

Three.js extra meshes

Updated by Tim 17/07/2018

Adding extra meshes to your three.js scene

The Three.js Playground has great documentation on how to add a single mesh to your scene and animate it, but doesn't go into much detail around how to add multiple meshes.

This is slightly different from using the 'loop (multiply) mesh' snippet, which is duplicating a single mesh.

A three.js mesh

A three.js mesh is made up of geometry (the shape) and a material. We can see this through the line of code:

```
mesh = new THREE.Mesh(geometry, material);
```

Each of these pieces (mesh, geometry, material) needs to be defined as a variable above the setup(), you should be able to find a line similar to:

```
var geometry, material, mesh;
```

Multiple three.js meshes

In order to have an extra mesh (with different geometry and material), we need to add extra variables. These variables can have any name, though it would make sense to give them logical names that you can remember. For example, if we were adding a new mesh that was a sphere, we might define them as:

```
var sphere_geometry, sphere_material, sphere_mesh;
```

From here, the code for defining and adding this mesh to the scene is the same as usual, but you would use these new variable names in place of the usual mesh, material, and geometry.

Usual code:

```
geometry = new THREE.CubeGeometry(200, 200, 200);  
material = new THREE.MeshNormalMaterial({shading:  
THREE.FlatShading});  
mesh = new THREE.Mesh(geometry, material);  
scene.add(mesh);
```

With additions (new variables highlighted):

```
geometry = new THREE.CubeGeometry(200, 200, 200);  
sphere_geometry = new THREE.CubeGeometry(200, 200, 200);  
material = new THREE.MeshNormalMaterial({shading:  
THREE.FlatShading});  
sphere_material = new THREE.MeshNormalMaterial({shading:  
THREE.FlatShading});  
mesh = new THREE.Mesh(geometry, material);  
sphere_mesh = new THREE.Mesh(sphere_geometry, sphere_material);  
scene.add(mesh);  
scene.add(sphere_mesh);
```

Rotating individual mesh objects in a for loop

In the snippets in the three.js playground, you may notice that the snippets to transform (move, rotate, scale) deal with either one mesh, or camera or scene. It is possible to translate objects individually i.e mesh1, mesh2, mesh3. But if you have created multiple objects, you can also use a for loop to do this.

The code below, will

Add this below snippet to the function draw() section of your code.

```
for ( var i = 0; i < scene.children.length; i ++ ) {
    var meshSel = scene.children[ i ];

    // add rotations, scales, position code here. Incorporate the
    // 'i' in the code to make the movements different, as with my
    // below example.

    meshSel.rotation.x = Math.sin( Date.now() * 0.0001 ) *i;
}
```

For a simple example, see:

[//threejsplaygnd.brangerbriz.com/s/?id=6174](https://threejsplaygnd.brangerbriz.com/s/?id=6174)

For a more complex example involving nested for loops to create waves, see:

[//threejsplaygnd.brangerbriz.com/s/?id=6175](https://threejsplaygnd.brangerbriz.com/s/?id=6175)