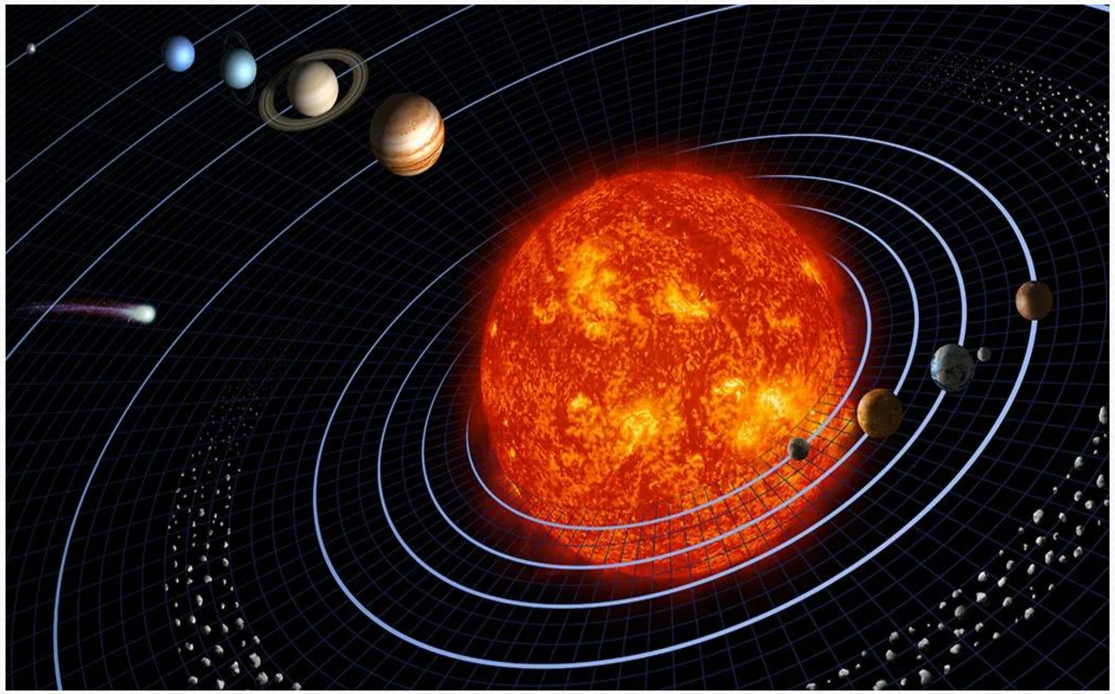
Solar System Simulation

Submitted by

Tasnim Jabir -(2022-3-60-283)
Md. Jubayer Ahmed-(2022-1-60-307)
Md.Ferdous-(2021-3-60-102)
Yesmin Akter-(2021-3-60-072)

Project Overview

- A simulation of the Solar System using OpenGL to visually represent planetary orbits, movements, and interactions.
- Users can interact with the simulation, adjusting speed and zoom levels, and can explore the solar system from different perspectives
- Users can interact with the simulation, adjusting speed and zoom levels, and can explore the solar system from different perspectives.



Feature

8 Planets
Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune.

Real-Time Animation
Planets and their moons orbit around the sun with accurate speed adjustments.

Starry Background
Simulated stars providing a dynamic, realistic space environment.

Programming Tools & Libraries

Core Technologies
• C++ (Primary Language)
• OpenGL (Graphics Rendering)
• GLUT (Window Management)
• Standard Template Library

Graphics Features Implemented
• 2D Transformations (Rotation, Translation, Scaling)

Development Environment
• Windows with OpenGL support
• GCC/G++ Compiler
• Code Blocks / Visual Studio

Methodology

System Architecture(Object-Oriented Design):

- Star structure stores properties like position and brightness for random star generation.
- Each planet has properties like orbit radius, size, speed, and color.

