**foggs ASSIGNMENT 2**

**NAME: Haroon Nasir**

**Student Number: 20016320**

**3D Game: Solar System**

Table of Contents

[Glossary 1](#_Toc70063339)

[Key Words 1](#_Toc70063340)

[Introduction 2](#_Toc70063341)

[Controls/Instructions 2](#_Toc70063342)

[UML Diagrams 3](#_Toc70063343)

[Critical Reflection 5](#_Toc70063344)

[Bibliography 7](#_Toc70063345)

[Appendices 8](#_Toc70063346)

# Glossary

UML Diagrams = **Unified Modelling Language Diagrams**

FPS **= Frame Per Seconds**

## Key Words

Solar System, Planets, Orbit, Earth, Mercury, Venus, Pluto, Saturn, Jupiter, Mars, Neptune, Sun.

# Introduction

Assignment 2 is all about learning a new language called OpenGL and implanting those skills learned to create either a 3D game or 3D scene. Skills like, from creating a Scene Window to creating objects through text files and opening through code, adding textures, and adding lighting etc, was the main objective of this assignment. Solar System plays a major role in our universe and recreating Solar System through OpenGL code was a level up in knowledge and it was an aesthetic treat to watch at. That is why Solar System was the choice for this assignment where 3D objects/Planets were imported through code in the 3D scene. Whole space atmospheric scene was created by using OpenGL language. Planets like Earth, Sun, Mars, Jupiter etc can be seen in in 3D scene orbiting around Sun just like the actual Solar System works. For extra features there is menu added to the game to make it look professional and interesting. Using the menu is not hard just press SPACE bar to pause or start the 3D scene. Controls are also mentioned in the control/instruction section. The 3D scene also contains stars which gives a space vibe and make it look realistic. Stars were important part of the Solar System to give that space aura. Controls are added to move around the space either with the given controls in the control section or use of mouse wheel to zoom in and zoom out of the planets. Texturing and lighting have been used on the planets to give a realistic touch to the planets and to make them look 3D. Overall, The Solar System works perfectly fine.

## Controls/Instructions

**Menu Controls:**

**SPACE BAR = PAUSE/START**

**X = ON THE TOP OF THE WINDOW TO EXIT THE SCREEN**

**Movements:**

**W = UP**

**S = DOWN**

**A = LEFT**

**D = Right**

**Mouse Wheel = ZOOM IN / ZOOM OUT**

# UML Diagrams

# Critical Reflection

Things that went well for this assignment are quite few. Rendering screen objects was quite successful where multiple objects were loaded in the 3D scene. In this case they were planets. To create those planets there is a 3D object in the source file of the Solar System that was loaded in the project through code. The code that was used to create the planet was GLUquadricOBJ. To render them on screen basic glusphere code is used. Camera movements went well. The 3D scene does move x, y axis to change directions for example right, left, up and down. Mouse wheel button was also used to zoom in and of the planets which made the scene more complexed and pleasing to watch. This was compulsory to add as with out movements 3D scene or any game looks unpleasant and boring to look art or play. So, by achieving the movements it makes the scene looks more pleasant and third dimension. Keyboard is not only used in 3D scene it is also used in pausing and starting the scene such as pressing the SPACE BAR key helps the scene to pause ad start rotating again which has been successful.

Text display is one of the things that was very well done as when the 3D scene is being run and SPACE BAR key is pressed, there is a text display that appears on top saying “Press SPACE BAR to Start/Pause” and if the key is pressed again the text display disappear which is quite fascinating to look at. Scene object movement without user input has also been successful as there are two animation/movement happening one the planets are orbiting around the Sun and other while they are rotating, they rotate up and down while going around the Sun, and there is no user input involve in it. All of this seems so pleasing and aesthetics as if the solar system were real. Other things like texturing and lightning went very well. Texture is implanted on a complex shape like sphere that went very well the texture make planets look realistic and put a depth in the planets that force them to look 3D which looks pleasurable. The way texture is applied is by having BMP file and converting them into Gimp application and changed it to raw file. This is because raw files are easy to load, and the skills learned in OpenGL helped a lot while adding textures on the planet. Complex lightning was used on the 3D scene. How lighting works on this scene is when the planets come in front of the Sun then the planets are more visible and as soon as they go off the Sun the lightning and shadows start to diffuse and once, they go back the Sun the turn in to black similarly how real Solar System work and how day and night work. Lastly, a timer was added to show Fps in the game this is a good feature that shows on what frame rate is you game 3D scene is running at. The higher the frame rate, better quality and better animation is expected which is a good feature in terms of graphic point of view.

Where things go successful in a project on the other hand some things do not go as planned. There are few things that were not successful in creation of Solar System. Planets were built with basic code, there were no text file used to load up planets that is one thing that did not go well this is because that way was easier and more understandable but using complex methods and trying new ways can be beneficial for the knowledge so for the future projects this will be considered as one the method to use to load up any object through text file. Not loading one but loading multiple objects from file can be useful for knowledge and a new way of working rather than working with hard code. Texturing on cube, pyramids and sphere are do able however adding textures on a complex shape is hard which also did not go well, for example loading a planet Saturn which is a complex shape either a text file was needed or object loader to lad that shape but that did not go well because there was lack of knowledge which led to this method go unsuccessful. For future either it is a game or a scene there will be an attempt on adding texture on complex shapes this can be done by researching online or watch YouTube tutorials that can help to gain that knowledge. This can make the future games and 3D scene look aesthetically pleasing and intriguing.

Next thing that did not go well was adding extra features, as there is a menu screen but not like proper main menu that could have done by searching online tutorials and learn from there. Main menu could have been complex, for example when running the 3D scene, a screen popping up saying press enter to start and press escape to exit. This could have been more interesting and well structured then just saying press a space bar to start. This could be an improvement where in future projects this feature can be utilised to make the scene look more professional. Other features like collision can be added when two things collide it change its colour or display any message like collision occurred. This could be a good feature to add in the scene or game to add a bit of spice in the game or scene where their movements happing just like in Solar System. In future if the solar system were to make again there are couple of small features could be added, for example, white orbit lines on which planets are orbiting on. This can look eye-pleasing and give a unique texture to the scene. Last thing would be a specific key that can be pressed to reveal the names of the planets while they are orbiting around the Sun and could add little information about them. This will make it look professional and unique.

# Bibliography

**Website**:

Solar System scope (2019) Textures [online]. Available from:

<https://www.solarsystemscope.com/textures/> [Accessed: 1st April 2021].

Code in code block (none) Simple method for texture mapping [online]. Available from:

<http://www.codeincodeblock.com/2012/05/simple-method-for-texture-mapping-on.html>

[Accessed: 1st April 2021].

# Appendices



