Computer Science

COMPSCI 372 S2 C – Exercise Sheet 3

4th August 2008

Q1: Download the “OpenGL Examples” demo at http://www.cs.auckland.ac.nz/compsci372s2c/resources/OpenGL_ExamplesNET.zip

(a) Run the project “fog” which demonstrates three different types of distance attenuation to simulate smoke/fog covering an object. Which type of attenuation makes a far away object look most blurry?

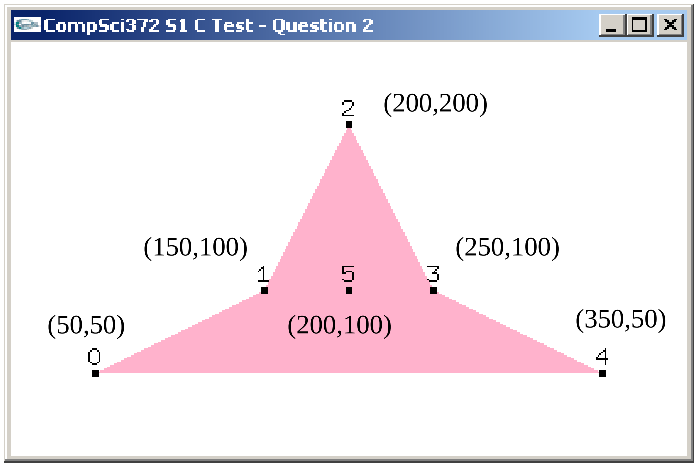
(b) Run the project “dof” which simulates the out-of-focus depth-of-field effect you have when using a real camera. For what type of applications would such an effect be useful? Find an example for this on the Internet.

Q2: **(a)** Compute the world-to-viewport mapping which maps the world window with the coordinates (1.5, 4.5, 0.0, 3.0) [(left, right, bottom, top)] onto a window on the screen (the viewport). The window has a width of 600 pixels, a height of 400 pixels and its left-top corner is at the pixel (100,100) on the screen.

(b) Assume you draw a triangle with the world coordinate vertices (0.0,1.0), (3.0, 1.0), and (3.0,2.0) using the world-to-viewport mapping from (a). What are the pixel positions of the triangle corners on the screen? Which part of the triangle is inside your drawing window (i.e. the viewport)?

(c) Write a program drawing the triangle from (b) using the world-to-viewport mapping from (a).

Q3: Draw the 2D object shown on the right. The vertices 0 to 5 of the object and the corresponding co-ordinates are shown in the image on the right. For each answer to the questions below use the most efficient representation. [This question is from the 2004 COMPSCI 372 Test]



(a) Complete the code fragment below so that it defines a global array containing the six 2D vertices of the above shape in the given order:

```
const int numVertices=6;
const float v[numVertices][2] =
```

(b) Complete the display function below so that it draws the given shape using the GL_TRIANGLE_FAN mode and the glVertex2fv command. The vertex numbers and the black dots in the above image have been inserted afterwards for clarity and you don't have to draw them.

```
void display(void)
{
      // clear all pixels in frame buffer
      glClear(GL_COLOR_BUFFER_BIT);
      glColor3f(1.0, 0.1, 0.2);     // reddish colour
      glBegin(GL_TRIANGLE_FAN);



      glEnd();
      glFlush();
}
```

(c) Complete the display function below so that it draws the given shape using the GL_TRIANGLE_STRIP mode and the glVertex2fv command.

```
void display(void)
{
      // clear all pixels in frame buffer
      glClear(GL_COLOR_BUFFER_BIT);
      glColor3f(1.0, 0.1, 0.2);     // reddish colour
      glBegin(GL_TRIANGLE_STRIP);



      glEnd();
      glFlush();
}
```

(d) Complete the display function below so that it draws the given shape using the GL_QUAD_STRIP mode and the glVertex2fv command.

```
void display(void)
{
      // clear all pixels in frame buffer
      glClear(GL_COLOR_BUFFER_BIT);
      glColor3f(1.0, 0.1, 0.2);     // reddish colour
      glBegin(GL_QUAD_STRIP);
```

```
      glEnd();
      glFlush();
}
```

(e) Write a program which displays your solutions to (b)-(d).

(f) Write a program which draws the above shape using an OpenGL primitive such that its colour varies smoothly from red at its centre to yellow at its boundary.