Josh Bell

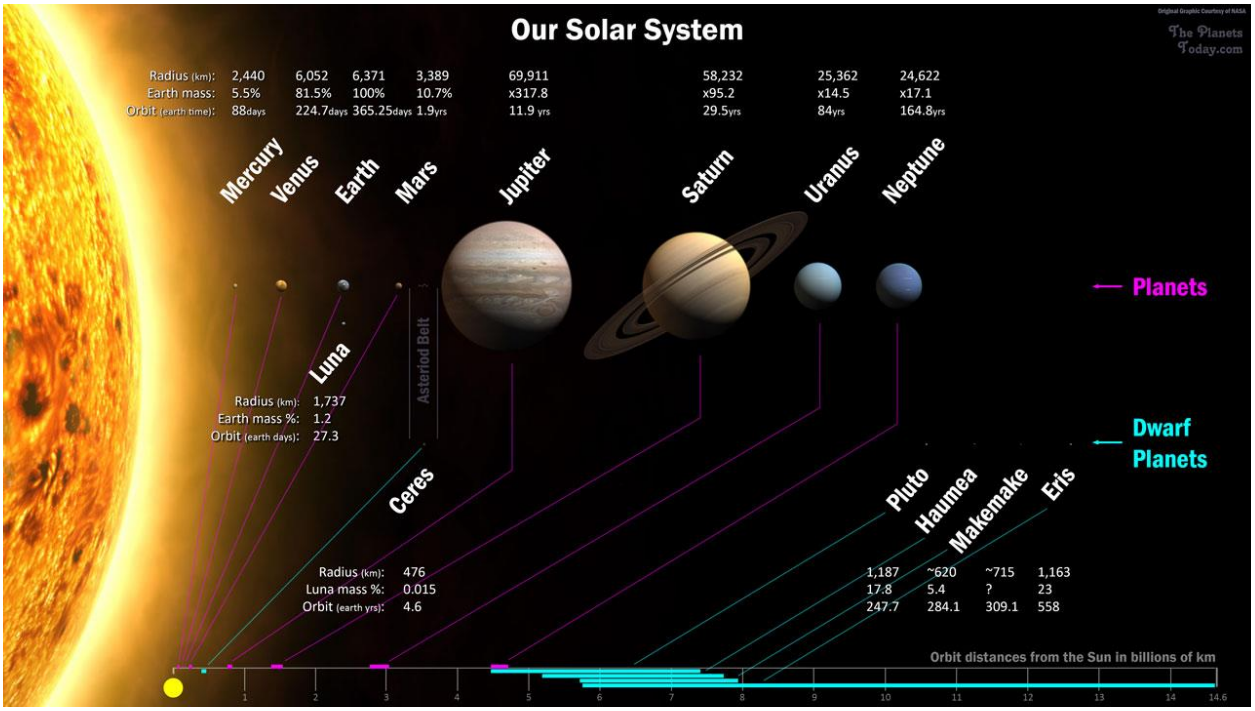
CS – 450

~~November 10th 2020~~

December 3rd 2020

Final Project Proposal

I would like to recreate our Solar System for the final project. I have reviewed the Final Project Proposal Comments *(Links 1)*, on the class website *(Links 2)*, and I have thought of some ways to implement these ideas. I wanted to implement the planets diameter and planet orbital radius by changing the exact size and putting it into my own size type *(Math 1)*. For the distance in between each planet *(Math 2)*. In terms or rotation around the sun I was planning on having each rotation being equal to one Earth year *(Math 3)*. I will be using the textures from *(Link 3)*. I will make the Sun a point light and will have ~~RGB: (255,228,132) = (1.0, 0.89, .52)~~. I made the light white so that the colors of the texture would not mess up. I added in the beginning a frozen space so you could look at the textures for the planets, case: ‘f’. Afterwards because time has elapsed the planets will automatically move as if it’s been has been moving the entire time.



Math

1. Planet Size Calculating:
   Since the sizes of the actual planets wont really work in openGL I wanted to make each 20,000 km = 1.0 in xyz coordinates, except for the sun otherwise it would be to big, the sun will be set to 10 xyz.
   ~~**Diameter:**~~ **Radius:**
   a. **Sun**: 10 xyz
   b. **Mercury**: 2,440 km = .122 xyz
   c. **Venus**: 6,052 km = .3026 xyz
   d. **Earth**: 6,371 km = .31855 xyz
   e. **Mars**: 3,390 km = .1695 xyz
   f. **Jupitar**: 69,911 km = 3.49555 xyz
   g. **Saturn**: 58,232 km = 2.9116 xyz
   h. **Uranus**: 25,362 km = 1.2681 xyz
   i. **Neptune**: 24,622 km = 1.2311 xyz

2. Distance between each planet will be 1 xyz for simplicity, for clarification the planets from the end of one planet to the beginning of another will be 1 xyz apart!

3. Planet Rotation:
   Using the Time mechanic

```
float Time;

#define MS_IN_THE_ANIMATION_CYCLE  10000

. . .

int ms = glutGet( GLUT_ELAPSED_TIME );      // milliseconds

~~ms  %=  MS_IN_THE_ANIMATION_CYCLE;~~ Commented this out so that time wont reset back to 0 after Time has reached 1.

Time = (float)ms  /  (float)MS_IN_THE_ANIMATION_CYCLE;        // [ 0., 1. )
```

I will make each second be equal to an Earth year, subject to change if its too fast or too slow will notify in the final paper. It was too fast to see some of the planets, I made it 10 times slower. Also for another clarification I had the rotation be 1 divided by the amount of seconds for each rotation, for example Mercury is (1/.2411) = 4.1477 so 4.1477*Time*360, and Neptune is (1/164.8) = 0.0061 so 0.0061*Time*360.

   a. **Mercury**: 88 days = .24110 seconds
   b. **Venus**: 224.7 days = .61562 seconds
   c. **Earth**: 365 days = 1 second

d. **Mars**: 1.9 years = 1.9 seconds
e. **Jupitar**: 11.9 years = 11.9 seconds
f. **Saturn**: 29.5 years = 29.5 seconds
g. **Uranus**: 84 years = 84 seconds
h. **Neptune**: 164.8 years = 164.8 years

Links

1. Final Project Proposal Comments: http://web.engr.oregonstate.edu/~mjb/cs550/Projects/fpcomments.html
2. Class Website: http://web.engr.oregonstate.edu/~mjb/cs550/
3. NASA Textures: Never used textures from NASA https://nasa3d.arc.nasa.gov/images
4. Solar System Scope Textures: I got high resolution jpg's from here https://www.solarsystemscope.com/textures/
5. Convertio JPG to BMP: Converted jpg to bmp here https://convertio.co/jpg-bmp/