Drawing a Sphere using three.js

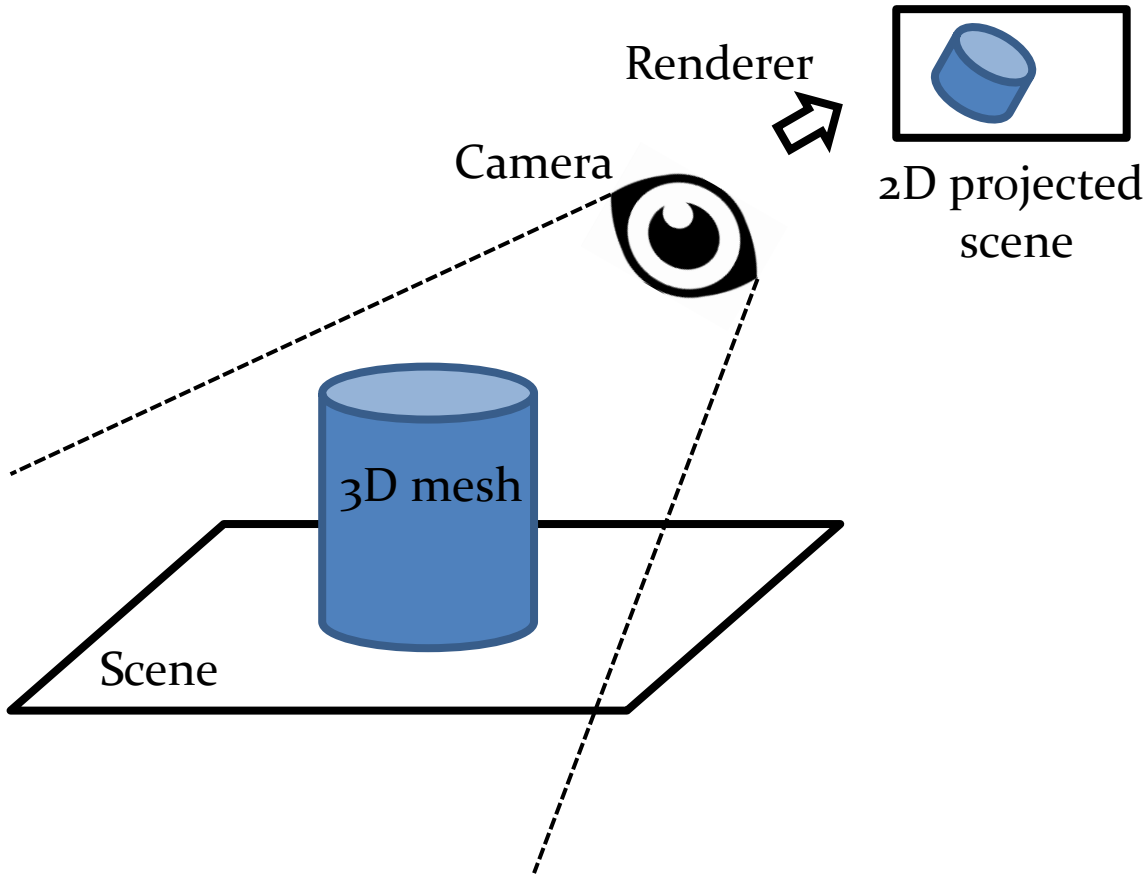
Shanfeng Hu (Lecturer)

shanfeng2.hu@northumbria.ac.uk

Department of Computer and Information Sciences
Northumbria University

three.js – Scene, Camera, Renderer

- Scene
  - Creating an environment to store all the 3D objects
- Camera
  - Adjusting the user viewpoint
  - Controlling camera's perspective
- Renderer
  - Converting the view from 3D camera into a 2D image
  - Drawing the stuff onto the canvas

