Three.js/WebGL

What is WebGL

- Standard for 3D graphics on the Web
- Past
  - Java Applets
    - Required JVM
  - Adobe (Flash,AIR)
    - Offered GPU hardware accelerated structure
    - Never adopted as a web standard
- OpenGL
  - Cross-platform API for 2D and 3D graphics
- WebGL
  - Derived from OpenGL

Advantages of WebGL

- Supported by all major web browsers (recent versions)
- Javascript programming
- No need to compile it
- Automatic memory management
- Easy to set up

HTML Canvas

- HTML-5 <canvas> tag provides an easy an powerful way to draw graphics with JS
- Important attributes
  - ID
  - Width
  - Height

```
<html>
<head>
<style> #mycanvas{border:1px solid red;} </style>
 </head>
<body>
<canvas id = "mycanvas" width = "100" height = "100"></canvas>
 </body>
</html>
```

Three.js

- "A *JavaScript* 3D Library which makes WebGL simpler."
- 3D World composed of
  1. Scene
  2. Rendered
  3. Camera
  4. One or multiple objects

Set up scene

```
// set the scene size
var WIDTH = 400,
  HEIGHT = 300;

// set some camera attributes
var VIEW_ANGLE = 45,
  ASPECT = WIDTH / HEIGHT,
  NEAR = 0.1,
  FAR = 10000;

// get the DOM element to attach to
var container = document.getElementById(“canvas-id");

// create a WebGL renderer, camera
// and a scene
var renderer = new THREE.WebGLRenderer();
var camera =
  new THREE.PerspectiveCamera(
    VIEW_ANGLE,
    ASPECT,
    NEAR,
    FAR);

var scene = new THREE.Scene();

// add the camera to the scene
scene.add(camera);

// the camera starts at 0,0,0
// so pull it back
camera.position.z = 300;

// start the renderer
renderer.setSize(WIDTH, HEIGHT);

// attach the render-supplied DOM element
container. appendChild(renderer.domElement);
```

Making a mesh

```
// set up the sphere vars
var radius = 50,
    segments = 16,
    rings = 16;

// create a new mesh with
// sphere geometry - we will cover
// the sphereMaterial next!
var sphere = new THREE.Mesh(

  new THREE.SphereGeometry(
    radius,
    segments,
    rings),

  sphereMaterial);

// add the sphere to the scene
scene.add(sphere);
```

Material

```
// create the sphere's material
var sphereMaterial =
  new THREE.MeshLambertMaterial(
    {
      color: 0xCC0000
    });
```

Adding Light

```
// create a point light
var pointLight =
  new THREE.PointLight(0xFFFFFF);

// set its position
pointLight.position.x = 10;
pointLight.position.y = 50;
pointLight.position.z = 130;

// add to the scene
scene.add(pointLight);
```

Finnaly

```
// draw!
renderer.render(scene, camera);
```

Three.js tutorial source:
https://aerotwist.com/static/tutorials/getting-started-with-three-js/sample.zip