OpenGL Project Documentation

Codruta Elen Jucan

March 11, 2025

Contents

1 Subject Specification

The project involves the creation of a 3D scene rendered using OpenGL. The scene includes grass, a lake, a castle, and Disney princesses, all modeled in Blender and imported as a single object. Additionally, Cinderella's fairy godmother is modeled as a separate object and programmed to fly in circles.

2 Scenario

2.1 Scene and Objects Description

The scene is a magical environment featuring:

- A grassy landscape surrounding a shimmering lake.
- A castle standing near the lake.
- Disney princesses standing at the table (with Mrs. Potts, Chip and Lumiere) or near the lake.
- Cinderella's fairy godmother, animated to fly in circular paths.
- Cinderella's step mother and sisters, standing next to the castle, spying on the princesses
- Rapunzel's horse
- Trees
- A skydome

2.2 Functionalities

- Dynamic lighting effects, including point and directional lights.
- Camera navigation and automated tours.
- Environmental fog and lightning effects.
- Interactive controls for toggling lighting, fog density, and camera modes.

3 Implementation Details

This section describes the OpenGL functionalities implemented in the project.

3.1 Lighting Model in OpenGL

Dynamic lighting is achieved using point lights and directional lights. The lighting model incorporates ambient, diffuse, and specular components for realistic effects. Lightning flashes are simulated by adjusting light intensity, triggered at random intervals.

Code snippet for point light calculation:

```
vec3 lightDir = normalize(pointLightPos - fPosition);
ambient = ambientStrength * pointLightColor;
float diff = max(dot(normalEye, lightDir), 0.0);
diffuse = diff * pointLightColor;
vec3 reflectDir = reflect(-lightDir, normalEye);
float spec = pow(max(dot(viewDir, reflectDir), 0.0), 32);
specular = specularStrength * spec * pointLightColor;
```

3.2 Camera Navigation

The camera system allows free movement using keyboard inputs and an automated tour through predefined positions and targets. Users can toggle between manual and automated modes.

Code snippet for camera movement:

```
void Camera::move(MOVE_DIRECTION direction, float speed) {
    switch (direction) {
        case MOVE_FORWARD:
            cameraPosition += speed * cameraFrontDirection;
            break;
        case MOVE_BACKWARD:
            cameraPosition -= speed * cameraFrontDirection;
            break;
        // Additional directions...
    }
    cameraTarget = cameraPosition + cameraFrontDirection;
}
```

3.3 Effects

Effects such as fog and lightning add depth to the scene. Fog density adjusts dynamically based on user input, while lightning is triggered programmatically, accompanied by sound effects.

Code snippet for fog density:

```
glUniform1f(fogDensityLoc, fogDensity);
finalColor = mix(fogColor, finalColor, fogFactor);
```

4 Graphical User Interface Presentation / User Manual

The application uses keyboard controls for interaction:

- `W, A, S, D`: Move the camera.
- `Q, E`: Move the camera vertically.
- `R`: Toggle wireframe mode.
- `UP/DOWN`: Adjust fog density.
- `T`: Start or stop the automated camera tour.
- `L`: Toggle directional light.
- `P`: Toggle point light.

5 Conclusions and Further Developments

The project successfully demonstrates various OpenGL functionalities, including dynamic lighting, camera navigation, and environmental effects. Future enhancements could include:

- Improved animations for characters and objects.
- Enhanced graphical effects such as reflections and shadows.
- Additional interactive elements within the scene.

6 References

- Blender: Blender Youtube Tutorials - Ing. Constantin Nandra, PhD.

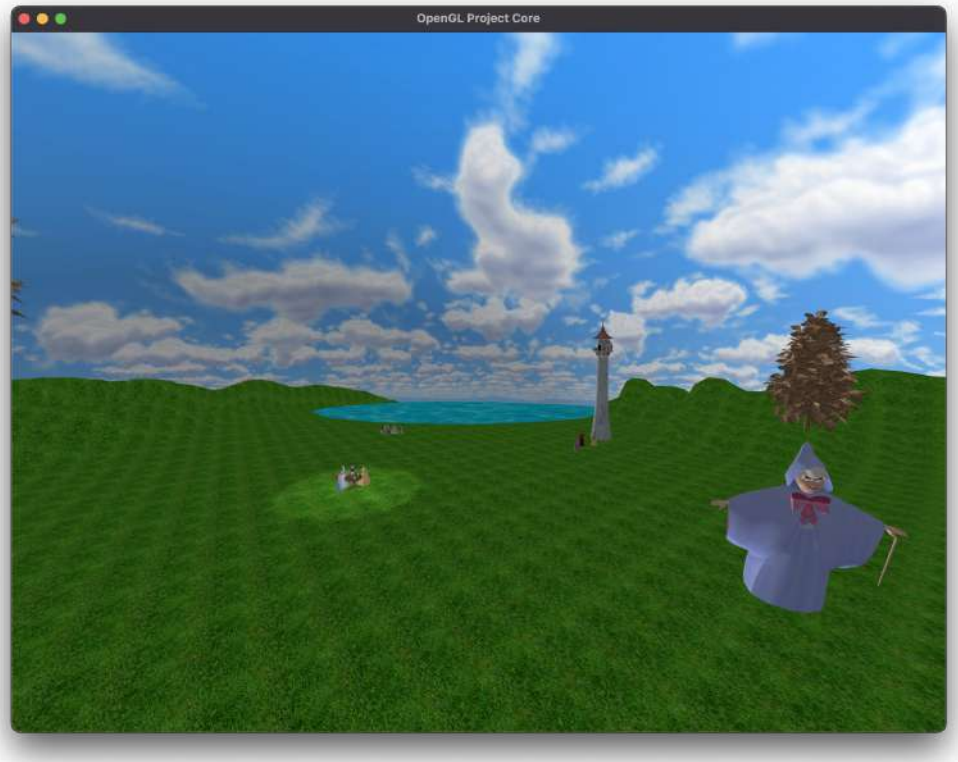

Figure 1: View of the scene

Figure 2: Princesses standing at the table

Figure 3: Cinderella's step mother and sisters

Figure 4: Princesses standing at the lake