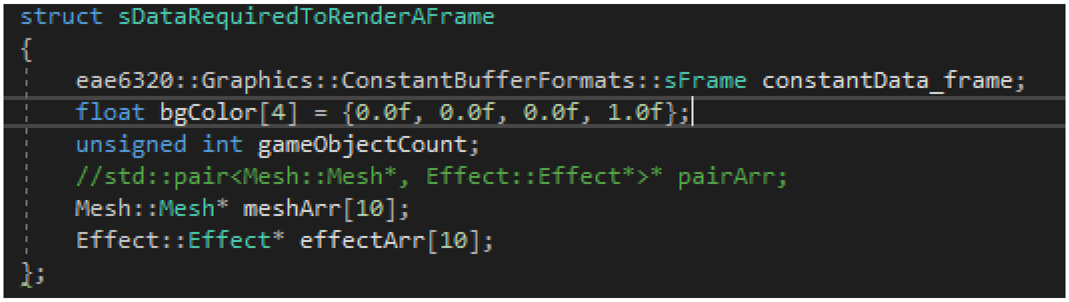
```
Opened log file "eae6320.log"
Initialized time
Registered main window class "Wei Zeng's EAE6320 Example Main Window Class"
The user settings file "settings.ini" doesn't exist. Using default settings instead.
Created main window "Wei Zeng's EAE6320 Example Game -- Direct3D"
Set main window resolution to 512 x 512
Size of app thread budget - 336
Size of render thread budget - 336
Size of Mesh - 40
Size of Effect - 56
My Game Initialized.
The application was successfully initialized
My Game Cleaned Up.
Unregistered main window class
Closing log file
```

(Before change)



```
Opened log file "eae6320.log"
Initialized time
Registered main window class "Wei Zeng's EAE6320 Example Main Window Class"
The user settings file "settings.ini" doesn't exist. Using default settings instead.
Created main window "Wei Zeng's EAE6320 Example Game -- Direct3D"
Set main window resolution to 512 x 512
Size of app thread budget - 328
Size of render thread budget - 328
Size of Mesh - 40
Size of Effect - 56
My Game Initialized.
The application was successfully initialized
My Game Cleaned Up.
Unregistered main window class
Closing log file
```

(After change)



```
struct sDataRequiredToRenderAFrame
{
    eae6320::Graphics::ConstantBufferFormats::sFrame constantData_frame;
    float bgColor[4] = {0.0f, 0.0f, 0.0f, 1.0f};
    unsigned int gameObjectCount;
    //std::pair<Mesh::Mesh*, Effect::Effect*>* pairArr;
    Mesh::Mesh* meshArr[10];
    Effect::Effect* effectArr[10];
};
```

(data structure)