MIP Mapping

Guided by Professor Michael Manzke
Implemented by: Yuzhou Shao
Student id: 19322035
Youtube Demo: https://youtu.be/OHzWwTk8drk

My Submission Includes:

This Report: CS7GV3-MIPMapping-Report.pdf

A Youtube Demo https://youtu.be/OHzWwTk8drk also mentioned above

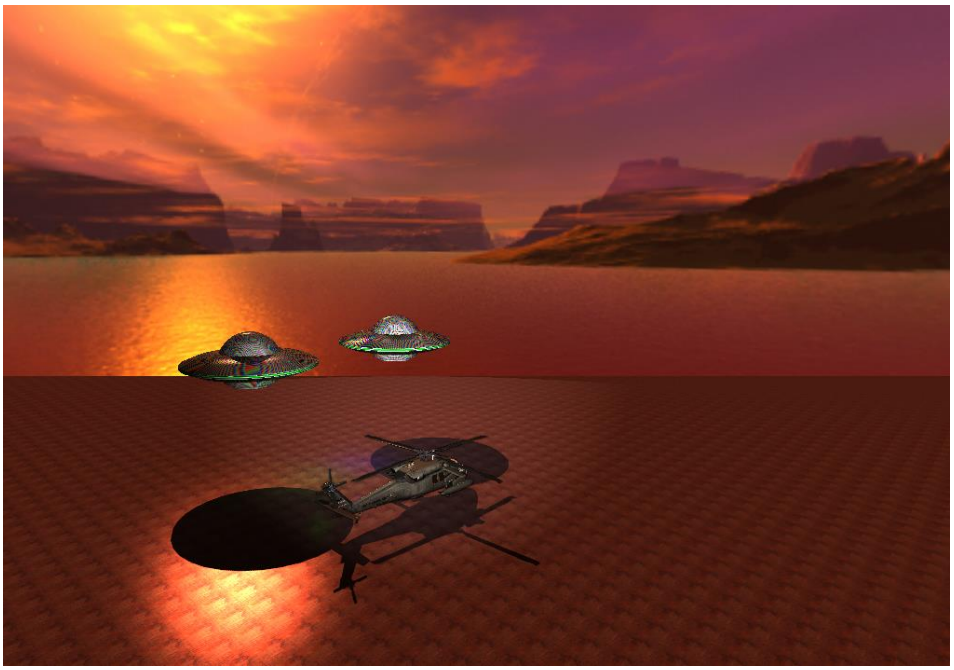
A Zip file includes:

Source Code: main.cpp, **Texture.cpp** and Texture.h

Shader Code: shader.frag, shader.vert omni_shadow_map.vert, omni_shadow_map.geom, omni_shadow_map.frag, directional_shadow_map.frag and directional_shadow_map.vert

Description of my scene

My scene includes an extended ground with repeated brick as texture mapping showing the **MIP mapping**, a static Blackhawk helicopter, two rotating UFOs, the shadow mapping of those three objects and a skybox.



External Libraries, 3rd party source code used:

As down left, I have utilized the **external libraries** including GLEW GLFW and glm. I have referenced **the 3rd party** classes **code** down right from an Udemy OpenGL tutorial OpenGL + C++: Modern Graphics for Groundbreaking Games | Udemy.

```
3   #include <stdio.h>
4   #include <string.h>
5   #include <cmath>
6   #include <vector>
7
8   #include <GL\glew.h>
9   #include <GLFW\glfw3.h>
10
11  #include <glm\glm.hpp>
12  #include <glm\gtc\matrix_transform.hpp>
13  #include <glm\gtc\type_ptr.hpp>
14
```

```
17  #include "Window.h"
18  #include "Mesh.h"
19  #include "Shader.h"
20  #include "Camera.h"
21  #include "Texture.h"
22  #include "DirectionalLight.h"
23  #include "PointLight.h"
24  #include "SpotLight.h"
25  #include "Material.h"
26
27  #include "Model.h"
```

I have used GLSL

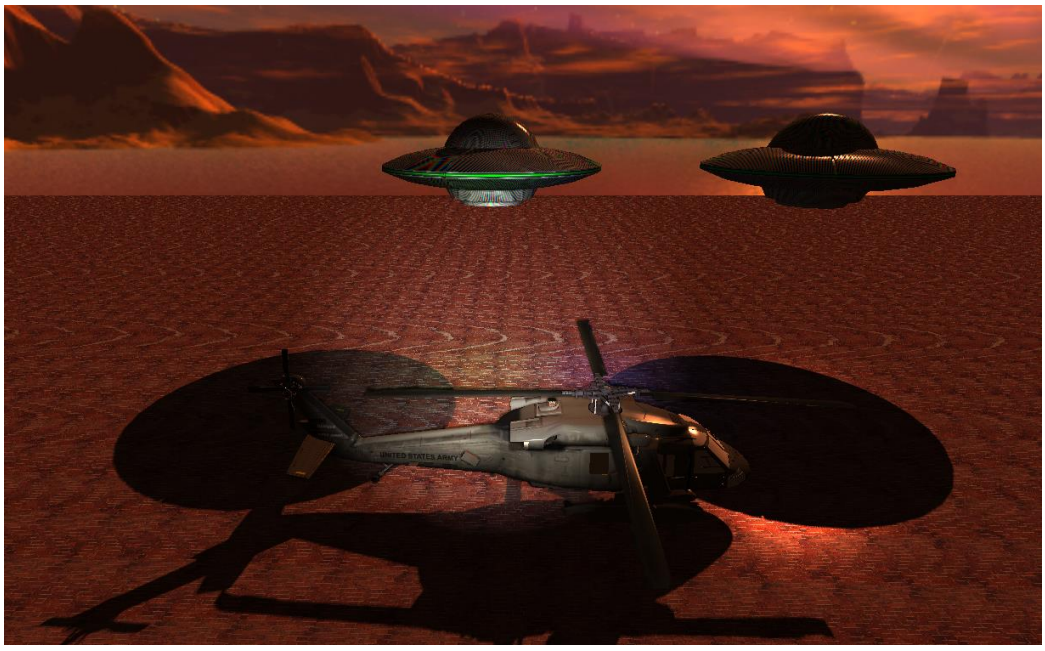
```
153 void CreateShaders()
154 {
155     Shader* shader1 = new Shader();
156     shader1->CreateFromFiles(vShader, fShader);
157     shaderList.push_back(*shader1);
158
159     directionalShadowShader.CreateFromFiles("Shaders/directional_shadow_map.vert", "Shaders/directional_shadow_map.frag");
160     omniShadowShader.CreateFromFiles("Shaders/omni_shadow_map.vert", "Shaders/omni_shadow_map.geom", "Shaders/omni_shadow_map.frag");
161 }
```

```
70  // Vertex Shader
71  static const char* vShader = "Shaders/shader.vert";
72
73  // Fragment Shader
74  static const char* fShader = "Shaders/shader.frag";
75
```

