CompSci 351-1 Introduction to Computer Graphics
Name: Spencer Fitch
Email: SpencerFitch2022@u.northwestern.edu
Demo Video: https://northwestern.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=01d14891-e6a3-414b-ba81-ac8a00632da0

Project C: Lighting and Shading

1 Introduction

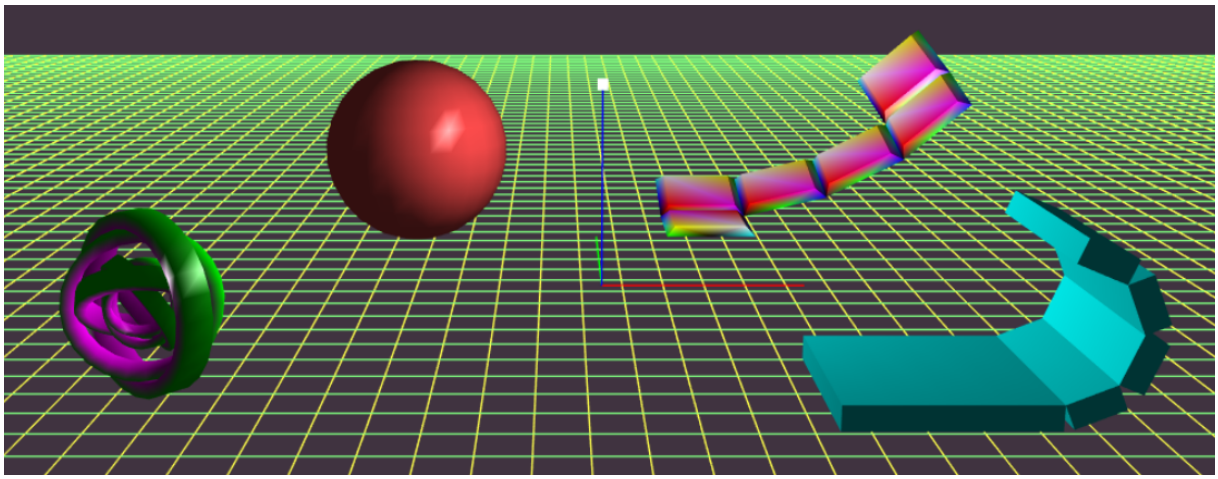


Figure 1: Full 3D scene for Project C

The main goal in completing this project was to gain a better understanding of how lighting and shading works within WebGL in order to create more visually compelling and technical scenes. Prior to this project, all of our scenes had "cartoony" uniform shading and lighting throughout the scene, rather than having different lighting and shading on the object based on its position and orientation. In this project, we update our WebGL vertex and fragment shaders in order to support more dynamic lighting based on a point light source in our scene, as shown above in Figure 1.

2 User Guide

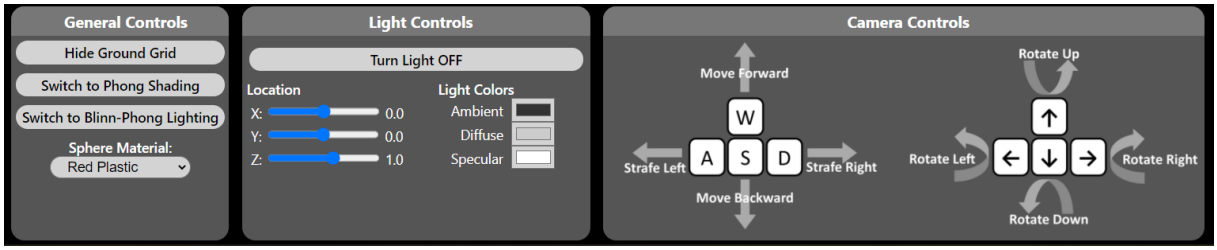


Figure 2: Instructions for controlling the shading, lighting, and camera

Figure 2 above shows all of the controls for this program that display below the WebGL canvas. In the leftmost section labeled "General Controls" we have several buttons as well as one drop down selector. The top button ("Hide Ground Grid") will toggle between hiding and showing the large ground plane, as well as the axis and light markers, that can be seen prominently in Figure 1. Below that are the buttons to toggle between the different shading modes (Gouraud and Phong) and lighting modes (Phong and Blinn-Phong). Lastly below that is a drop down list that changes the material of the slowly rotating sphere when using Phong shading. The list contains all of the 22 available materials from the provided *materials_Ayerdi.js* library.

