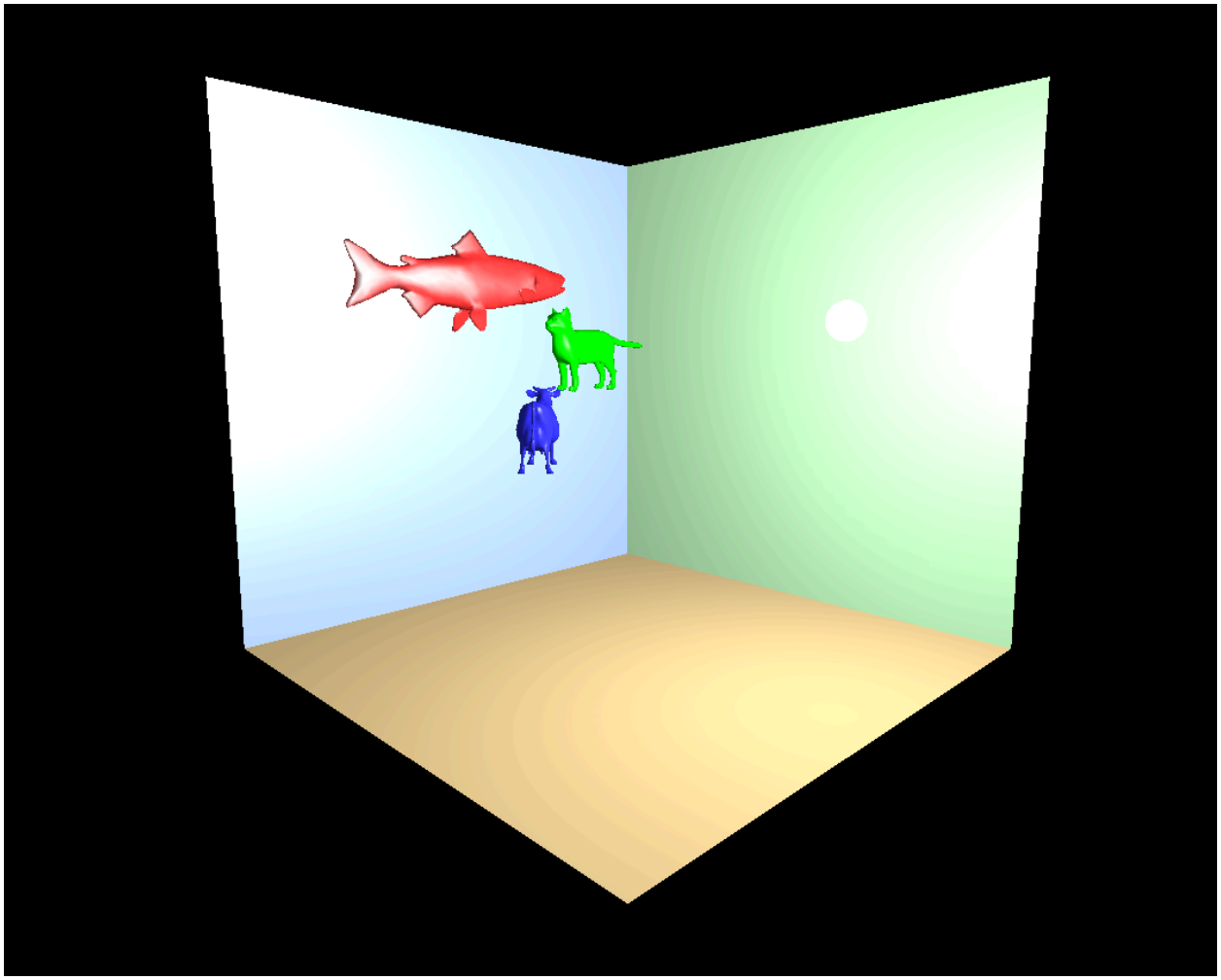
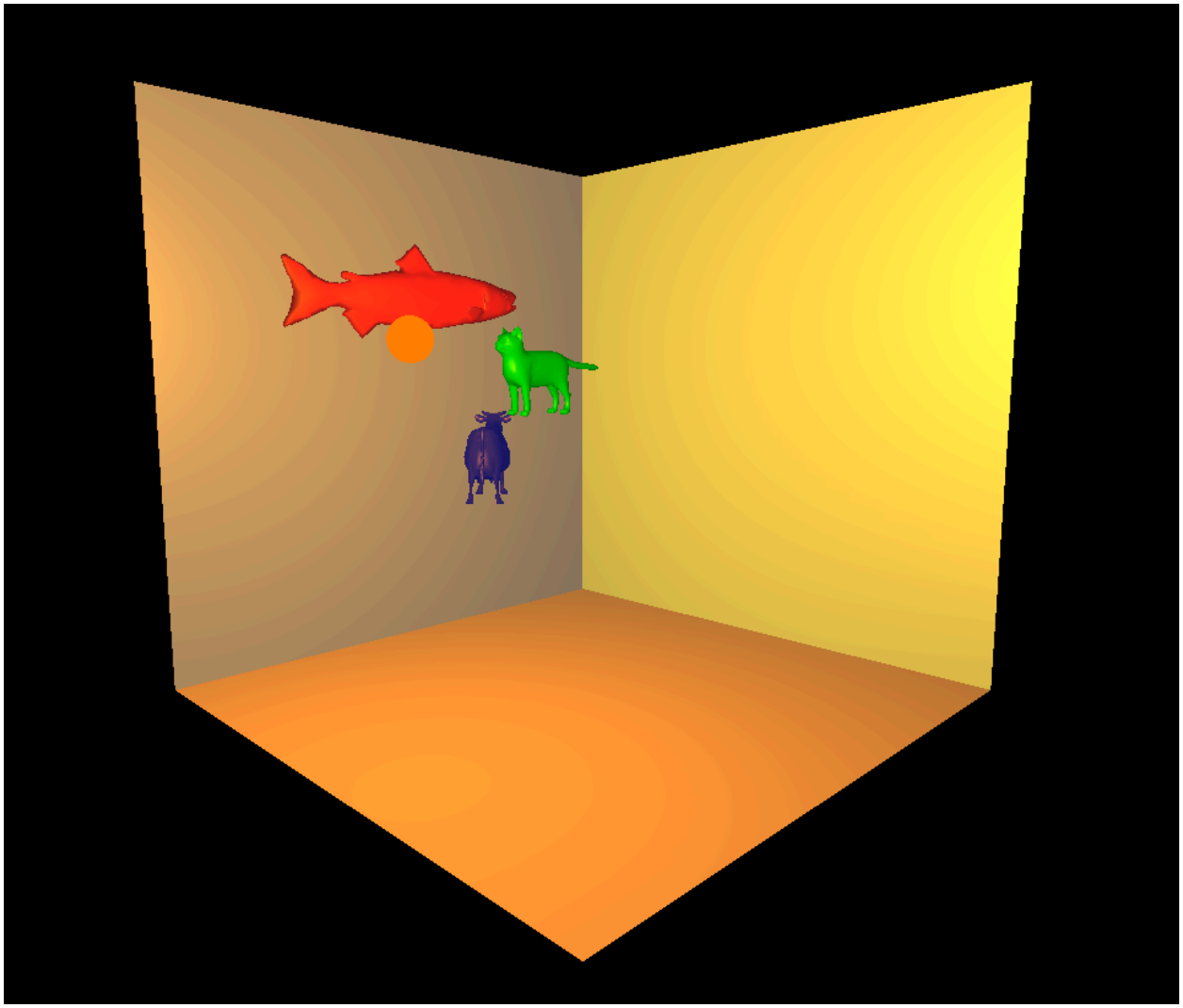
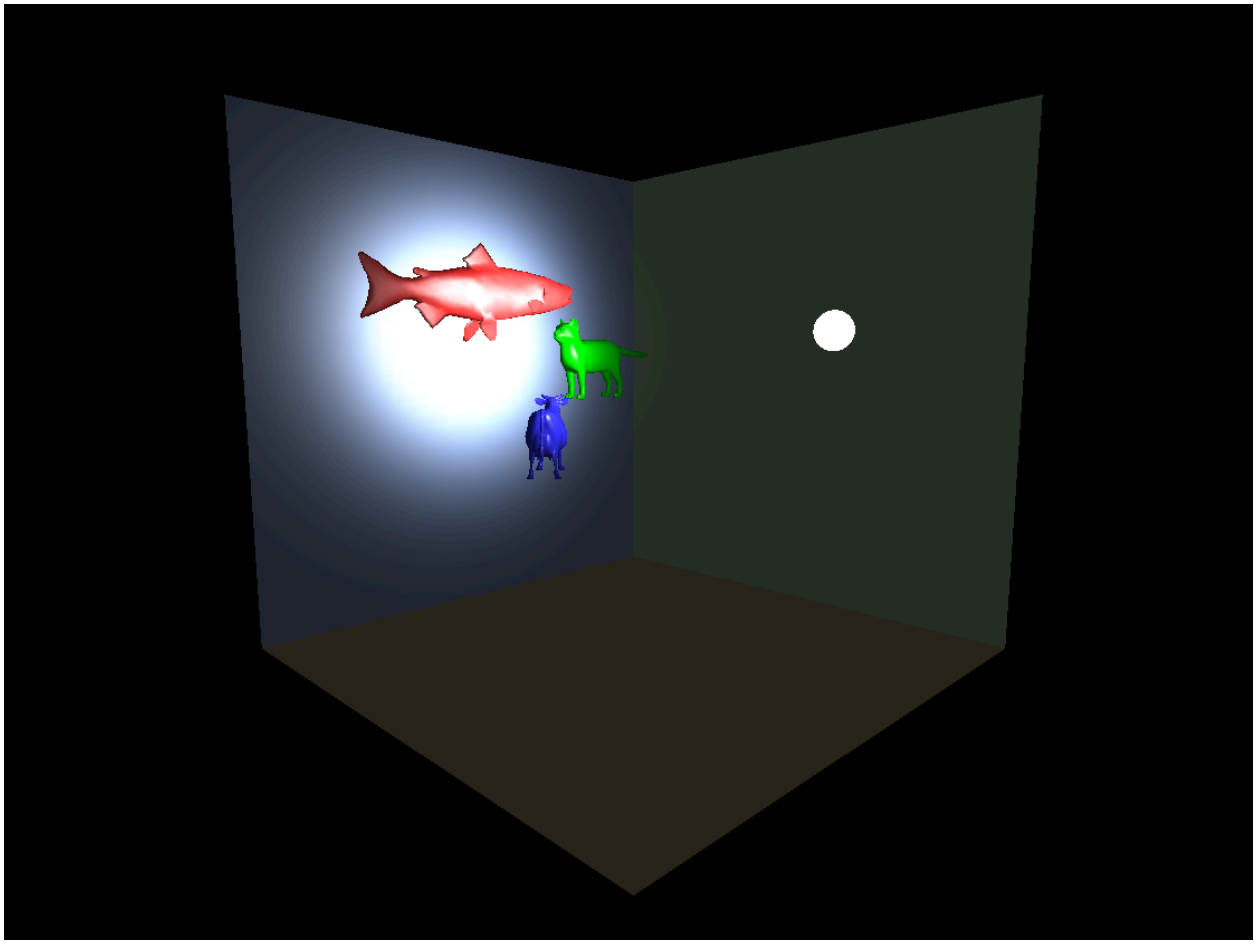
Wolfie Essink
essinkw@oregonstate.edu or wolfie.essink@gmail.com
Project #3 - Lighting

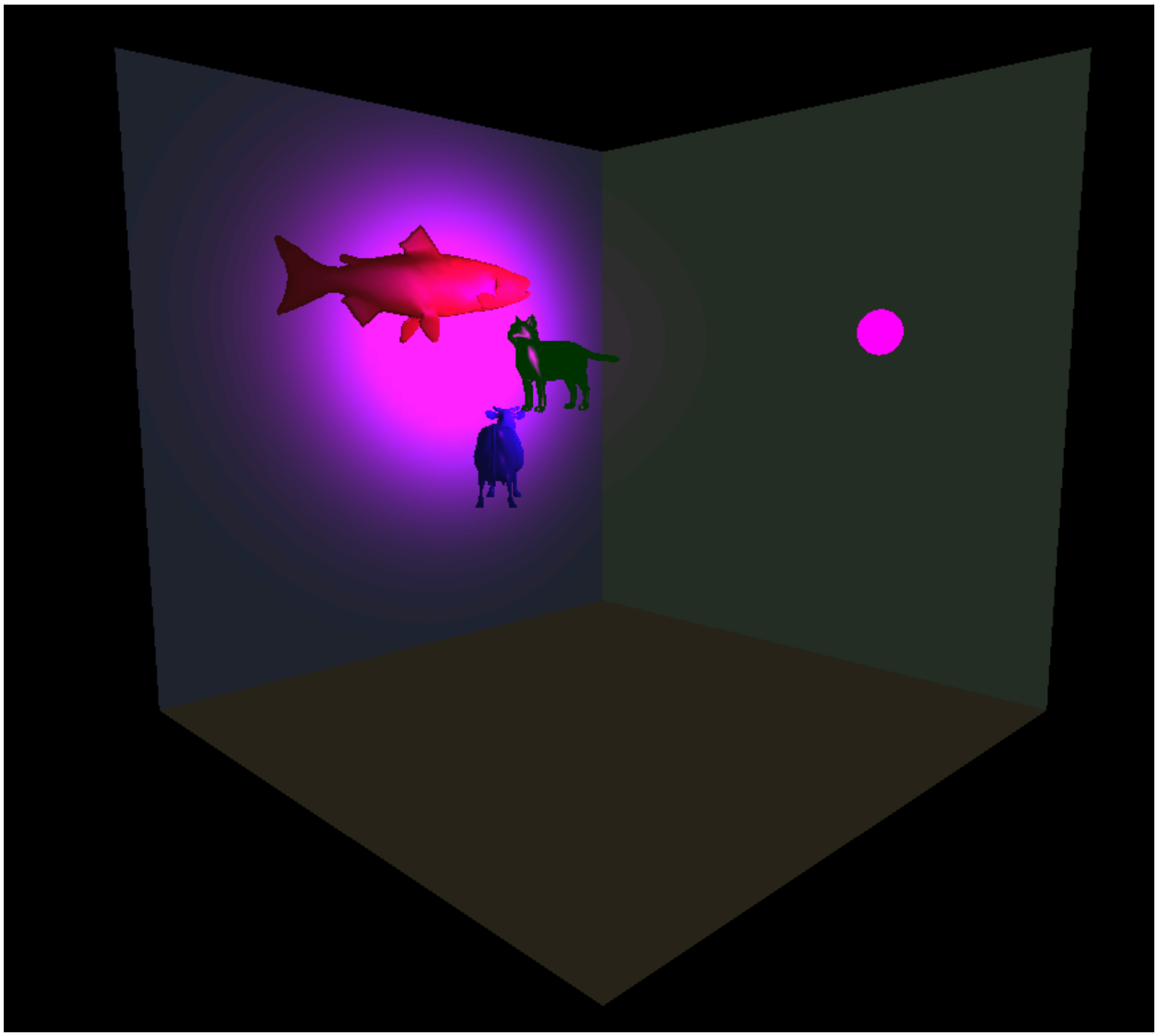
**What I did:**
First off, I started by adding my walls and floor to the scene. I did this by using the example code in the instructions to create their shape, along with making them smooth with glShadeModel(GL_SMOOTH), and gray, green, and brown like in the example image, using SetMaterial(). I then added my objects to showcase my lighting in the scene. I chose a cow, a cat, and a salmon and added their obj files to my project folder. Then I translated, rotated, scaled, colored, and