Name _____

GPU Programming
EE 4702-1
Midterm Examination

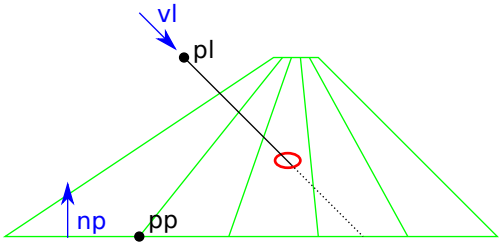
Wednesday, 18 October 2017 11:30–12:20 CDT

Problem 1	_____	(30 pts)
Problem 2	_____	(20 pts)
Problem 3	_____	(25 pts)
Problem 4	_____	(25 pts)
Exam Total	_____	(100 pts)

Alias _____

Good Luck!

Problem 1: [30 pts] The illustration to the right shows a ring surrounding the point at which a line crosses a plane. When completed the code fragment below will render the ring. The line is defined by parametric equation $P(t) = \texttt{pl} + t \times \texttt{vl}$. Point `pp` is on the plane and vector `np` is normal to the plane. The ring is circular and is on the plane.



(*a*) Modify the code below so that `t` is assigned the value for which `pc` is on the plane.

☐ Compute t such that $P(t)$ is on the plane.

(*b*) Modify the code below to render the ring. Use `r1` for the inner radius and `r2` for the outer radius. Render it using `slices` sets of triangles. Use variable `pc` even if you haven't yet solved the previous part.

☐ Include: ☐ Code to set up the rendering pass, ☐ to set the color (red), ☐ to set the correct normal, ☐ and to compute and set vertex coordinates.

```
World::render_ring( pCoor pl,  pVect vl,     // Line definition.
                    pCoor pp,  pNorm np,     // Plane definition.
                    float r1,  float r2 ){  // Ring radii.


  float t =                          ; // Fill In

  pCoor pc = pl + t * vl;









  const int slices = 100;
  const float delta_theta = 2 * M_PI / slices;
  for ( int i=0; i<=slices; i++ ) {
      const float theta = i * delta_theta;









    }
}
```

Problem 2: [20 pts] Recall that in Homework 2 a sphere was rendered using a spiral slice. The coordinates of the slice were placed in a buffer object and a sphere was rendered by performing multiple rendering passes each using the same buffer object but a different transformation. Let c denote the number of coordinates in the buffer object, and s denote the number of slices per sphere.

(a) What is the size of the buffer object, in bytes? *Hint: This is an easy question.*

☐ Size of buffer object, in bytes. ☐ State any assumptions made.

(b) Appearing below is the code rendering a sphere, adapted from the Homework 2 solution. Based on this code, estimate the amount of data send from the CPU to the GPU to render the sphere, **not counting the buffer object**.

```
pMatrix_Rotation rot_xz( pVect(0,1,0), delta_theta ),  rot_yz( pVect(1,0,0), M_PI );
// Code to render one sphere.
glTranslatefv( ball->position );
glScale1f( ball->radius );
glMultTransposeMatrixf( pMatrix_Rotation(ball->orientation) );

for ( int j=0; j<2; j++ ) {
     for ( int i=0; i<s/2; i++ ) {
        glColor3fv( i & 1 ? color_lsu_spirit_purple : ball->color );
        glDrawArrays(GL_TRIANGLE_STRIP,0,c);
        glMultTransposeMatrixf(rot_xz);
      }
    glMultTransposeMatrixf(rot_yz);
  }
```

☐ In terms of s, amount of data sent to GPU to render sphere.

(c) Buffer object BO2 holds coordinates to render an entire sphere, not just a slice, $s \times c$ coordinates total. Buffer object BO1 holds the slice discussed above. Suppose that initially all data is on the CPU. Let n be the number of sphere to be rendered. In terms of s and c, what is the smallest value of n for which using BO2 will require less data than BO1?

☐ Break-even value for n:

Problem 3: [25 pts] The diagram below shows a texture on the left-hand side and a scene to be rendered on the right. The scene consists of three triangles with the texture applied. The text in the texture, written by Mathy Vanhoef, is from the description of a widespread vulnerability in WiFi WPA2 implementations. See `https://www.krackattacks.com`, and please, Fall 2017 people, update your OS and wireless access point firmware. *Grading Note: The description of where the text came from was not in the original exam. The KRACK vulnerability became public two days before the midterm exam.*

Texture

We discovered serious weaknesses

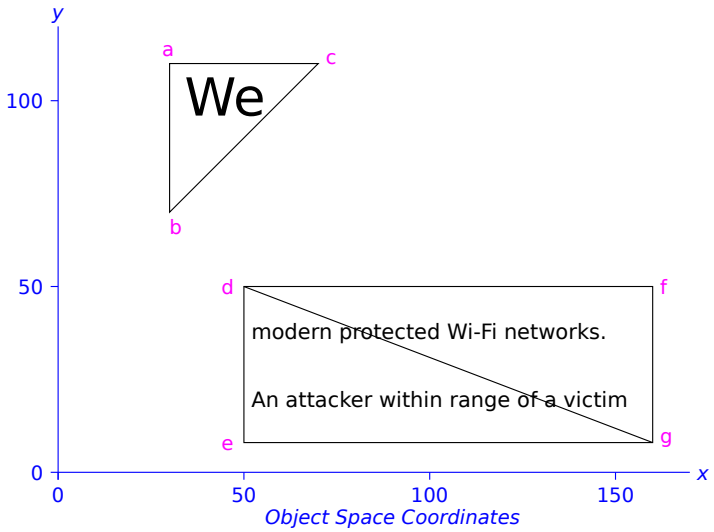
in WPA2, a protocol that secures all

modern protected Wi-Fi networks.

