Name:	Paramvir Gill
Email Address:	gillp@oregonstate.edu
Course:	CS 450 Introduction to Computer Graphics
Project Name:	Final Project: Modified Solar System

Introduction:

The objective of this project is to create an animated scene based on all the learnings from this course [1].

Results and Discussion:

I will address each of the questions for the report in the following text.

1. Here is the original text from my proposal:

I really liked the idea of creating a solar system after watching the video posted by Prof. Bailey: https://media.oregonstate.edu/media/t/1_ajd6br1u, which motivated me to create it but with some slight twists. I plan on having all the planets around the sun with a light at the sun shining around all planets to create a nice lighted shading feature on the other orbiting planets. I would also like to add texture to this from the NASA website (https://nasa3d.arc.nasa.gov/images) and create distortion. For example, distortion on the sun will give it a nice look and make it seem like the flames are moving around. I plan on scaling the planets relative to one another to make it as realistic as possible. It would also be cool to put a viewpoint on the Moon or the Earth (or both but have at least one) so that all the rotating planets can be shown from a different angle (like Project 2).

I hope this is satisfactory for the full 100 points. If not, please let me know and I can look into adding more things.

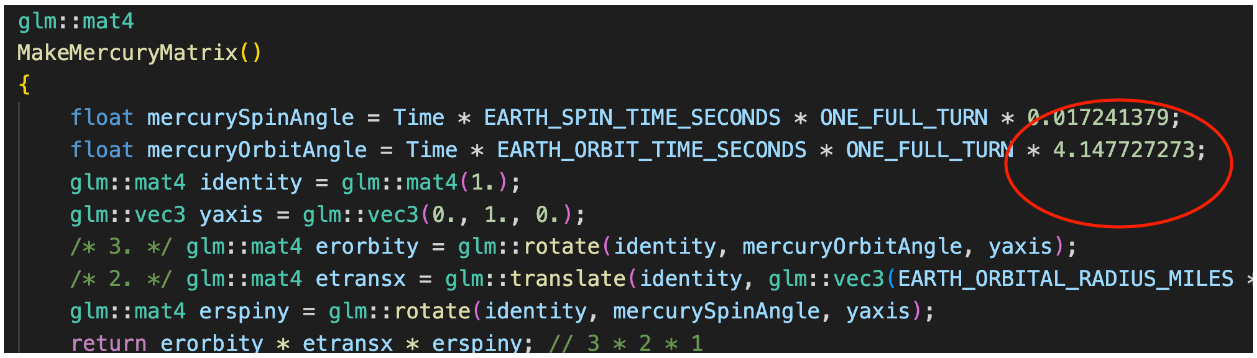
Thanks,
Paramvir (Vick) Gill

2. To demonstrate the learnings from this course, I chose to do a solar system that had texturing, distortion, lighting and different viewing options from Earth and Moon. I was able to fulfill all the requirements from my proposal. In order to get all the planets to fit on the view, skewing and a lot of research had to be done. Outside resources were used to determine things like a planet's radius, its orbital distance from the Sun etc. [2] [3]. All the planets (aside from the Sun) were modelled with respect to the Earth's parameters. The calculations for the OrbitAngle parameter (defined within the Matrix for each planet) are shown in Table 7.1.

Days	Factor to multiply OrbitAngle
88	4.147727273
225	1.622222222
365	1
687	0.531295488
4,333	0.084237249
10,759	0.033925086
30,687	0.011894287
60,190	0.00606413

