Brett Stelly

Direct Lightning

**Specular Implementation**

To calculate the specular light, you will need to create two uniform variables in the fragment shader. In the fragment shader, declare two uniform variable of type vec3 and call them “lightPosition” and “cameraPosition”. To send information to the uniform variables, we need to create handles and send information to the handles from the rendering program.

In the rendering program, create two variables of type int. These will be the handles for the light’s position and camera’s position. Initialize the handles using glGetUniformLocation and pass in the shader’s program and the name of the uniform variable that was created in the fragment shader.

Next, create a variable of type vec3. This variable will store the light’s position. Initialize this variable with a vec3. You will also need to do the same for the camera’s position

Once the handles and the variables for the light and camera positions are created, we must now send the information to the fragment shader. To do this, use the function glUniform3fv and pass in the handle as the first argument and the reference to the first index of the variable for both.

Now that the fragment shader contains the information for the light’s position, we can begin calculating the specular light. In the fragment shader, create a float for specular strength. Next create a vec3, call it “viewDir”, and assign it by calling normalize and passing in the cameraPosition – vPosition.xyz. Then create a new vec3 variable called “reflectDir” and assign it by calling the function reflect and passing in the normalized light position and the normalized vPosition.xyz. Then create a float called “spec”. Initialize spec with the pow function. Pass in the max function with the arguments of being the dot product of viewDir and reflectDir, and 0. Pass in a power of 2 as the second argument of the pow function. The higher the power the sharper the specular light will be. Finally, create a variable of type vec4 and call it “specular”. Initialize specular with a vec4 with all four arguments being 1 multiplied by vColor multiplied by spec.

To add the effects of specular to FragColor, multiply vColor by specular. You can also add specular to ambient and diffuse and then multiplying to add their effects.