

Wolfie Essink

essinkw@oregonstate.edu or wolfie.essink@gmail.com

Project #4 - Keytime Animation

What I did:

First off, I started by setting up my keytime constants, global variables, and values that I would use for animation. This included initializing my keytime object Xpos in initGraphics() and adding a series of time-value pairs from the project instructions using AddTimeValue() to define how the animation would progress over time. After that, I began creating my main scene objects. I first created the Sun at the origin and set it up as a light source by enabling GL_LIGHTING and GL_LIGHT0. I gave it a warm yellow color using SetMaterial() and positioned the light directly at the Sun's center with glLightfv() so it acted as the main source of illumination, similar to my point light from the previous project, for the rest of the scene. Next, I created my Earth and Moon using OsuSphere() to draw them as smaller spheres. I positioned the Earth slightly away from the Sun using glTranslatef(), and then made it orbit around the Sun by rotating it each frame based on the current time. I then did the same for the Moon, making it orbit around the Earth. I then made the Earth and Moon spin in place, similar to a globe. After adding this realized it was hard to tell they were actually spinning, so I added vertical "rods" through their centers using scaled red cubes created with OsuCube() that can be visually seen spinning. After that, I made the Earth and Moon scale in and out over time by using a scaleFactor that smoothly increased and decreased based on a sine function, making them look like they are slowly expanding and contracting. Then I worked on the camera movement. I used trigonometric functions to make the camera orbit slightly farther out than the Earth, while looking inward toward the Sun, following the Earth's orbit. I also made the camera slowly zoom in and out, as well as bob up and down over time to create a dynamic, floating effect. Finally, I added a second camera mode that can be toggled using keyboard input, which switches to a simple top-down view positioned above the Sun, looking directly downward at the entire scene. This is mainly used to properly view the Earth and Moon scaling in size because it is hard to tell in the animated camera angle.

Keytime Animations:

Object 1, Earth:

1. Orbits around the sun
2. Spins in place (like a globe)
3. Slowly scales larger, and then back to the original size

Object 2, Moon:

1. Orbits around the Earth

2. Spins in place (like a globe)
3. Slowly scales larger, and then back to the original size

Camera, gluLookAt():

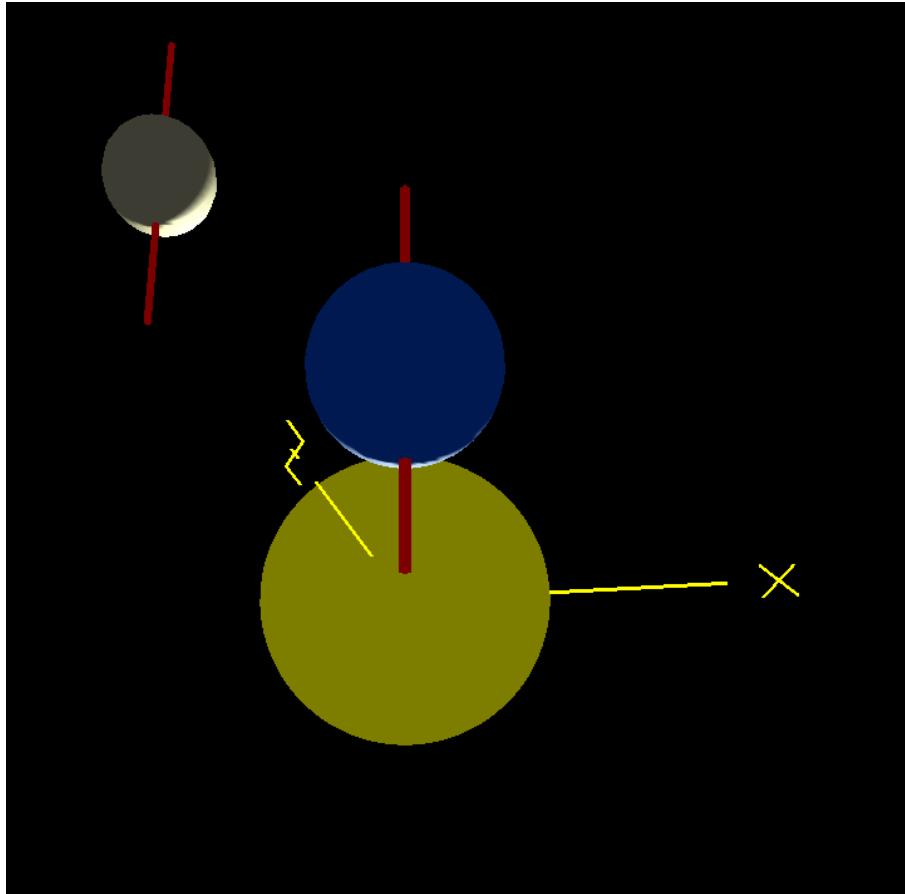
1. Orbits around the sun slightly further away than Earth, looking inward
2. Slowly move backwards, away from Earth, and then back to being closer to it
3. Slowly bobs up and down