Table 7.1 OrbitAngle Factors for scaling

Figure 7.1 snippet of code from sample.cpp that shows mercuryOrbitAngle being used in the MakeMercuryMatrix().



```
glm::mat4
MakeMercuryMatrix()
{
    float mercurySpinAngle = Time * EARTH_SPIN_TIME_SECONDS * ONE_FULL_TURN * 0.017241379;
    float mercuryOrbitAngle = Time * EARTH_ORBIT_TIME_SECONDS * ONE_FULL_TURN * 4.147727273;
    glm::mat4 identity = glm::mat4(1.);
    glm::vec3 yaxis = glm::vec3(0., 1., 0.);
    /* 3. */ glm::mat4 erorbity = glm::rotate(identity, mercuryOrbitAngle, yaxis);
    /* 2. */ glm::mat4 etransx = glm::translate(identity, glm::vec3(EARTH_ORBITAL_RADIUS_MILES *
    glm::mat4 erspiny = glm::rotate(identity, mercurySpinAngle, yaxis);
    return erorbity * etransx * erspiny; // 3 * 2 * 1
```

Figure 7.1 Snippet of code showing mercuryOrbitAngle parameter being multiplied by a factor with respect to Earth in MakeMercuryMatrix() within sample.cpp

Similarly, other parameters like mercurySpinAngle (as well as the orbit lines themselves) were also multiplied by a factor with respect to Earth to get the solar system to be nicely displayed, through research and a lot of trial and error.

The keyboard keys used for demonstration are as follows:

Kayboard Key	**Outcome**
2	Turns on pointlight from the Sun after it has been Textured into the display
e, E	EARTHVIEW
M, M	MOONVIEW
k, K	OUT, an outside view of the entire solar system
l, L	OUTSIDEVIEW

All the textures were taken from an outside resource [4]:
https://www.solarsystemscope.com/textures/

Please use the “HIGH RESOLUTION 2k” Download for each planet for consistent results that match the demonstration video for this project. The textures were converted from JPEG to BMP format using the following resource [5]: https://cloudconvert.com/jpeg-to-bmp

The working program is provided in the file titled “sample.cpp”. The program was compiled using the command “make” and then run using the command “./sample”.

A few screenshots of the program demonstrating the project requirements are also shown on the next pages in Figures 7.1, 7.2, 7.3 and 7.4.

