Introduction to three.js

https://github.com/Toronto-WebGL-Meetup/Talk_10-December-2015

About Me

- Gareth Marland (@garethmarland)
- British
- By day, Autodesk Revit Server
- By night, Dotstorming and Boardthing

Moving from 2D to 3D

- Traditional 2D web development is like creating a shareable document
- 3D web development is like creating a shareable movie scene

Creating your scene

- The light source
- The camera
- The renderer

Lights, Camera, Action!

example1.html

The light source

- Ambient Light
  - Applies a light globally. To every object in the scene
- Point Light
  - Shines light in all directions and lights up anything that is in range
- Directional Light
  - The most commonly used lighting
  - Works like a spotlight, in a position and shines in a direction

Light source examples

- Ambient Light

```
var ambLight = new THREE.AmbientLight(0x404040);
scene.add(ambLight);
```

- Point Light

```
var pointLight = new THREE.PointLight(0x0033ff, 3, 150);
pointLight.position.set(70, 5, 70);
scene.add(pointLight);
```

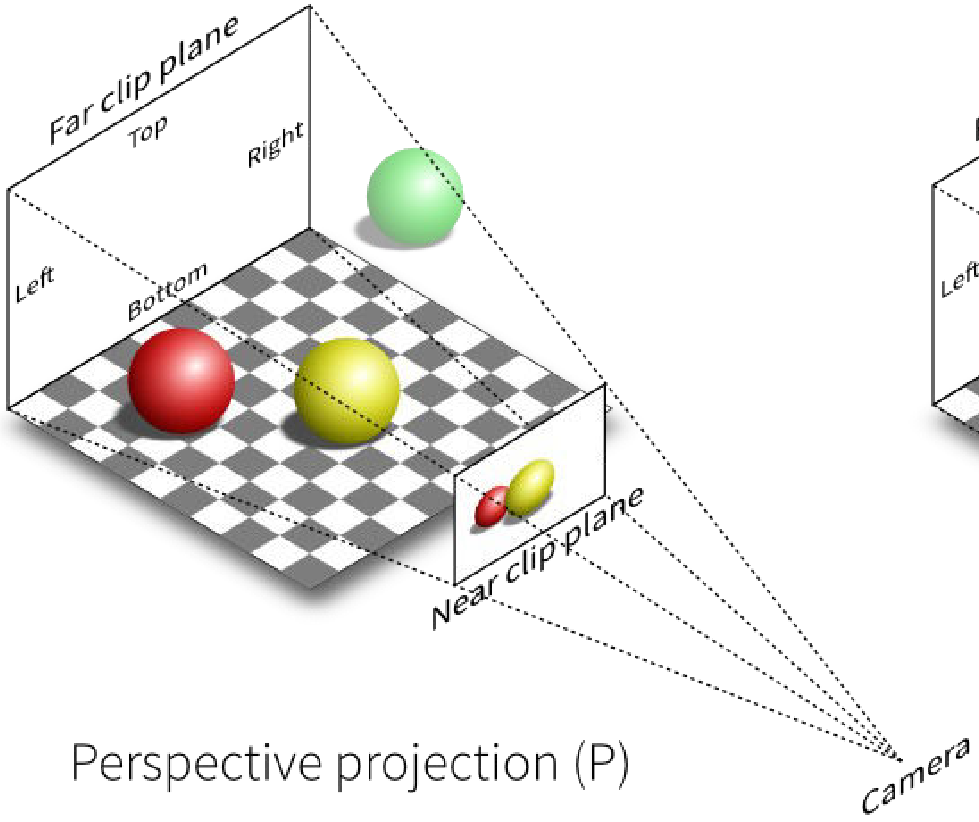
- Directional Light

```
var dirLight = new THREE.DirectionalLight(0xffffff, 1);
dirLight.position.set(100, 100, 50);
scene.add(dirLight);
```

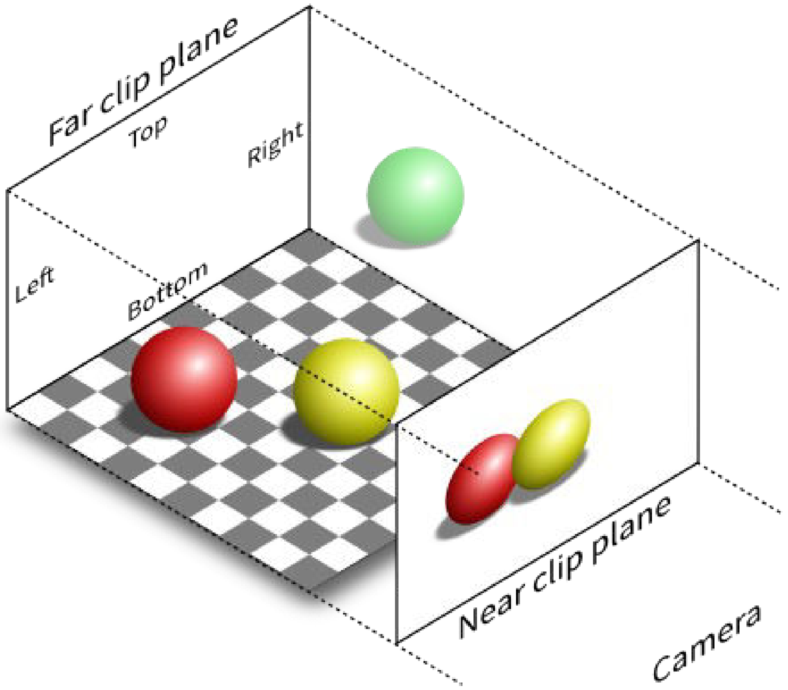
The camera

- Orthographic Camera
  - Does not take a objects distance from the camera into account when drawing
  - Created with top, bottom, left, right, near and far
- Perspective Camera
  - The most commonly used
  - Takes an objects distance from he camera into account
  - Created with fov, aspect ratio, near and far

