CS-341 Solar System with Biomes per Planet Project

03/05/2023

Will try and set up the solar system environment following the assignments we have previously made. From this we need to implement the different planets.

* The shading will be different seeing as the plants are going to be hexagonal.
* The sun we can keep.
* The camera moving around will be the same.

Using mulberry 32 as seed generation algorithm. From this seed the solar system should be generated.

The planets’ self-rotation speed is now also based on spin-orbit-coupling. This means that planets that are larger (and thus has more mass) will rotate slower than the ones with a smaller mass. Similarly, planets that are further away from the star (sun), will rotate faster due to the lack of gravitational pull that the star exudes on the planet.

The planets inclination is also varying between two fixed values to generate a more vibrant solar system.

Starting on creating planets based on Hexagons. This is a hard task, as we are starting from a cube and performing chamfering to increase the spherical nature of the cube. There are methods for this easily implemented in Blender / Unity, but doing this for WebGL has proven difficult, without using the “THREE” package.

Used this GitHub for generating hexaspheres in the solar system environment. <https://github.com/arscan/hexasphere.js> Ran into a problem while integrating it. Due to where the code was placed, the meshes for the planets were all identical, meaning that we were unable to change the meshes for planets that are larger / smaller, which otherwise would be beneficial. The code was moved, which fixed the above-mentioned problem, but now it produced lag as every separate mesh is generated every frame.

The environment is now generated with the option to change the view to each of the planets.

Working further with the shading, creating our own shader written in GLSL that should work for every planet in the solar system. We wish to implement the Blinn-Phong model as it is newest rendition of the ones we have been effectively introduced to. The model is popular due to its ability to produce smooth highlights without excessive compilations, and thus works great for real-time rendering.

Next:

* Add offset to planets so they don’t start in a line when the solar system is generated.
* Moons- as part of the solar system. It will not be a difficult task to implement moons but is left out for now as to focus on more important things.

Random ideas:

* Shooting stars / asteroids (either happens at set intervals or by button on interface). Can impact nearby planets.
* Deathstar – Texture mapping and changing the rendition of a planet to the deathstar (or maybe easier to add the deathstar as a new planet / object entirely). Play some music as well?