GL_SMOOTHed them until I was happy with how they looked and thought they were in a good location to show off my lighting. I also made sure to make the cow and cat more shiny, while the fish is more rough with SetMaterial() to showcase different textures with my lighting. Then I added a different default camera angle using gluLookAt(), along with Yrot and Xrot with glRotatef() until I found an angle I liked that better showcased my scene. After that, I started to work on the lighting. I did this by first defining all of the global variables I would need for light calculations, modes, and colors, and setting the default light mode to point and color to white. Next, I implemented the lighting for my scene. I started by enabling lighting and creating a single light source using GL_LIGHT0 that moves in a circular orbit around the scene, similarly to my cow in the previous assignment. I used trigonometric functions to calculate the light's X, Y, and Z position each frame so that it smoothly rotates over time. I also defined arrays for the light's diffuse, specular, and ambient components to control its brightness and color. Then, I created the two different lighting modes, point and spot light, that can be switched between using the keys p and s defined in the keyboard function. In point light mode, the light emits evenly in all directions without attenuation, while in spotlight mode, it focuses in a cone shape aimed at the back wall with gradual falloff using attenuation. I also added a small colored sphere using OsuSphere() at the light's current position so I could see where the light source was moving in the scene. After that, the last thing I added was the different colors. I did this by creating the keyboard shortcuts 'w', 'r', 'o', 'y', 'g', 'c', and 'm' in the keyboard function for the colors white, red, orange, yellow, green, cyan, and magenta. These keyboard shortcuts change the RGB values of my global LightColor variable used in my point and spot light code, therefore changing the color directly in the display function when the corresponding key is pressed.

**Screenshots:**









**Video:**
https://media.oregonstate.edu/media/t/1_n3i9u5t6