Initialization:

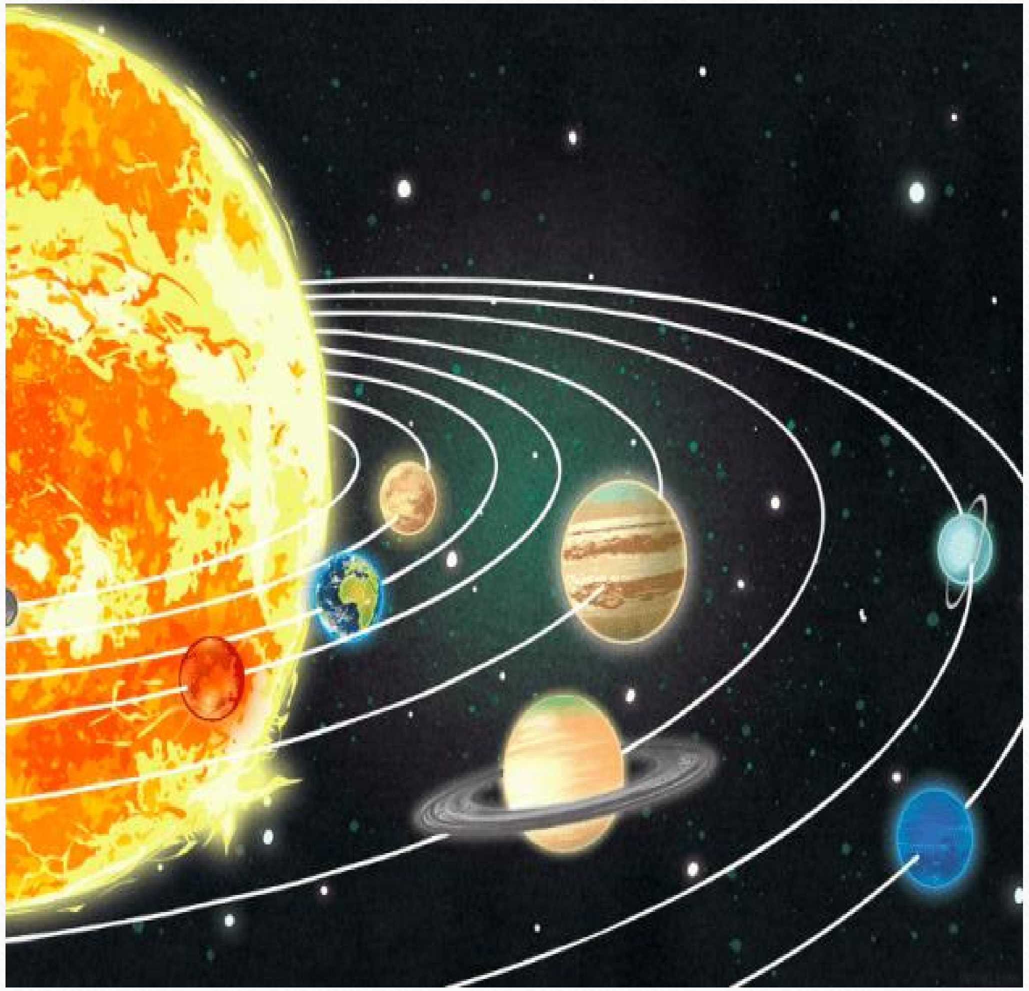
- Star Initialization: 400 stars are randomly positioned and assigned brightness values.
- Planetary Setup: Each planet is positioned with its own orbital radius and rotational speed.

Key Features:

Planetary Movement: Each planet orbits the Sun based on a speed factor, calculated through rotation transformations.
Moons: The Earth's moon orbits Earth, also controlled by orbital speed.

