

Solar System Simulation Report

Overview

This report documents the current implementation of a Solar System simulation using **Pygame** and **OpenGL**. The simulation visualizes celestial objects such as planets, their satellites, asteroid belts, and comets in a 3D environment with lighting, textures, and camera movement.

Features

Visuals

- **High-resolution textures (4K):** Applied to planets, the Sun, and other celestial objects to enhance realism.
- **Dynamic lighting:** The Sun serves as a light source, with diffuse and specular lighting affecting the scene.
- **Starry background:** A rotating textured sphere simulates a dynamic starry background.

Celestial Objects

1. Planets

- Each planet has properties including size, orbit radius, speed, axial tilt, and texture.
- Nine planets are included (Mercury, Venus, Earth, Mars, Jupiter, Saturn, Uranus, Neptune, and Pluto).
- Satellites are defined for Earth and Jupiter.

2. Satellites

- Earth has two satellites defined (textures to be added in the future).
- Jupiter includes four major satellites (Io, Europa, Ganymede, and Callisto).

3. Comets

- Elliptical orbits for comets (e.g., Halley and Encke).
- Defined with semi-major and semi-minor axes.

4. Asteroids

- An asteroid belt is included with randomized orbit radii and speeds.

Dynamic Environment

- **Camera Movement:**
 - Keyboard-controlled camera position (W, A, S, D, arrow keys).
 - Preset camera distances (R, T, Y, U keys).
- **Time Scaling:**

- Adjust time scale (1 to 4 keys) for orbit speeds.
- **Interactive Modes:**
 - Pause or resume time (0 key).
 - Camera reset options.
- **Quit**
 - Terminate program with ESC key.

Future Enhancements

- Support for toggling fullscreen and windowed modes.
 - Addition of planetary rings with textures (e.g., Saturn).
 - Enhanced satellite visuals with specific textures.
 - Improved comet tails and asteroid belt density.
-

Technical Details

Libraries Used

- **Pygame:** Manages the main event loop, keyboard inputs, and sound effects.
- **PyOpenGL:** Handles 3D rendering, including lighting, textures, and transformations.

Key Functionalities

1. Texture Loading

- `load_texture`: Loads and applies 2D textures to celestial objects.

2. Sphere Rendering

- `draw_sphere`: Generates textured spheres using OpenGL's `gluSphere`.

3. Saturn Rings Rendering

- `draw_rings`: Creates quad-strip-based rings for Saturn.

4. Comet Orbits

- Implements elliptical orbit paths with adjustable semi-major and semi-minor axes.

Event Handling

- **Keyboard Events:**
 - Camera controls and time scaling handled through `KEYDOWN` events.
-

Assets

Textures

- **Planets:** High-resolution images (`earth.jpg`, `moon.jpg`, etc.) stored in the `textures4k` directory. And also including `textures2k` for 2K images.
- **Stars:** Background image simulating outer space.

Audio

- **Background Music:** A looping MP3 file (`music/main.mp3`) provides ambiance.
-

Known Issues

- **Performance:** High-resolution textures may impact performance on less powerful hardware.
 - **Fullscreen Mode:** Currently fixed in fullscreen; toggling to windowed mode is not implemented.
 - **Lighting:** Intensity can sometimes oversaturate specific textures.
-

Conclusion

The Solar System simulation provides an interactive 3D environment showcasing celestial mechanics. Future iterations aim to enhance realism and interactivity, building on the robust framework established in this version.

Appendix

- **File Structure:**
 - `textures4k/`: High-resolution textures.
 - `music/`: Audio files.
- **Dependencies:**
 - Python 3.11.10
 - Pygame 2.6.1
 - PyOpenGL 3.1.7
 - SDL 2.28.4