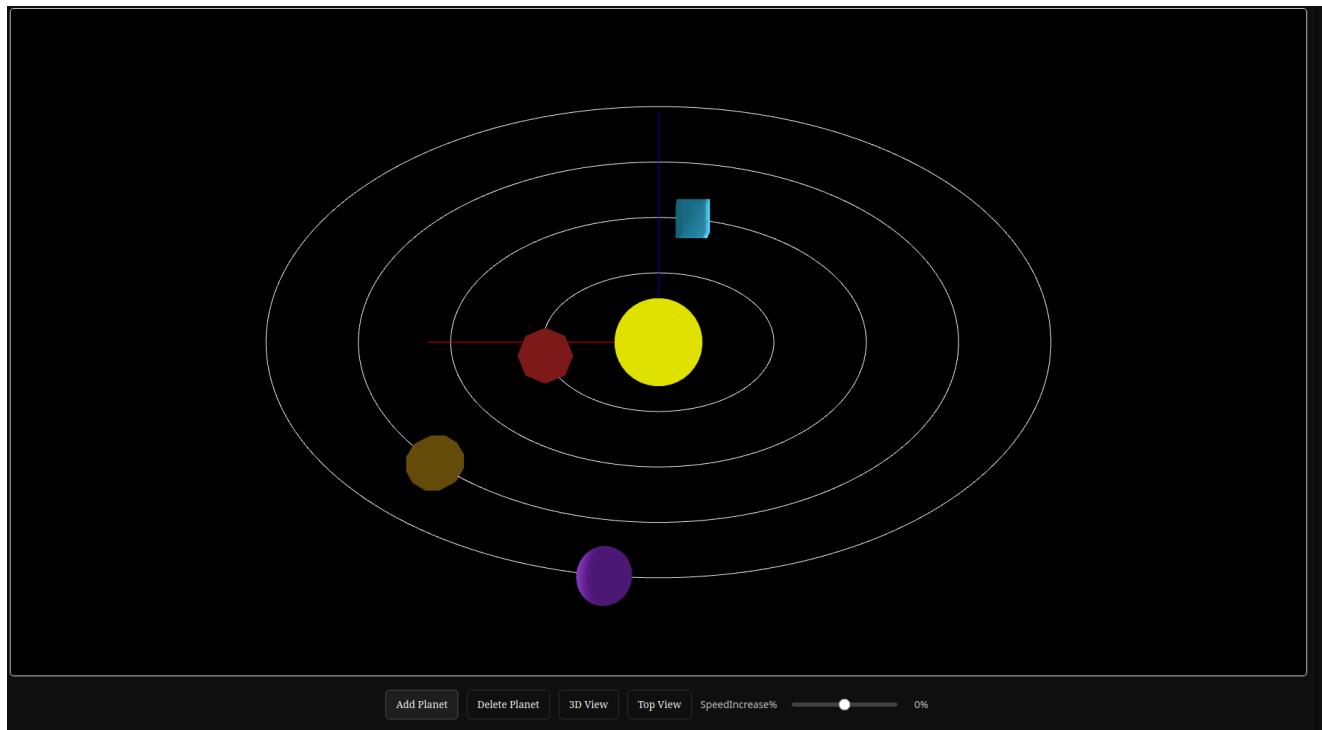
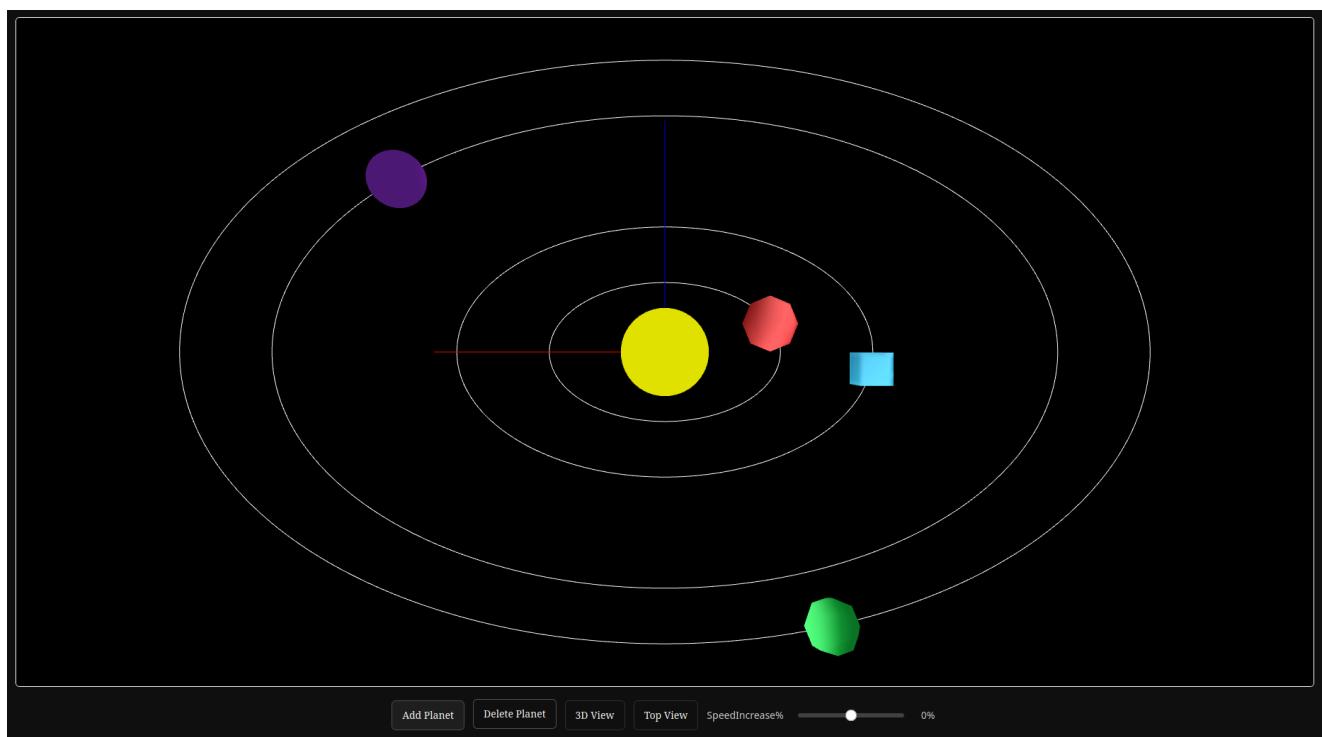


