CS 530 Final Project Report

Jared Wilson and Andrea Zapata Delgado
5/5/2021

Objectives

For our Final Project in CS 530, Jared and Andrea worked together to create a model of the Solar System. We decided to work together after determining that we both wanted to do something with visualizing outer space for the final project in this course. We decided the best use case for our project would be as an educational tool for visualizing the Solar System. Our main goals in this project were:

- Create an accurately scaled model of our universe
- Allow the user to be able to see each object modelled up close
- Allow the user to see the objects move in their orbit accurately
- Allow the user to choose a specific date to visualize the solar system

We were able to accomplish each of these goals! We modelled the 8 planets, Pluto, 5 asteroids, and the sun. The user can view these objects at their original size or choose to scale them up. The user can move the focal point of the camera to any object and view it up close. The user can also see the planets move in orbit or choose a specific date.

Datasets

Our datasets are publicly available at NASA's Jet Propulsion Laboratory (JPL).

Planets: https://ssd.jpl.nasa.gov/?planet_pos
Asteroids: https://ssd.jpl.nasa.gov/?sb_elem

The planet dataset contains the required information to compute the approximate positions of planets based on Keplerian elements. The Keplerian elements vary for two main time intervals, for this project we used the Keplerian elements for 1800 AD to 2050 AD. This dataset includes the Keplerian elements for all eight planets plus Pluto in the solar system: Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, and Pluto.

The asteroid dataset also gives us the Keplerian elements to compute the asteroid orbit information. JPL distinguishes among numbered and unnumbered asteroids for their datasets. We selected the five largest asteroids by diameter: Ceres, Vesta, Pallas, Hygiea, and Interamnia, all of which belong in the numbered asteroids dataset.

Both datasets contain roughly the same information, the following table describes the meaning of each one of the values:

VALUE	DATASETs

a	Semi major axis (AU)
e	Eccentricity.
I OR i	Inclination (DEG)
L (Planets only)	Mean longitude (DEG)
w OR long.peri	Longitude of perihelion (DEG)
Node or long.node	Longitude of the ascending node (DEG)
M (Asteroids only)	Mean anomaly (DEG)

Finally, we used spheres to model all elements of our solar system. JPL also has the main attributes for physical characteristics for planets and the JPL Small-Body Database Browser has the available attributes for all asteroids.

Planets physical characteristics: https://ssd.jpl.nasa.gov/?planet_phys_par

Asteroids physical characteristics: https://ssd.jpl.nasa.gov/sbdb.cgi#top

We also included the texture files from Solar System Scope available at: https://www.solarsystemscope.com/textures/

We decided to merge all this information in four csv files:

- planets_keplerian_elements
- planets_physical_characteristics
- asteroids_keplerian_elements
- asteroids_physical_characteristics

The physical characteristics files contain additional information that we did not use, but we decided to keep them in the files for future work.

Methodology

We decided to start with the initial modelling of the planets and asteroids. Andrea was collecting and preprocessing the datasets to easily include them in VTK. Jared started modelling the spheres and giving them the desired texture. We were able to model each object with their correct radius and texture and got them evenly spaced in a straight line to verify their differences in size were correct. Jared built each sphere with the custom class MySphere which we used for project 1. Jared modified the code to receive the radius, center and texture file to be able to reuse it for all objects. Andrea made sure our code was object oriented so that the visualization tasks would work the same for planets and asteroids. To do so, she created two main classes, Planets and Asteroid that contain the required fields from the keplerian elements and physical characteristics files.

Next step was visualizing the orbits for planets and asteroids. Instead of implementing the math to transform Keplerian elements to cartesian coordinates, we used the library pytwobodyorbit which provided us with this transformation. This library let us pick an initial time and build the orbit with the keplerian elements. Once we had the orbit, we were able to get the object position on the initial time and get 1000 evenly distributed points to visualize the orbit. We set the initial time to December 17 2020, as it was the initial time available in the asteroids dataset and we needed the planets and asteroids to be in their same relative positions for the initial state. For each orbit, we used the class `vtkPoints` to represent each one of the 1000 points and `vtkCellArray` to define the topology. We also set the cartesian position on the initial date as the center for each object.

