3D SCIENTIFIC SOLAR SYSTEM

Computer Graphics Course – 2022/2023

3rd Year – Second Term

AI Department

15/05/2023

Faculty of Computers and AI

Benha University

3D Scientific Solar System

Computer Graphics

2022 / 2023

Supervised by:

Dr. Rasha Orban

Eng. Mariam Mahmoud

يوسف طارق عبدالمعبود البارودى	
منة محمد إبراهيم ابوالخير	علي أسامة عابدين عمر
رضوى محمد سعيد عبدالشافي	رنا حسن بدراوي شهاب الدين

Contents

Table of Figures

1 Introduction

Our project for Computer Graphics Course was a little bit difficult, and after a lot of search about OpenGL projects we got into the Scientific Solar System in 3D manner, which is introduced in elementary schools for students, with features of moving into the space and discover more about our solar system, how the planets are moving around our sun, what is the shape of other planets, how an asteroid could be existed in our solar system and much more.

The project is divided into many topics that has been learned through the course, which is the main components of the project as the following:

1. **3D Coordinates**
2. **Animation**
3. **Textures**
4. **Collision**
5. **Keyboard/Mouse Event**

These components are applied perfectly in the project as we will see later through this documentation, starting with the Space designing with the sun and Planets to the rotations and lot more, and the source code is available to be modified later with the textures, which making it easy to improve it.

Note that the controlling of the Space World will be explained through this documentation in the Last Page.

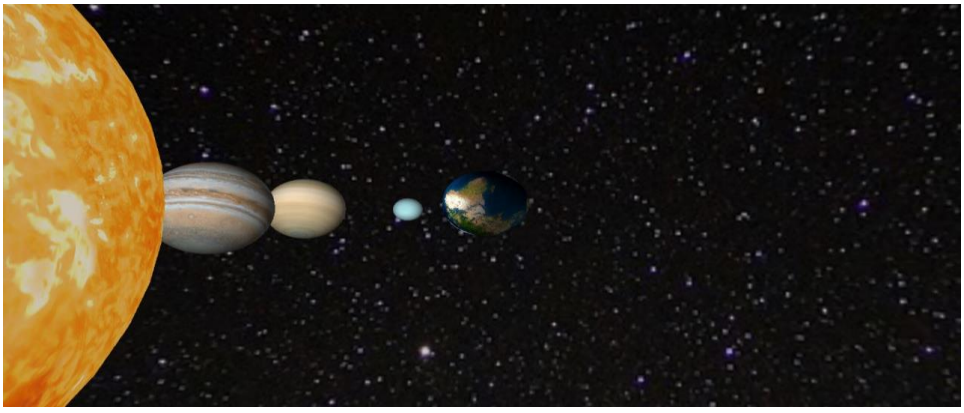

Figure 1: A Snapshot of the Space World

2 World Designing

The system is designed starting with the planets textures, for each planet such as Mars, Earth, Neptune, Jupiter and more, and sure with the Sun as the center of our Project, surely the planets and sun are Spherical shapes, and it is actually the same in C++ OpenGL, all planets, sun and universe are spherical.

2.1 Planets Textures

At this section, we will show how the textures of the planets is look like (not all planets will be shown),

