System Specifications

Erik Smith (40002747)

**Features**

|  |  |
| --- | --- |
| All spatial displacements (i.e. scaling, translation, rotation, ...) | Completed |
| Spatial data structure (via the *Part.h*, *complex.h* and *object.h* classes) | Completed |
| Object instancing | Completed |
| Lighting | Completed |
| Texturing + toggle | Completed |
| Collision detection | Completed |
| Horse “AI” | Completed |
| Camera mouvement | Completed |
| Shadows | Incompleted |
| Animations. (Joints had been implemented.) | Incompleted |
| Render modes (Lines, points, triangles) | Completed |
| Window Resizing | Completed |

\***Note**

For future classes, I believe it would be in the students best interest to go over shadows in detail during the Lab time as we did not cover this topic at all and is quite complex in the online tutorials. Animations were not implemented as my time was short for the second assignment. The task of animating the joins was tedious and time consuming, especially after getting lighting and texturing done. I had to stop the animation work half way through and decided to remove the unused code from the horse class. Assignment 2 should have been broken into 2 seperate assignments.

**What went right?**

The Code is very well devised into classes following a good OOP design and techniques we were taught in class. Most of the features are implemented and robust. The code follows a logical sequence with many relevant comments separating the chunks. After having a short break between the assignments, I was able to jump back into it easily thanks to the commenting I had done.

**Difficulties**

At the beginning of the class, we are taught the basics of openGL in the lectures which was fantastic, however we never fully went into details about how that correlates in the codebase leading to some time consuming debugging (I.E. I grasped the concept of VBO in class, but took me a very long time to figure out how to properly use them in the code despite being a simple mechanic). Going into the first lab, It was basically implied we already had a firm understanding of openGL’s codebase which was very much not the case.

**Clarifications**

It was unclear to me if the previous work done on the horse needed to be kept for the singular horse or needed to be implemented for all newly instanced horses. To compensate, I have done both with minimal code change.To apply all the previous work to the horses, simply uncomment in the lines 165 and 166 Main.cpp:

*>//eventableObjects.push\_back(horseCp);*

*>//horseCp->ai = false;*

**Note:** to control the horses, you need to turn off their AI thus they will not move around on their own and will not be able to respond to collision detection by design. This applies to any horse whose AI has been turned off.

**Extra features**

|  |  |
| --- | --- |
| Out of bounds detection for world map | Completed |
| Randomly making horse stand still for a random amount of time | Completed |
| Randomize horse speed at random intervals | Completed |
| FPS limiter (Global variable FPS in *main.cpp*) | Completed |