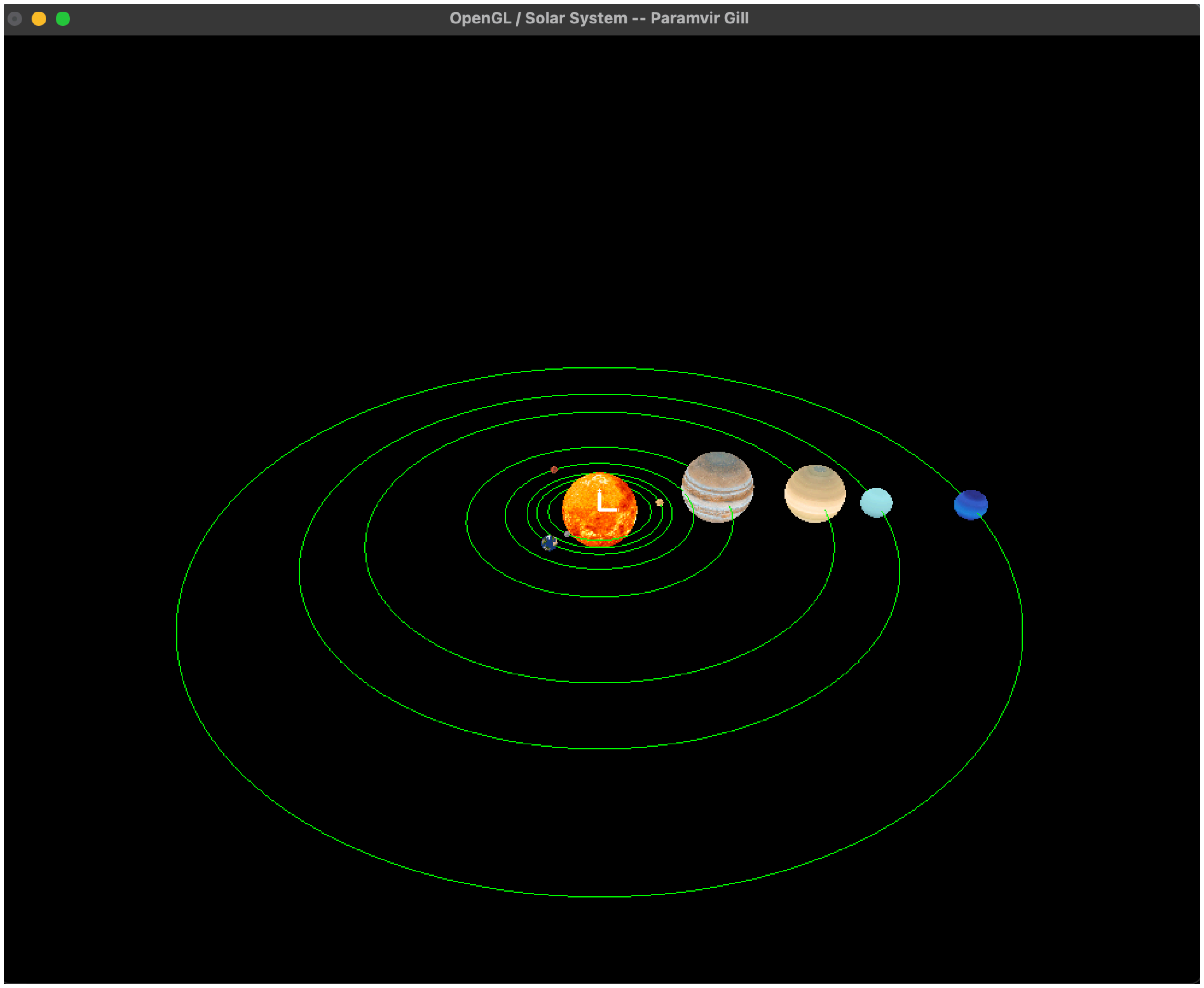


Figure 7.1 The entire solar system when the keyboard key ‘k’ or ‘K’ is pressed

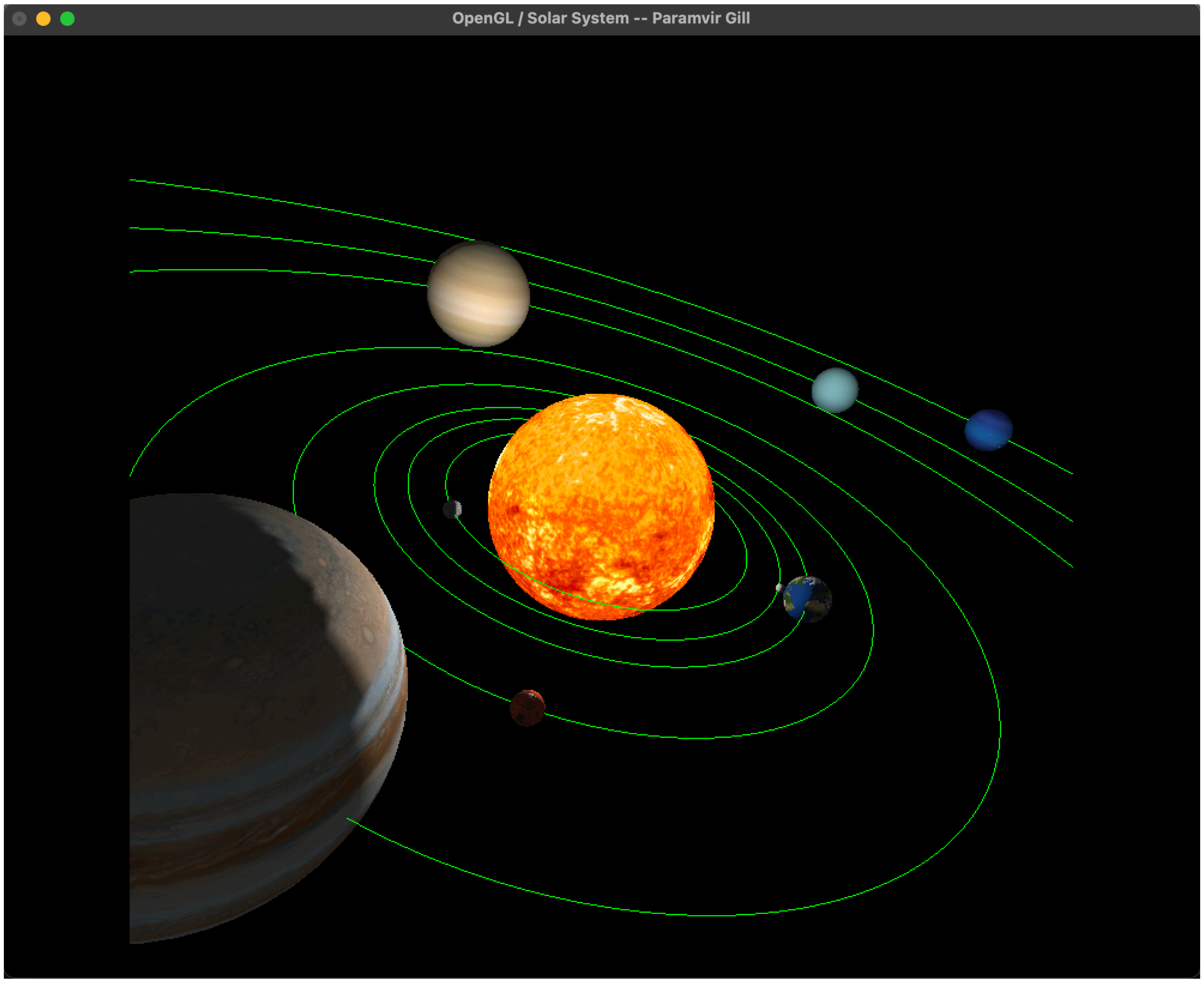


Figure 7.2 A little more inside view of the solar system showcasing the lighting feature (as can be seen visibly on Earth)