Figure 2: Jupiter Texture with the Common Landmark

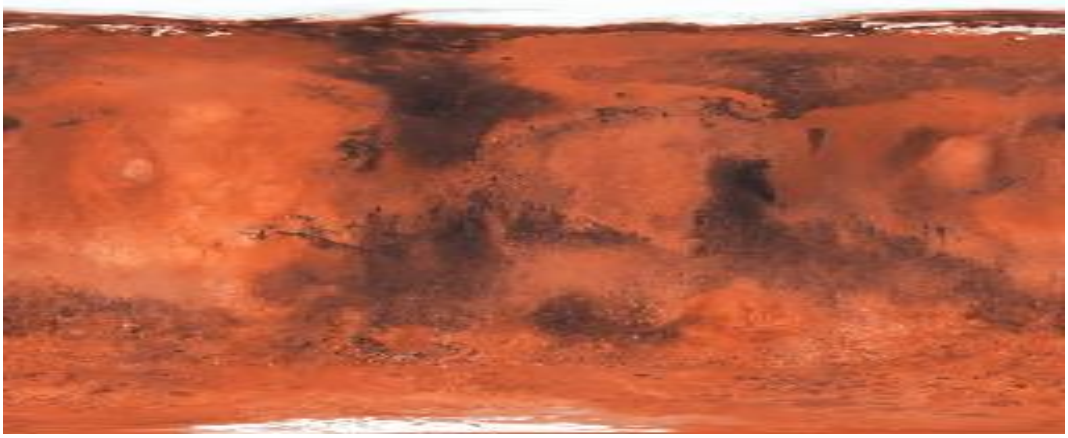

Figure 3: Mars Texture showing the red color of it

2.2 Sun and Universe Textures & Designing

The designing of the universe around the solar system was unique due the applied shape of it, for closer look, the universe is constructed as Sphere shape for smooth angles, applied with the required texture that makes the starts and black holes, it gives a real space effect.

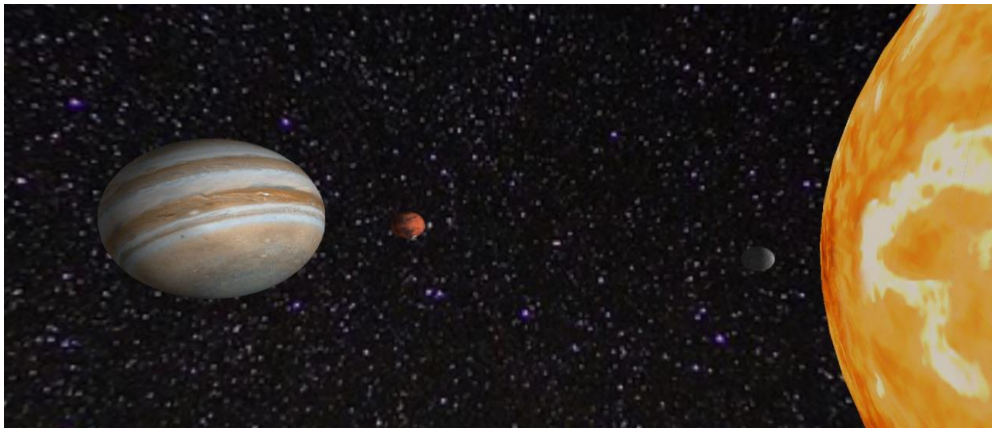

Figure 4: The Universe texture with millions of starts behind the solar system.

The same is for the sun. the sun is created at the center of the 3D World without translating it, and make its radius too large (as a scale to a real one) and putting a texture of the sun on it.

