CSC 240 Computer Graphics
Day 13: Three.js & WebGL

Nick Howe
Smith College
Announcements

Homework 6 is posted now, due April 8.

New TA session: Tuesdays 10-11 pm
Your Questions

Q. This is exciting!!!
A. Yes it is!!! 😊

Q. Can you explain more details about the scene object?
A. Think of the scene as a container – or an empty theater set. The main thing you do with it is add things to it.
Q. What is the renderer doing that we are not seeing?
A. Lots! All the math we’ve been studying – projection, line clipping, line drawing, area fill, etc. – is done by the renderer. Of course, we see the result, but not all the work that went into it.

Q. Is there an easy way to visualize what the camera coordinates should be and the near and far clipping plane?
A. For simple scenes, you can set near to a very small value, and far to a very large value – say 0.1 and 1000. The trickier choice is the field of view and camera position, which trade off in unintuitive ways.

Q. Does anything change if we import three.js instead of three.min.js?
A. No. They are functionally equivalent, except for load time.
Field of View

\[
\text{fov} = 30; \ z = 120
\]

\[
\text{fov} = 60; \ z = 60
\]

\[
\text{fov} = 30; \ z = 60
\]

\[
\text{fov} = 30; \ z = 120
\]
Your Questions

Q. Why material and geometry I found don't have THREE. before the name of the class?
A. The documentation may not show it, but you need it in the code.

Q. I'm not sure where to find the Three.js class in the documentation.
A. When you load three.min.js, you get the object THREE. This contains all the classes documented on the Three.js web site.

Q. I don't quite get what class is supposed to be for these functions in particular, as they the types of camera/geometry/... there are? It's also very confusing to navigate through three.js
A. Right now it is all very new. We’ll be learning by example to start.
Q. I thought lights illuminate only objects with complementary colors? For instance if you shine a blue light on several objects, the blue objects tend to disappear into the background but a yellow/orangeish object would show up really dark?

A. Technically, you are correct that illumination is done by component. White light contains all three RGB components, so lights everything. A red light will only light objects that have red in their surface color, green only lights green, and blue only lights blue. So a yellow surface (made of red and green components) will look yellow in white light, red in a red light, green in a green light, and black in a blue light.
Q. Are there any shortcuts or tricks for making complex objects? Specifying all the vertices and faces sounds hard for making something like a dinosaur.

A. You’re absolutely right! We’ll only create simple objects by hand.
   • Later in the semester we’ll study Blender, which is a tool for creating more complex 3D models.
   • You can also get a 3D model by laser-scanning a physical object.
Q. Is using { } a different way to define an array than [ ]?
A. { } can be used to initialize arrays all at once.
   [ ] is used to index individual elements

Q. Does the order of the calls matter? Not necessarily like defining the scene after the cube, but can you define the lighting and then render the cube?
A. Nothing is drawn until you call renderer.render(). So you can add lights, objects, etc. in any order so long as it is all done before you try to render the scene.
Q. Where do we decide the position of an object in the code?

A. Great question!
All objects that go in the scene are subtypes of `Object3D` which has a field called `position` which has three fields called `x`, `y`, and `z`. It also has a field called `rotation` which has three fields called `x`, `y`, and `z`. So to change the position of `myObject`, do one of these:

```javascript
myObject.position.x = 10;
myObject.position.y = -2;
myObject.position.z = 5;
myObject.position.set(8,8,8);
```
Q. How to put different colors on the same geometry (ex. gradient color)?

A. The color is a property of the material, not the geometry. You can apply a single material to the whole geometry, or different materials to each face.

Q. How do the geometries interact when they have overlapping faces?

A. Unlike physical objects, the faces in virtual geometries can intersect. The renderer will draw whichever face is closest to the camera at any point.
Your Questions

Q. Can you explain the code for create triangular faces?
A. We’ll practice this in lab today.
   • To create a 3D point: new THREE.Vector3(1, 1, -1)
     (Arguments are coordinate components)
   • Geometry will require an array of such points (all the vertices of the shape)
   • To create a face: new THREE.Face3(0, 1, 2)
     (Arguments are index into vertex array)
   • Geometry will require an array of such faces

Q. Can you go over the right hand rule?
A. I’ll try. You can also just experiment. 😊
Unlike real-world objects, in Three.js, faces point only in one direction. Reverse side is totally transparent! Actually, you won’t even see the outline.

Visible side is determined by **right hand rule**. 

Right hand rule: Fingers curl in order vertices are listed. Thumb side is exterior.

Looking at the side you want visible, list the points **counterclockwise**.

A-B-C points out of screen (also B-C-A or C-A-B)

A-C-B points into screen (also C-B-A or B-A-C)
Lab 6

A custom 3D object!