Assignment 1 : Creating a simple keyframe-based animation

NAME: JIANG SUYI
STUDENT NUMBER: 39421387
EMAIL: JIANGSY@SHANGHAITECH.EDU.CN

1 INTRODUCTION

1.1 Things To Be Done

- [must] A basic window-based program to load and render 3D mesh objects (OpenGL is recommended). The program should support frame-by-frame animation to demonstrate the result.
- [must] In the program, we can add and edit keyframes. The frames in-between are interpolated using non-linear methods. Specific requirements are listed in "Instructions" below.
- [optional] To create a smooth rate of change through keyframes, so that there is no sudden change of the object's velocity.
- [optional] A timeline GUI component to add and edit keyframes instead of hard coding.

1.2 Usage

After building the project, run the program in VS. A window will be opened. There is an UI below where you can operate. For example, the degree of B-spine, the lasting time of the animation, a slide to specify precise time and two buttons which are used to control recording of frames. One can use WASD to rotate model. If frames are recorded, the animation can be played and stopped using two buttons.

2 IMPLEMENTATION DETAILS

I build this project mainly based on CG project and use another pikaqiu model. The loading of the model are done by $tinyObjectLoader$ and redering of the model are done by shaders written by myself.

So the main purpose is to change yaw and pitch angle of the model through time and I mainly get idea from B-spline interpolation (https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/INT-APP/CURVE-INT-global.html). A pair of $(yaw, pitch)$ are seen as data point and each data point has a time parameter t. Suppose we have n data points, we can build an $n * 2$ matrix D:

$$D = \begin{bmatrix} yaw_0 & pitch_0 \\ .. & .. \\ yaw_n & pitch_n \end{bmatrix}$$

and with n time parameter t, an $n * n$ matrix N can be built:

$$N = \begin{bmatrix} N_{0,p}(t_0) & N_{1,p}(t_0) & .. & N_{n,p}(t_0) \\ N_{0,p}(t_1) & N_{1,p}(t_1) & .. & N_{n,p}(t_1) \\ .. & .. & .. & .. \\ N_{0,p}(t_n) & N_{1,p}(t_n) & .. & N_{n,p}(t_n) \end{bmatrix}$$

Here $N_{i,p}$ means B-spline basis function under degree p. It can be computed through following equation:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \le u < u_{i+1} \\ 0 & \text{otherwise,} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

Here U set is a uniform knot set from 0 to 1 generated automatically. With D and N computed, we can calculate out all control points in $n * 2$ matrix P through following equation:

$$D = N \cdot P$$

Once we have P, we can calculate any data point $(yaw, pitch)$ with any given time parameter t.

When it comes to key frame interpolation, I just cut the total animation time to pieces like 1/30 s. When 1/30 s passes, a new time parameter will be used to calculate a new pair of $(yaw, pitch)$, this data will be used to form a new model matrix which will be transfered to shader to control the rotation of a model.

I used imgui based on glfw and glad to build GUI. You can see the usage and vedio for detail.

3 RESULTS

Here is the demo vedio link:
https://www.bilibili.com/video/BV1tf4y1B7Yu