

CS 480 Computer Graphics

3D Solar System Model

Technical Manual

Developed by:

Enzo Arata

Evan Brown

Zachary Mcilwain

Overview

Dependencies

This assignment was designed and tested on a Linux operating system, and requires the use of third party packages and libraries. Instructions to install these packages on Ubuntu are found below.

GLEW, SDL2, GLM:

```
sudo apt-get install libglew-dev libsdl2-dev libglm-dev
```

Assimp:

```
sudo apt-get install libassimp-dev
```

ImageMagick and Magick++:

```
sudo apt-get install imagemagick libmagick++-dev
```

Extra Credit

- Live adjustment of simulation speed
- Option to go from actual data to scaled view

User Manual

Build Instructions

This assignment was compiled using the CMake build system. All the necessary CMake modules are on the project's repository, as well as the CMakeLists.txt file. The instructions to build and run the program are found below:

```
mkdir build
cd build
cmake ..
make
```

Keyboard Inputs

Game Controls

UP : Increase orbit scale

DOWN : Decrease orbit scale

LEFT : Decrease orbit speed

RIGHT : Increase orbit speed

Camera Controls

0 : Sun

1 : Mercury

2 : Venus

3 : Earth

4 : Mars

5 : Jupiter

6 : Saturn

7 : Uranus

8 : Neptune

9 : Pluto

w : Pan up

a : Pan left

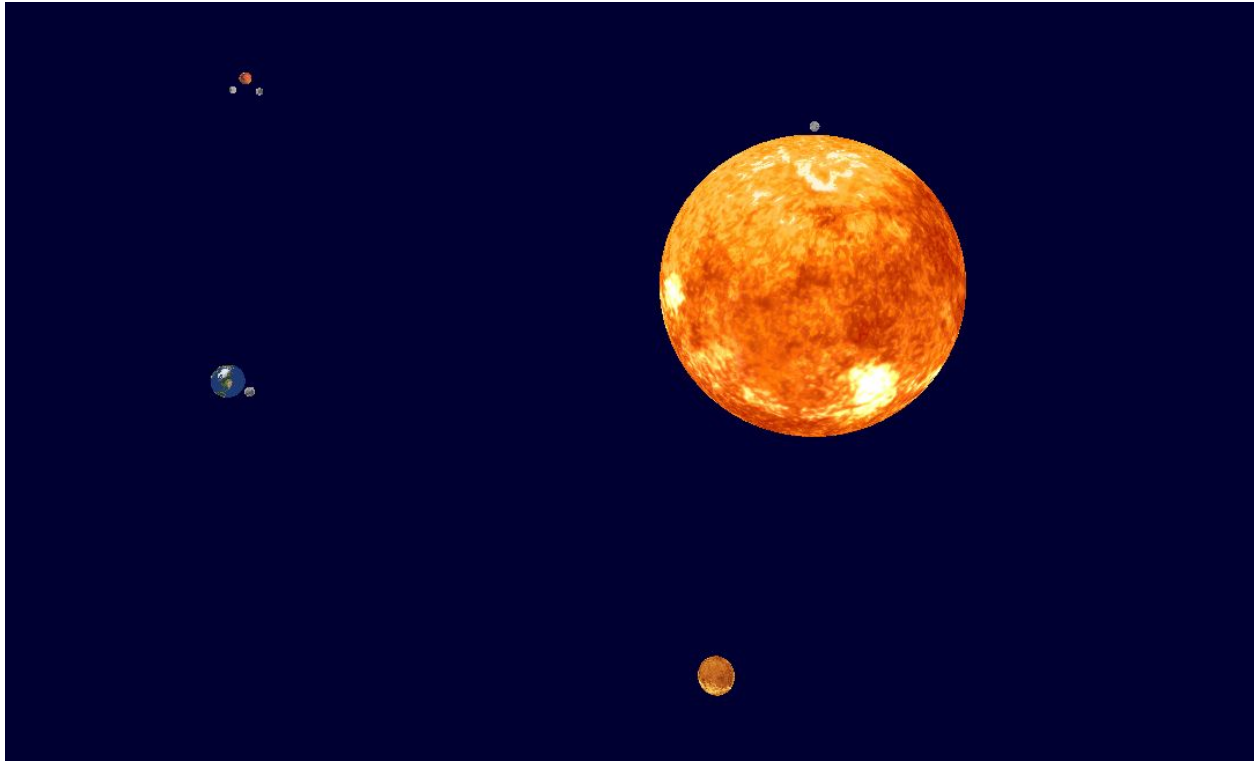
s : Pan down

d : Pan right

q : Pan in

e : Pan out

Figures



Technical Manual

Issues

The main issue we encountered while programming the assignment was being able to instance model data in GPU memory without having to load the same image every time we render it. Unfortunately, we could not figure out a way for this to work before the assignment was due. Although our project still works according to specification, it takes a few seconds to load before anything is rendered. If possible we will find a way to fix the speed issue in the future. It would remove all speed issues from our code.

Another small issue we had was getting a menuing system integrated into our codebase. We had one working before we were dealing with textures, however we could not get it to work within the time frame for this assignment. Getting menuing implemented is a future goal for this assignment and is necessary for future assignments to have a good user interface.

Things we would do differently

One thing we would do differently is copying the data for object files that have already been loaded in. This would drastically reduce load time for our program, we would do the same thing with textures as well. The textures for all of our moons are the same so it would make a big difference.

Another thing we would do is to load in a lower verticy count model for our moons. We could do this because the moons are small enough that it wouldn't affect the quality of our solar system but it would reduce load time.

Changes

- Q and E inputs for panning in and out
- Fixed existing memory leak in graphics module
- Uploaded tech manual