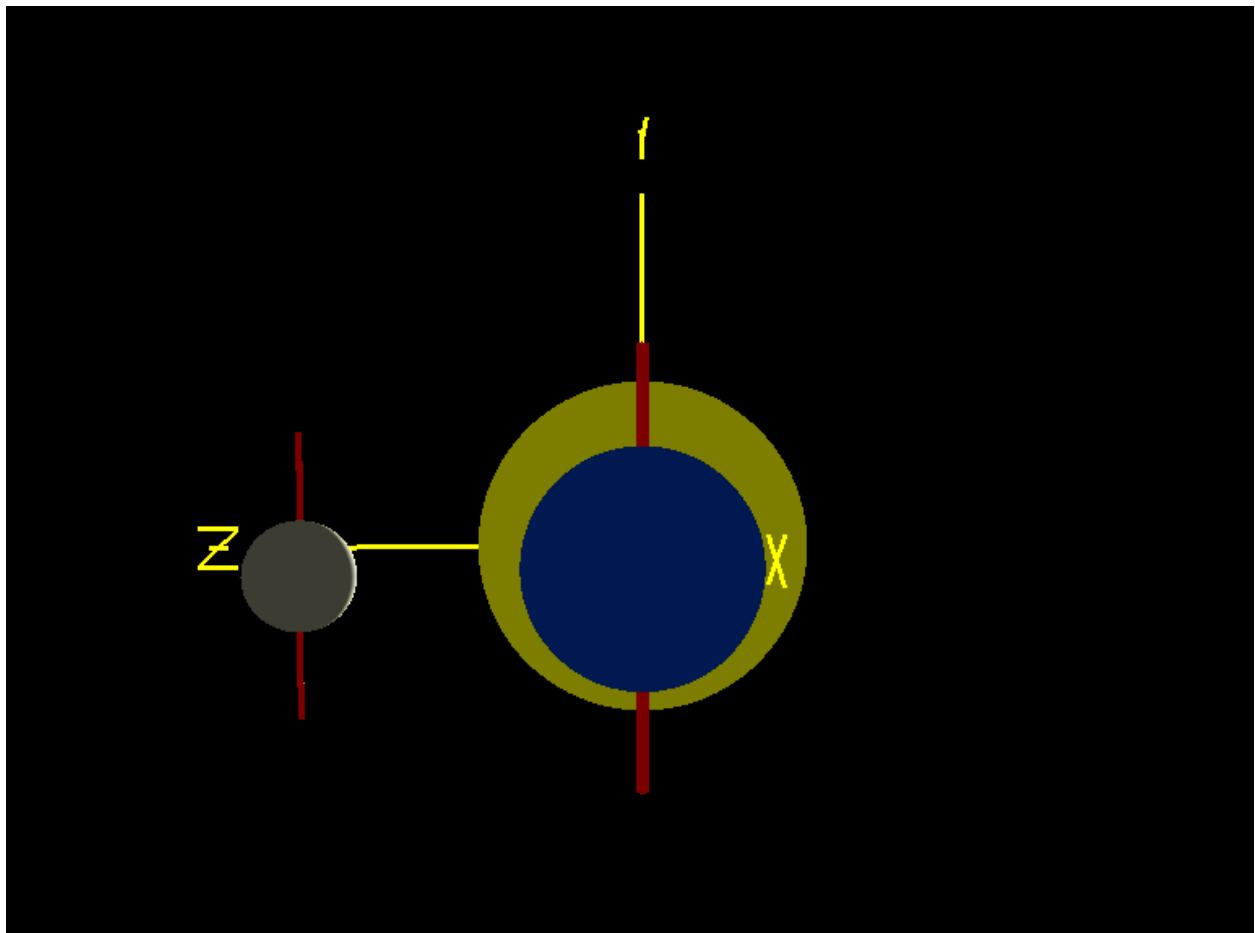
Keytime Value Table:

