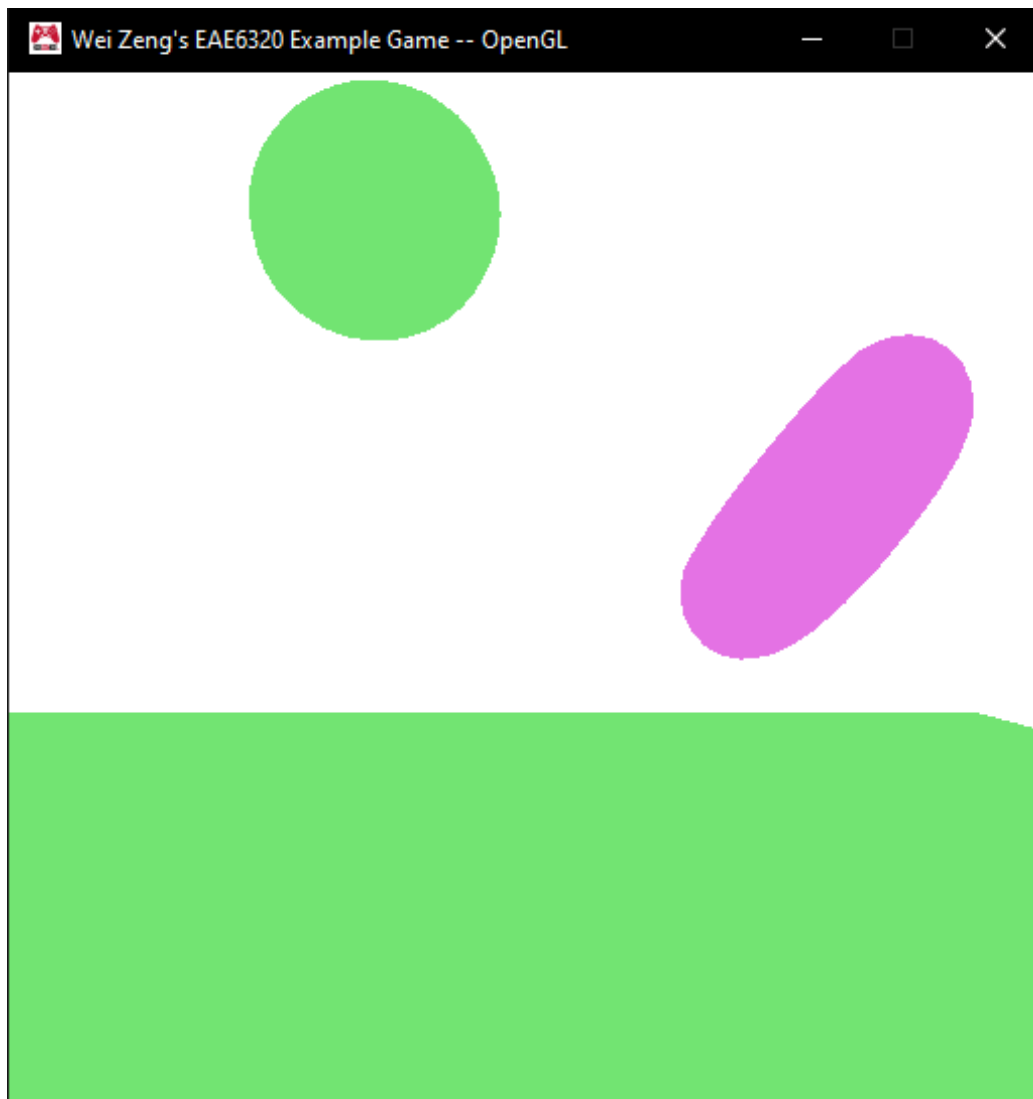


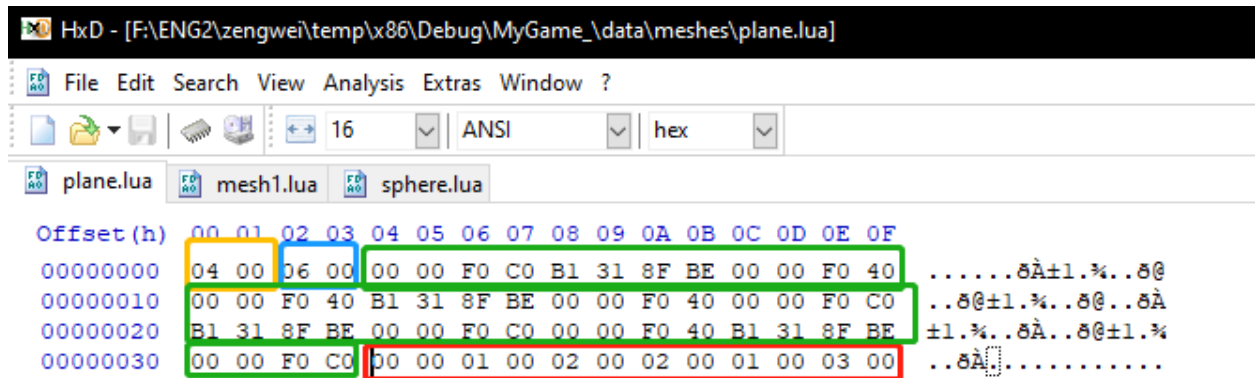
Assignment 8 Writeup

Game Executable:

Game running:



(Game Running)



(Plane binary in Hex Editor)

The order of the data is as follows:

1. Vertex count
2. Index count
3. Vertex data
4. Index data

First advantage that I could think of is security. Having a binary format could prevent people from easily modifying or accessing your assets to some degree but at the same time like we said in class it also discourages modding for the game. In addition, binary files compared to the original lua file could have faster loading times since with large lua files we need to loop a lot. Similarly, binary files will also have smaller size compared to the human readable files.

```

59
60 // Write to Binary
61 std::ofstream outfile(o_path, std::ofstream::binary);
62 // Write Vertex and Index Count
63 outfile.write((const char*)&vertexCount, sizeof(uint16_t));
64 outfile.write((const char*)&indexCount, sizeof(uint16_t));
65 // Write Vertex and Index Data
66 size_t vertexBufferSize = sizeof(eae6320::Graphics::VertexFormats::sVertex_mesh) * vertexCount;
67 size_t indexBufferSize = sizeof(uint16_t) * indexCount;
68 outfile.write((const char*)points, vertexBufferSize);
69 outfile.write((const char*)triangles, indexBufferSize);
70 outfile.close();
71

```

(Writing to Binary)

helix.lua	10/30/2022 6:49 PM	Lua 源文件	282 KB
-----------	--------------------	---------	--------

(Human Readably)