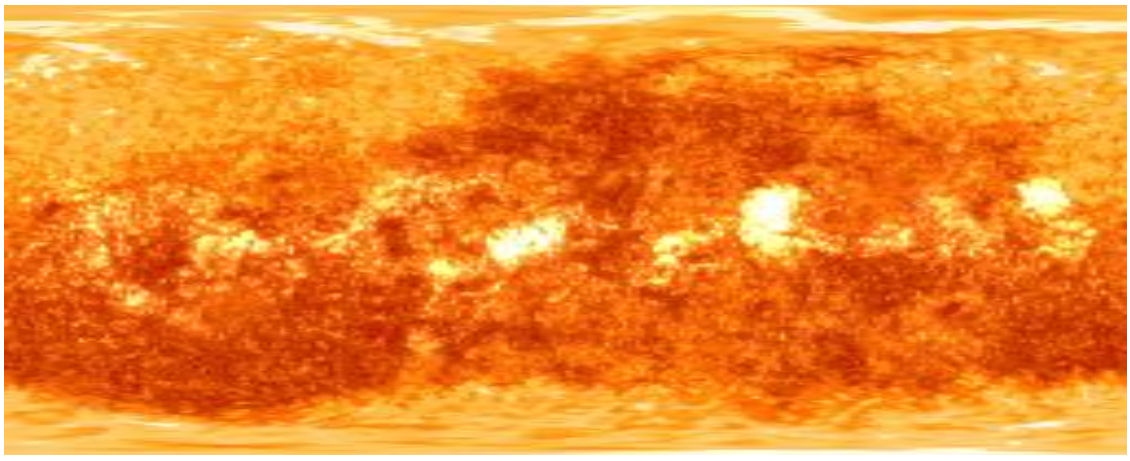

Figure 5: The Texture of the Sun

2.3 The Light Effect

The light effect is the most essential component of the Solar System, where the sun is source of illumination and planets are illuminated, but at our project, we faced some problems such as, programmatically the sun is an object not a light source, so how could we solve that ?

Putting the light source at the center (original) coordinates of Space makes it much easier, which means the position of the light source is the same as the Sun ! now by modifying the material settings of the Sun, it will be acting as a light source, and the material settings of the planets are modified to get the light-shadow effect on each planet.

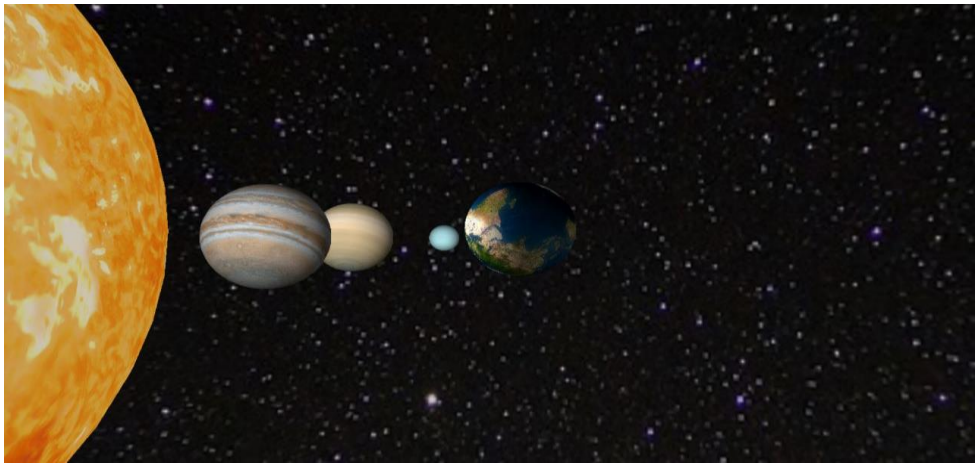

Figure 6: The light effect is shown on Earth.

3 Animation

The planets Motion is designed carefully with trigonometric laws to guarantee the efficient and correct motion for each plane individually, as known as the multiplication of rotational angle and the rotational speed. Further, each planet has its unique speed around the sun (the orbit) which is the radius of the orbit is the distance between the sun and each planet.

To simplify our task, when creating the planet, we will rotate the planet around the origin point (which is the center of sun) then translating N units on Z-Axis far from the origin of the world , and now we got a planet that rotate around the sun !

Now we can control the speed of each planet individually to determine how much it will take to complete a period around the sun (Time Period) which is defined by the speed of rotating.

These details are defined more clearly in the C++ source code of the project, which is organized with comments explaining each part of the code.

```
// دالة موحدة لرسم الكواكب ، مع تحديد نصف القطر ، بعدها عن المركز ، والزاوية الدورانية
void drawPlanet(GLfloat radius, GLfloat distance, GLfloat red, GLfloat green, GLfloat blue, GLfloat rot) {

    glPushMatrix();
    sphere = gluNewQuadric();
    gluQuadricTexture(sphere, GL_TRUE);
    gluQuadricDrawStyle(sphere, GLU_FILL);
    gluQuadricNormals(sphere, GLU_SMOOTH);
    glRotatef(rot, 0.0f, 1.0f, 0.0f);          // (عشان تلف حوالين السنتر الأساسي (الي هو الشمس
    glTranslatef(0.0f, 0.0f, distance);        // (بعدها عن السنتر (بعدها عن الشمس
    glRotatef(300, 1.0f, 0.0f, 0.0f);          // تدوير الكوكب حوالين محور إكس ، عشان اعدل شكل الكوكب
    glRotatef(rot, 0.0f, 0.0f, 1.0f);          // دوران الكوكب حوالين نفسه (محور زد
    glColor3f(red, green, blue);
    gluSphere(sphere, radius, 32, 32);         // إنشاء الكوكب بنصف القطر المحدّد
    gluDeleteQuadric(sphere);
    glPopMatrix();
}
```

