CSCE 4813 - Computer Graphics
Final Exam – Spring 2013

Student Name: _______________________

Student UAID: _______________________

I pledge that I have neither given nor received unauthorized help on this exam

Instructions: This is closed book exam. Students are allowed one 8.5 x 11 page of notes, but no calculators or other electronic devices. The exam worth a total of 100 points (5 questions worth 20 points each). Please read all questions before starting the test and schedule your time accordingly.

1) Computer Shading (20 points)

In the Phong shading model, the amount of light reflected from a surface has three components: the diffuse term D, the specular term S, and the ambient term A.

a) Draw a side view diagram that shows how light illuminates a surface. Include the surface normal N, the vector to the light source L, the light reflection vector R, and the vector to the viewer V.

b) What is the formula for calculating the diffuse component of the Phong shading model?

c) What is the formula for calculating the specular component of the Phong shading model?

d) When a polygon is part of a curved surface like a sphere, the points on the polygon will typically have different surface normals. Describe one way we can use this information to perform smooth shading on the polygon.

2) Line Drawing (20 points)

Consider the following line drawing code:

```c
int Image[10][10];
void draw_line(int x1, int y1, int x2, int y2, int value)
{
    double dx = x2 - x1;
    double dy = y2 - y1;
```
if (dx < dy)
{
    double x = x1 + 0.5;
    for (int y = y1; y <= y2; y++)
    {
        Image[y][(int)x] = value;
        x += dx/dy;
    }
}
else
{
    double y = y1 + 0.5;
    for (int x = x1; x <= x2; x++)
    {
        Image[(int)y][x] = value;
        y += dy/dx;
    }
}

a) Consider what happens if we call “draw_line(3, 2, 7, 7, 1)” from the main program. What will the values of dx and dy be? Show the resulting line in the image above.

b) Consider what happens if we call “draw_line(2, 9, 4, 1, 2)” from the main program. What will the values of dx and dy be? Show the resulting line in the image above.

c) What is the primary advantage of the Bresenham line drawing algorithm compared to the simple DDA approach above?
3) Drawing Circles (20 points)

In many interactive drawing applications, the user can select a “paint brush” and move their mouse around on the screen to create a “painting”. To make this work, we must be able to rapidly draw colored circles at different locations.

a) Describe how you could modify the polygon “flood fill” rendering algorithm to draw colored circles on the screen.

b) Describe how you could modify the polygon “scan line” rendering algorithm to draw colored circles on the screen.

c) Give one advantage of the “flood fill” algorithm versus the “scan line” algorithm.

d) Give one advantage of the “scan line” algorithm versus the “flood fill” algorithm.

4) Texture Mapping (20 points)

As you know, texture mapping is used to make simple polygon models look more realistic without having to create millions of polygons with different colors. In this question, we will discuss some of the implementation details for texture mapping. Assume that you are given the following image of the diet coke logo, and your task is to texture map this image onto a cylinder, so you can have diet coke cans in your graphics scene.

![Diet Coke](image)

a) How would you break your cylinder model into N polygons? Draw a diagram that shows a side view of your cylinder to illustrate your polygon model.

b) How would you map the texture onto these polygons? Be specific about how you would map polygon coordinates into texture coordinates. Where do you specify this mapping in a typical OpenGL texture mapping application?

c) What happens if the polygon in your image is larger than your texture map image? How are the pixel color values calculated for the polygon?
5) Ray Tracing (20 points)

Ray tracing was invented in the early 1980s to create realistic images, and now they are widely used in animation sequences in television and movies. In this question, we will discuss some of the implementation issues with ray tracing. Assume that you have a scene made up of N spheres with different sizes, locations, colors and reflection properties.

a) Draw a small diagram that shows the focus point of the camera, the imaging plane, and one ray being traced into the scene that hits a sphere.

b) If we have L light sources in the scene, how do we decide how much light from each light source is directly illuminating the sphere where the ray intersected the sphere? How is this different (better) than what we would get using traditional scan-line approaches?

c) Draw a diagram that shows what happens to your ray when it intersects a sphere made of shiny metal. How do we decide what the pixel color should be for this ray?

d) Draw a diagram that shows what happens to your ray when it intersects a sphere made of clear glass. How do we decide what the pixel color should be for this ray?