The next section, titled "Light Controls", contains all of the controls for the point light used to light the scene. The first control in this section is the large "Turn Light OFF/ON" button that spans the entire width of the section and enables the light to be turned on and off to see the effect that the point light has on the scene. Next, the three slider controls under the "Location" header on the left side change the X, Y, and Z position of the point light in the 3D space. Lastly, the three light inputs on the right side enable the user to change the color of the Ambient, Diffuse, and Specular terms of the point light. By changing these, it will appear as if the scene is being lit with a different color light bulb; by default the scene is just lit with a standard white light, but it can be customized to be any RGB color.

The last controls section, titled "Camera Controls", contains all of the keyboard controls needed to move and orient the viewport camera in 3D space. The *WASD* keys change the camera eye position, while the arrow keys change the camera look-at position. When used in conjunction, these controls enable the user to move anywhere within the scene's 3D world.

3 Scene Graph

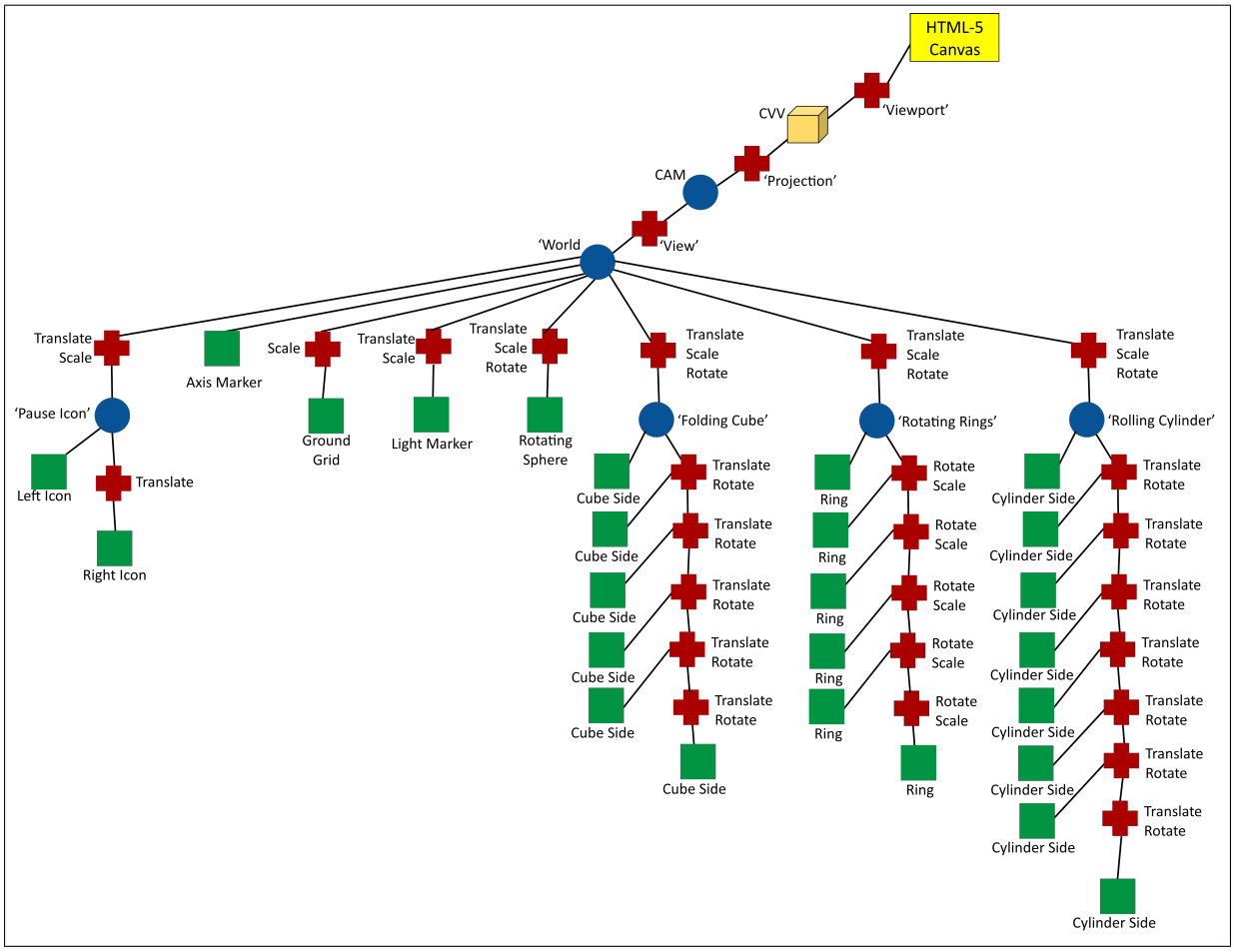


Figure 3: Full scene graph for Project C

Figure 3 above shows the full scene graph for the full scene drawn in the canvas. In this, we can see the many transformations and drawing calls that are made in order to construct the full scene shown on screen.

