Yazhuo Liu
Lab 6

Cross product & dot product:
A = (3, 0, 2) B = (4, 1, 8)

i	j	k	0k	
3	0	2	-2i	
4	1	8	-24j	0i
i	j	k	3k	
3	0	2	8j	

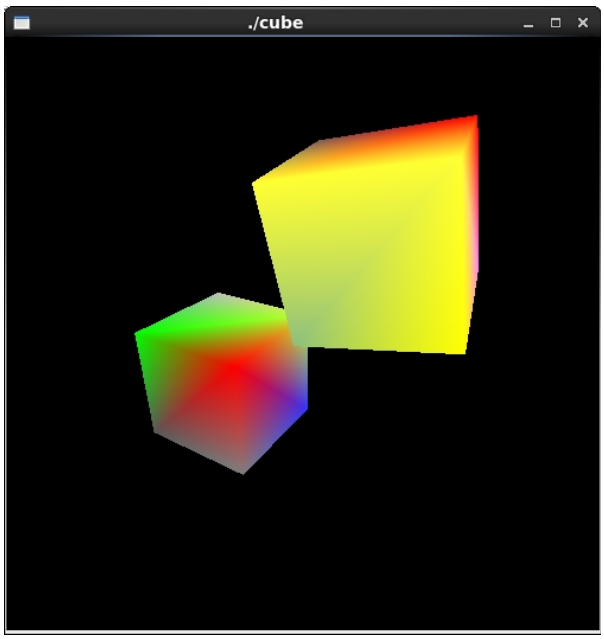
A x B = (-2, -16, 3)
B x A = - (A x B) = (2, 16, -3)

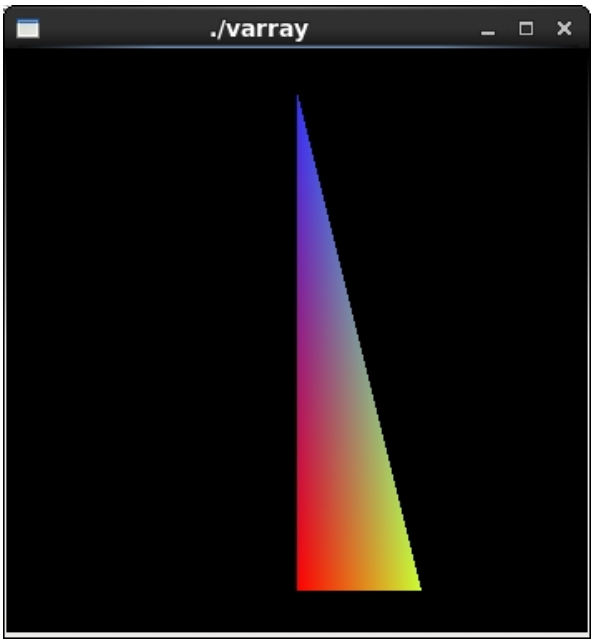
A . B = AxBx + AyBy + AzBz = 12 + 0 + 16 = 28

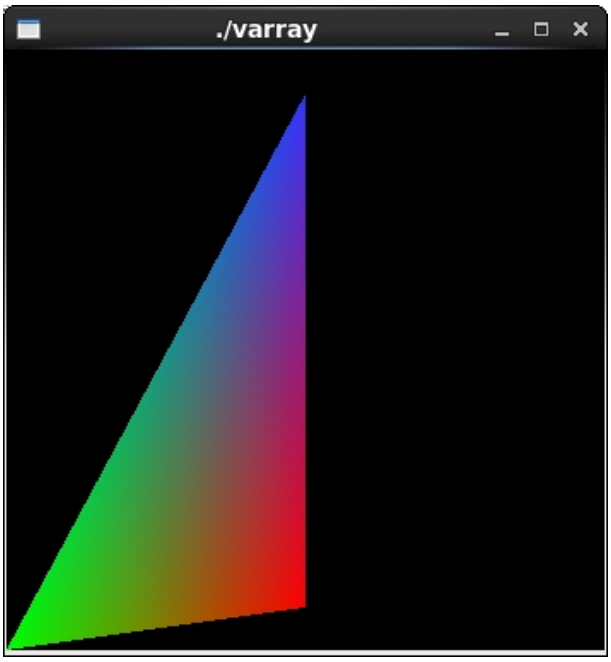
Normal vector:
P1 = (1, 1, 1) P2 = (1, 2, 1) P3 = (3, 0, 4)
A = P2 – P1 = (0, 1, 0) B = P3 – P1 = (2, 1, -3)

