CS475 Assignment 3 : FMX Rendering and Animation

Harsh Shah   **Neeraj Jadhav**
200050049   200050086

Contents

Figure 1: Global camera

1 Camera setup

There are 3 cameras in total, one global camera which is can be controlled independently of the rider, and other two cameras(one behind the rider and other for first-person-view) which move with the rider. Perspective transformation is used for all the three cameras.

1.1 Global

The Global camera is initially placed at the top of the track. The global camera is movable using the key presses mentioned at the end of the report.

1.2 First Person View

This camera is placed above the head of the rider and moves with the rider. It is constrained to move the rider.

1.3 Rider view

The rider view camera is situated at the back of the bike and is again constrained to move with the rider.

2 Lighting setup

In total 4 lights are added to the scene, involving two points lights to light up the environment, one light for headlight and last for spotlight. **Per-pixel shader** is used for all the lights. To implement spotlight and headlight, a cone in the vision of the light is defined and only those fragments which have angle between light vector and the optical axis less than a threshold(defined by the cone) is are lit with that
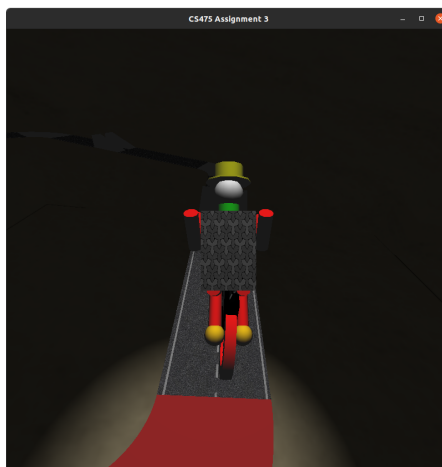
Figure 2: FPV camera

Figure 3: Rider camera

Figure 4: Spotlight

Figure 5: Cubemap

respective light. For lighting textures, separate glsl programs are created to define the color of the pixel, with remaining things being same.

3 Texture

Textures are instantiated in the `hnode` class, by defining a new constructor, using `.bmp` images. 4 textures are used in total: rider, bike, track and skybox.
For rendering texture of skybox, each of the face is mapped with corresponding texture coordinate, leaving the black spaces in the cubemap unmapped.

4 Animation

4.1 Loading and Saving

For saving the state of scene the following parameters are **appended** to a file(`keyframes.txt`):

- Timestamp

- Light states

- Light position

- Camera state and position and orientation of global camera

- Global translation and rotations

- Rotation parameters of all Hnodes(recursively)

For loading the keyframes, a 2-D array is maintained, which stores all the parameters of a particular timestep in a row. The file `keyframes.txt` is read to fill up the array and used for interpolation when animation is to be played.

4.2 Interpolation

The keyframes loaded are then used for interpolation. The number of frames between two key frames is decided by the fps of rendering. Only continuous parameters are interpolated and other parameters(like state of light on/off, etc.) remain fixed during a particular interpolation.

4.3 Playing animation

The frames, after being interpolated, are played at a specific fps using `glfwWaitEventsTimeout()`. During the animation, if the frames are to be saved then the canvas is converted to png format and saved to a folder.

5 Saving frames

In order to generate the video, all the frames(key frames as well as interpolated frames) are saved into `png` format images, and then converted to `.mp4` video using `ffmpeg`. For saving the canvas to `png` images, `stbi` library is used.

5.1 Link to video

The link to video uploaded to youtube is here.

5.2   Script

A FMX lover steals **Darth Vader's** bike and wishes to finish the most difficult track, infamous for being the **doom** of many riders. The rider, being confident of himself takes up this challenge and plans to finish the track.

First obstacle is a ramp, which is a child's play to him....or so it seems. Mid-way in the air he does a **steering turn** (after overshooting the ramp). However, upon landing, he loses control on the steering, and due to steep turn near the edge of cliff, disastrously falls and meets an *unfortunate end*!!

Moral of the story : Animation is not easy.

6 Some other images

Figure 6: Scene 1

Figure 7: Scene 2