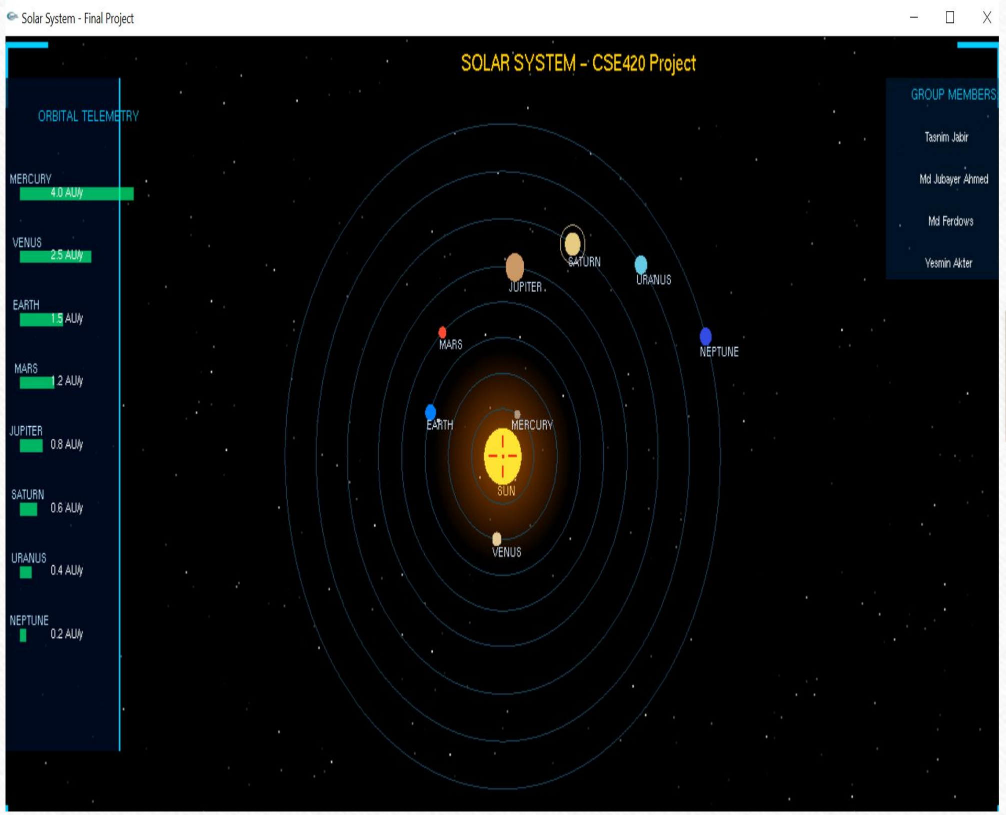
Sun Representation: The Sun is drawn centrally with a glowing effect.

User Interaction:
Camera Controls: Arrow keys (or WASD) for moving the camera, and zoom controlled by W/S keys.
Simulation Speed: +/- keys adjust the speed of the planetary movements.



Display:

- Real-time status display showing group members, simulation speed, and planet data (speed, distance).
- Static stars are drawn in the background to simulate the night sky, twinkling based on the global angle.
- Input handling system for camera controls
- Text rendering system for labels
- Orbital path visualization
- Implementation of transformation hierarchy
- Development of planet rendering system


Solar System - Final Project

SOLAR SYSTEM - CSE420 Project

ORBITAL TELEMETRY

MERCURY	4.0 AU/y
VENUS	2.5 AU/y
EARTH	1.5 AU/y
MARS	1.2 AU/y
JUPITER	0.8 AU/y
SATURN	0.6 AU/y
URANUS	0.4 AU/y
NEPTUNE	0.2 AU/y

GROUP MEMBERS

Tasnim Jabir

Md Jubayer Ahmed

Md Ferdows

Yesmin Akter

CORE ALGORITHMS & FORMULAS

- **Orbital Motion:**

✓ CurrentAngle = GlobalAngle × PlanetSpeed × SimulationFactor •

✓ Planet Position: glRotatef(angle) →glTranslatef(orbitRadius)

- **Circle Generation:**

✓ 60-segment polygon approximation

✓ radius × cos(2πi/60)

