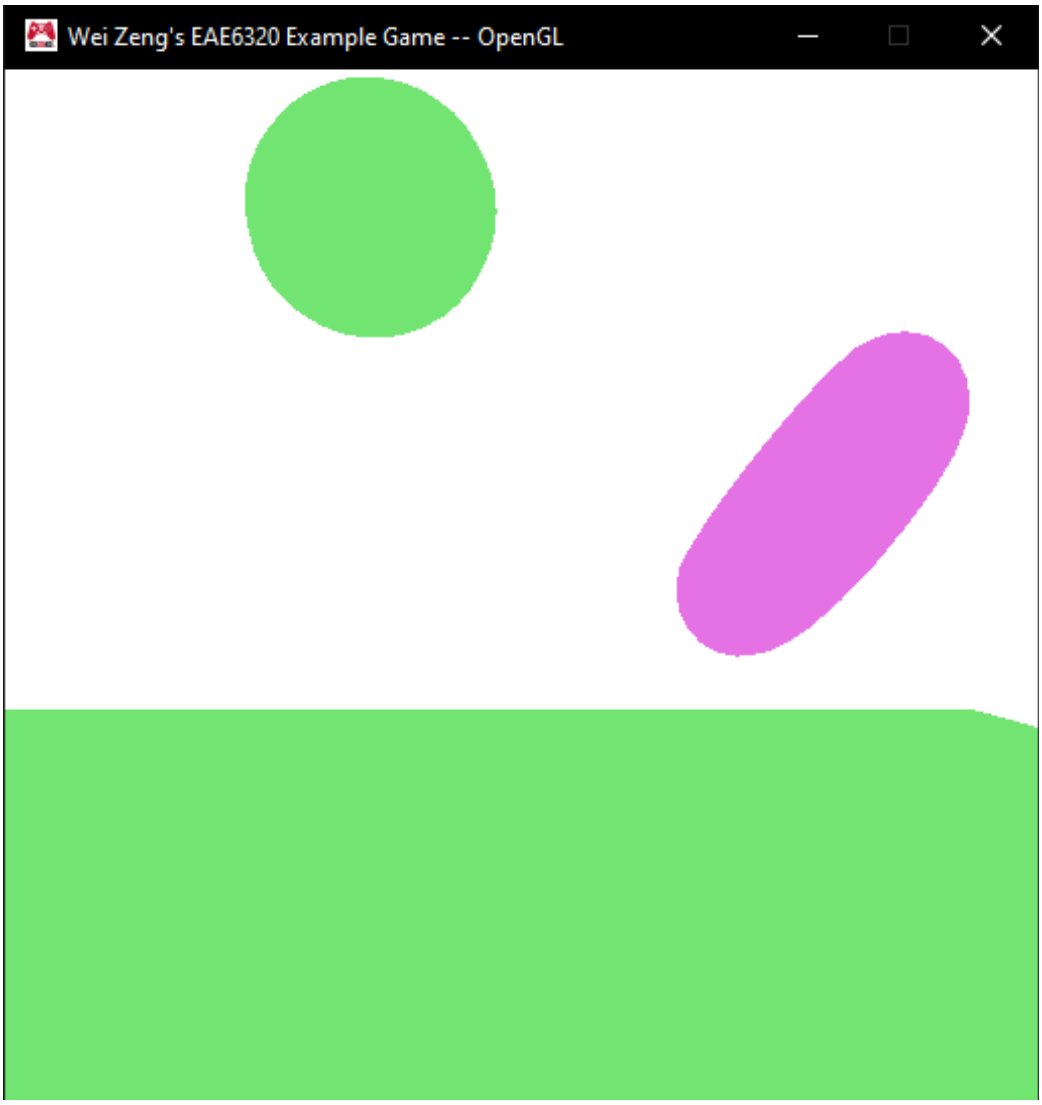
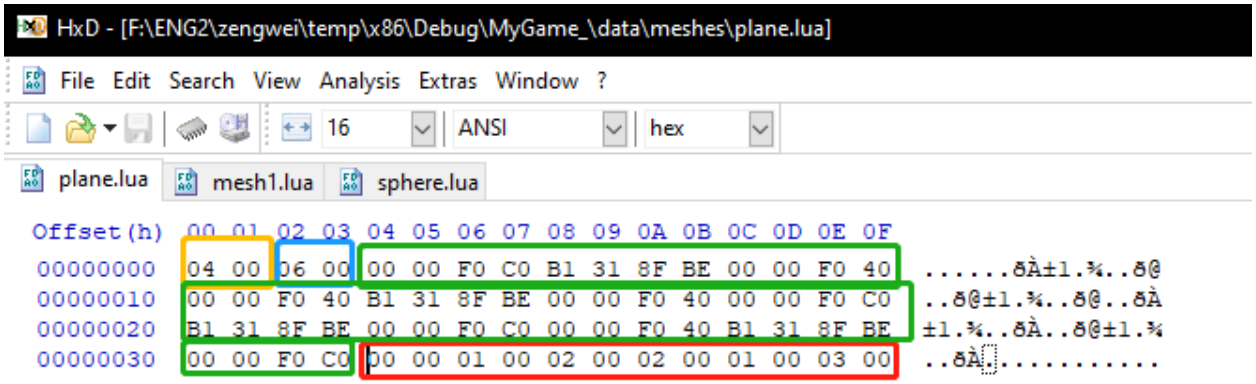
Assignment 8 Writeup

