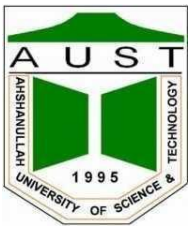
Ahsanullah University of Science & Technology
Department of Computer Science & Engineering



Into the Space: Earth and Its Satellite Buddy

Computer Graphics Lab (CSE 4204)

Project Final Report

Submitted By:	
Yasir Arafah	***20200104042***
Sumaiya Shejin	***20200104043***

Project Requirements:

3D Objects:

1. A Satellite [with texture]
2. An Earth [with texture]

Keyboard Interaction:

- Camera will move around the satellite.

Mouse Interaction:

- Texture of the satellite will change.

Animation:

- Satellite will orbit around the Earth.

Software Platform:

1. **VS Code (Visual Studio Code) - Version 1.91**

   Visual Studio Code (VS Code) is a preferred tool for developing and debugging the project's code. Its robust support for JavaScript, coupled with features such as IntelliSense, seamless Git integration, and a plethora of extensions, fosters an efficient and productive development environment.

2. **Node.js - Version 20.15.0**

   Node.js is a powerful tool for server-side development, enabling the execution of JavaScript code beyond the confines of a web browser. Its comprehensive library of modules provides essential functionalities such as file system management, network communication, and data streaming, ensuring optimal performance for our 3D application.

3. **Adobe Photoshop**

   In the process of developing the 3D models, we utilized Adobe Photoshop's robust capabilities in photo editing and graphic design to create and modify textures. This software's advanced tools were instrumental in generating high-quality visuals, which were crucial for achieving the project's realistic rendering objectives.

Project Features:

This project incorporates several advanced techniques to create an interactive and visually engaging 3D environment.

- Custom shaders are employed for rendering the satellite's body, solar panels, and antenna. These shaders improve the visual fidelity of the satellite by enabling dynamic color changes based on user interaction via mouse clicks. The shaders utilize a basic diffuse lighting model to produce realistic shading effects on the satellite, enhancing its appearance in the scene.
- The lighting system comprises both ambient and directional light sources. Ambient lighting ensures uniform illumination across the entire scene, while the directional light simulates sunlight, creating realistic shadow effects and illuminating one hemisphere of the Earth. This combination provides depth and realism to the scene, making it visually compelling.
- Perspective projection is implemented to achieve realistic depth and spatial relationships between the camera, Earth, and satellite. This projection model closely mimics human visual perception, where objects appear smaller as their distance from the camera increases. The camera moves dynamically around the Earth and satellite, offering multiple viewing angles.
- Texturing is applied to enhance the visual quality of various objects. High-resolution textures are utilized for the Earth, featuring distinct day and night representations, specular highlights for water bodies, and blended cloud layers for

added realism. A starry texture encapsulates the entire scene, providing a realistic background to the simulation.

- Animation plays a central role in the project, with the Earth and its cloud layer rotating continuously. The satellite follows an elliptical orbit around the Earth and adjusts its orientation to maintain a constant view of the planet. These animations contribute to the overall dynamism and realism of the scene.
- Interactive controls allow users to manipulate the camera's position and orientation using both mouse and keyboard inputs. The user can rotate, zoom, and pan around the scene, as well as dynamically alter the satellite's color scheme through mouse clicks.
- The color-changing feature adds an interactive component, allowing the satellite's body and solar panels to alter their colors with each mouse click. The satellite's body cycles through predefined colors, while the solar panels are assigned random colors, adding an element of variability and engagement.

