2017/11/10

EECS 267 LAB2 - README.PDF - Yusuke Niiro

//Description

In this project, I rendered a teapot. There are several UI (User Interface) which shows you several things that you want to see.

< UI Buttons / Sliders >

-Triangles: By clicking the number that are displayed in the screen you can change your number of triangles that is used to render this teapot.

-Cage: By clicking this button, it will show the cage that is made from control point coordinates.

-Normal: By clicking this button, it will show the normal vectors for all vertices in the teapot.

-Shadow: By clicking this button, it will show the shadow that is made from virtual light source.

-Light Control: By changing these parameters, you can control the coordinates of virtual light source.

-Smooth: By clicking this button, it will change the texture of teapot by using the normal.

< Key Board >

You can change the number of triangles also by clicking certain button.

If you click “q”, it will decrease the number of triangles. And if you click “w”, it will increase the number of triangles.

//Processes

1) Load the coordinates from “teapotCGA.txt”.
2) Find the curved teapot coordinates by calculating the Bezier Curve equation.
   Use coordinates that we got from process 1 as control points.
3) Get the normal vectors for each vertex in the curved teapot.
   First, calculate the derivative vector for two directions by using Bezier Curve.
   And, get the normal vector by calculating the cross product of those two derivative vectors.

4) Get the coordinates for making shadow.
   First, decide the coordinate for virtual light source and height of projection plane.
   And, get the coordinates of shadow by using ratio of (difference between virtual light source and teapot) and (difference between teapot and plane).
5) Render the teapot, normal vectors, cage made from control points and shadow.
6) Make the UI buttons , UI sliders, Key Board function that were explained in Description.

//Evaluation
Compare the Building Time of displaying a cage in 2 ways and find out which one is the better way to display it.

I used the gs_time() function to calculate the building time. I put this function in to each uievent of Cage Rendering 1 and Cage Rendering 2.

Building time will show up if you click the UI Button “Cage Rendering 1” or “Cage Rendering 2”.

1. Cage Rendering 1
   ON: Delete the previous scene and build a new scene with cage
   OFF: Delete the previous scene and build a new scene without cage

2. Cage Rendering 2
   ON: Use SnNode visible function and display the cage
   OFF: Use SnNode visible function and don’t display the cage

Table 1: Average Building Time to make a Cage with 2 Ways

| | Cage Rendering 1 (sec) | Cage Rendering 2 (sec) |
|---|---|---|
| ON | 0.00724331 | 5.01501E-07 |
| OFF | 0.006481245 | 5.26165E-07 |

Average Building Time: Average of 30 samples

By looking at the table, I figured out 2nd way of displaying the cage was much better way to do it. If you use 1st way to display the cage, it will take about 10,000 times longer time than 2nd way.

From the evaluation, if you want to make a UI button that switch the display something that is stable on and off, you should use SnNode visible function to make that UI work.