CSE423: Final Project Report

Spring 2023

Submitted by:

Mahmudul Hasan Mitul
ID: 18101066

Shah Md. Shakhawath Hossain
ID: 18101133

Md. Asiful islam
ID: 17201077

Group ID:
11

Section:
01

Solar System with OpenGL

Introduction:

The Solar System is an intriguing and complex concept. Using Pygame and OpenGL, our project seeks to produce a simulation of the solar system. Pygame is a multi-platform package created for Python game development. An open-source graphics library called OpenGL offers 3D graphics and other visual effects. In this project, we simulate the solar system in two dimensions using Pygame, OpenGL Library specified shapes, Mid point lines, Mid point circles, and their transformations. The solar system consists of the sun and the four planets Mercury, Venus, Earth, and Mars.

Project Description:

The simulation consists of 1000 randomly produced stars on a dark blue background. Using the draw_circle function, which creates a circle with the specified center, radius, and color, the four planets and the sun are depicted. The sun and the planets are in distinct locations, hues, and radii. The planets have smaller radii than the sun, which is yellow and has the biggest radius. The placements, colors, and rotational velocities of the planets vary. A miniature spaceship that the user can pilot using the keyboard's arrow keys is also included in the simulation. The draw_triangle function is used to create a triangle with the specified position, size, and color to represent the spaceship.

The simulation is interactive, and the user can control the movement of the triangle using the arrow keys. The user can move the triangle in all four directions: left, right, up, and down.

Implementation:

The implementation of the project is done using Python 3.7, Pygame 2.0.1, and OpenGL 3.3. The project is organized into several functions that handle different aspects of the simulation. The draw_circle function is used to draw circles, and the draw_triangle function is used to draw triangles. The main function is the main loop of the simulation, which handles the events and updates the positions of the planets and the triangle. The glOrtho function is used to set up the projection matrix, and the glBlendFunc function is used to enable alpha blending. In this code, we have used various algorithms and transformations for the animation. For drawing circles, we

have used the midpoint circle algorithm, which is an efficient way to draw circles with pixels. For drawing triangles, we have used the line drawing algorithm with Bresenham's algorithm. To create the animation of the planets, we have used transformations in the form of translations and rotations.

The simulation uses a clock to control the frame rate of the simulation. The clock is created using the pygame.time.Clock function, and it is set to run at 60 frames per second.

Results:

The simulation is a visually appealing and interactive program that simulates the Solar System in 2D. The planets move in circular orbits around the sun, and they rotate around their own axes. The sun also rotates around its axis. The triangle moves smoothly in all four directions and can be controlled by the user. The randomly generated stars in the background create an immersive experience.

Conclusion:

In conclusion, the Solar System simulation using Pygame and OpenGL is an interesting project that demonstrates the capabilities of these libraries. The simulation is an interactive program that simulates the Solar System in 2D. The program is visually appealing, and the randomly generated stars in the background create an immersive experience. The project can be extended to include more planets, moons, and asteroids, and it can be used as an educational tool for teaching astronomy to students.

Graphical representation:

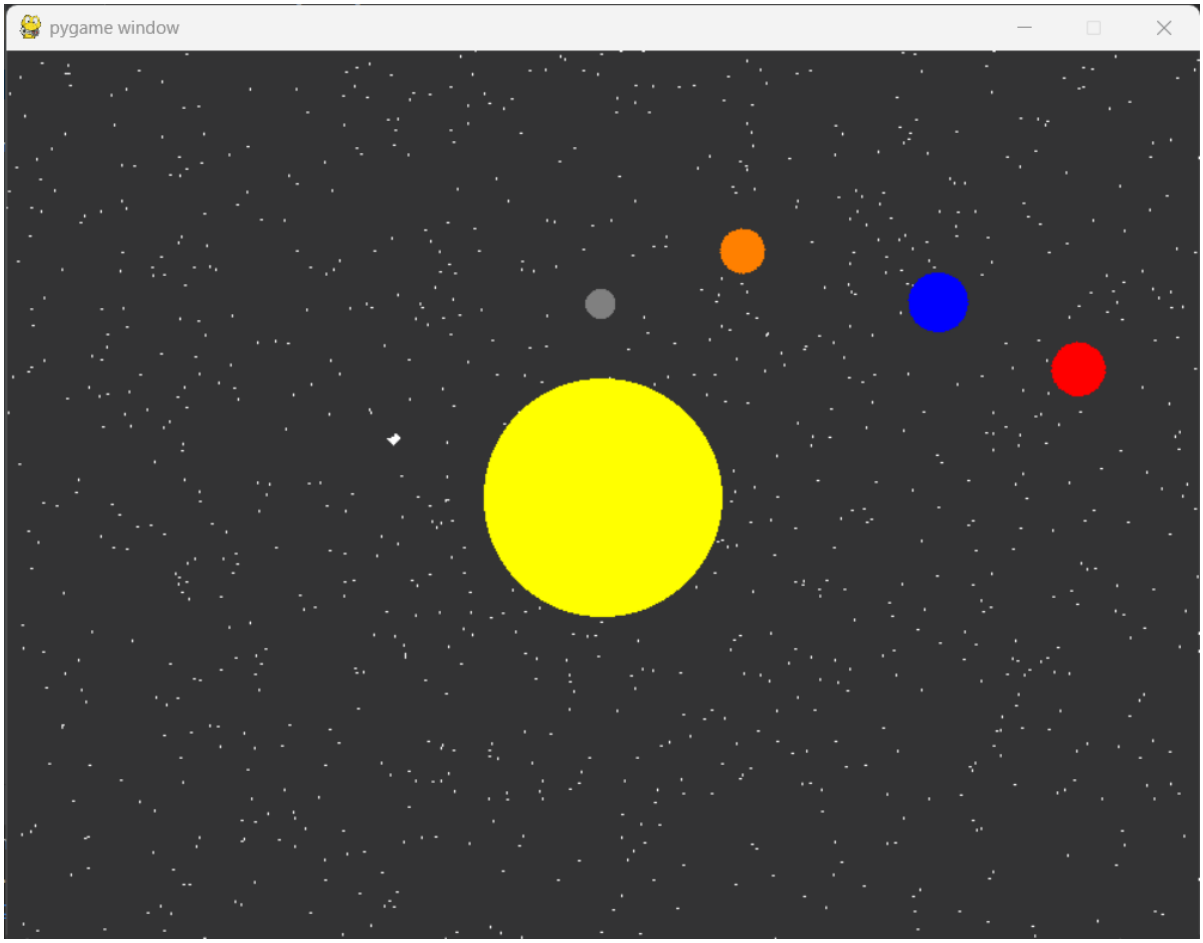

Fig: Solar System with OpenGL