As soon as we got our orbits working, we realized that we needed to scale the planets to be able to see them at the same time that we were displaying the orbits. The easiest way to do this was multiplying their radius by a constant value. Jared soon realized that we needed to set some maximum values for the scaling. This especially became evident after including the sun which took over our entire visualization when we increased the scale a small 50 times. After some trial and error, we found that the maximum scale factors that gave us the best visualization were: 25 for the sun, 3800 for the planets in the inner solar system, 2500 for the planets in the outer solar system and 50000 for asteroids.

Our next goal was get our objects moving, pytwobodyorbit let us get the position of each object on a specific date, so Andrea implemented two ways for the user to interact with our visualization. The first one was a slider with a long range of dates that allows us to see the movement on planets and asteroids around their orbit. This feature specifically succeeds at comparing the different velocities of the objects, where we can see the inner planets completing one revolution around the sun faster than the outer planets. The second way was with a calendar input, where the user could select any date between November 17 1850 to December 31 2050 and get the position of planets and asteroids for that specific date. This feature highlights how much each object in our solar system moves compared to a time period we understand well. For example, moving the selected date by one month shows the earth moving 1/12th around the sun and also allows the user to see how much all the other objects move in that time period.

Our final step was centering the camera on different objects. We included this feature to allow the user to be able to “zoom in” on any object we modelled and see the texture which we gave each object up close. We also found that this feature allows for some very cool visuals when moving the planets because the focal point stays centered on the selected object. Specifically, the asteroids are especially interesting when moving due to their abnormal orbits.

The final decision we made was to give the orbits different colors. We did this so that each orbit would be distinguishable and easy to tell apart from the other nearby orbits.

Problems

Fortunately, we did not run into any major problems or issues during this project. Definitely the most challenging part of this project was figuring out the scope of the project at the beginning as well as figuring out how to use the keplerian elements correctly.

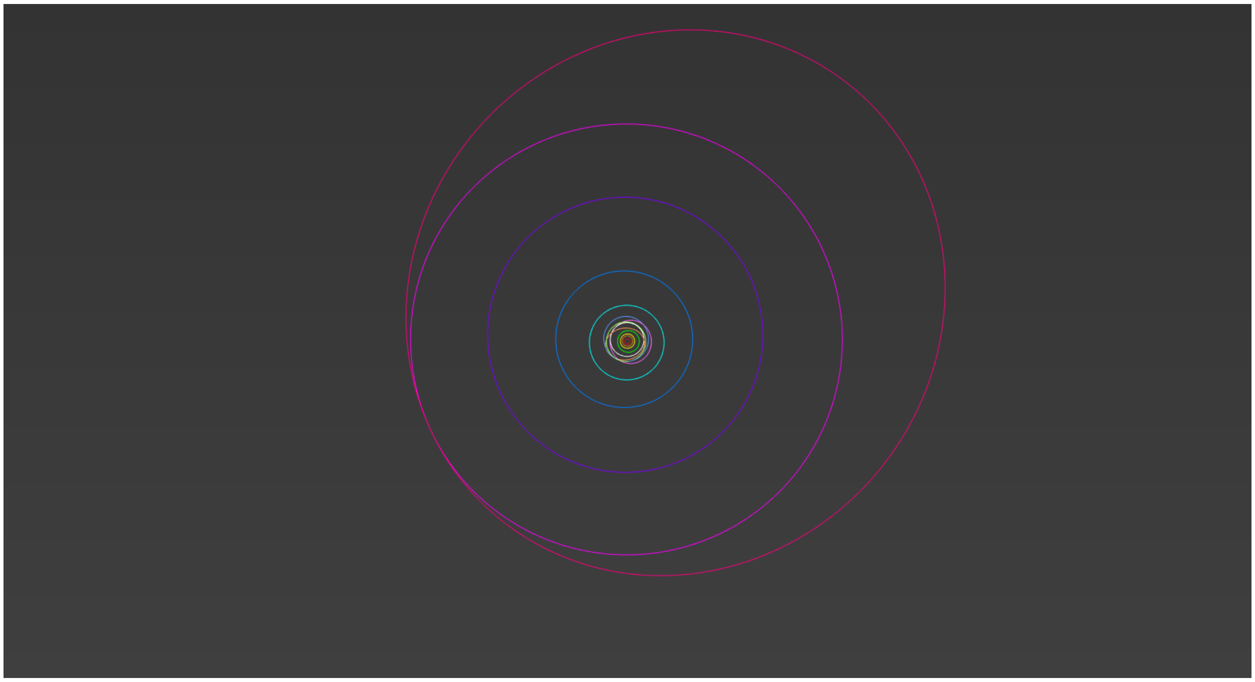
For scope, we were pretty quickly able to narrow it down to the four bullet points in the Objectives section above. We knew we wanted our model to be an interesting single model of the Solar System contained in a single program. In an even broader sense we wanted our model to be something anyone could run and spend 10 minutes or so playing around with and finding cool features. This thought process helped us get our objectives.

