Advanced Graphics

Lab 3 – Using light

Maximum points: 20

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

Objective for this lab:

- To be more comfortable creating a threejs application.
- To appreciate the convenience of using the dat.gui library to manipulate variables.
- To allow you to interact with your application.
- You will build a threejs application with a plane, sphere and cube having all the supported lights.
- You will provide an interface to disable each light.
- You will also provide an interface to change the color of each light.
- 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

In future, we will be using VS Code as our code editor. For now hosting will not be a problem because additional content will not be used. When that time come we will figure out something.

As in the previous lab, it is recommended that you add a new javascript function to setup the dat.Gui widget.

Tasks:

1 Mark

You will use your own template for a threejs application. (One with a plane and trackballControl properly initialized.)

You are not locked into my color, you may customized the colors as per your taste.

Ensure that the plane is constructed from a lambert material

```
new THREE.MeshPhongMaterial({ color: 0xeeeeee });
```

1 Mark

Add a cube about the same size and position as the screen shop on the following page. You should use `MeshLambertMaterial`. (Please note that cube is not resting on the plane).

```
new THREE.MeshLambertMaterial({ color: 0xeeeeee });
```

1 Mark Add a sphere about the same size and position as the screen shop on the following page. You should use `MeshPhongMaterial`. (Please note that the sphere is not resting on the plane).

```
new THREE.MeshPhongMaterial(
    {
        color: 0x66aa66, specular: 0x0000ff
    });
```

1 Marks Add an AmbientLight.

2 Marks Add a SpotLight.

2 Marks Add a PointLight.

2 Marks Add a DirectionalLight.

2 Marks Add a RectAreaLight.

3 Marks Add a HemiSphereLight.

Remember to configure the following when working with shadows:

renderer:

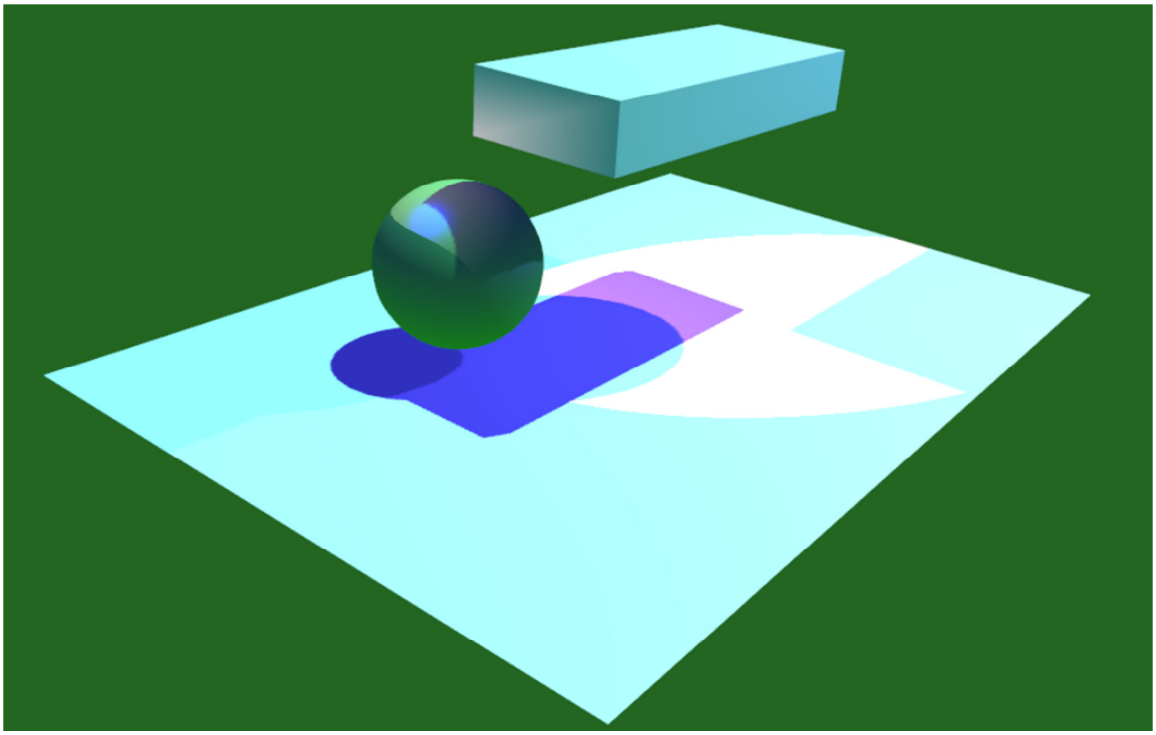
```
renderer.shadowMap.enabled = true;
```

light:

```
.castShadow = true;
```

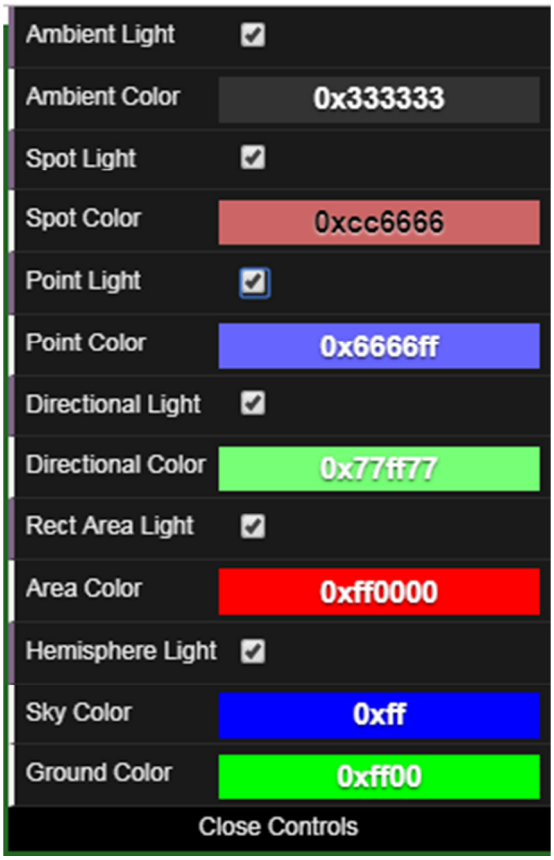
objects:

```
.castShadow = true;
.receiveShadow = true;
```



5 Marks

For each of the lights, you need to be able to turn it off and to change its color:



When assigning setting the colors in your controls object, do not use «`light`».`color`, because that will return a `THREE.Object`. Use a string or hex number to set it.

It is recommended to code this part of the solution in a single method and then call it from the `window.onload` handler.

`https://threejs.org/docs/index.html#api/en/lights/AmbientLight`

Appendices:

Html code

```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="utf-8" />
 <title>COMP392: Lab 3 - using lights</title>
 <link rel="stylesheet" href="app.css" type="text/css" />
 <script src="../libs/three.min.js"></script>         <!--or the correct path to library-->
 <script src="../libs/dat.gui.min.js"></script>       <!--or the correct path to library-->
 <script src="../libs/TrackballControls.js"></script> <!--or the correct path to library-->
 <script src="03-lab-lights.js"></script>
</head>
<body>
</body>
</html>
```

N.B. Unless instructed otherwise, this will be the structure of all of your html files.

Javascript code

```
//to add a color changer and set the display name and to listen to changes to the color value
gui.addColor(controls, 'ambientColor')
    .name('Ambient Color')
    .onChange((c) => {
    ambientLight.color = new THREE.Color(c);
});
```