Programming Assignment 2

Date of announcement: Session #5
Submission Deadline: Session #7

Description

In this homework you will be learning about Catmull-Rom splines and basic model animation and how to use these features to develop an OpenGL application.

You are required to develop an application using OpenGL which takes as input a set of control points entered using a mouse by the user and creates a cubic Catmull-Rom spline based on these points. You will then perform translation sweep using two such splines, or rotation sweeps using one spline depending on the user's input.

Implementation Specifications - Grading Criteria (100 pts)

Develop an OpenGL application with the following functionalities and features:

1. User decides whether to have a translational "T" or rotational sweep "R" (through console)
2. The user then marks control points in the window with the mouse (assume XZ plane, Y=0). The X and Z coordinates of each point should then relate to the window coordinates where the user has clicked.
3. User can then press "Enter" which computes and draws the spline.
   a. If the user selected translational sweep, then repeat steps 2-3 for the second set of control points resulting in two separate splines (this time assume XY plane, Z=0). The X and Y coordinates of each point should then relate to the window coordinates where the user has clicked.
4. Write the points of the resulting spline/s to an output file. The output file should have the same description format as Assignment #1.
5. Load the output file in Assignment #1 and generate smooth surfaces based on the input splines.

- Create a GLFW window of size 800x800 with double buffering support.
- Use an orthographic projection for simplicity.
- You should use the subdivision technique presented in class for drawing the spline.
- Input Handling :
  - [Left Mouse Button] - marks control points in the window
  - [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 entered so far
  - [Key 'Backspace'] - resets the application i.e. removes all control points, interpolated points etc.

- Window resize Handling:
    - The application should handle window resize events.

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: 6/10) (**15 pts**)
- Merge Assignment 1 into the code and compute the sweep surfaces. For this, the sweep surfaces should be displayed in a new perspective window. (difficulty level:5/10) (**10 pts**)

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.
- A demo video of the basic functionality [no extra credit implementation] can be found here:
    - Video 1 - https://youtu.be/cCHfGiOq_Ag
    - Video 2 - https://youtu.be/8SqD4IuaVYI

Evaluation Procedure

You MUST demonstrate your solution program to the lab instructor during lab hours. You must run your program, demonstrate its full functionality and answer questions about OpenGl programming of your solution. Major marking is done on the spot during demonstration. Your code will be further checked for structure, non-plagiarism, etc. However, ONLY demonstrated submissions will receive marks. Other submissions will not be marked.