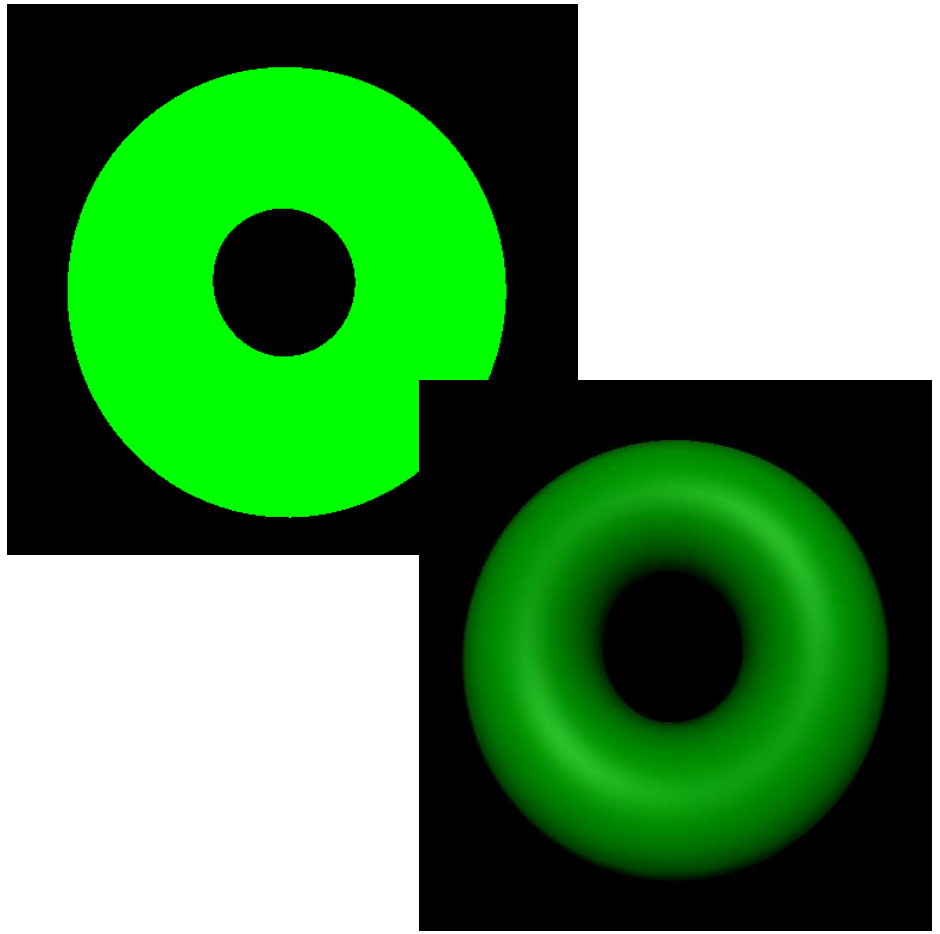


three.js – Scene, Camera, Renderer

```
var scene = new THREE.Scene();

var camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 0.1, 1000 ); // Perspective projection parameters
camera.position.x = 0;
camera.position.y = 0;
camera.position.z = 10;

var renderer = new THREE.WebGLRenderer();
renderer.setSize(window.innerWidth, window.innerHeight); // Size of the 2D projection
document.body.appendChild(renderer.domElement); // Connecting to the canvas
```

three.js – Mesh, Geometry, Material

- Mesh
  - A 3D object
  - Consisting of 1 geometry and 1 material
- Geometry
  - The 3D shape of the mesh
  - three.js supports functions for basic shapes, such as cube, sphere, torus
- Material
  - The appearance of the mesh
  - E.g. color and texture
  - three.js supports function for basic flat material, or more advance shaded material



three.js – Mesh, Geometry, Material

```
var scene = new THREE.Scene();


var camera = new THREE.PerspectiveCamera( 75, window.innerWidth /
window.innerHeight, 0.1, 1000 ); // Perspective projection parameters

camera.position.x = 0;

camera.position.y = 0;

camera.position.z = 10;


var renderer = new THREE.WebGLRenderer();

renderer.setSize(window.innerWidth, window.innerHeight); // Size of the 2D
projection

document.body.appendChild(renderer.domElement); // Connecting to the canvas


var geometry1 = new THREE.SphereGeometry(2, 18, 18); // Sphere shape geometry

var material1 = new THREE.MeshBasicMaterial( { color: 0x00ff00 } ); // Basic material
with a color

var mesh1 = new THREE.Mesh(geometry1, material1); // Link up the geometry and the
material to the mesh

scene.add( mesh1 );
```

