CS633 Computer Animation

Programming Assignment 1 (40 points)

Due: 3/6/2018

Your job for the first programming assignment is to design and implement an openGL program that can perform the following tasks:

1. For a set of given points $P_0, P_1, \ldots P_n$ on a 3D sphere S your program can generate a closed path (space) curve $C(u)$ on S that passes through these points. $C(u)$ should be at least C^1-continuous
2. Your program can build an arc length table for $C(u)$ so that
   (1) for any given u_1 and u_2 in the parameter space of $C(u)$ one can use the arc length table to find the length of $C(u)$ between u_1 and u_2
   (2) for a given u_1 in the parameter space of $C(u)$ and a given real number s, one can use the arc length table to find a u_2 in the parameter space of $C(u)$ so that the length of $C(u)$ between u_1 and u_2 equals s
3. Your program can control the motion of a toy car (or something) along the space curve $C(u)$ such as in constant speed or constant acceleration

For the first task, the quaternion based $slerp$ technique or an equivalent technique has to be used in the construction of the path curve $C(u)$. Note that the quaternion based $slerp$ technique can only give you a circular arc between two consecutive points P_{i-1} and P_i. Therefore using this technique alone you wouldn't be able to generate a smooth path curve. You need to use either 3D composite Bezier curve or 3D B-spline curve as a representation of the path curve, and use $slerp$ in the computation of the control polygon of the path curve.

For the second task, you have three choices in building the arc length table:

(1) Forward differencing
(2) Adaptive subdivision
(3) Uniform (or, adaptive) Gaussian integration

However, if a cubic composite Bezier curve is used as the representation of the path curve, the $de\ Casteljau$ algorithm is actually the best technique in computing curve points of the path curve $C(u)$.

For the third task, you need to know how to set your $distance - time$ function $s(t)$. Possible choices are:

(1) Constant speed

(2) Ease-in/ease-out
(3) Constant acceleration
(4) General distance-time functions

Each of these choices is basically a re-parametrization of the path curve. The core is the mapping from time t to the parameter space u of the path curve. For instance, for constant speed, $s(t)$ should just be ct for some constant c.

A detailed description of things you need to know (especially about the first task) when doing design and implementation of this project will be mailed to you separately. However, don't wait for the detailed description to start your work, you need to start your work for this project immediately.