4 Results

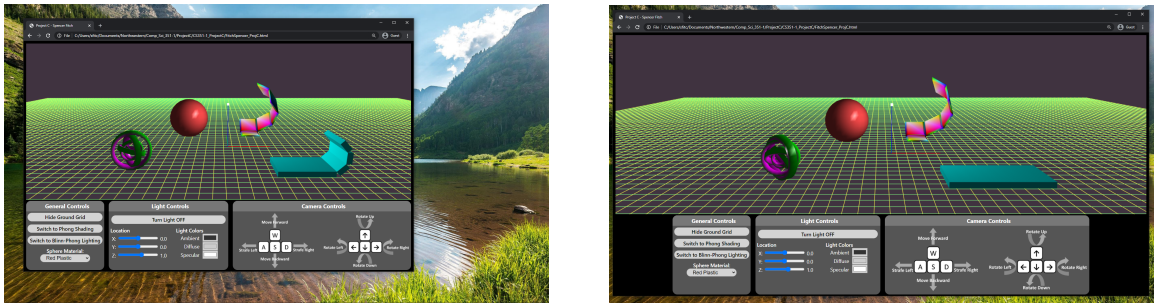
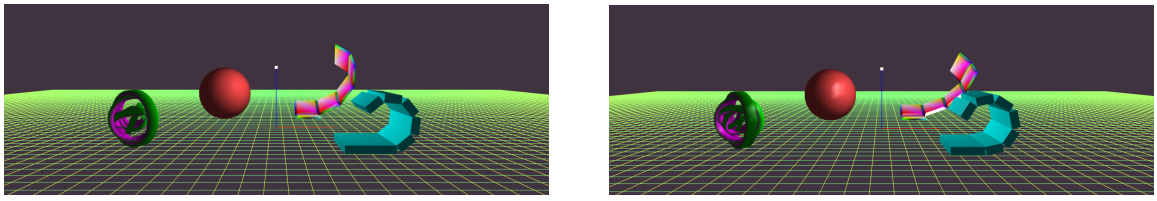


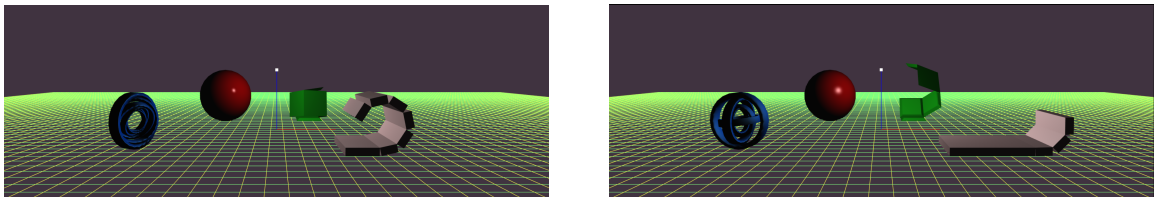
Figure 4: Demonstration of canvas resizing based on window size

Figure 4 demonstrates how the WebGL canvas dynamically resizes so that it always takes up the full width and 70% of the height of the browser window, minus a small margin in each direction. In addition to the canvas resizing, the instructions and controls on the lower 30% of the page are wrapped in a flexbox HTML object so they will also dynamically resize to fill the remainder of the available browser window. The combination of the canvas resizing and flexbox object means that the full page will never need any scroll bars to display properly, though the flexbox item itself may have scroll bars added when the window is sized incredibly small so that it can still display all of its contents adequately.