helix.lua	10/30/2022 6:55 PM	Lua 源文件	118 KB
-----------	--------------------	---------	--------

(Binary File)

The human readable format for the helix is more than 2 times larger than the binary format.

```

2>"F:\ENG2\zengwei\temp\x64\Debug\output\MeshBuilder.exe" "F:
2>Getting Vertex Count:
2>Getting Index Count:
2>Iterating through every vertex data:
2>Iterating through every index data:
2>Time to load human readable: 38777 (ms)
2>Built F:\ENG2\zengwei\MyGame_\Content\meshes\helix.lua

```

(Time to Load Helix Using Human Readable Format)

With the human readable format, loading a helix took me 38777 ms as we are just looping for so many vertices and indices.

```
eae6320.log - Notepad
File Edit Format View Help
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 -- Debug"
Set main window resolution to 512 x 512
Size of app thread budget - 968
Size of render thread budget - 968
Time used to load binary file for:
  data/meshes/Plane.lua
81 (ms)
Time used to load binary file for:
  data/meshes/helix.lua
162 (ms)
Time used to load binary file for:
  data/meshes/tube.lua
63 (ms)
Size of Mesh - 40
Size of Effect - 56
Size of RenderObject - 88
Size of RigidBody - 68
My Game Initialized.
The application was successfully initialized
My Game Cleaned Up.
Unregistered main window class
Closing log file
```

(Time to Load Helix Using Binary Format)

However, with the binary format, it only took me 162 ms to load all the data since all operations have $O(1)$ time complexity.