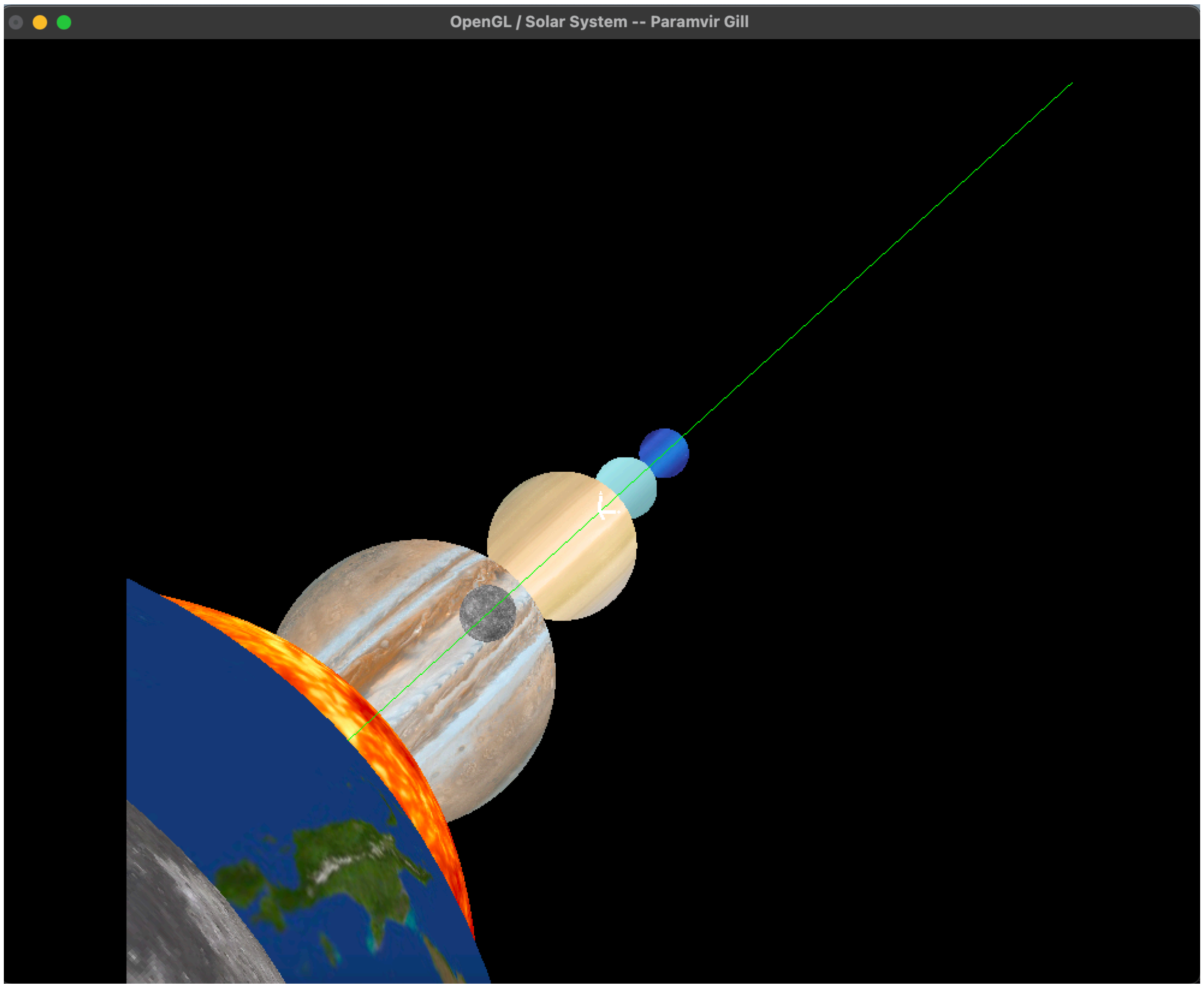


Figure 7.3 View from the Moon using the keyboard key 'm' or 'M'