(a) Phong Lighting (b) Blinn-Phong Lighting

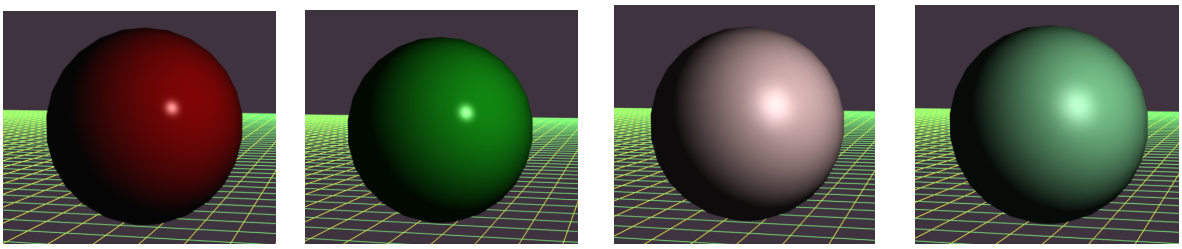
Figure 5: Gouraud Shading



(a) Phong Lighting (b) Blinn-Phong Lighting

Figure 6: Phong Shading

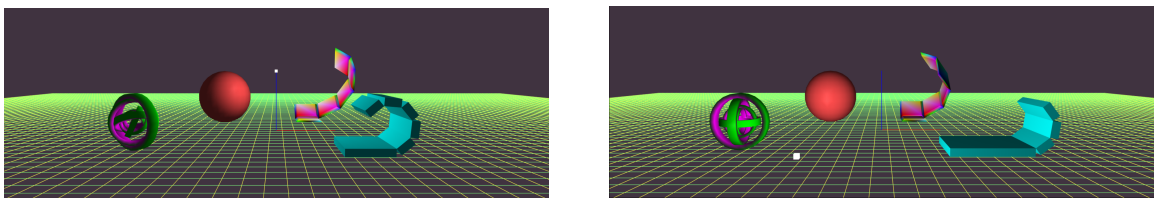
Figure 5 and Figure 6 above show how the scene appears with the different shading and lighting combinations. We can see that the specular highlight is smaller for both of the shading methods when using the Phong lighting than it is when we use the Blinn-Phong lighting, which is what we would expect given that Blinn-Phong is an approximation of the more precise Phong lighting, meaning its specular highlights will have a larger region of error than Phong lighting.



(a) Red Plastic (b) Emerald (c) Pearl (d) Jade

Figure 7: Slowly rotating sphere with different materials

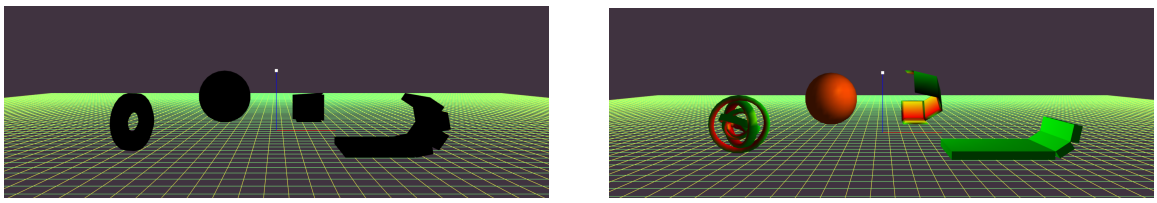
Figure 7 above shows the slowly rotating sphere being drawn with several different materials from the *materials_Ayerdi.js* library. We can see that by changing the selected material in the controls below we change not only the color of the sphere, but also the reflectance characteristics of the surface as well, which results in the surface having different specular highlights and overall shading across the different materials.



(a) Light at (0, 0, 1) (b) Light at (-0.8, -2.7, 0.2)

Figure 8: Effect of changing light position in scene

Figure 8 demonstrates how moving the position of the point light in the scene changes the way that the objects within the scene are lit. We can see that not only does the overall shading of the objects change, but the specular highlights and ambient lighting as well.



(a) Lights turned off (b) All light colors yellow shifted

Figure 9: Effect of changing different lighting options

Figure 9 shows the effects of using the other controls in the "Light Controls" section below the canvas. In Figure 9a we can see that turning the light off causes all of the objects that use the new lighting shaders to be completely black. This is what we would expect to happened when we completely remove the light from the scene, as it is impossible to see anything in a scene where there is no light. The reason we still see the ground plane, axis marker, and light marker is that those objects were intentionally drawn using a different shader such that they would not be affected by any changes we make to our lighting or shaders. Figure 9b demonstrates how changing the color of the different lighting terms affects how our objects appear in the scene. We can see in the figure that by changing the color of the light, we change how the colors of the objects appear to us, which is the same thing that happens in real life with colored lights.