Daniel Meyer

CSE 420-01

Homework 2

Homework 2 Report

Part 1: (success)

a) (1, 1, 1), (1, 2, 1), (3, 0, 4)

A = P2 – P1 = (1, 2, 1) – (1, 1, 1) = (0, 1, 0)

B = P3 – P1 = (3, 0 ,4) – (1, 1, 1) = (2, -1, 3)

N = A X B = (3 – 0)i + (0 – 0)j + (0 – 2)k = <u>3i + 0j – 2k</u>

b) (6, 3, -4), (0, 0, 0), (2, 1, -1)

A = P2 – P1 = (0, 0, 0) – (6, 3, -4) = (-6, -3, 4)

B = P3 – P1 = (2, 1, -1) – (6, 3, -4) = (-4, -2, 3)

N = A X B = (-9 + 8)i + (-16 + 18)j + (12 - 12)k = <u>-1i + -2j + 0k</u>

Plane -> 5x – 3y + 6z = 7

Unit normal = N/|N| = (5, -3, 6) / sqrt($5^2 - 3^2 + 6^2$) = (5, -3, 6) / sqrt(70)

= <u>(5/sqrt(70), -3/sqrt(70), 6/sqrt(70))</u>

F(P1) = F(1, 5, 2) = 5(1) – 3(5) + 6(2) – 7 = -5 -> -5 < 0 so behind the plane

F(P2) = F(-3, -1, -2) = 5(-3) – 3(-1) + 6(-2) – 7 = -7 -> -7 < 0 so behind the plane

<u>Both points are behind the plane</u>

Part 2: (success)

P = (1, 2, 3)

a) Sphere: $x^2 + y^2 + z^2 = 14$

N = (1, 2, 3) - (0, 0, 0) = (1, 2, 3)

Unit normal = (1, 2, 3) / sqrt($1^2 + 2^2 + 3^2$) = (1, 2, 3) / sqrt(14)

= <u>(1/ sqrt(14), 2/sqrt(14), 3/sqrt(14))</u>

b) Plane: 3x – 4y + 2z – 1 = 0

N = (3, -4, 2)

Unit Normal = (3, -4, 2) / sqrt($3^2 - 4^2 + 2^2$) = (3, -4, 2) / sqrt(29)

= <u>(3/sqrt(29), -4/sqrt(29), 2/sqrt(29)</u>

Part 3: (success)

a) Q1 using Newell's method

Nx = (1 - 2)(1 + 1) + (2 - 0)(1 + 4) + (0 - 1)(4 + 1) = 3

Ny = (1 - 1)(1 + 1) + (1 – 4)(1 + 3) + (4 - 1)(3 + 1) = 0

Nz = (1 - 1)(1 + 2) + (1 - 3)(2 + 0) + (3 - 1)(0 + 1) = -2

<u>Normal = (3, 0 ,-2), same answer</u>

b) (1, 1, 2), (2, 0, 5), (5, 1, 4), (6, 0, 7)

Nx = (1 – 0)(2 + 5) + (0 – 1)(5 + 4) + (1 – 0)(4 + 7) + (0 – 1)(7 + 2) = 0

Ny = (2 – 5)(1 + 2) + (5 – 4)(2 +5) + (4 – 7)(5 + 6) + (7 – 2)(6 + 1) = 0

Nz = (1 – 2)(1 + 0) + (2 – 5)(0 + 1) + (5 – 6)(1 + 0) + (6 – 1)(0 + 1) = 0

<u>Normal = (0, 0, 0)</u>

Part 4: (success)

A = $(2, -1, 1)^T$

B = $(1, 1, -1)^T$

a) |A||B|Cos(theta) = A.B

A.B = (2)(1) + (-1)(1) + (1)(-1) = 2 – 1 – 1 = 0

|A| = sqrt(2^2 - 1^2 + 1^2) = sqrt(6)

|B| = sqrt(1^2 + 1^2 - 1^2) = sqrt(3)