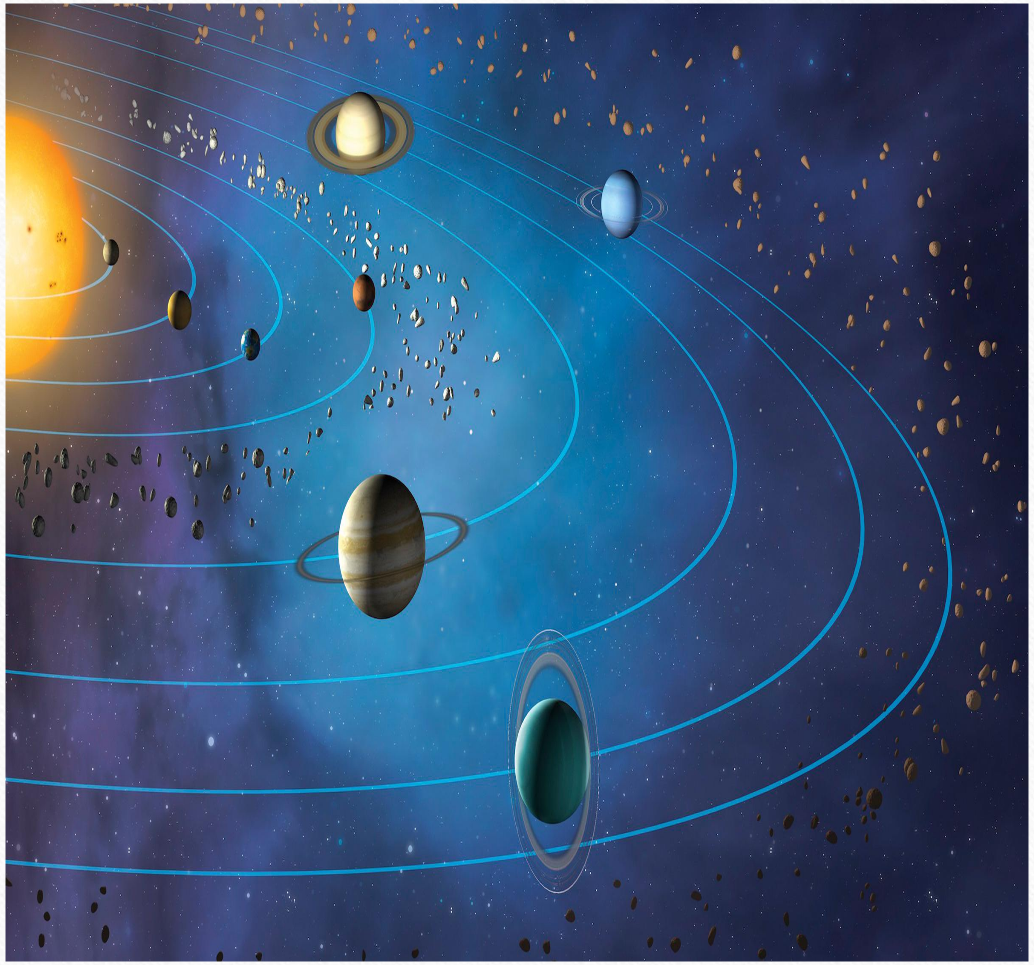
✓ y = radius × sin(2πi/60)

✓ Used for planets, sun, and orbit.

