Advanced Graphics

Lab 4 – Using material

Maximum points: 20

Due: At the end of the lab (demonstration only)

Objective for this lab:

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

- To be more comfortable creating a threejs application.
- To allow you to interact with your application.
- You will build a threejs application with a plane and a single geometry having all the supported lights.
- You will provide an interface to adjust 15 properties of the material that you are using.
- Do all the assigned problems on your own.

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:

5 Marks You will use your own template for a threejs application. (One with a plane, a rotating geometry (of your choice), lights and trackballControl properly initialized.)

It is strongly recommended that you use Lab3 as the starting point for this lab because the various lights are already present so you can enhance the aesthetics of your application.

Each student in a seating pod will work with a different material (`Normal`, `Toon`, `Lambert`, `Phong`, `Standard` or `Physical`) and a different geometry (`Cone`, `DodecahedronGeometry`, `IcosahedronGeometry`, `Torus`, `TorusKnot` or `Tube`)

15 Marks Using the dat.Gui library provide an interface to modify 15 properties of the material excluding the following: visibility, color, name, id,

```
https://threejs.org/docs/index.html#api/en/geometries/BoxBufferGeometry
https://threejs.org/docs/index.html#api/en/materials/MeshLambertMateria
l
```

Appendices:

Javascript code

```
//declare a variable to drive the rotation
let angle = 0;

//in the render function
scene.rotation.y = angle += 0.1;
```