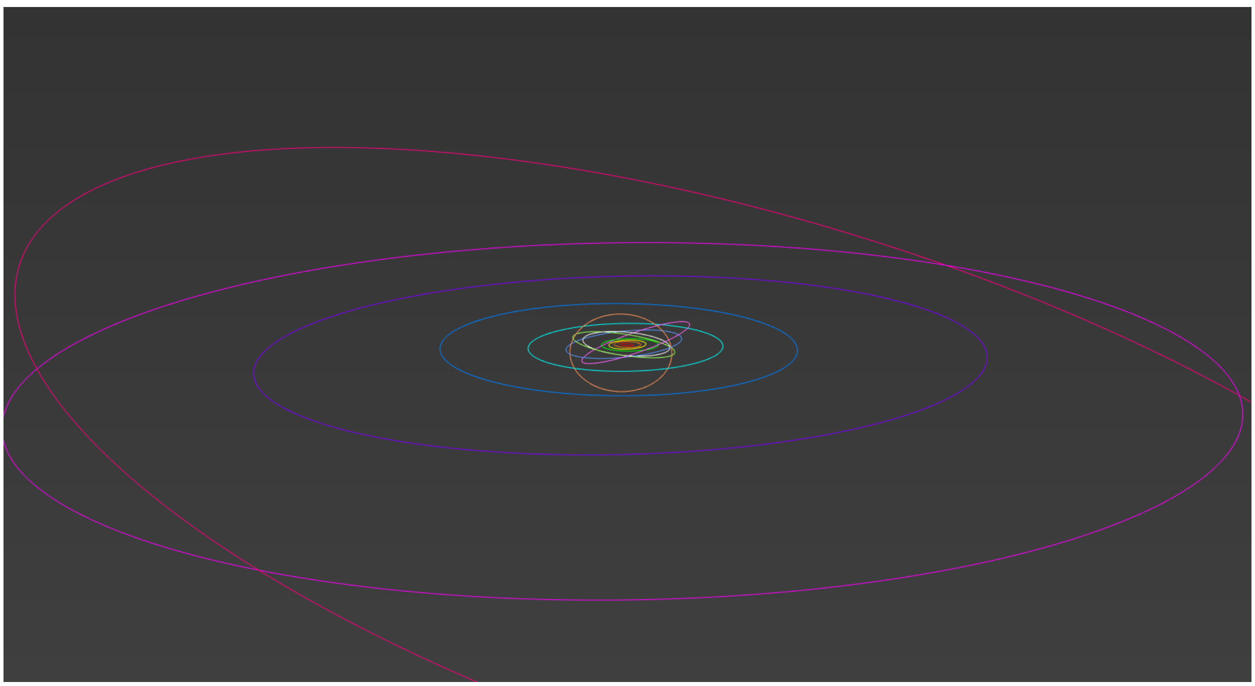
For the keplerian elements, we did lots of research about what each element means and why that element is needed to determine orbits. During our research, we found an extremely useful python module on pypi and Github called pytwobodyorbit. In this repo we found the math needed to get the orbit information from the elements written in python. This module helped us speed up the work on the project greatly due to not having to code all the math ourselves now.