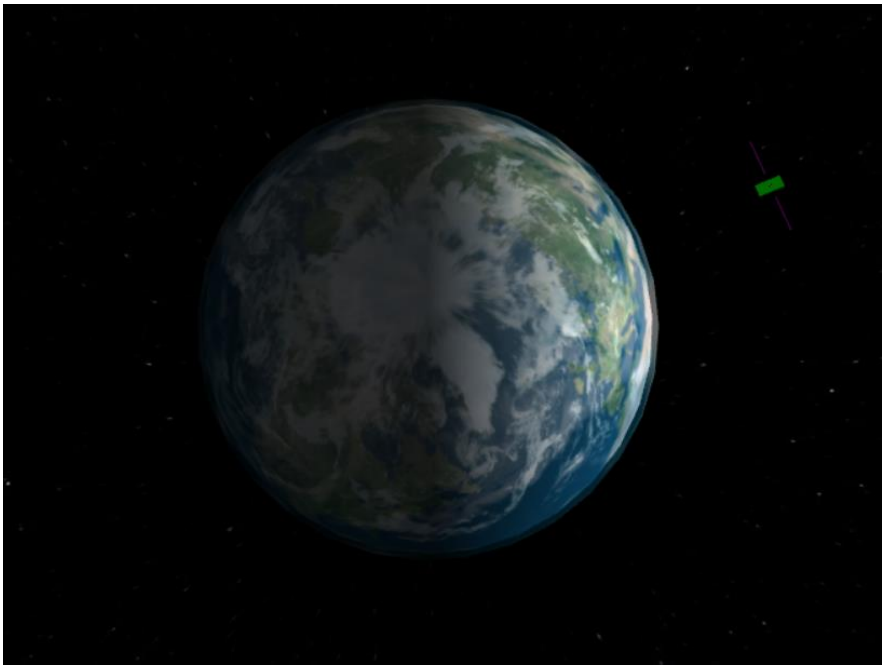
The below table classifies the features into three categories as either implemented, partially implemented or not implemented,

#	Features	Status
1	Custom Shaders	Implemented
2	Implementation of Lighting	Implemented
3	Perspective Projection	Implemented
4	Texture for each object	Partially Implemented
5	Mouse and Keyboard Interaction	Implemented
6	Animation	Implemented

