

CS 6323 Computer Animation & Gaming

Assignment 4 (Grade: 10 points)

Path Following with Orientation Interpolation

Now you already have a program to move the shuttle object along the curve path through eight points, please write a program to rotate the shuttle object. The orientation of the shuttle object is required to interpolate the orientation of the eight points, which are given by quaternions. When the shuttle object is moving along the curved path, use Slerp to interpolate the orientation between two quaternions along the corresponding line segment.

Initialization:

1. Set coefficient matrices for each segment using Catmull-Rom spline formulation. (Done in Assignment 2)
2. For each segment, loop through points, summing linear distances to create a table of parametric values and summed linear distances to approximate arc length. Because we are using multiple segments, the table might look like (Done in Assignment 3):

segment #	u- value	length	OR	point	length
0	0.0	0.0		x,y,z	0.0
0	0.01	0.12		x,y,z,	0.12
...
0	0.99	5.35		x,y,z	5.35
0	1.0	5.4		x,y,z	5.4
1	0.01	5.45		x,y,z	5.45
...
1	1.0	7.3		x,y,z	7.3
...
...
7	0.01	12.0		x,y,z	12.0
...
7	1.0	12.5		x,y,z	12.5

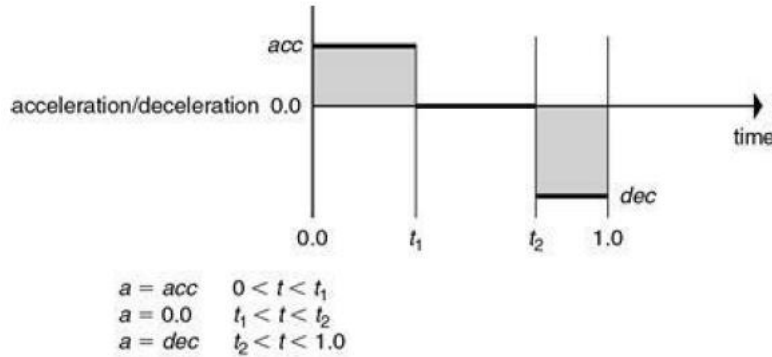
3. Compute at least 200 point per segment. If you normalize the lengths in the table so the total length is 1.0 and if the ease-in/ease-out function i/o is also normalized to go from **0.0** to **1.0**, then the code is easier to reuse with other ease-in/ease-out procedures. (Done in Assignment 3).
4. Draw the cubic curve (Done in Assignment 2).
Please keep your assignment 2 implementations in this assignment. The cubic curve should still be rendered.

Keep your assignment 3 ease-in/ease-out implementation.

Simulation:

1. Increment a time value t , that goes from zero to one as the curve is traversed (Done in Assignment 3).

2. Apply an ease function, $s = \text{ease}(t)$, using constant acceleration assumption. Support interactive change t_1 and t_2 (Done in Assignment 3).



3. Search the table created by the initialization routine for the entries s in between (Done in Assignment 3)
4. Compute the fraction that s is between two entries (Done in Assignment 3)
5. Use the computed fraction to interpolate between the points recorded in the table, $u = \text{table}(s)$ (Done in Assignment 3)
6. Evaluate the interpolation function to produce a point along the curve, $p(x,y,z) = P(u(s(t)))$ (Done in Assignment 3)
7. Based on the computed fraction u and two quaternions of end points of the line segment, interpolate the quaternion $q(t)$ using Slerp.
8. Rotate the object according to $q(t)$ and move the object according to $p(x, y, z)$.

Note: because t monotonically increases, so does s and, therefore, so do the indices of the entries retrieved from the table. This should make your search more efficient since you can start the next search from the point of the previous search.

Requirements:

1. The quaternion $q = s + ix + jy + kz$ for the eight points are given in the starting code (s, x, y, z). The starting code draws a local coordinate axis for each point, which are all identical to world coordinate system. Rotate the cubes and their local coordinate axis at each point according to the given q . (Grade: 2 points)
2. The moving shuttle object and its axis are initially placed at the first point and the orientation is identical to the orientation of the first point. (Grade: 1 points)
3. Add a button to GUI to enable/disable the ease-in/ease-out control. Add a button to GUI to enable/disable the rotation control. Note that rotation will be on when both ease-in/ease-out and rotation control are enabled. (Grade: 1 points)
4. When rotation control is enabled, the shuttle object adjusts its pose based on Slerp interpolation. The local coordinate axis should be moving and rotating with it. (Grade: 6 points)

Note: Quaternion to Rotation matrix:

Suppose the quaternion $q = s + ix + jy + kz$, then the corresponding rotation matrix M_q is:

$$M_q = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2sz & 2xz + 2sy \\ 2xy + 2sz & 1 - 2x^2 - 2z^2 & 2yz - 2sx \\ 2xz - 2sy & 2yz + 2sx & 1 - 2x^2 - 2y^2 \end{bmatrix}$$