- **Text Positioning:**

✓ Center offset = (text_length × 0.01) / 2

✓ Counter-rotation for planet labels



- **Dynamic Effects:**

✓ Star twinkling: brightness = base + sin(time × 0.05 + index)

- **Sun glow:**

✓ Radial gradient using triangle fan

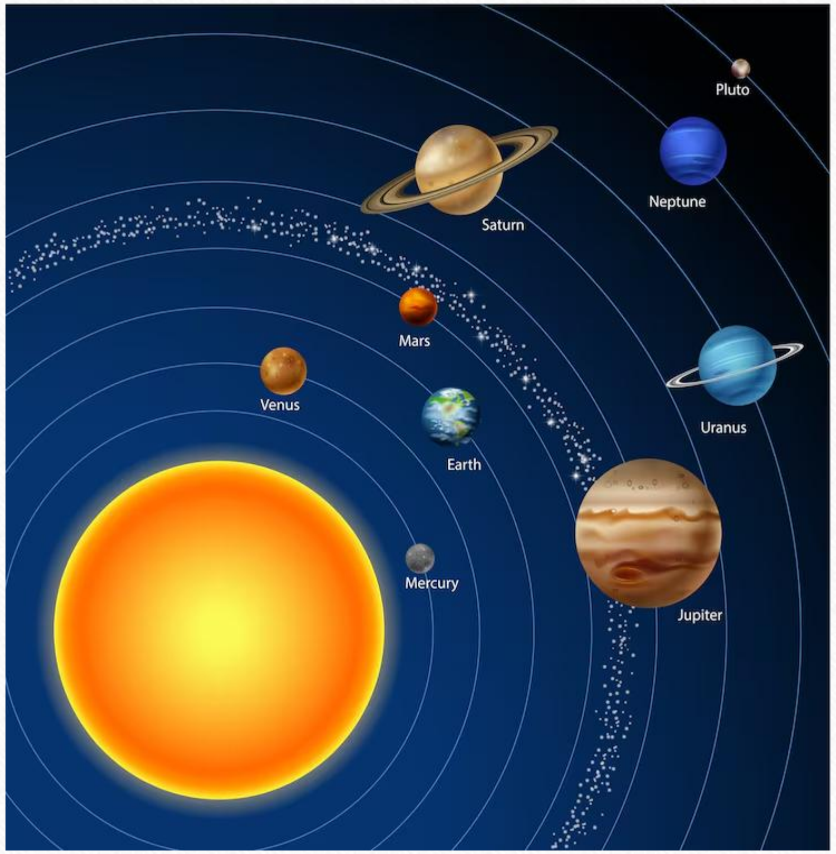
✓ Pan speed = 0.1 / zoomScale (adaptive control)