Goals

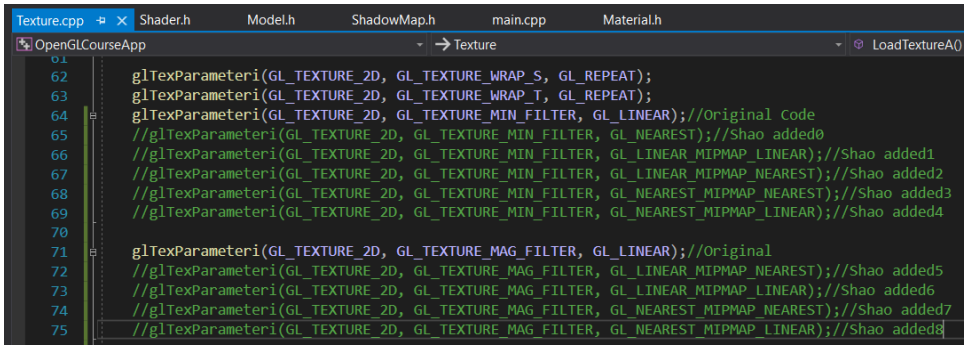
Implement a program that Demonstrates Mip mapping

Without MIP Mapping



With MIP Mapping

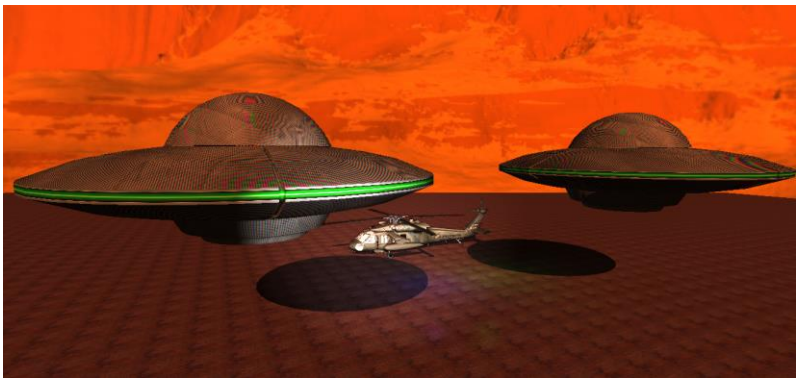
In my Texture Class, in line 64 or 71, by changing the last parameter GL_LINEAR to GL_LINEAR_MIPMAP_NEAREST, GL_LINEAR_MIPMAP_LINEAR , GL_NEAREST _MIPMAP_LINEAR or GL_NEAREST_MIPMAP_NEAREST respectively, the effect of MIP Mapping shown above can be implemented successfully.



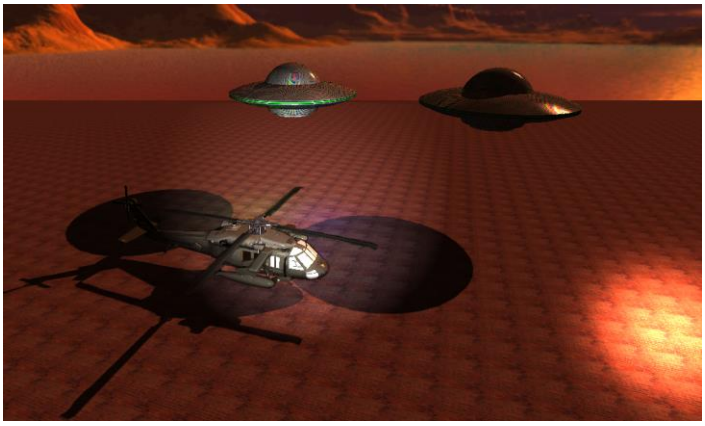
Secondary Objectives:

There are two **rotating Objects**(UFOs) implemented.

Besides, there is a Blackhawk helicopter implemented between the two UFOs as **variation in models**, to **make the demo slightly unique.**

The extended and repeated brick images as the texture of the ground, the skybox, and the shadow mapping implemented make the scene **as photorealistic as possible**.




Reference:

The OpenGL® Programming Guide 9th Edition
http://www.opengl-redbook.com/

LearnOpenGL - Textures

OpenGL + C++: Modern Graphics for Groundbreaking Games | Udemy