

CS 3451 Final Project Report

Akina Castellano & Jiwon Lee

Image Overview

Because our theme was “Journey”, we wanted our scene to reflect the wildest journey, and we thought illustrating our world in outer space was the best reflection of a stunning journey. However, just a solar system with a black background with a couple stars would be boring, so we wanted to implement a skybox to fully encapsulate the beauty of the endless galaxy we live in. We implemented Phong shading so that the planets would look as realistic as possible and truly highlight the unique planet textures we selected to map onto the planet objects. We wanted to fully encapsulate the beauty of the universe in which Earth belongs while keeping our planet as being the focal point, so we utilized the billboard shader to keep the viewer focused on our planet and to differentiate it from the other planets. This was to truly depict how Earth is truly the center of attention in this scene since it is our planet, in which we live and can see the surrounding universe from. Additionally, the implementation of spacecrafts included a semi-invisible UFO and spaceship both made from complex geometric objects and unique texture files for their base colors and normal mappings. This UFO represents the unknown mysteries that space holds, which are yet to be discovered. The spaceship was chosen to be added for the purpose of representing how our human curiosity has motivated us to create such vehicles of transportation that allow us to journey to places we would’ve never expected us to reach before. Then lastly, we added a collection of star particles placed in random locations on the skybox to add even more realism and illuminated beauty to the scene.

Technical Implementation Details

To begin, we started by implementing the desired obj files and texture packs we wanted to have in our space scene. This included some basic geometric shapes being spheres for the planets but also some more advanced complex shapes being our spaceship hull and UFO spacecraft we downloaded from online sources. For the complex shapes especially, it was necessary to adjust the transform Matrices for scaling and positioning purposes to make the scene visually appealing and ensure all objects were able to be seen clearly along with their unique textures. Regarding the phong lighting algorithm in the basic shader, that was

implemented utilizing similar code that was previously implemented in one of our past assignments with some slight modifications due to variable names and accessing the correct return types of those variables. After successful implementation of that algorithm, we set each object's corresponding material properties and our singular light's properties that made our scene visually appealing. We applied the billboard shader to our Earth object due to the fact that we kept our world as the main focal point of the scene and wanted to keep the viewer's attention centered on it by always having it face the camera. We added the basic shader to all the other surrounding planets and the spaceship as well, but for the UFO object we applied the environment shader to give the effect of it blending into the background and giving the illusion of it being partially invisible. For our skybox, we customized the skybox implementation in the main code by incorporating our own galaxy cuebmap files for the correct positions that would be mapped into the environment. Then for the particle system, we utilized the stars shader file and added the functions from the stars shader file to the skybox shaders to implement the particles onto the skybox. Then, we had to add the color of the stars to the skybox.

Solved Challenges

We encountered multiple technical challenges throughout the implementation phase of our project, the first being in finding the preferred objects to be used in our scene and their corresponding texture downloads. The challenge was in finding suitable objects online that would fully render after downloading and also had texture mappings of their color and normals in the format of jpegs and pngs. Some of the base color and normal maps provided for an object wouldn't map correctly onto the obj file after being implemented into our code, so the primary solution to this problem was simply a lot of trial and error and selective choosing of textures that map correctly onto the more complex obj files.

The second challenge was regarding the addition of a particle system for the bright stars into the skybox shader background. We implemented our skybox with a space galaxy consisting of some small stars so we decided to add our brighter star particles for a more aesthetic visual. The biggest challenge for the particles was determining whether to use particles on the skybox or the particles in the 3D environment. However, we decided to add the particles to the skybox because it wouldn't flood the scene with a bunch of stars and cover up the planets. In the end, the

star particles were successfully implemented onto the skybox and fit well with the background surrounding the other objects in the scene.

The last significant challenge was implementing the phong lighting in the basic shader. There was some trouble with figuring out how to use our previous algorithm we implemented in an old class assignment and modifying its implementation slightly to work into this final project due to the difference in return types of the method (vec3 vs vec4). However after going through the variables more in-depth that were to be used in the formulas, it became quickly apparent how to access the correct variables necessary with the correct return types. Adjusting the material properties of the objects and properties of the light itself was also a process but we were eventually able to find a good balance that was visually appealing and effectively highlighted the beauty of our universe.

Contributions

Akina: I implemented the skybox background by modifying the implemented skybox shader provided in the code so that proper customization took place to create the limitless universe background consisting of the galaxy and stars. In the provided default implementation of sky box, I found our desired cubemap online of space with a dark blue galaxy and tiny stars scattered throughout. I then edited the cubemap_files so that they corresponded to the correct cubemap locations (+X, -X, +Y, -Y, +Z, -Z). Our skybox was then successfully rendered giving the illusion of an endless environment in the background of our solar system.

Jiwon: I added the particle system to the skybox and made sure that it would not slow down the framerate of the scene using GPU sided particle generation.