24-783 Problem Set 8

Before starting update your public, data, and course_files directories by typing:

svn update ~/24783/src/public

svn update ~/24783/data

svn update ~/24783/src/course_files

If you haven't checked out data files for testing your program, files can be downloaded by:

```
svn checkout https://ramennoodle.me.cmu.edu/svn/teaching/data
```

Deadline: 05/01 (Mon) 23:59

Preparation: Set up CMake projects

You first create projects for the two problem sets.

1. In the command line window, change directory to:
   ~/24783/src/*yourAndrewId*
2. Update the course_files, data, and public repositories.
3. Use "svn copy" to copy base code to your directory:
   ```
   svn copy ~/24783/src/course_files/ps8 .
   ```
4. Add ps8/ps8_1 and ps8/ps8_2 sub-directories to your top-level CMakeLists.txt
5. Run CMake, compile, and run ps6 executable.
6. Commit to the SVN server.

Optional: Check out your directory in a different location and see if all the files are in the server.

PS8-1 Plain 2D Renderer

In this assignment, you write a GLSL program (a pair of vertex and fragment shaders) called Plain2DRenderer. The skeleton of the class is already written in renderer.h and renderer.cpp. You implement the GLSL shaders and Plain2DRenderer class.

This GLSL program takes two attributes, vertex and color. A vertex is a 2D coordinate (x,y), and a color is a 4D color (r,g,b,a). In the vertex shader, (0,0), (800,0), (0,600), (800,600) are transformed to (-1,1), (1,1), (-1,-1), (1,-1), respectively. (Leave z,w components 0 and 1).

The fragment shader just outputs a color passed from the vertex shader.

Using this GLSL program, replace drawing functions in the Cannonball game. The GLSL program must be made ready in Initialize function. Remove all glColorXX, glVertexXX, glBegin, glEnd, and replace them with appropriate functions.

PS8-2 Drawing Cubic Bezier Surface with a GLSL program

Cubic Bezier surfaces are very often used in CAD packages. It defines a surface with 16 control points. A point on the surface is a function of two parameters (s,t).

The base code main.cpp draws a Bezier surface by calculating points on the surface in the C++ program. It uses Gouraud3dRenderer to draw the surface with triangular elements.

Your task in PS8-2 is to write a GLSL program that takes:

16 control points, projection and modelView matrices as uniforms, and

parameters (s,t) as an attribute.

and draws a Bezier surface.

The vertex shader must calculate a point on the surface from the 16 control points and a parameter. The point then must be transformed by modelView and projection matrices and passed as an output.

The fragment shader must assign a pixel color based on the y coordinate of the point. The color must be calculated as:

Y<=0	R 0	G 0	B 1	A 1
y=0.25	R 0	G 1	B 1	A 1
y=0.5	R 0	G 1	B 0	A 1
y=0.75	R 1	G 1	B 0	A 1
y>=1.0	R 1	G 0	B 0	A 1

R,G,B values must be linearly interpolated in between.

Replace the drawing functions in the basecode main.cpp so that it uses your renderer instead of Gouraud3d renderer.

If you successfully implement the renderer and send the input uniforms and attributes correctly, you will see an image like the following.