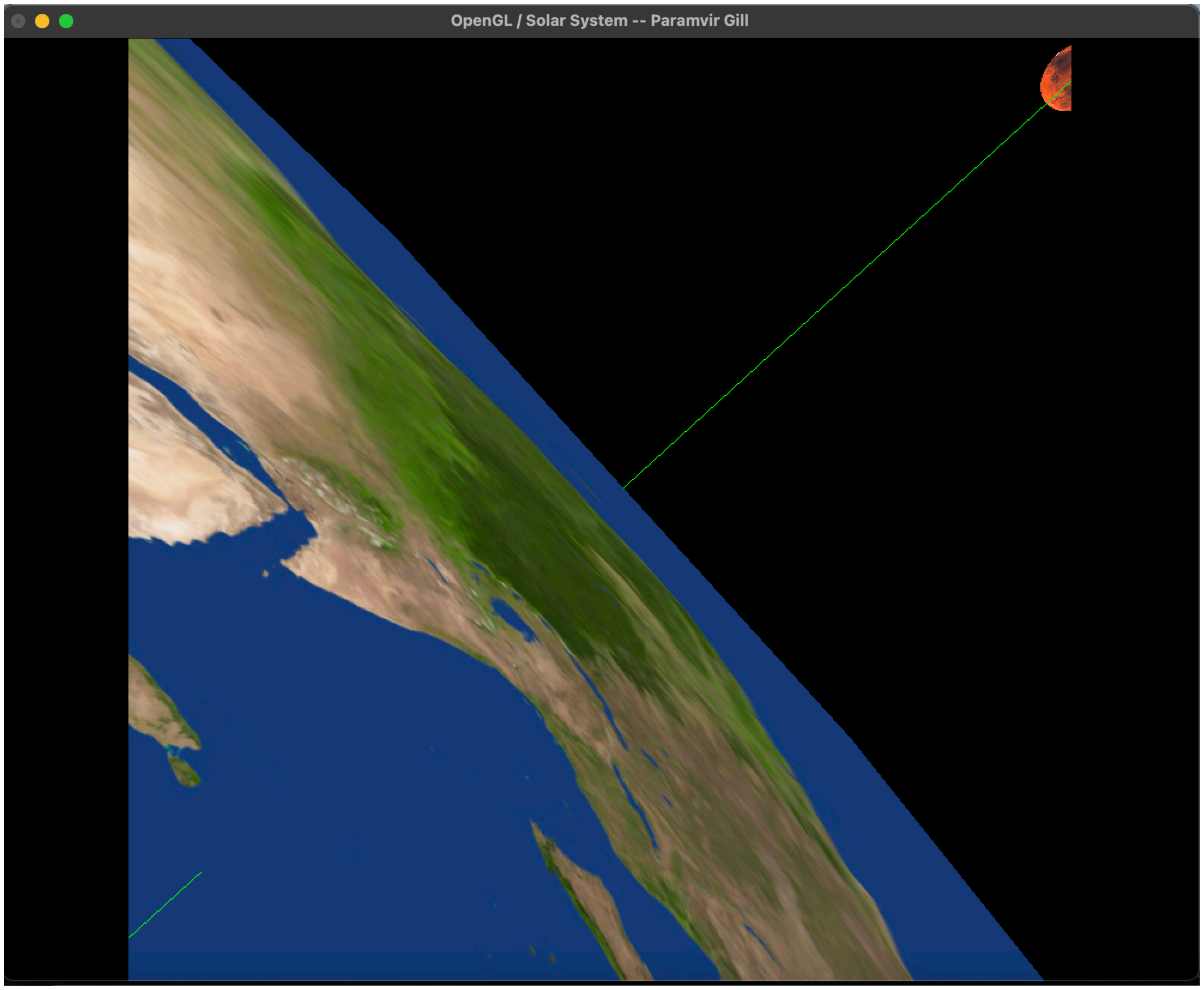


Figure 7.4 View from the Earth using the keyboard key 'e' or 'E'

3. I managed to do all the things listed in my original proposal. The only slight variation is the source I used for my textures [4]: https://www.solarsystemscope.com/textures/, instead of https://nasa3d.arc.nasa.gov/images, like I had originally proposed.

4. I liked the view I was getting from EARTHVIEW and MOONVIEW so I kept them as is and didn't want to change it to how it was shown in the sample code from the lecture notes "Putting the Eye Position on an Orbiting Body" because my views (Figures 7.3 and 7.4) show a tilted view from Moon and Earth, which was hard to get from any of the other views normally, like the OUTSIDEVIEW or OUT view (listed in Table 7.1).

Another thing I did was only allow for the pointlight to be on from the Sun when there actually is a Sun that is textured. I feel like it didn't make sense for me to allow lighting when there was no Sun that was textured. Hence why every time the view is called using "./sample", there is a

Sun that is not textured. It needs to be textured or Distorted for the pointlight to be turned on from it.

5. I learned about switch and case statements. That was new to me. They make it very easy to call-up newly implemented views by simply assigning a keyboard key, as opposed to creating an entire submenu like I did for applying distortion to the Sun. Another neat thing I learned was the application of viewpoints on moving objects, in this case, the Earth and the Moon.

6. Please refer to Figures 7.1, 7.2, 7.3 and 7.4.

7. Video showing off my project can be seen here:
https://media.oregonstate.edu/edit/1_0h5978da

Conclusion:

The given screenshot and the video provided from the given link in the Results and Discussion section cover all the requirements outlined in the project.

References:

[1] Bailey, M. (n.d.). *CS 450/550 -- Fall Quarter 2022 Final Project* CS 450/550 Final Project. Retrieved December 7, 2022, from https://web.engr.oregonstate.edu/~mjb/cs550/Projects/finalproject.html

[2] https://nssdc.gsfc.nasa.gov/planetary/factsheet/

[3] https://sos.noaa.gov/catalog/datasets/planet-rotations/#:~:text=Earth%3A%2023h%2056m%2C%201574%20km,10h%2033m%2C%2036%2C840%20km%2Fh

[4] https://www.solarsystemscope.com/textures/

[5] https://cloudconvert.com/jpeg-to-bmp