

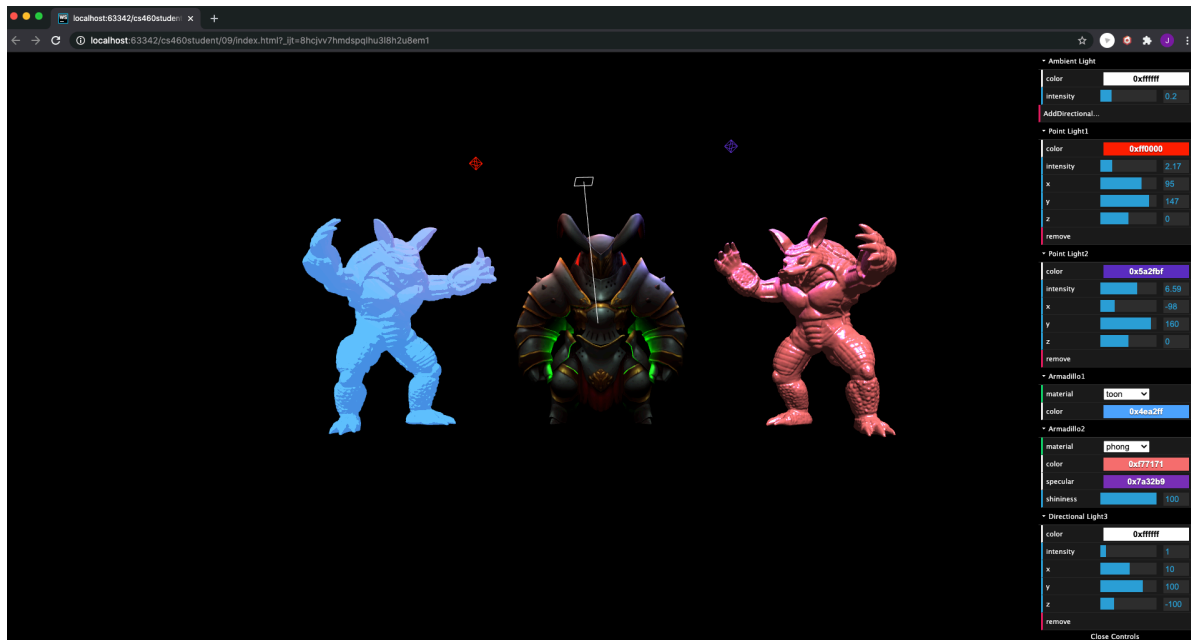
CS460 Fall 2020

Github Username: JamesEdMichaud

Due Date: 11/30/2020

Assignment 9: Geometry, Materials, and Lighting!

We will load our favorite mesh from a file, try out different materials, and play around with light settings.



Starter code for assignment 9. After pulling from upstream, there is the folder 09 in your fork. If you run a webserver and access the file, you will see a sad single armadillo in the scene.

Part 1 (14 points): The armadillo needs a friend! Please load a second mesh from a file using a THREE.js loader. This could be any mesh you find online in any format THREE.js supports - or you could load the armadillo again. Please modify the positions so that the meshes do not overlap.

This was really hard! I tried .obj, .fbx, .tga, and .ply. I eventually found a mesh/texture that I could load successfully. I found a .fbx file on Sketchfab.com, but was unable to get that working correctly. I used Sketchfab's auto-convert feature to turn it into a glTF file.

At first there were some clipping issues (some parts of the texture were being clipped, with parts from the back showing through the front), but using 'material.depthWrite = true;' seemed to fix that. Then I had trouble getting the texture to react to light sources, and learned that that had to do with an extension that was baked into the design of the mesh file (KHR_materials_unlit). Removing that caused the material to be made as MeshStandardMaterial instead of Basic (which doesn't react to light), solving that issue.

Using a texture prevents me from completing the extra credit, so I've added a second armadillo for those parts.

Part 2 (15 points): Please configure the second mesh from above with a different material of your choice (not MeshToonMaterial again!).

I used the texture that came with the mesh I found. As mentioned above, the default material was MeshBasicMaterial, but I was able to change it to MeshStandardMaterial. Each armadillo is assigned a random material when created.

Part 3 (10 points): Please add two point light sources to the scene.

I added a red light inside the knight's head and a green light in the torso. The result is a somewhat subtle glow coming out of the knight. I set the intensity very high with a rather short decay distance to keep the lighting local to the knight. I added two additional point lights that will be controlled in part 4.

Part 4 (15 points): The starter code includes the following snippet to control the color and position of the directional light.

```
var directionalFolder = gui.addFolder('Directional Light');
directionalFolder.addColor(controller, 'color').onChange( function(value) {
  directionalLight.color.setHex(value);
});
directionalFolder.add(directionalLight.position, 'x', -100, 100);
directionalFolder.add(directionalLight.position, 'y', -100, 100);
directionalFolder.add(directionalLight.position, 'z', -100, 100);
directionalFolder.open();
```

Please setup dat.GUI to control position and color of the two point lights with similar code.

I made function to do gui folder creation for lights, so this was a simple matter of passing the light and folder name to the function. I also included helpers for each of the lights for added visibility.

Part 5 (15 points): Please setup dat.GUI to control the color of both materials.

Since I included MeshNormalMaterial as one of the material options, I added some functionality to add or remove options based on the current material of each armadillo.

Part 6 (30 points): Please play around with the lights and try to understand why the toon material seems to work *sometimes*. What are your observations?

I assume this refers to how the material can look washed out, or seem to have no shading.

I'm not 100% certain, but I believe the toon material works by quantizing the color (shading) of the material into discrete steps. When the light intensity is too great, there is not enough variation in the color to produce discrete steps with any significant detail, resulting in a uniform color over the whole mesh.

Part 9 (1 points): Please update the screenshot above with your own and then post the github pages url here:

<https://jamesedmichaud.github.io>

Bonus (33 points):

Part 1 (11 points): Please add dat.GUI elements that allow to switch the material for the two meshes. Here is an example of a combobox in dat.GUI:

```
// Choose from accepted values
gui.add(controller, 'material', [ 'toon', 'standard', 'phong' ] ).onChange( function(value) {

  if (value == 'phong') {
    // TODO
  }

});
```

I used an array of functions which generate materials based on the selected choice.

Part 2 (22 points): Please make adding lights to the scene dynamic: Add dat.GUI buttons to add new directional lights that then also add a dat.GUI folder to the menu that allows to control (color and position), and remove the light. There's an 'AddDirectionalLight' button near the top of the gui, and each light has a remove button.