- **Matrix Operations:**

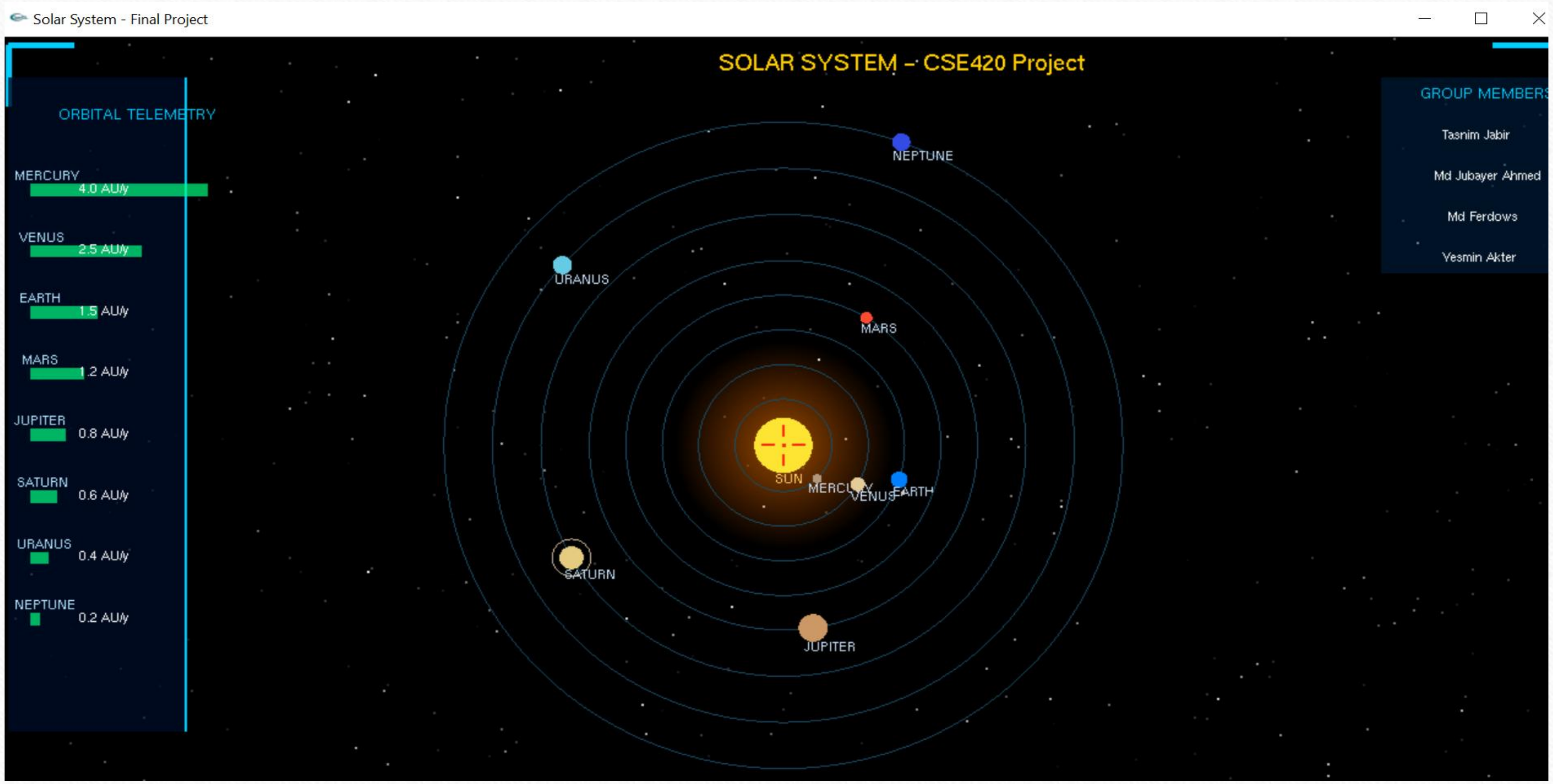
✓ Push/Pop Matrix for hierarchical transformations

✓ Model-View matrix for object positioning

✓ Projection matrix for HUD overlay



Screen shot and Demo:



SOLAR SYSTEM – CSE420 Project

ORBITAL TELEMETRY

MERCURY	4.0 AU/y
VENUS	2.5 AU/y
EARTH	1.5 AU/y
MARS	1.2 AU/y
JUPITER	0.8 AU/y
SATURN	0.6 AU/y
URANUS	0.4 AU/y
NEPTUNE	0.2 AU/y