Cos(theta) = 0 / (sqrt(6) * sqrt(3)) = 0

Theta = $\text{Cos}^{-1}(0)$ = <u>90 degrees</u>

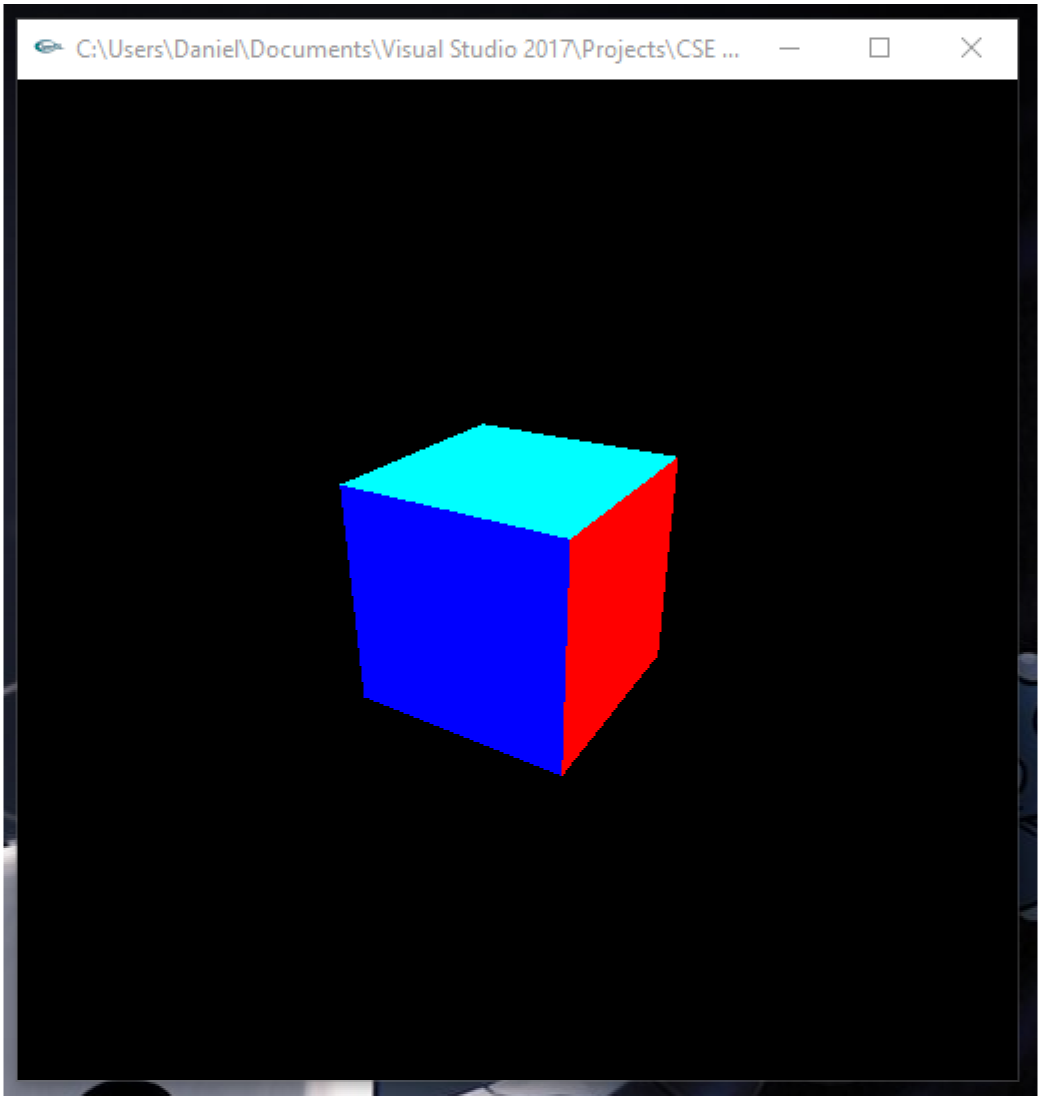
b) Unit normal = N / |N| = A X B / |A X B|

A X B = (1 – 1)i + (1 + 2)j + (2 + 1)k = 0i + 3j + 3k

Unit normal = (0, 3, 3) / sqrt(0^2 + 3^2 - 3^2) = (0, 3, 3) / sqrt(18)