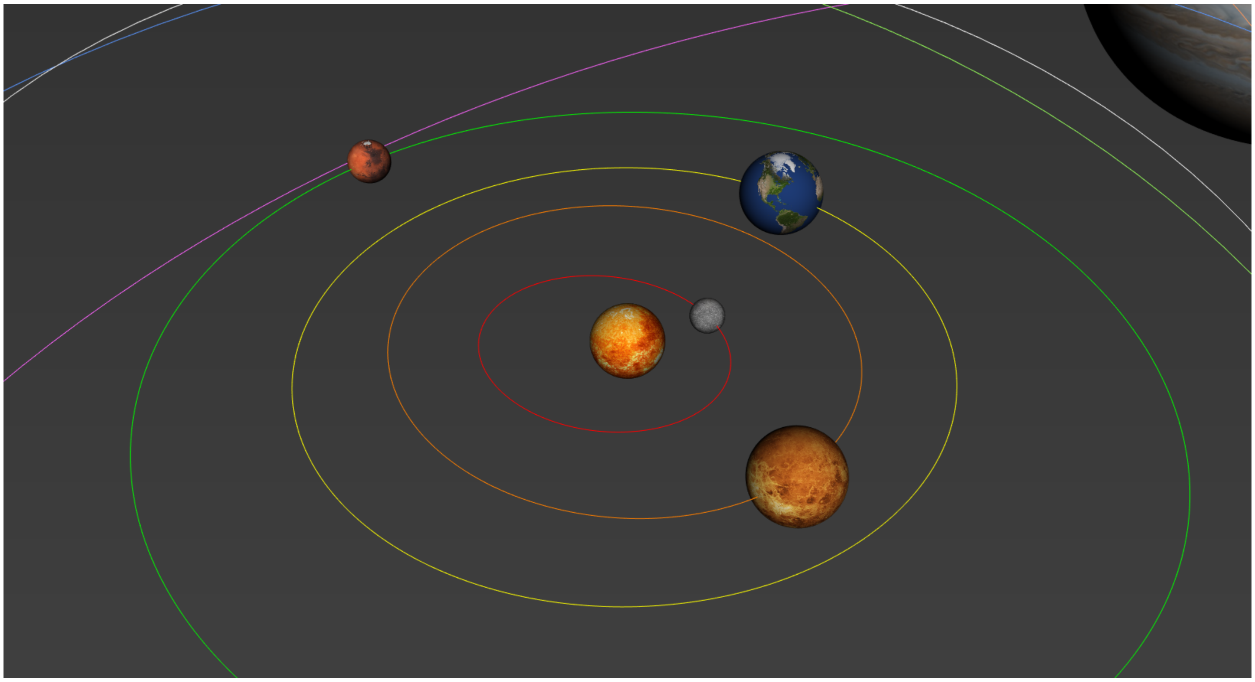
The last problem we ran into that we spent a lot of time dealing with for this project was unit conversion. We had to do lots of converting for this project to make it all work. Some datasets were in meters, kilometers, and astronomical units. We converted all of those to meters. For using the calendar date picker, we had to use Julian Date, Modern Julian Date, and seconds. While not being the most complicated issue, it was a problem we had to be diligent with.