Assingment2-CG-Report

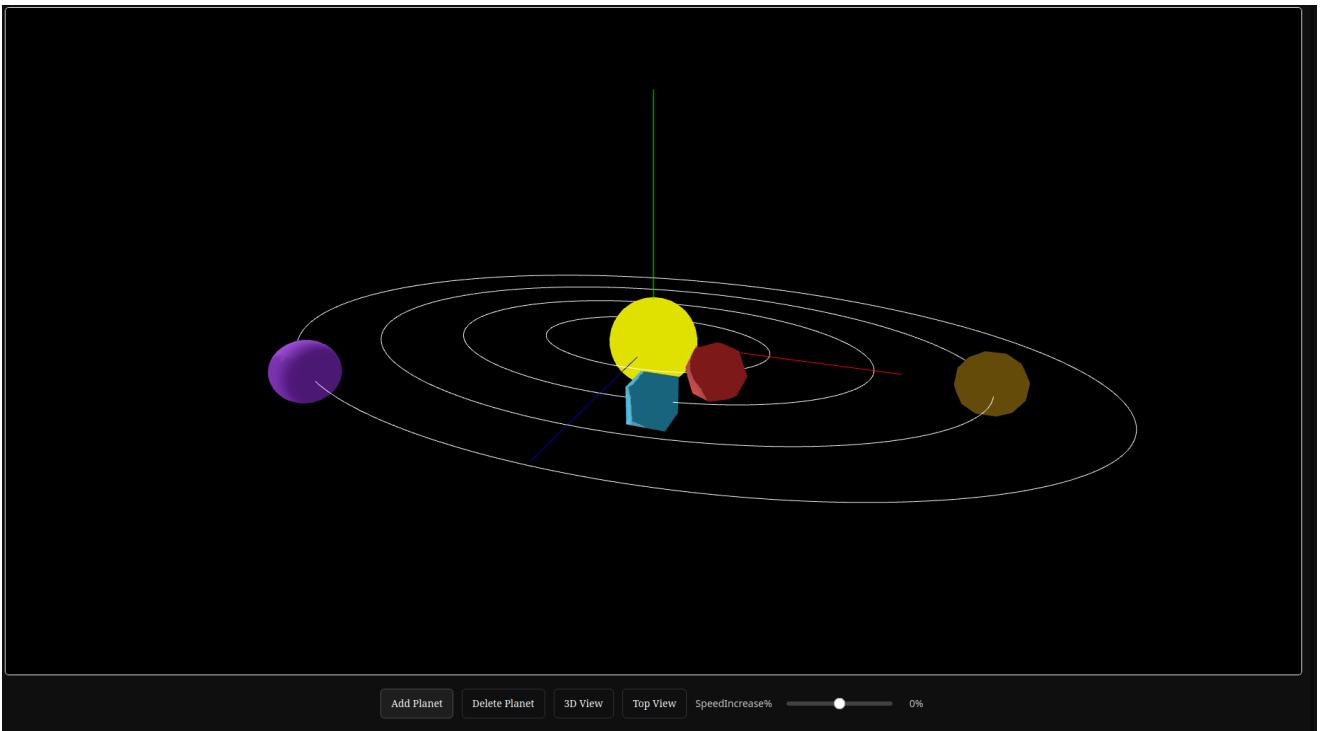
ScreenShots



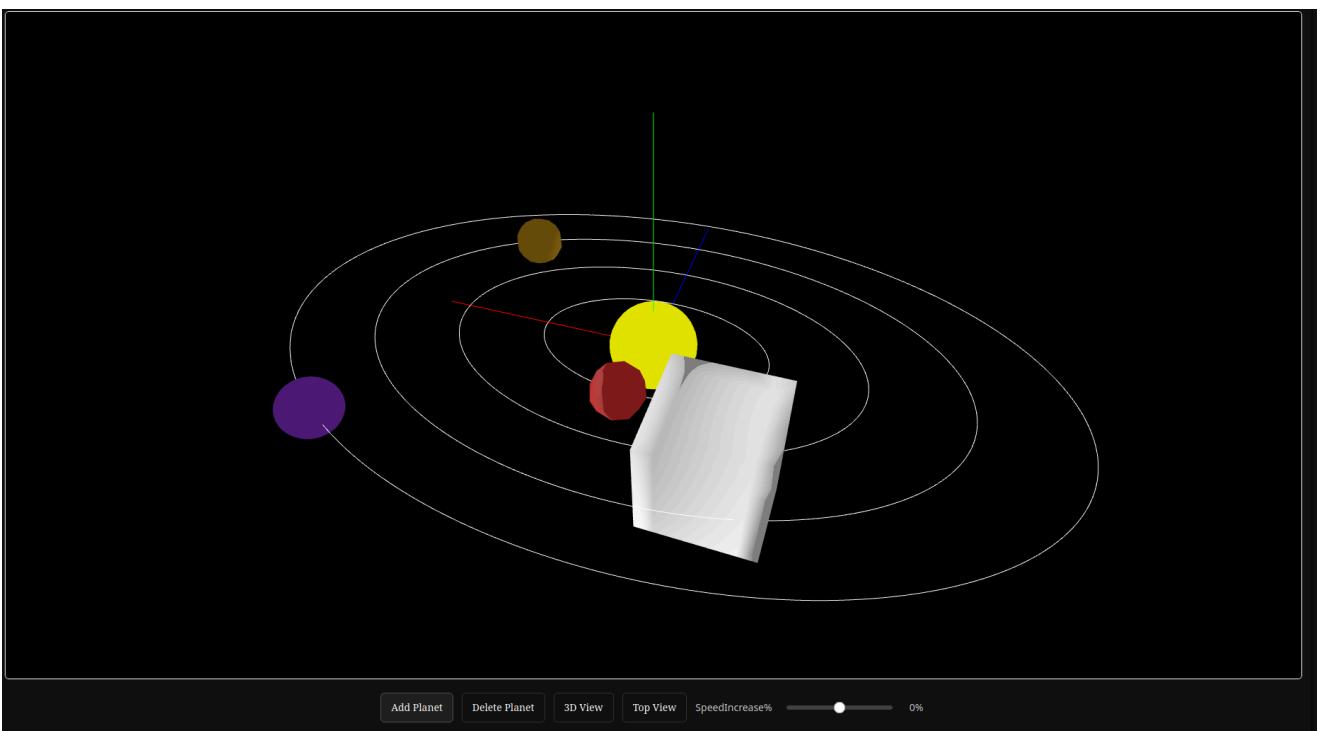
Top View from camera



Adding and Deleting Planets



3d view that allow camera rotations



transformations of planets

Implementation Overview

Design Choices

- Perspective Projection
- Smooth Shading Models for planets
- modeled planets, star by using blender

- modeled orbits as a parametric curve
- ray picking for detection of object clicked on the screen

Classes

- *Renderer* - For loading and drawing object buffers and setting up uniforms and attributes
- *EventRecord*- for handling of user inputs
- *Planets,Star,Orbit* - for storing object specific details and for drawing logic
- *Camera* - for generation of view and projection matrix along with camera transformations
- *PointSourceLight* - for generation of Illumination Matrix

1. To what extent were you able to reuse code from Assignment I?

code for the following was reused in this assignment

- The code for handling translation,scaling and rotation of objects that included code for calculation of model matrix.
- code that includes loading up the buffer arrays and drawing the arrays for most of the objects
- code for parsing a obj file was just copied from assignment I.

2. What were the primary changes in the use of WebGL in moving from 2D to 3D?

- The first major change was the handling and usage in the vertexshader program of two new matrices projection and view matrix compared to just having just a one model matrix in 2d
- Not just now in 3D having vertex positions but also vertex normals to handle and load up as attributes in shader programs
- Another major change was implementing the smooth shading model in the fragment Shader program along with defining models for LightSource and material properties for objects

3. How were the translate, scale and rotate matrices arranged? Can your implementation allow rotations and scaling during the movement?

*modelMatrix = T * R * S . First all three translation,scaling and rotation matrices were generated then multiplied in above order to get model matrix. Yes my implementation allows rotation and scaling during the movement but as the assignment pdf suggest i have disabled rotated and scaled planet when it rotates.*

4. How did you ensure that there are no conflicts when adding/deleting a planet along with its orbit

Orbit is made a attribute of the planet so the adding and deletion operations on planets are therefore applied on the orbit as well. The draw operation of planet has a function call to the draw function of orbit.

Credits

1. gl-matrix- <https://cdn.skypack.dev/cdt2d> used for matrix and vector representation and calculations.