Final Project

Delyar Tabatabai

Car in Dark Night

[tabatase@oregonstate.edu](mailto:tabatase@oregonstate.edu)

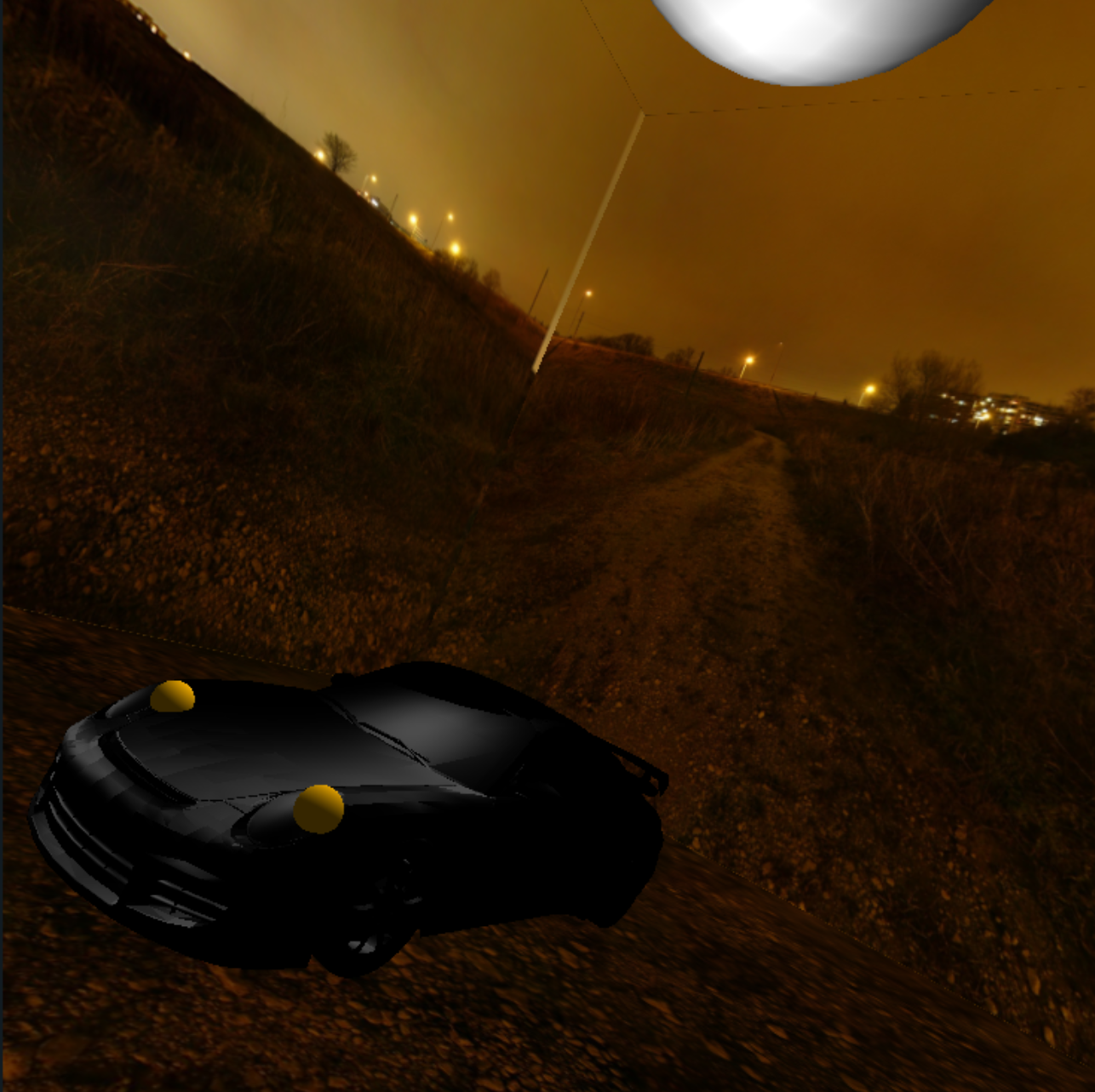
Video: <https://media.oregonstate.edu/media/t/1_qov81dkk>

In this project, I aim to create a realistic 3D scene of a car driving at night. The scene will be rendered using OpenGL shaders with the implementation of cube mapping and car object rendering. The main features of this project are the use of 6 images for cube mapping to create a dark night atmosphere, rendering a car obj file, and displaying the car's front beam lights and their reflection on the road.

This GLIB file defines a 3D scene with a cube map background, a car object, two spheres as lights attached to the car, and a moon object. It also defines two shader programs for rendering the scene, one for the car object with lighting and specular highlights, and one for the cube map with texture coordinates.

The Perspective command sets the field of view for the camera to 70 degrees, and the LookAt command positions the camera at (0,0,3) looking at the origin with the up vector (0,1,0).







The PushMatrix and PopMatrix commands are used to apply transformations to objects in the scene. Translate, Scale, and Rotate are used to move, resize, and rotate objects, respectively. The Obj command is used to load 3D models from OBJ files.

The Color command sets the color for subsequent objects to be drawn, and the Sphere command draws a sphere with a given radius and number of subdivisions.

The CubeMap command sets up a cube map texture for the scene, using six BMP images for the six faces of the cube.

The Vertex and Fragment commands define shader programs for rendering objects in the scene. The program is used for rendering the car object, and the Texture program is used for rendering the cube map.

The program takes in several uniform variables as inputs, including the light positions, material properties, and the color and specular color of the object. It applies Phong shading to calculate the lighting and specular highlights for the object.

My goal was to create the feeling of the light bulbs and show the reflections on the road. However it did not turn out as good as I thought it would.

The shader program is performing per-fragment lighting using three different light sources: Light0, Light1, and Light2. The normal vector is calculated in the vertex shader and passed to the fragment shader as an interpolated value, along with other variables such as the texture coordinates, eye position, and material properties.

In the fragment shader, the ambient component is calculated based on the ambient reflectance coefficient (uKa) and the material color (uColor). The diffuse component is calculated by taking the dot product of the normal vector with the sum of the normalized light vectors (Light0+Light1+Light2) and multiplying it by the diffuse reflectance coefficient (uKd) and material color. The specular component is calculated using the Blinn-Phong model, which involves calculating a reflection vector and taking the dot product of the reflection vector with the normalized eye vector. The resulting value is raised to the power of the shininess coefficient (uShininess) and multiplied by the specular reflectance coefficient (uKs) and the specular color (uSpecularColor).

The final color is the sum of the ambient, diffuse, and specular components, and is output as the fragment color. The alpha value is set to 1.0 in this case, but could be modified based on the needs of the application.