CSCI 6555 Computer Animation

**Lab 1 Report**

Lingfeng Zhou

GWid: G38336549

Professor：James Hahn

The George Washington University

**I. Brief description of the system**

The system use a set of control points and two kinds of interpolation splines as input, and output the animated view of the object. Each control point is presented as 7 parameters in Quaternion format or 6 parameters as Euler Angles. The user can choose the splines as Catmull-Rom spline or B spline. Also, the system allows speed control of the animation.

This lab is based on lab0 given by professor. The mainly change is in the render function which provides the transformation matrix and timer function which is used to control animation speed.

**II. Description of code**

Firstly, create the 4\*4 matrix of Catmull-Rom spline and B spline which will be used in interpolation. And then create some control point. In this system, there are 6 control points. In Euler angle format, there are 6 parameters for each point, in quaternion format, there are 7 parameters, respectively for orientation and position.

GLfloat CatmullRomMatrix[16]

GLfloat BSplineMatrix[16]

GLfloat EulerAngle[6][6]

GLfloat Quaternion[6][7]

* **For quaternion:**
* A function to achieve matrix multiplication to computer Q=TMG

GLfloat ComputeQ(GLfloat T[4],GLfloat M[16],GLfloat G[4])

* Computer in-between point. The main task for this part is to get Geometric of in-between points using the 4 points,,,. Use the equation Q=TMC to get the 7 parameters of in-between point.

void Interpolation (GLfloat MatrixType[16],GLfloat Object[6][7],GLfloat Q[7])

* When using quaternion, normalize it to get the same rotation scalar.

void Normalize(GLfloat ParameterArray[7])

* Convert the 4 parameters of quaternion to corresponding rotation matrix, add the 3 parameters of position to form the 4\*4 transform matrix.

void getMatrix (GLfloat ParameterArray[7],GLfloat Matrix[16])

* Load the transform matrix and use the solid teapot to get animation.

glLoadMatrixf(TransformMatrix);

glutSolidTeapot(1.0);

* **For Euler angler:**
* For Euler angle, use the same way to get in-between point. The difference with quaternion is Euler angle has 6 parameters rather than 7.

void Interpolation (GLfloat MatrixType[16],GLfloat Object[6][6],GLfloat Q[7])

* Then simply transform Euler angle to quaternion, and normalize quaternion, get transform matrix, load the matrix and finally get animation.

void EulerToQuaternion (GLfloat ParameterArray[7])

void Normalize(GLfloat ParameterArray[7])

void getMatrix (GLfloat ParameterArray[7],GLfloat Matrix[16])

glLoadMatrixf(TransformMatrix);

glutSolidTeapot(1.0);

* **Speed control:**

To control speed, map parameter t to time, and when t exceed 1, move to next point.

void timer( int value ) {

// render

glutPostRedisplay();

// map t parameter to time to achieve speed control

t=t+speed;

if(t>=1){

t=0;

if(PointNumber<PNumber-1){

PointNumber++;

}

else{

PointNumber=0;

}

}

// 16 ms per frame ( about 60 frames per second )

glutTimerFunc( 16, timer, 0 );

}

If Link error 2019 occurs when running the project, check the project attribute, there may be some lib file missing.