Middle East Technical University | Department of Computer Engineering

CENG 477

Introduction to Computer Graphics

Fall '2014-2015

Assignment 2 - Surfaces and Transformations in OpenGL

Due date: November 24th 2014, Monday, 23:55

1 Objective

This assignment aims you to get familiar with OpenGL programming, transformations and Bézier surfaces using the OpenGL API.

2 Specifications

In this assignment, you are expected to implement a basic graphics tool that loads a Bézier surface and applies some basic transformations to the rendered surface. Specifications of this OpenGL program in this assignment are listed below:

- In the beginning of the program, the properties of the surface is read from an input file. This input file includes total numbers of sample points in x and y axis, vertex coordinates of the control points of the Bézier surface, and color palette that is to be used for coloring the surface. More detailed information about the input scene file is provided below:

```
SampleR SampleC
M N
x11 y11 z11 x12 y12 z12 ... x1N y1N z1N
x21 y21 z21 x22 y22 z22 ... x2N y2N z2N
...
xM1 yM1 zM1 xM2 yM2 zM2 ... xMN yMN zMN
K
P1r P1g P1b
P2r P2g P2b
...
PKr PKg PKb
```

- `SampleR` and `SampleC` are the total number of samples of the Bézier surface. `SampleR` number of points is used for sampling row and `SampleC` number of points is used for sampling column of the Bézier surface, respectively. In other words, `SampleR` and `SampleC` are the numbers of points that is to be used for row and column index of the given control points, respectively. Their data types are integers. `M` and `N` are the total row and column numbers of control points array respectively and their data types are also integer. `xij`, `yij`, and `zij` are the x, y, and z components of the Bézier control point coordinates in (i, j)th position. All of the control point coordinates are given in float type. `K` is the length of the color palette and its type is integer. `Pkr`, `Pkg`, and `Pkb` are R, G, and B values of the colors in color palette. They have integer types and their values are between 0 and 255.

- The control points are used for drawing a two dimensional Bézier surface in the scene. It is defined as a parametric surface where the position of a point $\mathbf{p}$ as a function of the parametric coordinates u, v is given by:

$$\mathbf{p}(u,v) = \sum_{i=0}^{m} \sum_{j=0}^{n} B_i^m(u) B_j^n(v) \mathbf{k}_{i,j} \tag{1}$$

  where,

$$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i} \tag{2}$$

  is a Bernstein polynomial, and

$$\binom{n}{i} = \frac{n!}{i!\,(n-i)!} \tag{3}$$

  is the binomial coefficient. This formula is defined for $(m+1)(n+1)$ control points $k_{i,j}$. u and v are the parametric coordinates of the surface points $\mathbf{p}_{i,j}$ and their values are in $[0,1]$ interval.

- After reading the input file, the program draws the surface by triangulating the output sample points of the Bézier surface. Triangulation simply means to draw triangles by using a set of m-by-n vertices. In this technique, any 4 points each of which has exactly 2 4-neighborhood among this 4 points set is used for drawing 2 triangles which have 2 common vertices. Triangulation process allows us to create surface by interpolating a set of points in 3D space.

- Lighting is disabled in the scene. However, the triangles of the surface is assigned to a set of different colors according to the average y values of the triangle vertices. The program uses input file order as color palette indices. The program assigns the color value with the largest index to the triangle with the largest average y value and visa versa. All color indices are assigned to the triangles by scaling color palette indices to the average y values of all triangles in the surface.

- At startup, your camera should point to (0,0,0) coordinate from (100,100,100) point. Up vector of the camera is (0,1,0) and perspective projection is used as the projection mode of the scene. By clicking left button of the mouse, the camera changes the direction by mouse movement as long as left button of mouse is pressed.

- The program runs in two forms named **Transformation Mode** and **Surface Editing Mode**. '1' button activates Transformation Mode and '2' button activates Surface Editing Mode. The program runs in Transformation Mode as default. The specifications of these modes are given in the following sections:

2.1 Transformation Mode

- This mode allows the program to perform some transformations on the surface object. Functionalities of this mode are given as follows:

- **Translate:** The program allows user to translate the surface object along x, y, and z axes. 'q' and 'w' buttons translate the object in -x and +x directions, respectively. 'a' and 's' buttons translate the object in -y and +y directions, respectively. 'z' and 'x' buttons translate the object in -z and +z directions, respectively.

- **Rotate:** The program allows user to translate the surface object along +y direction. 'r' button rotates the surface object in counter clockwise along +y direction and 't' button rotates the surface object in clockwise along +y direction. The surface object rotates around the position which is the average point of all the sampling points.

2.2 Surface Editing Mode

- This mode allows the program to change the shape of the surface object by altering the coordinates of the control points. When this mode is activated, the control points are also drawn in the scene with the surface object. (Please, draw control points large enough to be recognized easily, *see glPointSize()*) They are in the same color except the selected control point.

- First this mode is activated, the selected control point is the point in (0, 0)th location in the input file. The selected point is highlighted by assigning a different color than the other control points so that the control point which is currently selected is distinguished from the other control points easily.

- The arrow keys change selected control points. In other words, the user is able to change currently selected point by using arrow keys. Control points are given in the input file as M-by-N array. This mode uses the array locations specified in the input file as control point coordinates. Arrow keys are used for increment/decrement control point coordinates in order to change selected control point.

- If the selected control point has (i, j) control point coordinates then right and left arrow keys increment and decrement i value of the selected control point coordinate (row index), respectively. Similarly, up and down arrow keys increment and decrement j value of the selected control point coordinate (column index).

- Selected control point is translated with the buttons used for object translation in Transformation Mode. That means, 'q' and 'w' buttons translate the selected control point in -x and +x directions, respectively. 'a' and 's' buttons translate the selected control point in -y and +y directions, respectively. 'z' and 'x' buttons translate the selected control point in -z and +z directions, respectively. Please note that after modifying the selected control point coordinates, you should update the surface object with respect to the modification.

- Number of sample points is changed by via keyboard input. 'i' botton decreases row sampling by dividing its value by two and 'o' button increases row sampling by multiplying its value by two (SampleR value from the input file). Similarly, 'k' botton decreases column sampling by dividing its value by two and 'l' button increases column sampling by multiplying its value by two (SampleC value from the input file).

- Please note that mouse event handling is active in both Transformation and Surface Editing Mode.

3 Hints & Tips

- The program will be implemented in C++ with OpenGL API.
- Your codes will be tested on departmental machines using "g++". Please make sure to run tests on ineks.
- You can use the example source codes provided in the course webpage on COW.

4 Submission

Submission will be done via COW. Please upload a single zipped file called "hw2.zip". The file will contain your source code and a makefile. A sample makefile is uploaded on COW. Your executable should have the name "hw2". We will test your code as:

```
$unzip hw2.zip
$make
$./hw2 any_surface_file.txt
```

Note that the command sequence does not work if your submitted zip file extracts to a directory. Also note that you do not have to include any input file in your submission. Follow the newsgroup for further details.

Late submissions are allowed for this homework, regarding to the policy on the course's web site.

5 Cheating Policy

We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations. See the course website for more information.