

Computer Graphics: Assignment 3 Part 2

Inner Solar System Simulator

Enrico Fini

1 Report

1.1 Usage

To run the simulation:

```
./assign3_part2
```

Keyboard controls:

1 : object viewer camera;

2 : world object viewer camera;

3 : custom camera that follows planets;

S : toggle Sun texture (procedural/mix);

9 : planets rotation slows down;

0 : planet rotation speeds up.

M : camera follows Mercury (in custom camera mode);

V : camera follows Venus (in custom camera mode);

E : camera follows Earth (in custom camera mode);

N : camera follows Mars (in custom camera mode);

UP : camera translates up perpendicularly to the planets plane(in custom camera mode);

DOWN : camera translates down perpendicularly to the planets plane(in custom camera mode);

Z : camera zooms in (in custom camera mode);

Furthermore, to enable sound in a secure way (I couldn't test on Windows or Linux), if you are running MacOS you can open *model-view.cpp* file and edit line 31 with "ON".

1.2 Features

Here are the features of the simulator:

- **Point Light:** the Sun is a point light located at the center of the world coordinate system. The planets feature texture and material properties;
- **Multiple Cameras:** the simulation features 3 different cameras:
 - ObjectViewer: provided by you;
 - WorldObjectViewer: provided by you;
 - FollowViewer: the camera is located on the rotation axes of the solar system and follows the selected planet.
- **Texture Mapping:** for every planet both diffuse texture and specular texture are provided. For what concerns the sun I couldn't find a satisfying texture, so I decided to mix two different textures at runtime, the result can be shown pressing "S";
- **Skybox:** a skybox with stars has been set up in the skybox shader;
- **Multiple Vertex/Fragment Shaders** 3 shaders:
 - mview shader: provides texturing, lighting;
 - skybox shader: enables the skybox;
 - procedural shader: create the procedural texture for the sun.
- **Procedural Generation of Textures:** I used Simplex Noise to generate a procedural texture for the Sun. Since it is a 2D texture but it has to be placed on a sphere I computed longitude and latitude and used Mercator projection to generate x and y coordinates from the angles. To make the texture as uniform as possible I divided the sphere in patches using coordinates and generated the texture on the most suitable coordinates. You can find the code in the file *procedural.frag*;
- **Particle System:** I formalized the revolution movement of the planets using a particle system. My implementation of gravitational force is similar to the repulsion force you showed us. I also tried to add code to take into account mass but the system started to behave randomly and I couldn't find out why. In this implementation the mass of the planets is considered unitary.
- **Sound:** I added a background song to the simulation to make it more majestic. Unfortunately I don't own any Windows or Linux laptops so, if you have a MacOS machine and want to enable this feature you can edit line 31 in *model-view.cpp* file.

NOTE: only the ratios between the sizes of the planets are meaningful but not wrt the size of the Sun. I tried to draw the objects in real scale but the planets ended up to be too small and almost impossible to see.