CS 6323 Computer Animation & Gaming

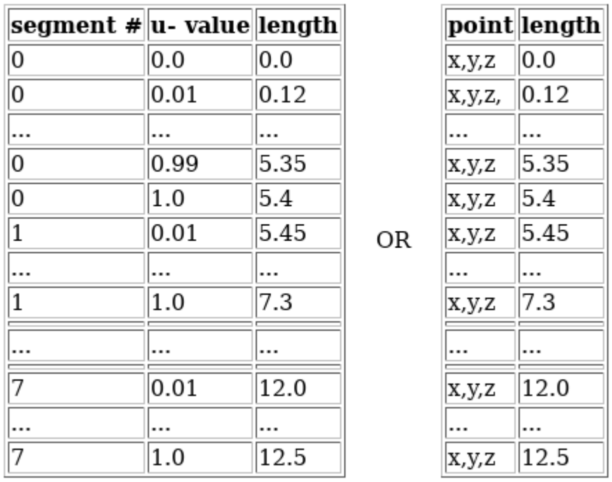
Assignment 3 (Grade: 10 points)

Path following with ease-in/ease-out control

Write a program to move a shuttle object that follows a curved path through eight points starting from the first point, accelerating to some maximum speed at t1, keeps constant speed till t2, and then decelerating to stop at the initial position.

Initialization:

1. Set coefficient matrices for each segment using Catmull-Rom spline formulation. (Already finished in your 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:



segment #	u- value	length
0	0.0	0.0
0	0.01	0.12
...	...	...
0	0.99	5.35
0	1.0	5.4
1	0.01	5.45
...	...	...
1	1.0	7.3
...	...	...
7	0.01	12.0
...	...	...
7	1.0	12.5

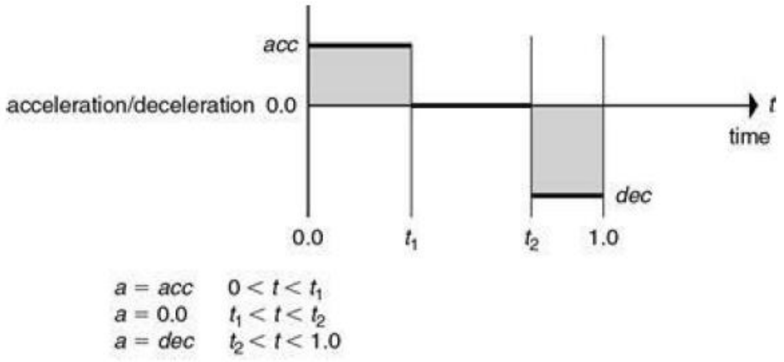
OR

point	length
x,y,z	0.0
x,y,z,	0.12
...	...
x,y,z	5.35
x,y,z	5.4
x,y,z	5.45
...	...
x,y,z	7.3
...	...
x,y,z	12.0
...	...
x,y,z	12.5

   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.
3. Draw the cubic curve (Already finished in your Assignment 2).
   Please keep your assignment 2 implementation in this assignment. So the cubic curve is still rendered.

Simulation:

1. Increment a time value ***t***, that goes from zero to one as the curve is traversed
2. Apply an ease function, ***s = ease(t)***, using constant acceleration assumption. Support interactive change of t1 and t2.



3. Search the table created by the initialization routine for the entries ***s*** in between
4. Compute the fraction that ***s*** is between two entries
5. Use the computed fraction to interpolate between the points recorded in the table, ***u = table(s)***
6. Evaluate the interpolation function to produce a point along the curve, ***p(x,y,z) = P(u(s(t)))***

Note that, because ***t*** is 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.

You do NOT need to control the orientation of the shuttle object in this assignment.

The starting code draws the eight cubes and a shuttle object at (0,0,0). The shuttle object is defined in shuttle.obj, which specifies the vertices and triangles of the model. Model::load_obj() is added to load the model.