Assignment 1: Display of 3D Model

Student Name: Kevin Pruvost (潘凯文)

Student ID: 2021400603

Table of Contents

Introduction

For this assignment, the main theme was `Display of 3D Model`.

The objectives here were:

Display of 3D model

- 1) Loading a 3D mesh model and display it on the screen
- 2) Four modes are supported: wireframe mode, vertex mode, face mode, face and edge mode. Mode is switched by keyboard. U
- 3) Rotate and translate the model by keyboard
- 4) Change color of wireframes under wireframe mode by keyboard.

Demonstration

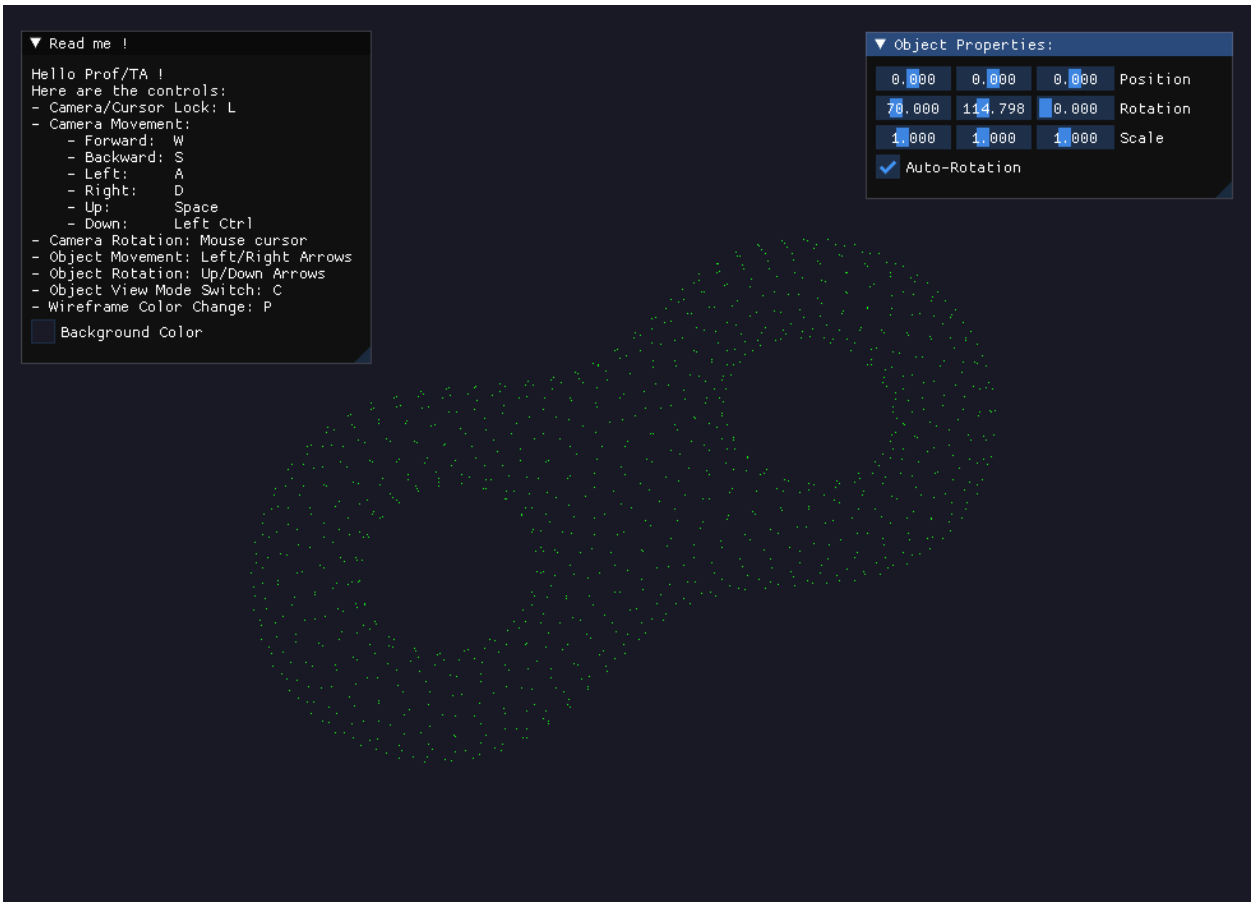
Content

For my Assignment, I got all these points covered:

- 3D Object Rendering:
    - Vertex Mode
    - Wireframe Mode
    - Face Mode
    - Wireframe/Face Mode
    - Wireframe Color Change (both on Wireframe Mode & Wireframe/Face Mode)
- Bonus:
    - Camera Movement + Rotation (can be enabled/disabled)

- Background Collor change
- 3D Object Movement + Rotation + Scaling
- Auto-Rotation (can be disabled)

How to use it



You can launch the `Assignment1.exe` directly, if you already have Visual C++ Redistribuable.

The program was compiled in Release mode.

As it is displayed in the program, here are the controls by order of priority for the assignment:

Controls

- Object View Mode Switch: C
- Wireframe Color Change: P
- Object Movement/Rotation:
  - Movement on X axis: ⬅️ ➡️
  - Rotation on X axis: ⬆️ ⬇️
- Camera/Cursor Lock: L
- Camera Movement:
  - Forward: W (or Z on AZERTY layout)

- Backward: S
- Left: A (or Q on AZERTY layout)
- Right: D
- Up: Space
- Down: Left CTRL

Additional Interactions

- You can change the background color by clicking in the square next to `Background Color`.
- As DearImgui library allows it, you can move the `Object Properties` window, I locked the `Read me !` window so that it's less bothering.
- You can modify Position, Rotation & Scaling with sliders on the X,Y,Z axis.
- You can enable/disable automatic Auto-Rotation.

Code Architecture

The Complete Documentation is available in the project, in the Code Architecture part will only be explained a small description of what classes do.

Documentation

There is a documentation available in `docs/index.html` if you want to have a better view on the classes I made.

Camera

Manages everything about the point of view we need for the view matrix.

Entity

Manages key composants of an entity such as position, rotation, ...

Also returns model matrix.

GUI

Manages the overall GUI, mostly written with https://github.com/ocornut/imgui.

Input

Manages everything related to input from keyboard and mouse.

Mesh / Mesh_Base

`Mesh` & `Mesh_Base` were created for the sole purpose of optimization, a database made of Mesh_Base is set so that we can load meshes once.

Then the `Mesh` class uses this database and links himself to a Mesh_Base, this method makes the use of meshes much lighter as `Mesh_Base` weighs 24 bytes because of the information it contains, while `Mesh` weighs only 2 bytes.

Obj

.obj files Parsing/Loading class, for now, only vertices & triangle faces are handled.

Shader

Contains utilities & informations about shaders.

Window

Inherits Input and manages everything about the OpenGL window.