Game Executable:

Game running:



(Game Running)



(Plane binary in Hex Editor)

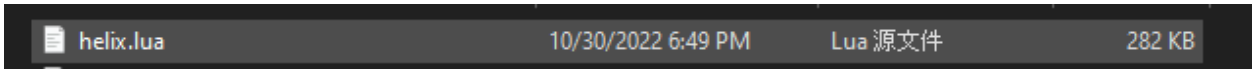
The order of the data is as follows:

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

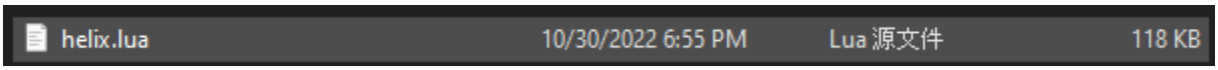
First advantage that I could think of is security. Having a binary format could prevent people from easily modifying or accessing your assets to some degree but at the same time like we said in class it also discourages modding for the game. In addition, binary files compared to the original lua file 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)



(Human Readably)



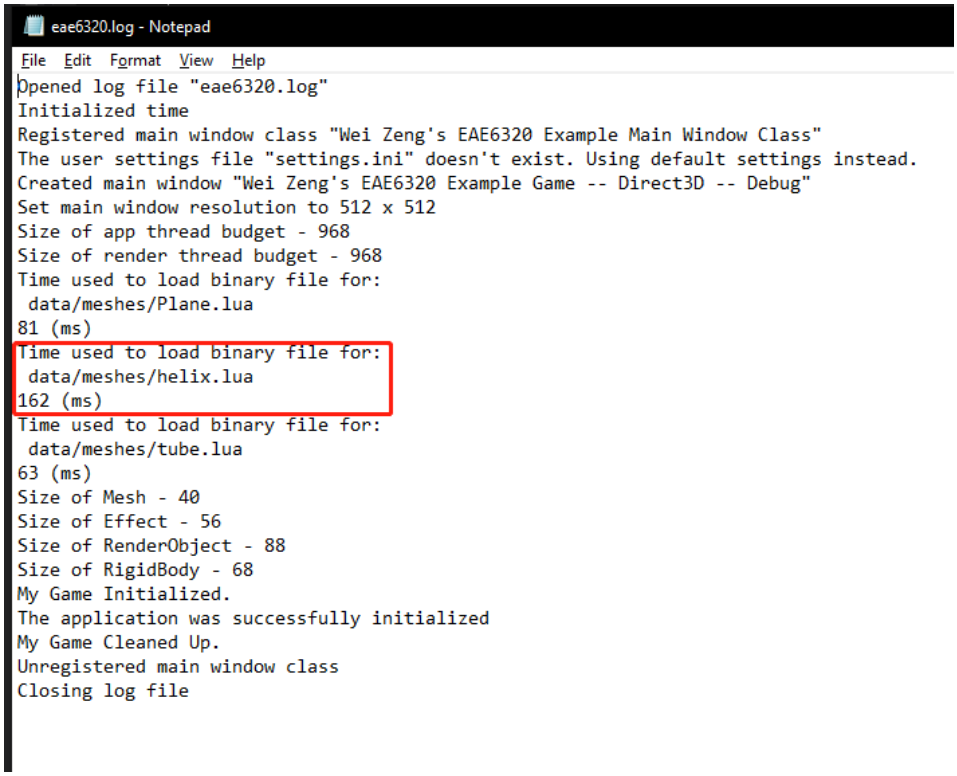
(Binary File)

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.



(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.