KOÇ UNIVERSITY

Solar System Simulation in OpenGL

Muhammad Aditya Sasongko
Najeeb Ahmad
Burak Bastem

Koç University, Istanbul, Turkey

Project Overview

- Solar system simulation
  - Simulates the Sun and eight planets revolving around the Sun
  - Simulates satellites for two of the planets, namely, Earth and Jupiter
  - Allows user navigation and simulation speed control

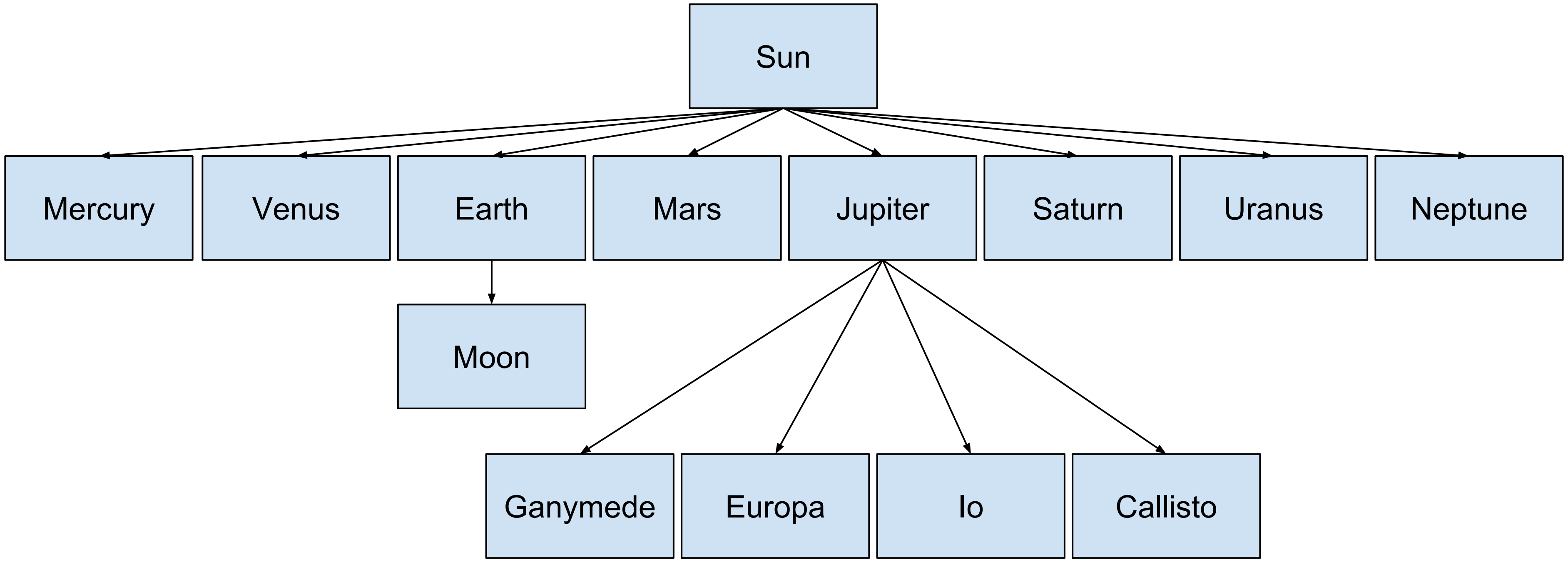
Project Motivation

- To apply most of the techniques learnt in the course
  - Object instantiation
  - Translation, rotation and scaling
  - Texture mapping
  - Lighting and Shading
  - Transformations
  - Hierarchical Modeling

Project Team

- Burak Bastem
  - Hierarchical modeling, navigation
- Najeeb Ahmad
  - Texture mapping, shading
- Muhammad Aditya Sasongko
  - View Transformations, picking, shading

Hierarchical Modeling



Navigation

- Camera movement
  - Forward, backward (Zoom in/out)
  - Right, Left, Up, Down
- Camera rotation
  - Up, Down, Right, Left