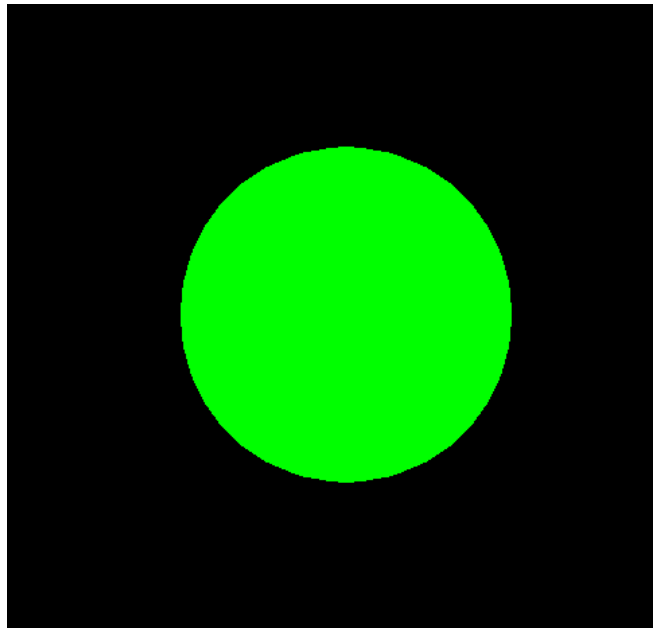
three.js – Animation Function

- A function to repeatedly draw the image frame by frame
  - Asking the renderer to draw a frame during each iteration
  - Callback function – scheduling another call of the function when the current iteration finishes

three.js – Animation Function

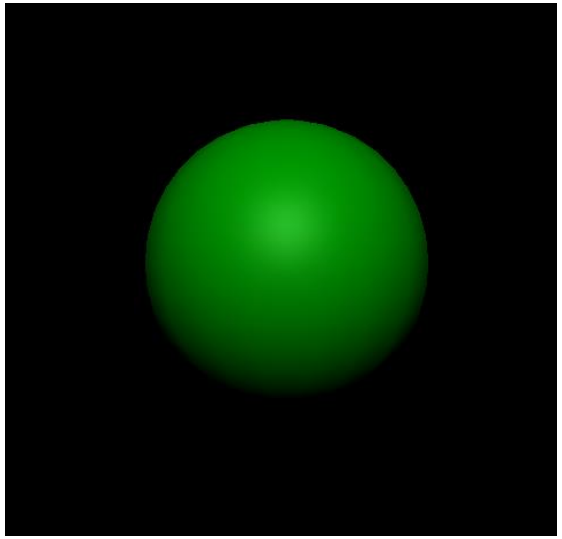
```
var scene = new THREE.Scene();


var camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 0.1, 1000 ); // Perspective projection parameters

camera.position.x = 0;

camera.position.y = 0;

camera.position.z = 10;


var renderer = new THREE.WebGLRenderer();

renderer.setSize(window.innerWidth, window.innerHeight); // Size of the 2D projection

document.body.appendChild(renderer.domElement); // Connecting to the canvas


var geometry1 = new THREE.SphereGeometry(2, 18, 18); // Sphere shape geometry

var material1 = new THREE.MeshBasicMaterial( { color: 0x00ff00 } ); // Basic material with a color

var mesh1 = new THREE.Mesh(geometry1, material1); // Link up the geometry and the material to the mesh

scene.add( mesh1 );
```

```
function animate()
{
   requestAnimationFrame(animate);
   renderer.render(scene, camera);
}
animate();
```



That is All for This Lesson!

- But if you are interested in challenges…
- Changing the material into a shaded material
- Adding a point light



```
var scene = new THREE.Scene();


var camera = new THREE.PerspectiveCamera( 75, window.innerWidth / window.innerHeight, 0.1, 1000 ); // Perspective projection parameters

camera.position.x = 0;

camera.position.y = 0;

camera.position.z = 10;


var renderer = new THREE.WebGLRenderer();

renderer.setSize(window.innerWidth, window.innerHeight); // Size of the 2D projection

document.body.appendChild(renderer.domElement); // Connecting to the canvas


var geometry1 = new THREE.SphereGeometry(2, 18, 18); // Sphere shape geometry

var material1 = new THREE.MeshPhongMaterial( { color: 0x00ff00 } ); // Advanced material with shading

var mesh1 = new THREE.Mesh(geometry1, material1); // Link up the geometry and the material to the mesh

scene.add( mesh1 );


var spotLight = new THREE.SpotLight(0xffffff);

spotLight.position.set(-0, 30, 60);

spotLight.intensity = 0.6;

scene.add(spotLight);


function animate()

{

   requestAnimationFrame(animate);

   renderer.render(scene, camera);

}

animate();
```

Conclusion

- three.js Programming
  - Scene, camera, renderer
  - Mesh, geometry, material
  - The animation function

Further Reading

- three.js Function List & Basic Tutorials
  - https://threejs.org/docs/#manual/en/introduction/Creating-a-scene

The End

Any Questions?