

Lab 05: Ambient Lighting

CS423: Computer Graphics

Contents

1 Overview	1
2 Instructions	1
2.1 Setting the scene	2
2.2 Lighting effects	4
2.3 Your turn	4
3 Submission instructions	4

1 Overview

2 Instructions

We start with our basic HTML file, adjusted to work to bring in the appropriate Javascript file:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Example 03.01 – Ambient Light</title>
5   <script type="text/javascript" src="../libs/three.js"></script>
6
7   <script type="text/javascript" src="../libs/stats.min.js"></script>
8   <script type="text/javascript" src="../libs/dat.gui.min.js"></script>
9   <style>
10     body {
11       /* set margin to 0 and overflow to hidden, to go fullscreen */
12       margin: 0;
13       overflow: hidden;
14     }
15   </style>
16 </head>
17
18 <body>
19 <div id="Stats-output">
20 </div>
21 <!-- Div which will hold the Output -->
22 <div id="WebGL-output">
23 </div>
24 <!-- Javascript code that runs our Three.js examples -->
25 <script type="text/javascript" src="03-ambient-light.js">
26 </script>
27 </body>
28 </html>
```

Start a new file 03-ambient-light.js. Add the basic material to get things moving with THREE.js:

```
2 function init() {
```

```

4   var stats = initStats();

6   // create a scene, that will hold all our elements such as objects, cameras and lights.
   var scene = new THREE.Scene();

8   // create a camera, which defines where we're looking at.
   var camera = new THREE.PerspectiveCamera(45, window.innerWidth / window.innerHeight, 0.1,
10      1000);

12  // create a render and set the size
   var renderer = new THREE.WebGLRenderer();

14  renderer.setClearColor(new THREE.Color(0xEEEEEE, 1.0));
   renderer.setSize(window.innerWidth, window.innerHeight);
16  renderer.shadowMapEnabled = true;

18  function render() {
      stats.update();

20      // render using requestAnimationFrame
      requestAnimationFrame(render);
      renderer.render(scene, camera);
24  }

26

28  function initStats() {

      var stats = new Stats();

30      stats.setMode(0); // 0: fps, 1: ms

32      // Align top-left
      stats.domElement.style.position = 'absolute';
      stats.domElement.style.left = '0px';
      stats.domElement.style.top = '0px';

38      document.getElementById("Stats-output").appendChild(stats.domElement);

40      return stats;
   }
42 }

44 window.onload = init

```

2.1 Setting the scene

The first element to be added to the scene is a ground plane. One thing we're going to do differently here is use a material that uses Lambert reflection. Insert this code immediately after you finish configuring the renderer:

```

2   // create the ground plane
   var planeGeometry = new THREE.PlaneGeometry(60, 20, 1, 1);
   var planeMaterial = new THREE.MeshLambertMaterial({ color: 0xffffff });
4   var plane = new THREE.Mesh(planeGeometry, planeMaterial);
   plane.receiveShadow = true;

6

8   // rotate and position the plane
   plane.rotation.x = -0.5 * Math.PI;
   plane.position.x = 15;
10  plane.position.y = 0;
   plane.position.z = 0;

```

And let's add a spinning cube and spinning sphere:

```
1 // create a cube
2 var cubeGeometry = new THREE.BoxGeometry(4, 4, 4);
3 var cubeMaterial = new THREE.MeshLambertMaterial({ color: 0xff0000 });
4 var cube = new THREE.Mesh(cubeGeometry, cubeMaterial);
5 cube.castShadow = true;
6
7 // position the cube
8 cube.position.x = -4;
9 cube.position.y = 3;
10 cube.position.z = 0;
11
12 // add the cube to the scene
13 scene.add(cube);
14
15 var sphereGeometry = new THREE.SphereGeometry(4, 20, 20);
16 var sphereMaterial = new THREE.MeshLambertMaterial({ color: 0x7777ff });
17 var sphere = new THREE.Mesh(sphereGeometry, sphereMaterial);
18
19 // position the sphere
20 sphere.position.x = 20;
21 sphere.position.y = 0;
22 sphere.position.z = 2;
23 sphere.castShadow = true;
24
25 // add the sphere to the scene
26 scene.add(sphere);
27
28 // create a cube
29 var cubeGeometry = new THREE.BoxGeometry(4, 4, 4);
30 var cubeMaterial = new THREE.MeshLambertMaterial({ color: 0xff0000 });
31 var cube = new THREE.Mesh(cubeGeometry, cubeMaterial);
32 cube.castShadow = true;
33
34 // position the cube
35 cube.position.x = -4;
36 cube.position.y = 3;
37 cube.position.z = 0;
38
39 // add the cube to the scene
40 scene.add(cube);
41
42 var sphereGeometry = new THREE.SphereGeometry(4, 20, 20);
43 var sphereMaterial = new THREE.MeshLambertMaterial({ color: 0x7777ff });
44 var sphere = new THREE.Mesh(sphereGeometry, sphereMaterial);
45
46 // position the sphere
47 sphere.position.x = 20;
48 sphere.position.y = 0;
49 sphere.position.z = 2;
50 sphere.castShadow = true;
51
52 // add the sphere to the scene
53 scene.add(sphere);
```

Make the rotation happen by adding the following code snippet to the render function:

```
1 // rotate the cube around its axes
2 cube.rotation.x += 0.02;
3 cube.rotation.y += 0.02;
4 cube.rotation.z += 0.02;
5
6 // bounce the sphere up and down
7 step += 0.03;
8 sphere.position.x = 20 + ( 10 * (Math.cos(step)));
```

```
9 sphere.position.y = 2 + ( 10 * Math.abs(Math.sin(step)));
```

Save both the HTML and JS files and confirm correct operation of your code.

2.2 Lighting effects

Now we add two lights to the scene: an ambient light and a spotlight:

```
1 // add subtle ambient lighting
2 var ambiColor = "#0c0c0c";
3 var ambientLight = new THREE.AmbientLight(ambiColor);
4 scene.add(ambientLight);
5
6 // add spotlight for the shadows
7 var spotLight = new THREE.SpotLight(0xffffff);
8 spotLight.position.set(-40, 60, -10);
9 spotLight.castShadow = true;
10 scene.add(spotLight);
```

Add a `dat.GUI` controls to change the color of the ambient light and turn the spotlight on or off:

```
1 // call the render function
2 var step = 0;
3
4 var controls = new function () {
5     this.rotationSpeed = 0.02;
6     this.bouncingSpeed = 0.03;
7     this.ambientColor = ambiColor;
8     this.disableSpotlight = false;
9 };
10
11 var gui = new dat.GUI();
12 gui.addColor(controls, 'ambientColor').onChange(function (e) {
13     ambientLight.color = new THREE.Color(e);
14 });
15 gui.add(controls, 'disableSpotlight').onChange(function (e) {
16     spotLight.visible = !e;
17 });
```

Change the constants in the render function for rotation speed and bouncing speed to use the appropriate fields in the control.

2.3 Your turn

A few things for you to try:

1. Using the GUI controls, change the color being thrown by the ambient lighting. Can you explain the effect?
2. Change the material using in the ground plane to use a basic material and note the differences.

3 Submission instructions

Please create a PDF file with the following:

- A screen-shot of both your webapps displayed in the browser.
- HTML and JS files for each webapp

Attach this PDF file to the submission link in Blackboard.