Programming Assignment 2

Date of announcement: 4th February 2016
Submission Deadline: 25th February 2016

Description

In this homework you will be learning about Hermite splines. One parameter splines are commonly used as motion paths in order to animate objects along their path. Bi-parametric splines are used to create complex curved surface shapes in films, games, engineering, manufacturing, etc. In this assignment you will be concerned only with one parameter splines.

You are required to develop an application using OpenGL which

1. Takes as input a set of control points and their tangents entered by the user and creates a cubic Hermite spline interpolating those points.
2. Draw a simple triangular object that follows the motion path defined by the spline.
3. Orient the simple triangular object such that it follows the curve's local orientation.

Implementation Specifications - Grading Criteria

Develop an OpenGL application with the following functionalities and features:

- create a GLFW window of size 800x800 with double buffering support.
- use an orthographic projection for simplicity.
- the application should request the number of points **N** from the user at the beginning of the execution. The user will then have to click **N** times for the control points P_i and another **N** times for computing the tangents T_i. The tangent at each point is calculated as T_i - P_i.
- You can assume that the Z coordinate of the control points is fixed e.g. 0. The X and Y coordinates of each point should then relate to the window coordinates where the user has clicked.
- Window resize Handling:
    - The application should handle window resize events
- Input Handling :
    - [Left Mouse Button]
        - marks control points/tangents in the window
    - [Key 'RIGHT', 'LEFT', 'UP', 'DOWN']
        - Move the camera right, left, up, down respectively.
    - [Key 'L']

        - Render spline using line strips
    - [Key ‘P’]
        - Render spline using points
    - [Key ‘Enter’]
        - compute and draw the spline based on the control points and tangents entered so far
    - [Key ‘Backspace’]
        - resets the application i.e. removes all control points, tangents, interpolated points etc, and asks the user for a new number of points **N**.
- You should use the subdivision technique presented in class for drawing the spline.
- Draw a simple triangular object that follows the motion path defined by the spline.
- The application should use OpenGL 3.0 and above, and include comments explaining each step.

For extra credit and further intellectual/programming challenge:

- Use the subdivision algorithm taking into account the curvature (rather than distance) for drawing the curve (difficulty level: 8.5/10)
- Consider the profile and trajectory polylines output by your assignment 1 program as the input points to your spline interpolator and draw the curves. You should be able to see smoother versions of your sweep surfaces. (difficulty level: 7/10)

Submission

Please create a zip file containing your C/C++ code, vertex shader, fragment shader, a readme text file (.txt). In the readme file document the features and functionality of the application, describe any extra credit work (if applicable), and anything else you want the grader to know i.e. control keys, keyboard/mouse shortcuts, etc. Submit your file using Moodle.

Additional Information

- You can use the skeleton code provided during the lab sessions to get started.