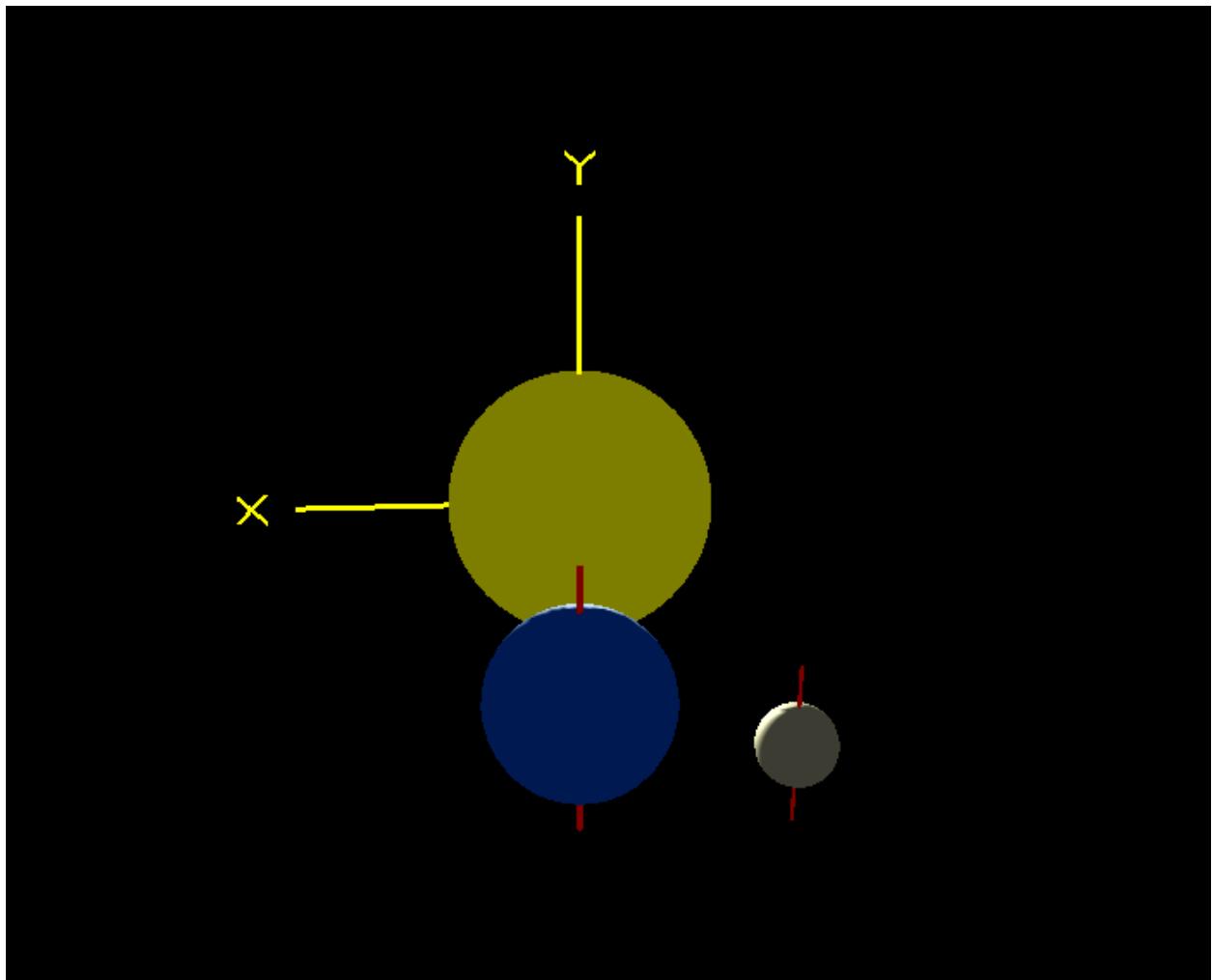
(Mine are the same as the ones in the project instructions)