Figure 7: A part of code showing the creation of planets and how they are oriented around the sun (orbit)

4 Normal Mode & Collision Mode

As part of the simulated space, we made a simulation for an asteroid hitting mars with simple way, which divides the project into two modes: **The Normal Mode** and **Collision Mode**.

4.1 Normal Mode

At the normal mode, the planets are rotating around the sun, with a Free Camera that you can control, and discover the space as much as you want to.

If you are about to toggle to the Normal Mode (from Collision Mode) then all your settings of the world will be restored normally.

4.2 Collision Mode

The collision mode is a simple simulation of an asteroid which is getting closer and hitting Mars. It is too simple but enough for kids to understand how this could happen.

In a short duration of time, an asteroid will move toward mars until hitting it, and simply a message will be shown to get return to the Normal Mode.

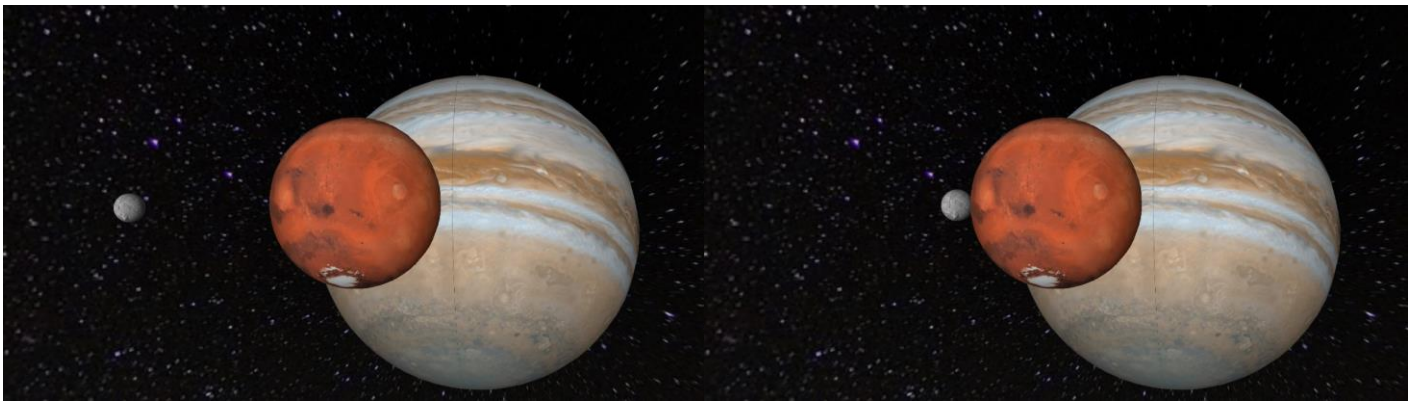

Figure 8: On the left, the asteroid is moving toward mars and getting closer. then on the right, when the asteroid reaching mars

5 Control

Key	Description
Ctrl	Decreasing the speed of the Camera Movement
Shift	Increasing the speed of the Camera Movement
w	Moving the camera forward
s	Moving the camera backward
a	Moving the camera to the left side
d	Moving The camera to the right side
PAGE UP	Move the camera to up (Y-Axis)
PAGE DOWN	Move the camera to down (Y-Axis)
p	Increasing the rotational speed of the Planets
o	Decreasing the rotational speed of the Planets
m	Toggle to the Collision Mode
n	Toggle to the Normal Mode

NOTE ! in normal mode, it is free camera, you could control it with mouse movements with static sensitivity.