**HW4 document**

Huayan Sun

I tested my program in Zoolab before submitting and my program accomplishes everything exactly and correctly required by HW4. Please click the “GLSLExperiment.sln” which should open Visual Studio, and then run the program (Ctrl + F5).

I started HW4 on base of the previous homework HW3 with the starting codes given by Professor, which contains GLEW/GLUT as well as some starting header files and source files that I need. I have a folder named "data" which contains 7 .ply files and the .bmp files. My main program is in example1.cpp, and I have a vertex shader (vshader1.glsl) and a fragment shader (fshader1.glsl). I also added helper.h and help.cpp file where I defined my structs as well as matrix and stack operations. And there is bmpread.h and bmpread.cpp for .bmp file reading.

My sculpture consists of 7 unique meshes with 7 different colors and 3 levels of hierarchy just as the example given on HW3 webpage. All meshes rotate counter-clockwise about their own Y axis. The arm of Level 1 of the hierarchy rotates clockwise, the arm of level 2 of the hierarchy rotates counter-clockwise, and the arm of level 3 rotates clockwise.

Program starts at main() in example1.cpp. It first initializes basic settings and creates a window. Then in myInit(), it reads in all data as well as sets up shaders and GPU buffer. Then it registers callback functions – display, keyboard and idle. Finally it enters the drawing loop glutMainLoop().

To deal with the performance issues, I read in all data at the beginning once, such as points and normal, and store them on GPU via VBO, thus completely avoid unnecessary re-initialization, buffer copying or file reopening that could slow down my program considerably. In display (), it sets up projection matricies, light, and hierarchy transformations. In keyboard(), it switches to different modes or actions based on the key pressed. When there is no event, program enters idle(). In idle(), I define frame rate as 30 FPS and animate my model at that speed.

My programs can do everything exactly the way required by HW4 on course website. When program starts, it just shows plain walls with objects in original own colors. Through keyboard operations, you can draw shadows, wall textures, reflection and refraction as a toggle.