= <u>(0, 3/sqrt(18), 3/sqrt(18)</u>

Part 5: (success)




```
void drawCube()
{
	glLoadIdentity();		/* clear the matrix */
			/* viewing transformation  */
	gluLookAt(3.0, 3.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

	static GLint vertices[] = { -1, -1, -1,   //0
					1, -1, -1,
					1, 1, -1,
					-1, 1, -1,        //3
		-1, -1, 1,
		1, -1, 1,
		1, 1, 1,        //6
		-1, 1, 1 };       //7
```

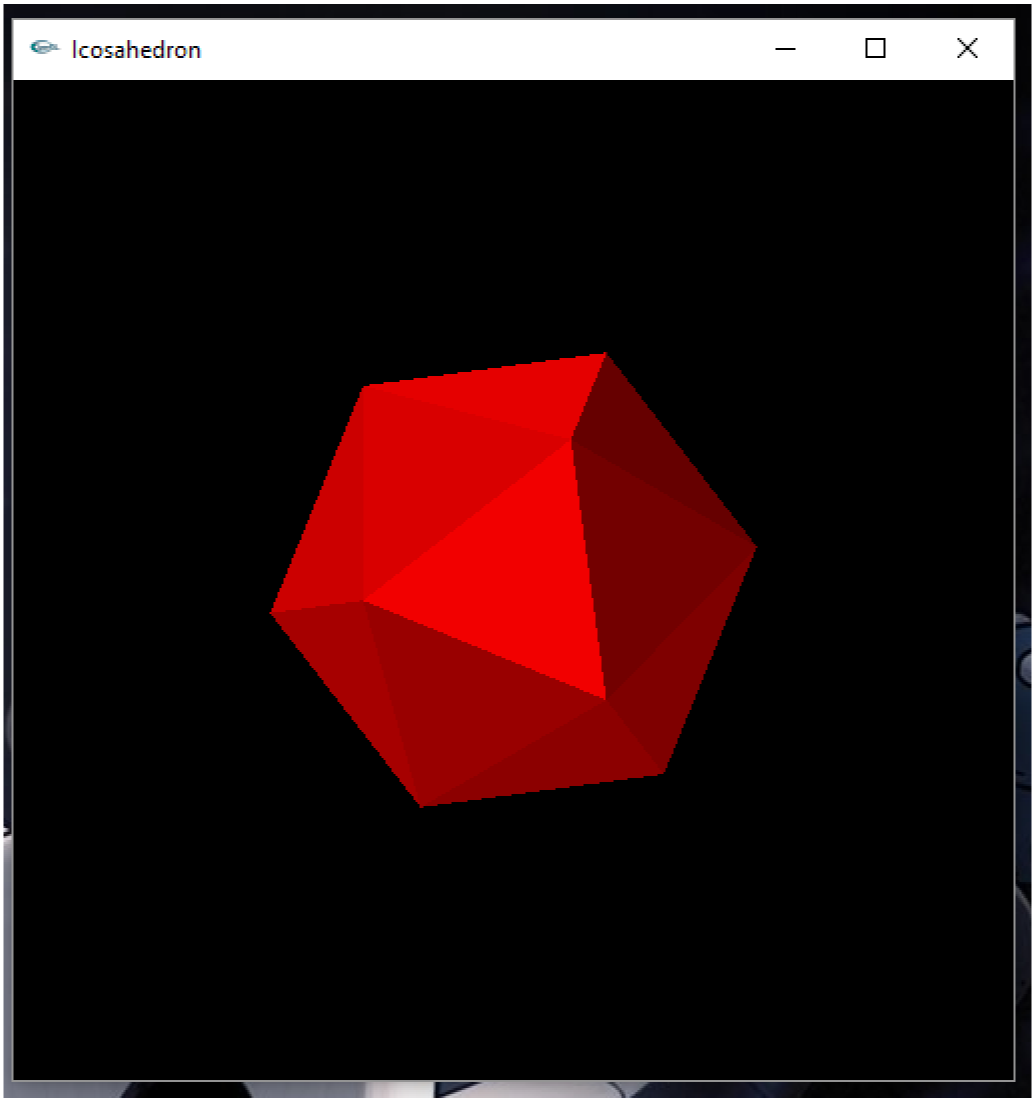
```
    glVertexPointer(3, GL_INT, 0, vertices);

    glEnable(GL_CULL_FACE);
    glCullFace(GL_BACK);

    static GLubyte frontIndices[] = { 4, 5, 6, 7 };
    static GLubyte rightIndices[] = { 1, 2, 6, 5 };
    static GLubyte bottomIndices[] = { 0, 1, 5, 4 };
    static GLubyte backIndices[] = { 0, 3, 2, 1 };
    static GLubyte leftIndices[] = { 0, 4, 7, 3 };
    static GLubyte topIndices[] = { 2, 3, 7, 6 };

    glColor3f(0.0, 0.0, 1.0);
    glDrawElements(GL_QUADS, 4, GL_UNSIGNED_BYTE, frontIndices);
    glColor3f(1.0, 0.0, 0.0);
    glDrawElements(GL_QUADS, 4, GL_UNSIGNED_BYTE, rightIndices);
    glColor3f(1.0, 1.0, 0.0);
    glDrawElements(GL_QUADS, 4, GL_UNSIGNED_BYTE, bottomIndices);
    glColor3f(1.0, 0.0, 1.0);
    glDrawElements(GL_QUADS, 4, GL_UNSIGNED_BYTE, backIndices);
    glColor3f(0.0, 1.0, 0.0);
    glDrawElements(GL_QUADS, 4, GL_UNSIGNED_BYTE, leftIndices);
    glColor3f(0.0, 1.0, 1.0);
    glDrawElements(GL_QUADS, 4, GL_UNSIGNED_BYTE, topIndices);
}
```