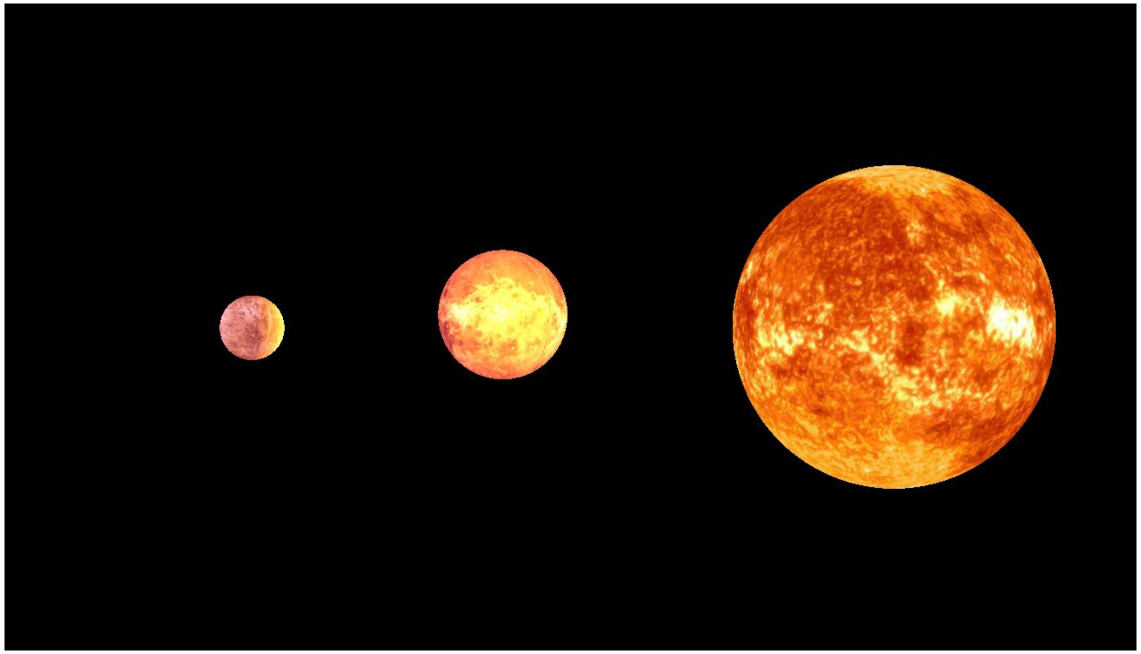
- View: LookAt(eye, at, up)
  - eye point is determined with polar coordinates.
    - elevation
    - azimuth
- Projection: Orthogonal

Texture Mapping

- JPEG images for planets/satellites
- Used FreeImage library for image input
- Texture function
  - Loads image
  - Generates texture object
  - Sets texture parameters
  - Assigns texture objects to TexID field of each astronomical object
  - Bind appropriate texture object just before drawing the object

Shading

- Sun as point source of light
- Phong shading and reflection models
- Distance term



Transformations

- Transformations rely on the attribute values of AstronomicalObject instance
  - equatorial_radius for scaling,
  - rotation_period and RotationTheta for rotation around object's own axis,
  - TiltingAngle for tilting object with respect to world coordinate,
  - average_orbit_distance for translating object from origin,
  - orbit_period and RevolutionTheta for revolution around parent object.

Transformations

- Transformations for planet:
  - planet_last_transformations = revolution_rotation_matrix * translation_from_origin
  - planet_transformations = planet_last_transformations * tilting_rotation_matrix * rotation_matrix * scaling_matrix

Transformations

- Transformations for satellite
  - satellite_transformations = planet_last_transformations * revolution_rotation_matrix * translation_from_origin * tilting_rotation_matrix * rotation_matrix * scaling_matrix

Picking

- Implemented with mouse click event callback function
- Mechanism:
  - when mouse is clicked, each object will be rendered briefly with a single color.
  - Each object will be rendered with a different color.
  - RGB value of each object will be read with glReadPixels function.
  - Since each object has a unique color, the retrieved RGB value can be used to identify that object.

Demo