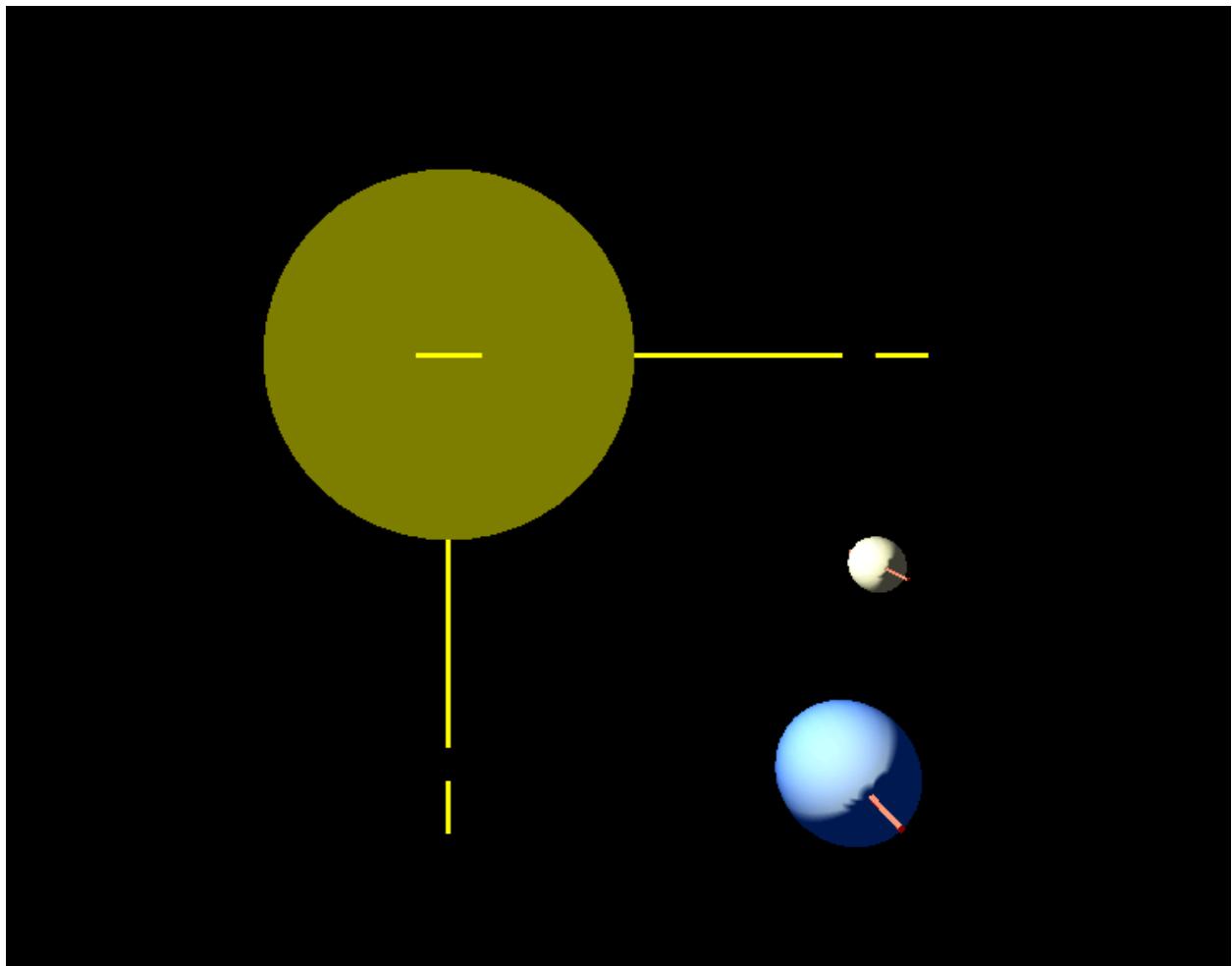
```
Xpos.Init();  
Xpos.AddTimeValue(0.0, 0.000);  
Xpos.AddTimeValue(0.5, 2.718);  
Xpos.AddTimeValue(2.0, 0.333);  
Xpos.AddTimeValue(5.0, 3.142);  
Xpos.AddTimeValue(8.0, 2.718);  
Xpos.AddTimeValue(10.0, 0.000);
```

Screenshots:









Video:

https://media.oregonstate.edu/media/t/1_n7pmpgxr