An attacker within range of a victim

can exploit these weaknesses using

key reinstallation attacks (KRACKs).

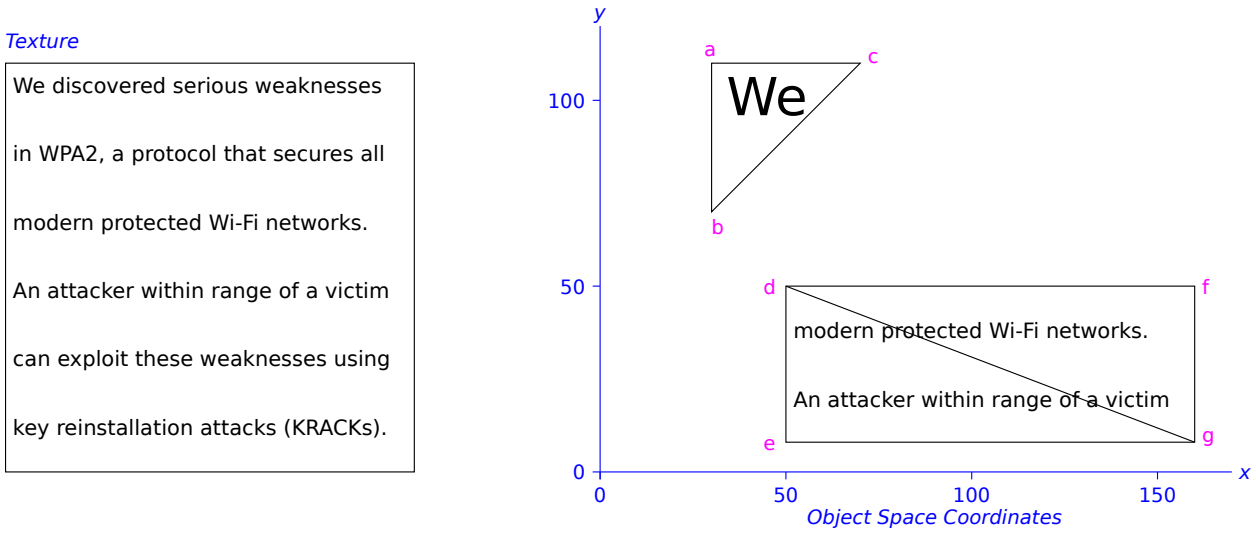


(*a*) Complete the code below so that it renders the triangles with the texture applied as shown. The color and normal have already been set, just specify vertex and texture coordinates. Use abbreviation `glV` for `glVertex2f` and `glT` for `glTexCoord2f`. Only specify x and y components for the vertex coordinates.

☐ Specify ☐ texture and ☐ vertex coordinates so that the scene is rendered as shown.

```
glBegin(GL_TRIANGLES);
glNormal3f(0,0,1);
glColor3f(0.5,0.5,0.5);

glV(30,110);   // Vertex a









glEnd();
```

Problem 3, continued:



(*b*) Suppose that defg in the diagram above is four pixels across and two pixels down. Suppose that texturing were set up to use only a single mipmap level and nearest filtering. Explain how defg could appear all white in that case.

☐ At 4×2 pixels defg all white because:

(*c*) Explain how the 4×2 rectangle would appear with linear filtering and mipmap levels. Explain the key points of linear texture filtering.

☐ Appearance with mipap levels and linear filtering.

☐ Key points of filtering.

Problem 4: [25 pts] Answer each question below.

(*a*) Show a transformation matrix M that will translate a homogeneous coordinate P to $P + \begin{bmatrix} 7 \\ 6 \\ 5 \end{bmatrix}$. (That is, find M such that $MP = P + \begin{bmatrix} 7 \\ 6 \\ 5 \end{bmatrix}$.) Only show nonzero array elements.

☐ $M =$

(*b*) Describe the "from" and "to" coordinate spaces for the following OpenGL matrices:

☐ The ModelView matrix maps a coordinate from ☐ _____ space to ☐ _____ space.

☐ The Projection matrix maps a coordinate from ☐ _____ space to ☐ _____ space.

(*c*) Show two drawings, each consisting of four triangles. The one should be well-suited for rendering using a triangle strip, for the other a triangle strip should make no difference.

☐ Show a 4-triangle example well-suited to triangle strips and one in which a ☐ triangle strip doesn't help at all.

(*d*) Suppose that buffer object o30 has 30 sphere coordinates and buffer object o1k has 1000 sphere coordinates. In both buffer objects the sphere coordinates are well chosen. They will be used to render a sphere which is fully visible. *Note: In the original exam the sizes were 1000 and 10,000.*

☐ Describe a situation in which the sphere would look better using o1k than using o30 and ☐ describe a situation in which the sphere would look the same with o1k as with o30. ☐ Explain.

Suppose that it takes much more time to render a sphere using o1k than with o30. The vertex, geometry, and fragment shader stages are running custom written code which could use some tuning.

☐ Which shaders should be tuned to potentially fix the performance problem? ☐ Which shaders be ignored? ☐ Explain.