The camera



Perspective projection (P)



Orthographic projection (O)

The renderer

- Draws the scene
- Can be set up to loop in order to animate
- Needs to be attached to an HTML element

```
var renderer = new THREE.WebGLRenderer({ antialias: true });
renderer.setSize(containerWidth, containerHeight);

document.getElementById(“container”).appendChild(renderer.domElement);
var renderScene = function () {
    requestAnimationFrame( renderScene );
    renderer.render(scene, camera);
};
renderScene()
```

Putting it together

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

var camera = new THREE.PerspectiveCamera(75, containerWidth/containerHeight, 0.1, far);
scene.add(camera);

var pointLight = new THREE.PointLight(0xffffff, 1);
pointLight.position.set(0,5,5);
scene.add(pointLight);

var renderer = new THREE.WebGLRenderer({ antialias: true });
renderer.setSize(containerWidth, containerHeight);
document.getElementById(“container”).appendChild(renderer.domElement);

var renderScene = function () {
   requestAnimationFrame( renderScene );
   renderer.render(scene, camera);
};
renderScene();
```

Creating an object

- Define the geometry
  - Use a predefined three.js object
  - Define your own vertices
  - Use a modelling tool
- Define the material
  - Basic
  - Lambert
  - Phong
  - Custom
- Create the mesh
- Add it to the scene

example2.html

Animating an object

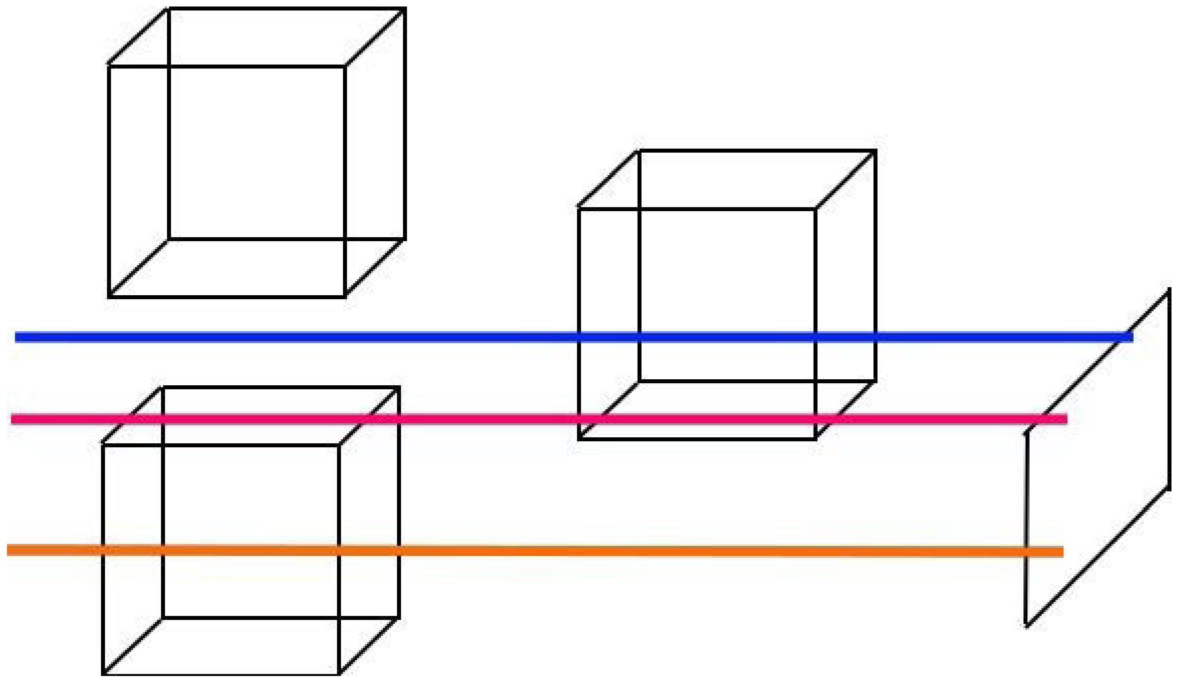
- Should be updated within the rendering method
- The rendering only needs to be called when a change needs to be made

```
var render = function () {
     var that = this;

     var updateScene = function() {
          // Do your updates to the object here
     }

     var renderScene = function () {
          requestAnimationFrame( renderScene );
          updateScene();
          renderer.render(scene, camera);
     };

     renderScene();
}
```

example3.html

Selecting objects

- Selecting needs to work differently in 3D
- We using something called “raycasting”
- It's basically shooting lazers in space and seeing what we hit!



Selecting objects example

```
var mousePosition = new THREE.Vector2();
mousePosition.x = 2 * (mouseEvent.x / renderer.domElement.clientWidth) - 1;
mousePosition.y = 1 - 2 * (mouseEvent.y / renderer.domElement.clientHeight );

var raycaster = new THREE.Raycaster();
raycaster.setFromCamera(mousePosition, camera);

var intersected = raycaster.intersectObject(scene, true);
```

example4.html

Controlling the camera

- Bind the camera to a mouse event
- Include a library such as OrbitControls.js

example5.html

Advantages of the medium

- Can be opened in a browser
- Can be mixed and matched with other web based technologies
  - Websockets
  - HTML Canvas
- Fairly easy to understand

fin