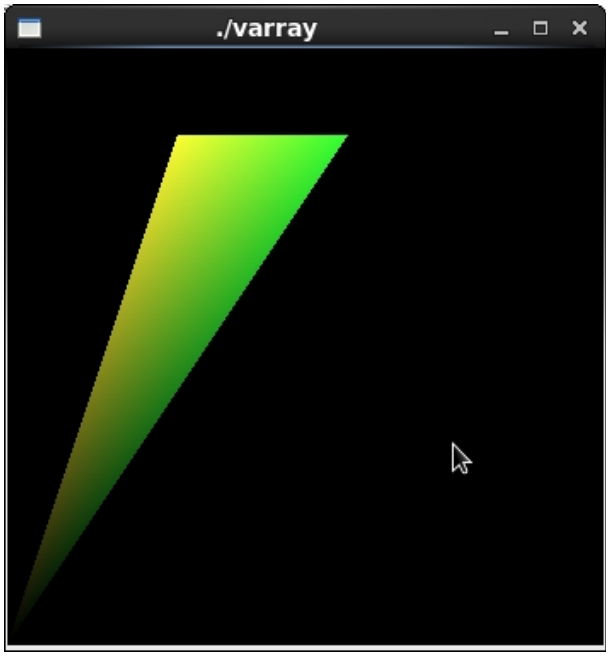
n = A x B = (3, 0, -2)

Output:









Partial Code:
//Cube.cpp
...

```
  static GLfloat colors[] = {1.0, 0.2, 0.2,
                    0.2, 0.2, 1.0,
                    0.8, 1.0, 0.2,
                    0.75, 0.75, 0.75,
                    0.35, 0.35, 0.35,
                    0.5, 0.5, 0.5,
                              1.0, 0.0, 0.0,
                              0.0, 1.0, 0.0
                              };

  static GLfloat diff_colors[] = {1.5, 1.2, 0.0,
                    1.5, 0.5, 1.5,
                    1.2, 0.0, 0.0,
                    0.5, 0.35, 0.35,
                    0.55, 0.75, 0.5,
                    1.5, 1.5, 0.0,
                              1.5, 1.0, 0.2,
                              1.0, 1.2, 0.2
                              };


  glVertexPointer (3, GL_INT, 0, vertices);
  glColorPointer (3, GL_FLOAT, 0, colors);
  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};

   glTranslatef ( -1.5, -1.2, 0 );
   glDrawElements(GL_QUADS, 4, GL_UNSIGNED_BYTE, frontIndices);
   glDrawElements(GL_QUADS, 4, GL_UNSIGNED_BYTE, rightIndices);
   glDrawElements(GL_QUADS, 4, GL_UNSIGNED_BYTE, bottomIndices);
   glDrawElements(GL_QUADS, 4, GL_UNSIGNED_BYTE, backIndices);
   glDrawElements(GL_QUADS, 4, GL_UNSIGNED_BYTE, leftIndices);
   glDrawElements(GL_QUADS, 4, GL_UNSIGNED_BYTE, topIndices);

   glColorPointer (3, GL_FLOAT, 0, diff_colors);
   glRotatef ( 15, 1, 1, 1 );
   glTranslatef ( 3, 2.4, 0 );
   static GLubyte allIndices[] = {4, 5, 6, 7, 1, 2, 6, 5,
          0, 1, 5, 4, 0, 3, 2, 1,
          0, 4, 7, 3, 2, 3, 7, 6};
...

//varray.cpp
...
void display(void)
{
   glClear (GL_COLOR_BUFFER_BIT);
   if (derefMethod == DRAWARRAY)
     glDrawArrays (GL_TRIANGLES, 0, 6);
   else if (derefMethod == ARRAYELEMENT) {
     glBegin (GL_TRIANGLES);
     glArrayElement (2);   //note: vertices 1, 3, 5 is a straight line
     glArrayElement (3);
     glArrayElement (6);
     glEnd ();
   }
   else if (derefMethod == DRAWELEMENTS) {
    GLuint indices[4] = {0, 1, 3, 4};

     glDrawElements (GL_POLYGON, 4, GL_UNSIGNED_INT, indices);
   }
   glFlush ();
}
...
```

Report:

Finding the cross product and dot product is not hard, neither is finding the normal vector if you know the formula. I looked through the lecture notes and found the equations for normal vector. For the cube.cpp, I simply just added another set of colors with a different function name, and I added one line

before drawing the second cube to make the second cube using the different set of colors. For the triangles, I modified the glArrayElement so it displays a different triangle. I didn't have time to work on the extra credit because a professor was having a class in the lab and threw everybody out. Other than that, I think I finished this lab successfully.