Emre Konak
31218

CS405 PROJECT 3 REPORT

Introduction

The goal of this project is to create a simple solar system simulation using WebGL and scene graph techniques. By modeling the Sun, Earth, Moon, and Mars, the project demonstrates hierarchical transformations to represent orbits and rotations. A key focus is implementing realistic lighting through the Phong reflection model, enhancing the visual representation of celestial bodies. The simulation uses textures for each planet and moon to create a more immersive experience. This project highlights the importance of scene graphs in managing complex object relationships and transforming them efficiently in 3D space.

Project Overview

The project simulates a solar system using WebGL and scene graphs to model the Sun, Earth, Moon, and Mars. Each celestial body is represented as a textured sphere with transformations applied for orbit and rotation. The Sun acts as the root node, with Earth and Mars orbiting it, while the Moon orbits Earth. Lighting and shading are handled through vertex and fragment shaders, creating realistic visual effects. The project highlights hierarchical transformations, texture mapping, and dynamic object rendering to create a simple yet effective solar system simulation.

Methodology

The project is structured around three key tasks, each contributing to the development of a functional and visually accurate solar system simulation.

In **Task 1**, the focus was on implementing the draw method within the scene graph. This method ensures that transformations applied to parent nodes propagate correctly to their children, allowing hierarchical relationships like the Sun, Earth, and Moon to reflect real-world orbits and rotations. By recursively applying model-view-projection matrices, each node is drawn in the correct position relative to its parent. This forms the backbone of the scene graph, ensuring smooth and consistent object rendering.

Task 2 involved enhancing the fragment shader to implement diffuse and specular lighting using the Phong reflection model. Initially, the lighting calculations only supported ambient light, resulting in flat, unrealistic shading. By incorporating light direction, surface normals, and view reflection, the shader dynamically adjusts surface brightness and highlights based on the light's position. This step significantly improved the realism of the scene, ensuring that celestial bodies react accurately to the Sun's light source.

In **Task 3**, Mars was added to the scene as an additional child node of the Sun. Mars was positioned by translating it along the X-axis and scaled to match its real-world proportions. A unique texture was applied to Mars, and rotational logic was implemented to make Mars rotate 1.5 times faster than the Sun. This task demonstrated how new objects can be seamlessly integrated into the existing scene graph, reinforcing the modularity and scalability of the simulation.

Tasks Breakdown

Task 1 focused on implementing the **draw method** within the **SceneNode** class to handle scene graph propagation. This method ensures that transformations applied to parent nodes, such as the Sun, correctly affect child nodes like Earth and the Moon. By recursively applying transformations, each node inherits its parent's position, rotation, and scale, allowing for hierarchical rendering. This task was crucial in modeling orbits and rotations, ensuring that planetary bodies moved naturally in relation to one another.

Task 2 involved enhancing the fragment shader to implement diffuse and specular lighting using the Phong reflection model. Initially, the scene only supported ambient lighting, resulting in flat shading. By calculating the light direction and reflections, the shader dynamically adjusted the brightness and highlights on celestial bodies, creating a more realistic and visually appealing representation. This task added depth and detail to the simulation, making planets and moons react accurately to the Sun's light.

Task 3 added Mars to the solar system as a child of the Sun. Mars was positioned by translating it -6 units along the X-axis and scaled to 0.35 of its original size. A texture representing Mars was applied, and rotational logic was added to make Mars rotate 1.5 times faster than the Sun. This task demonstrated the extensibility of the scene graph, showing how new planets could be easily integrated while inheriting transformations from their parent nodes.

Results

The solar system simulation accurately models the Sun, Earth, Moon, and Mars with realistic orbits and rotations. The scene graph ensures smooth transformation propagation, allowing each celestial body to move naturally in relation to its parent.

Lighting enhancements using the Phong reflection model add depth and realism, with the Sun acting as a fixed light source that casts highlights and shadows on the planets. Mars integrates seamlessly, correctly positioned, textured, and rotating independently.

The project successfully demonstrates key 3D graphics concepts, resulting in a functional and visually realistic solar system.