FMX Rendering and Animation

Kadoo Aditya Anil & Kajal

21.11.2022
200050055 & 200050056

INTRODUCTION

We have followed an Object Oriented approach and created classes for different kinds of graphics models that are required in this project (Same as Assignment 1 and Assignment 2).This Assignment is a continuation of Assignment 2 so the same hierarchy and base codes (which contains the structures and information about hierarchical structure of the assignment) have been used to start with. The main program is present in the file fmx-animate.cpp. It initializes the objects such as track, bike and the rider with all the lights, shading and texture .

MODEL CLASS

Hierarchy node:

- It's the file having implementation of hierarchy node class. It basically contains all the parameters of an object (all objects are essentially nodes) like the position, rotation, pointers to it's children and handles to it's VAO and VBO.
- It also has two important functions to render the node with it's children recursively.

USER INPUTS

The GLFW key_callback function is modified from the cs475 namespace to handle user input for various events.New event handlers have been added for the extra operations like switching camera/node or saving the state. This allows us to implement interactive functionalities easily with the help of global state variables.

NAVIGATION

Various types of navigation have been handles using the following key-operation mapping (specified in README file):

Global Rotation:-

- Right/Left : Rotation about x-axis.
- Up/Down : Rotation about y-axis.
- [/] : Rotation about z-axis.

Model Translation:-

- D/G : Translation about x-axis.
- R/F : Translation about y-axis.
- E/T : Translation about z-axis.

Model Rotation:-

- Y/I : Rotation about x-axis.
- U/J : Rotation about y-axis.
- H/K : Rotation about z-axis.

Model Controls:-

- 1, 2, 3,... 9, 0 and - : Switch between the components of rider.
- =, L, O and P : Switch between the components of bike.

Camera handling:-

- Z : Normal camera kept at a distance.
- X : Camera capturing what helmet sees.
- C : Camera moving with (at some distance) helmet.

Lighting:-

- V : Point Light 1 toggle
- B : Point Light 2 toggle
- N : Bike headlight toggle
- M : Rider spotlight toggle

RENDERING

We have implemented Gouraud shading model for light rendering. Each vertex is assigned a colour value in the vertex shader using this model as a sum of intensities of ambient, diffused and specular components. This shading is passed to the fragment shader where the colours are multiplied by texture values if applicable.

We have used 4 different lights. These include 2 point lights situated at a height from the track. There is a spotlight that follows the rider's root component. Beside this there is a headlight on the bike attached to the handle.

ANIMATION

For animation we have implemented all the required functionalities but please take note that the keys have been remapped for easy of use during animation process as follows:-

- S : Saves the current state in keyframes_w.txt file. This includes the camera being used, lights on/off, camera rotation parameters and translation and rotation parameters of all the nodes in the model.
- A : Loads keyframes from keyframes_r.txt file.
- Q : Interpolates keyframes to generate the animation. Cameras and lights remain constant between 2 frames. Everything else gets linearly interpolated for smooth animation.
- W : Saves frames from the animation as .tga files. These can be converted to .mp4 video format using ffmpeg program.

RESULTS

Video : https://youtu.be/NblwkuvLFhM