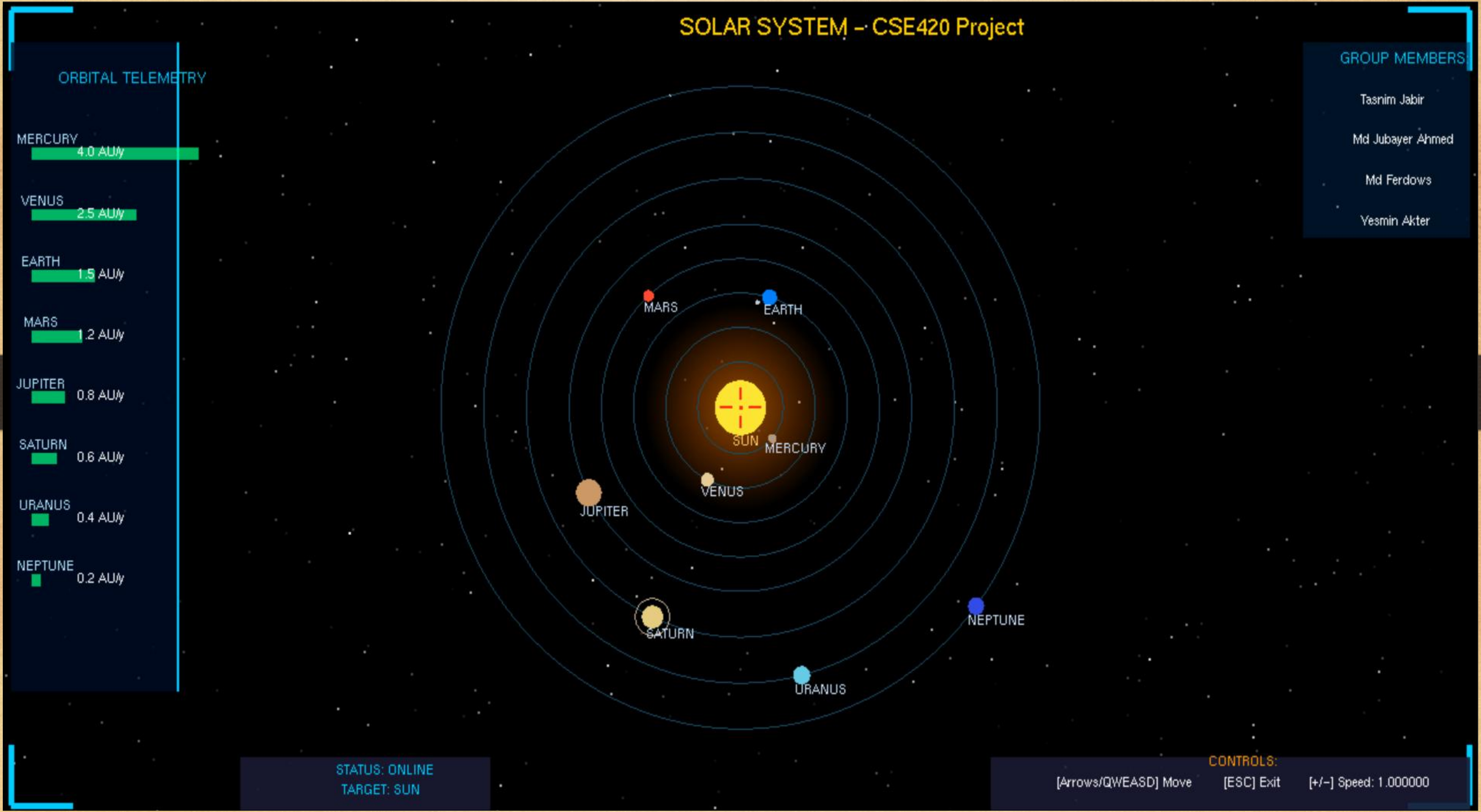
GROUP MEMBERS

Tasnim Jabir

Md Jubayer Ahmed

Md Ferdows

Yesmin Akter



SOLAR SYSTEM – CSE420 Project

ORBITAL TELEMETRY

MERCURY	4.0 AU/y
VENUS	2.5 AU/y
EARTH	1.5 AU/y
MARS	1.2 AU/y
JUPITER	0.8 AU/y
SATURN	0.6 AU/y
URANUS	0.4 AU/y
NEPTUNE	0.2 AU/y

GROUP MEMBERS

Tasnim Jabir

Md Jubayer Ahmed

Md Ferdows

Yesmin Akter

STATUS: ONLINE
TARGET: SUN

CONTROLS:
[Arrows/QWEASD] Move [ESC] Exit [+/–] Speed: 1.000000

Conclusion

This Solar System project successfully implements core computer graphics principles using OpenGL/GLUT. The code demonstrates effective hierarchical transformations for planetary motion and professional UI design with comprehensive HUD elements.It achieve smooth FPS animation through efficient rendering techniques and provides intuitive camera controls for user interaction. The project showcases practical application of mathematical concepts in computer graphics while maintaining clean, modular code structure suitable for educational demonstration.