Advanced Graphics

Assignment 1 – Using Geometry, Material and Lights to simulate our solar system.

Maximum points: 20

Due: At 11:59pm Friday February 15th 2019. (Upload your single javascript «**`assignment01.js`**» file to dropbox)

Objective for this assignment:

- To be able to create basic geometry.
- To use the built-in material.
- To use lights
- You will provide an interface to adjust the speed of rotation of the planets up to 0 (stopped)
- Do all the assigned problems on your own.

To prevent compatibility issues in marking and versioning, we will only use r100 of the three.js library

The workflow for all of the labs in this will comprise of the following:

1. Create an appropriate folder structure for VS Code.
2. Add the necessary javascript libraries to the html page
3. Code the required javascript statements to complete the lab is a separate javascript file

Tasks:

8 Marks Build a threejs application having:
nine planets
and their moons
with the sun at the center.
Your geometry and position must be relatively to scale. Absolute scale will not be appropriate because of the distances and sizes involved.
You must use the trackball controller to provide zooming, panning and dragging.
[See the url at the end of this document for information on our solar system.]

One mark for coding style

One mark for aesthetics

1 Mark Orbiting planets (nine planets including Pluto). You may assume that the paths are circular instead of elliptical. You may also assume that all the planets orbit around the same solar equator.

2 Marks Different orbit speeds for each planet.

1 Mark Different appearance for each planet. Do not use textures in this assignment; just use different materials that are configured differently.

3 Marks Orbiting moons (Earth: 1, Jupiter: 5, Saturn: 3). You may assume that all the moons of a particular planets are orbiting at the same speed but not in the same plane.

2 Marks Adjusting the speed to rotation of the planets around the sun

1 Mark Adjusting the orbiting speeds of the moons

```
https://nssdc.gsfc.nasa.gov/planetary/factsheet/
https://www.bobthealien.co.uk/solarsystem/table.htm
```

Hints:

1. Calculation the positions of the planets will be tedious. Put them in containers (either **`THREE.Object3D`**, **`THREE.Group`**, or a mesh) and rotate the containers
2. Put all the containers in a collection such as an array, so it is easier to process them
3. Each container’s speed is lined to a global value, so changing this global value will affect the entire system