Part 6: (success)




```
#define X .525731112119133606
#define Z .850650808352039932

static GLfloat vdata[12][3] = {
   {-X, 0.0, Z}, {X, 0.0, Z}, {-X, 0.0, -Z}, {X, 0.0, -Z},
   {0.0, Z, X}, {0.0, Z, -X}, {0.0, -Z, X}, {0.0, -Z, -X},
   {Z, X, 0.0}, {-Z, X, 0.0}, {Z, -X, 0.0}, {-Z, -X, 0.0}
};
static GLuint tindices[20][3] = {
   {0,4,1}, {0,9,4}, {9,5,4}, {4,5,8}, {4,8,1},
   {8,10,1}, {8,3,10}, {5,3,8}, {5,2,3}, {2,7,3},
   {7,10,3}, {7,6,10}, {7,11,6}, {11,0,6}, {0,1,6},
```

```
   {6,1,10}, {9,0,11}, {9,11,2}, {9,2,5}, {7,2,11} };

void display(void)
{
      glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
      glMatrixMode(GL_MODELVIEW); // position and aim the camera
      glLoadIdentity();
      gluLookAt(8.0, 8.0, 8.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

      glEnable(GL_CULL_FACE);
      glCullFace(GL_BACK);

      glBegin(GL_TRIANGLES);
      for (int i = 0; i < 20; i++)
      {
            glColor3f(i / 20.0, 0.0, 0.0);   //has no effect if Light
enabled
            glVertex3fv(&vdata[tindices[i][0]][0]);
            glVertex3fv(&vdata[tindices[i][1]][0]);
            glVertex3fv(&vdata[tindices[i][2]][0]);
      }
      glEnd();
      glFlush();
}
```

Summary:

This assignment focused on finding the dot product, cross product (normal), and unit normal of various vectors, a plane, and a sphere. I was able to calculate these as seen by the steps shown above and also used a new method provided called Newell's method to find the normal of two different sets of vectors. The next half of the assignment was to create a program that draws a cube with different colored faces of which I was able to create, compile, and run without errors. The next part of the programming section was to create either an icosahedron or a dodecahedron of which I made the former and was able to compile and run without errors. I believe I have earned the full 60 points for the assignment.