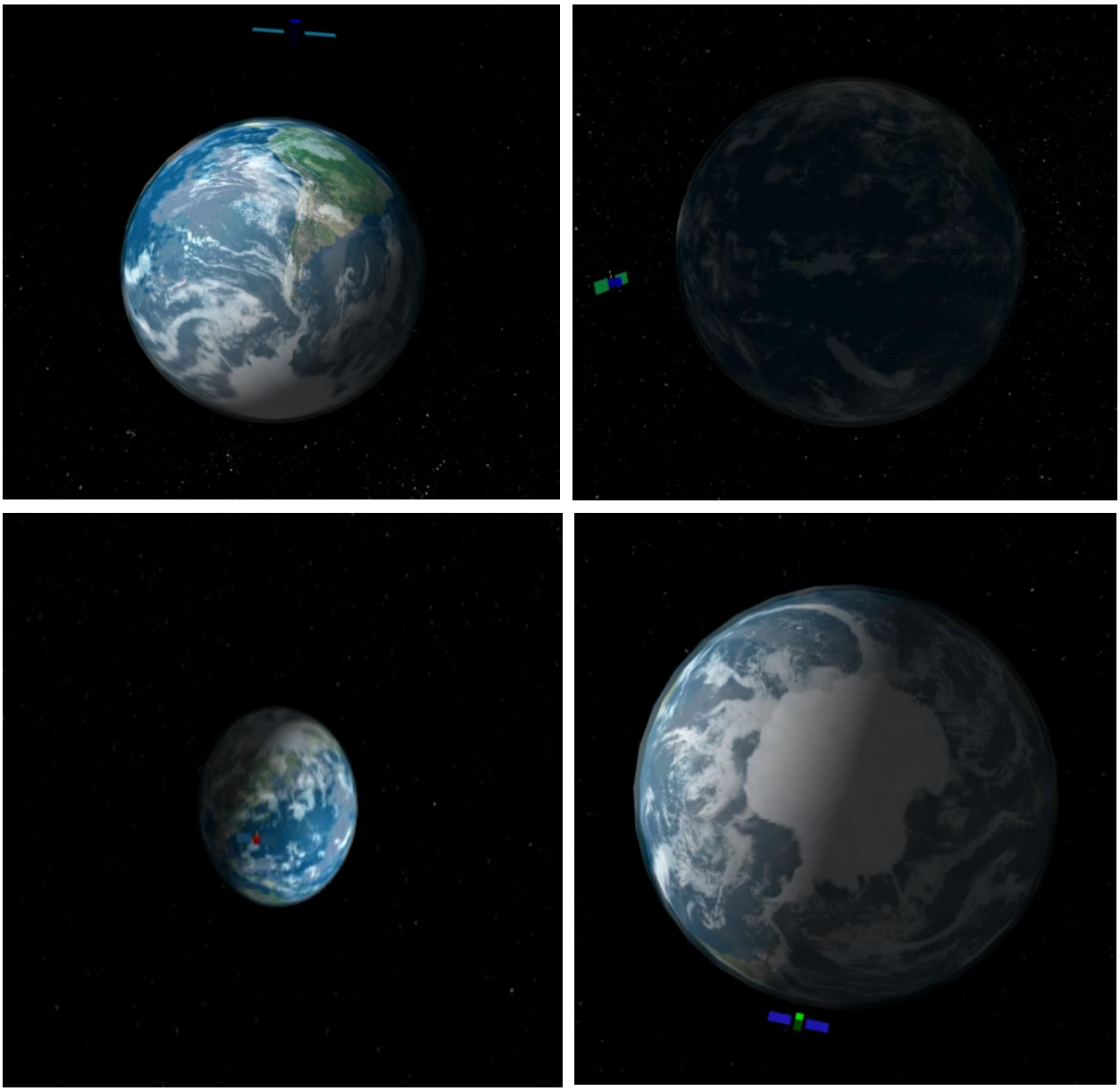
Table 01: Project Feature Table

Snapshots:

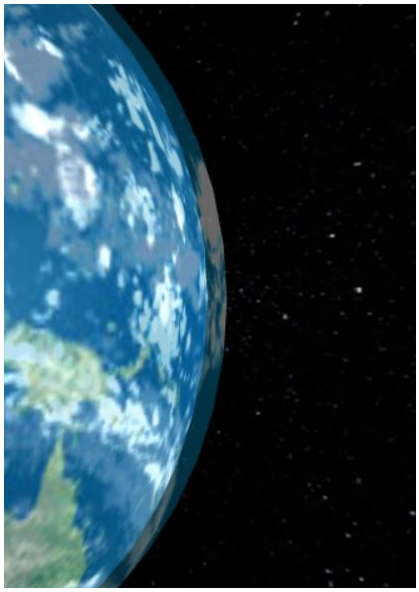
Implementation of Lighting

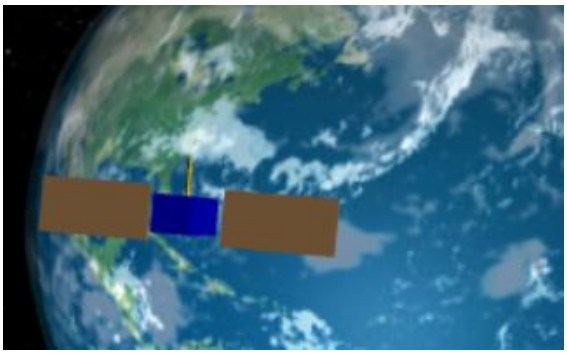


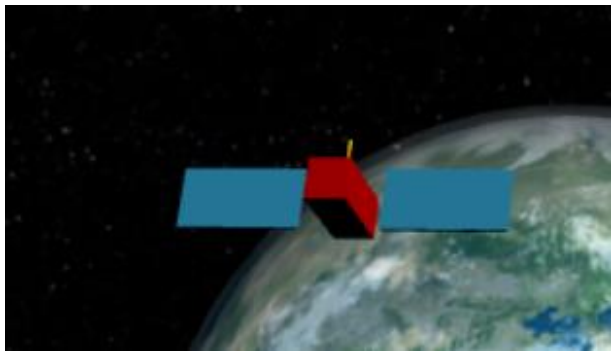
Perspective Projection:

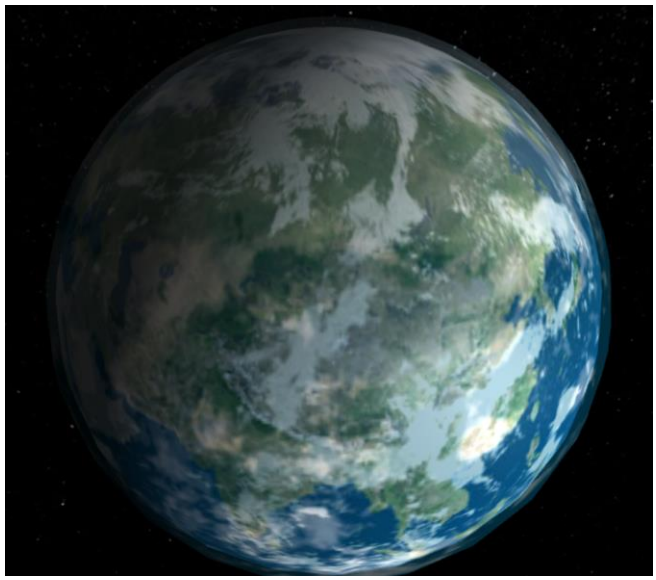


Texture for each object:

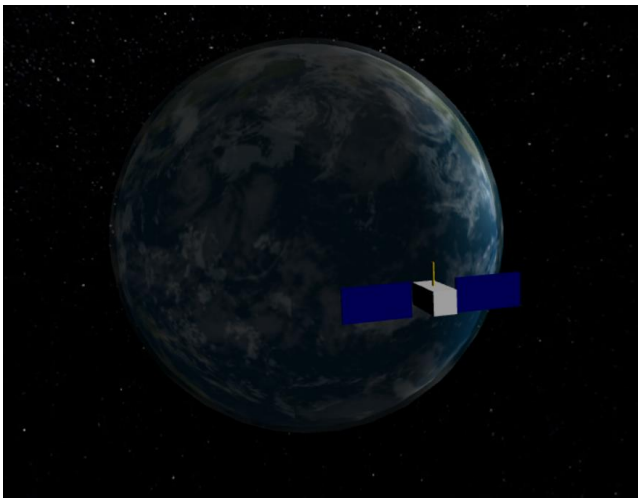


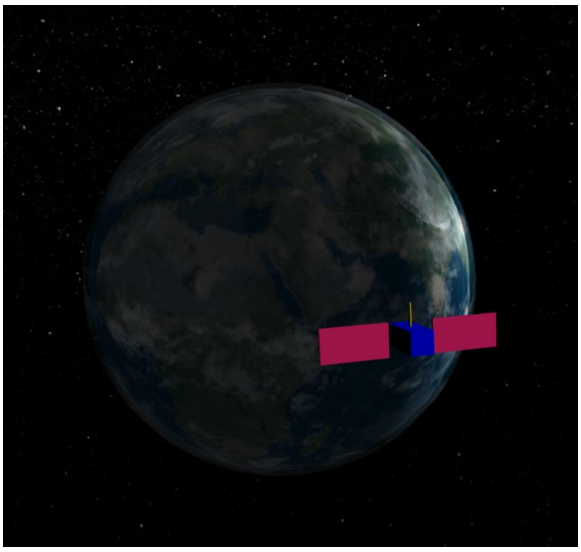


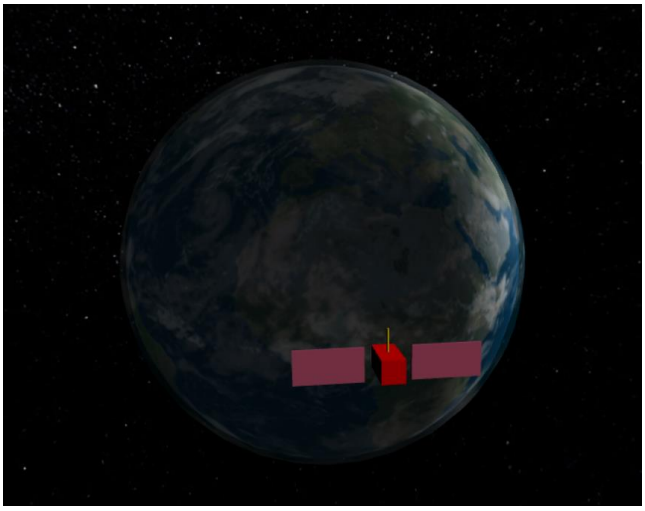


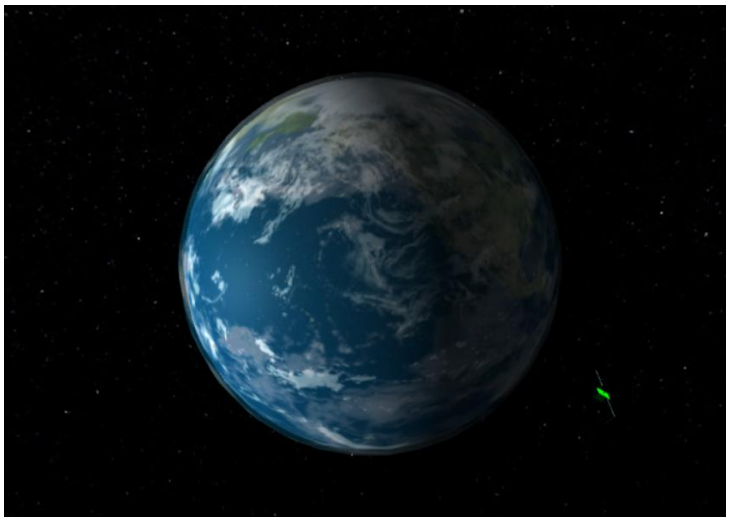


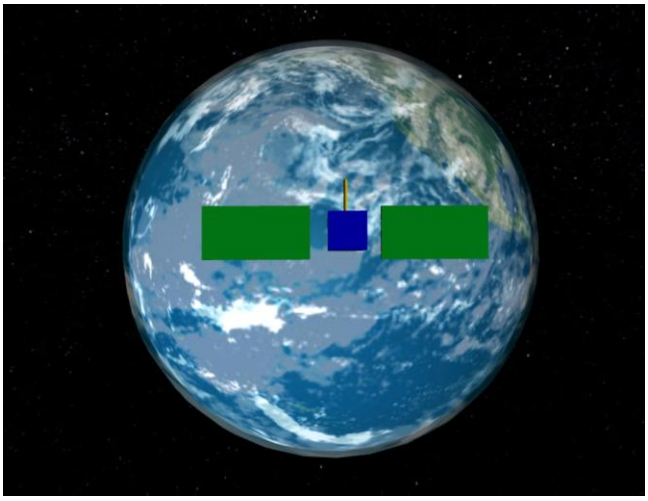
Mouse and Keyboard Interaction:



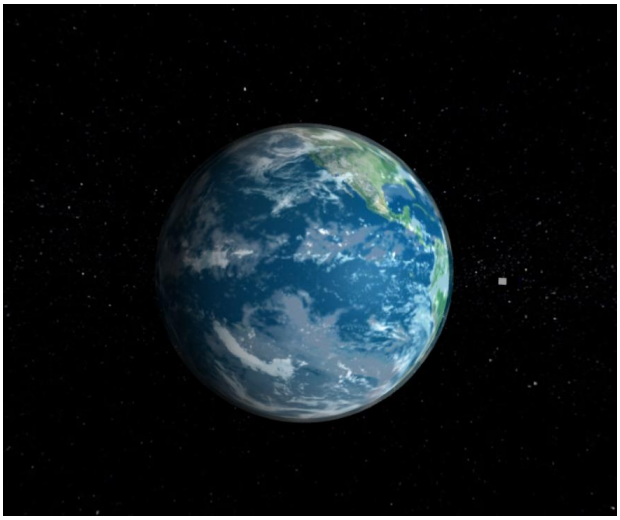


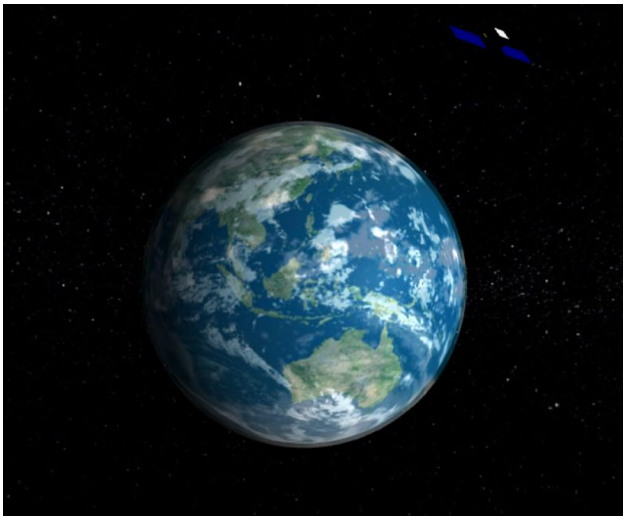


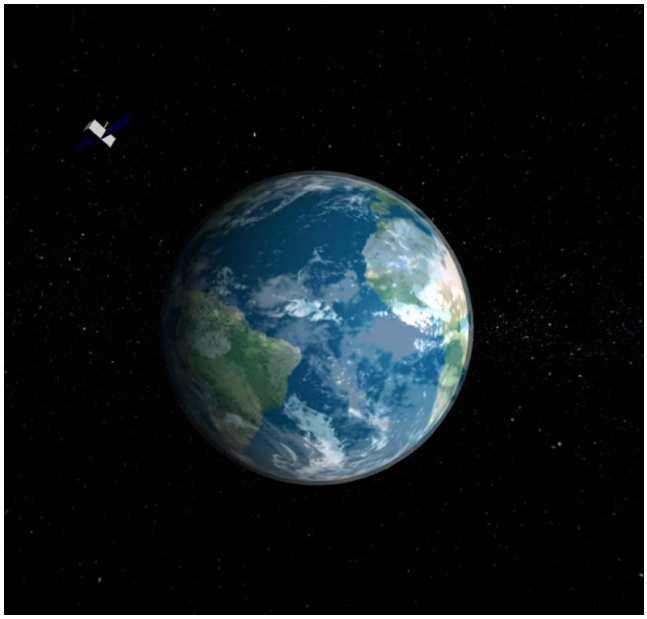




Animation:







Contribution:

20200104042 (**Yasir Arafah**)**:** Implementation of the project, Structuring the 3D models, Designing the scene, Report writing.

20200104043 (**Sumaiya Shejin**)**:** Setting up the ambient and directional lights, Modeling the Satellite, Report Writing.

Future Work:

In future iterations of this project, several enhancements could be made to improve the user experience and visual realism. One key enhancement would be the introduction of dynamic environmental effects, such as real-time weather systems, which could be synchronized with real-world data for added realism. Additionally, improving the satellite's functionality, such as providing real-time information or interactive controls for modifying its orbit or trajectory, could deepen user engagement. Sound integration, including background music and ambient space sounds, could create a more immersive atmosphere. Expanding the interactive features, such as allowing users to customize the appearance of the Earth and satellite or adjust lighting conditions, would provide greater personalization options. Finally, optimizing the scene's performance, by implementing level-of-detail (LOD) techniques and simplifying textures and shaders for distant objects, would ensure the project runs smoothly on a wide range of devices without compromising visual quality.