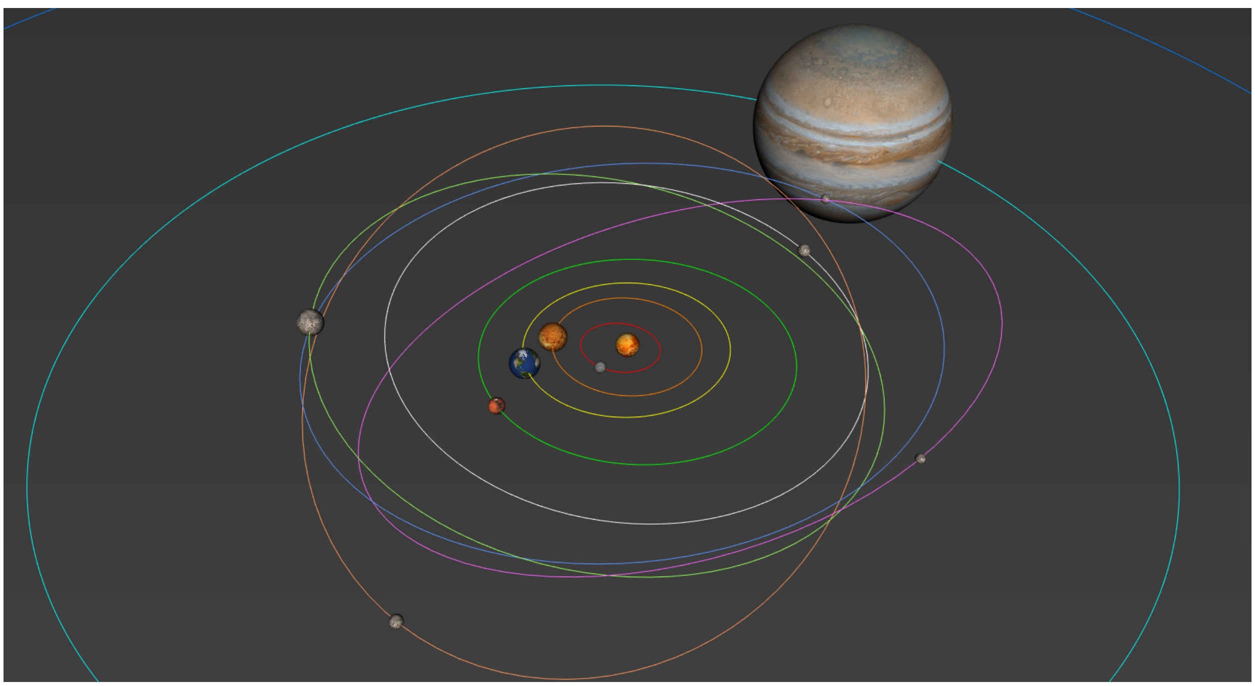
Results

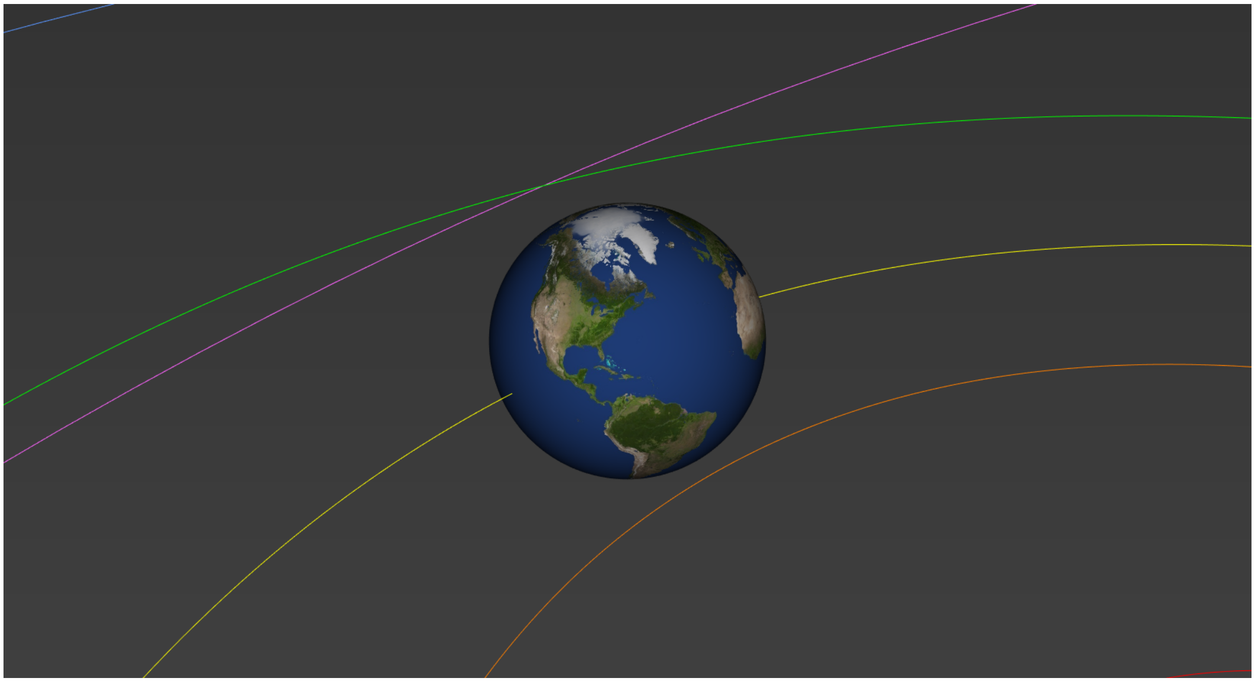
Below are some pictures of our results. While these pictures are a good representation of our results, we think the best way to experience our model is to run it yourself and try out each of the different features!

