Animation in THREE

Some details to know

The Animation Loop

```
let lastTimestamp; // undefined to start
function animate(timestamp) {
  let timeDelta = 0.001 * (lastTimestamp ? timestamp - lastTimestamp : 0);
  lastTimestamp = timestamp;

  cube.rotation.x += 0.5 * timeDelta;
  cube.rotation.y += 0.5 * timeDelta;

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

The new pieces

Update objects: (change their transformations)

```
cube.rotation.x += 0.5 * timeDelta;
cube.rotation.y += 0.5 * timeDelta;
```

Redraw:

```
renderer.render(scene, camera);
```

Animation in THREE

- it's a scene graph
- we update the objects
- we ask three to redrawn the world

Warning:

- not everything is easy to change
- hard to understand unless we know what is happening inside
- We are not talking about THREE's animation system

What is easy to animate?

Easy

Change a transformation

Change a material property (*)

Change a light property

Properties designed to be animated

- small number of numbers
- specialized mesh operations

Hard

Change points in a Mesh

Change a material

Change a light type

- Anything that requires sending large data to the **hardware**
- Anything that requites recompiling a **shader**

Transformations

Put objects in places

Make objects move by transforming them

```
    cube.rotation.x += 0.5*timeDelta;
    cube.rotation.y += 0.5*timeDelta;
```

Do not move objects by modifying vertices!

- too many vertices to change
- need to rebuild data structures
- need to send data to graphics card

Summary

1. Use animation loops with THREE
2. Update the scene and re-render
3. Only change what is easy to change
    - move objects by transformation!