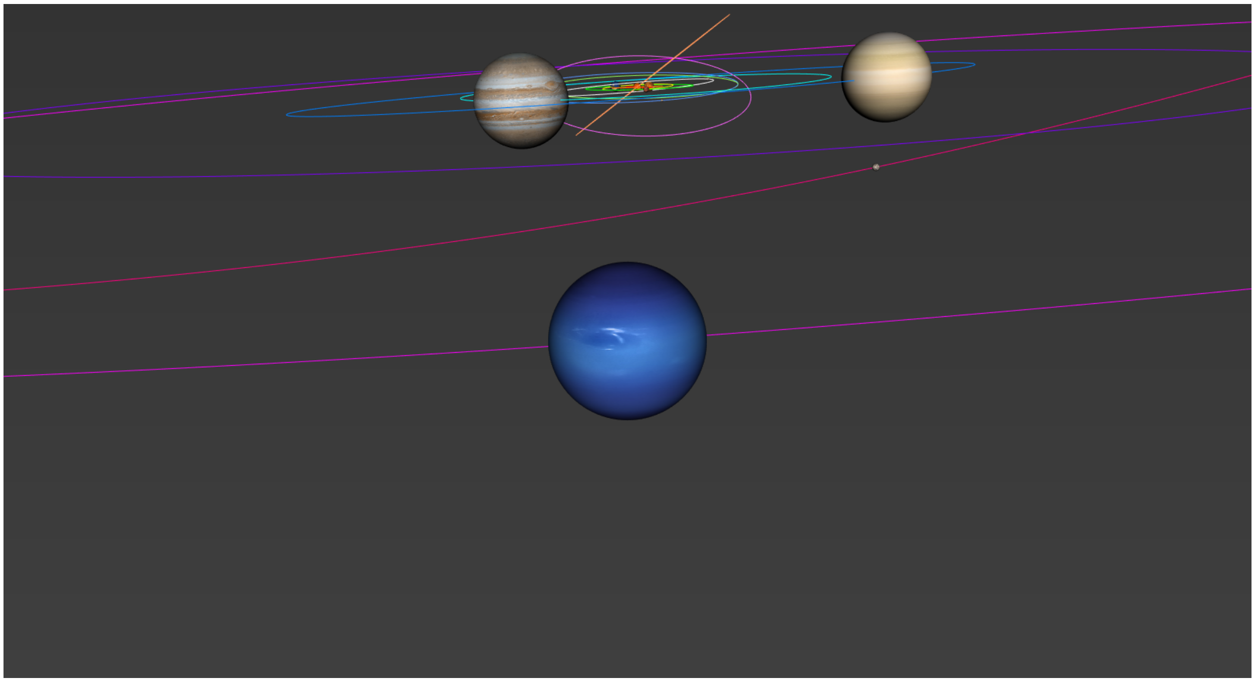


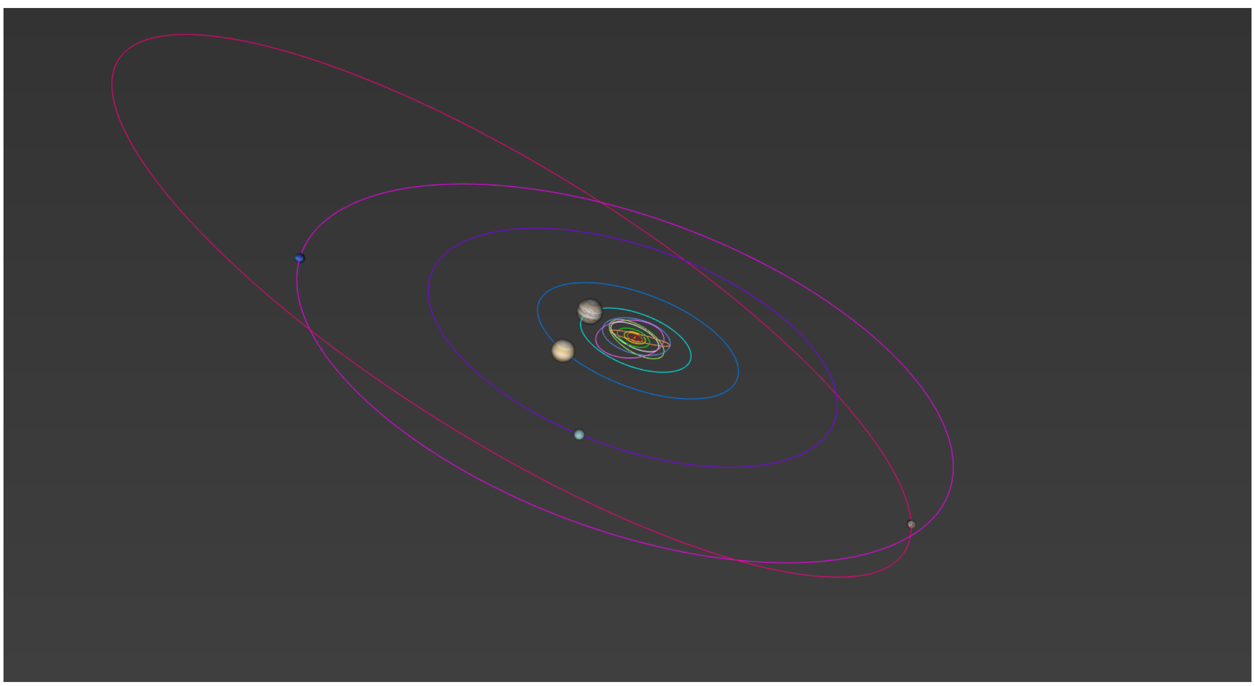












Conclusions

Our project concludes with lots of possibilities for future work. The first things we would choose to expand on would be adding more physical features to the objects. Right now they do not have their correct axial tilt or rotational period. Another cool possible feature to add would be to include some realistic lighting (this was inspired by Eman's cool use of lighting in her presentation). Overall, we are very happy with the result of our project and were very excited to be able to accomplish our goals. One of our main takeaways for the future was the importance of clearly defining the scope for our project and deciding what each member of the team needs to do. It was very important to us that our resulting visualization was very intuitive and open to everyone. Anyone from a 6 year old kid learning about the solar system for the first time to curious professors wondering about where the Earth and Mars were on the recent Mars Rover launching, can interact and enjoy it.