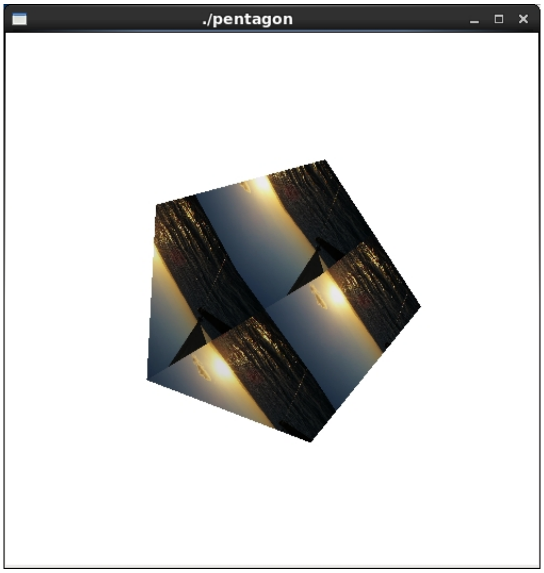
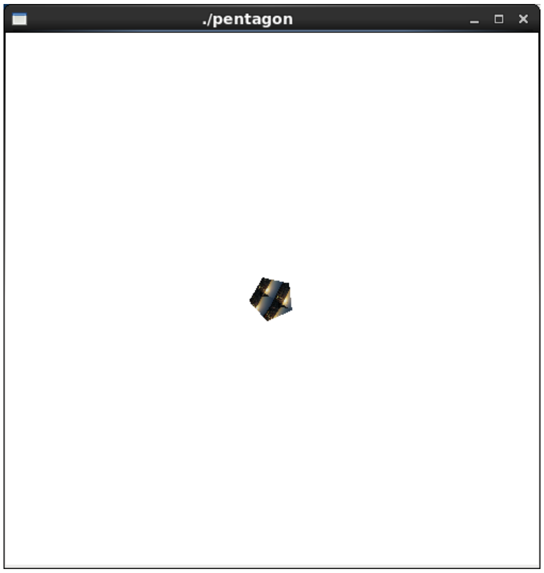
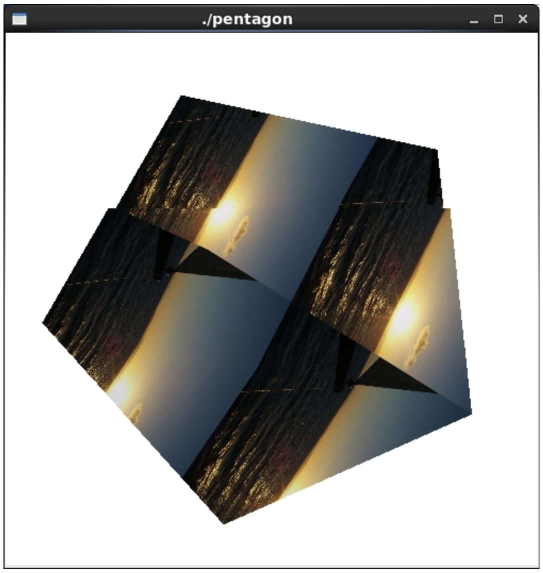
Yazhuo Liu
Midterm







Code:

```
//pentagon.cpp

...
static void Idle(void)
{
  glUniform1f( timeParam, glutGet ( GLUT_ELAPSED_TIME ) );
  glutPostRedisplay();
}
...
```

```
void display(void)
{
   GLfloat vec[4];

   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
   glClearColor( 1.0, 1.0, 1.0, 0.0 );               //get white background color

   glPushMatrix();

   glRotatef( anglez, 0.0, 0.0, 1.0);                   //rotate along z-axis

   glActiveTexture(GL_TEXTURE0);
   glBindTexture(GL_TEXTURE_2D, texName);

  float a = 2 * 3.14159265 / 5;      //angle subtended by one side
    float a1 = 0;
    glBegin ( GL_POLYGON );

      glTexCoord2f ( -0.7, -1.0 );
      glVertex2f (cos (a1 ), sin (a1) );
      a1 += a;
      glTexCoord2f ( -1.0, 0.3 );
      glVertex2f (cos (a1 ), sin (a1) );
      a1 += a;
      glTexCoord2f ( 0.0, 1.0 );
      glVertex2f (cos (a1 ), sin (a1) );
      a1 += a;
      glTexCoord2f ( 1.0, 0.3 );
      glVertex2f (cos (a1 ), sin (a1) );
      a1 += a;
      glTexCoord2f ( 0.7, -1.0 );
      glVertex2f (cos (a1 ), sin (a1) );
      a1 += a;
  glEnd();

   glPopMatrix();
   glutSwapBuffers();
   glFlush();
}

//pentagon.vert

varying float LightIntensity;
uniform vec3 LightPosition;
uniform float time;

//uniform float radius;

attribute float radius;
```

```
const float specularContribution = 0.1;
const float diffuseContribution  = 1.0 - specularContribution;

void main(void)
{
   vec3 ecPosition = vec3 (gl_ModelViewMatrix * gl_Vertex);
   vec3 tnorm      = normalize(gl_NormalMatrix * gl_Normal);
   vec3 lightVec   = normalize(LightPosition - ecPosition);
   vec3 reflectVec = reflect(-lightVec, tnorm);
   vec3 viewVec    = normalize(-ecPosition);

   float spec      = clamp(dot(reflectVec, viewVec), 0.0, 1.0);
   spec            = pow(spec, 16.0);

   LightIntensity  = diffuseContribution * max(dot(lightVec, tnorm), 0.0)
                   + specularContribution * spec;



   float s = 0;
   s = s + 2.0 * sin ( 0.0005 * time );
   radius = radius - sin( 0.0005 * time );

   gl_TexCoord[0]  = gl_MultiTexCoord0;
   gl_Position = gl_ModelViewProjectionMatrix * (vec4(s, s, s, 1.0)  * gl_Vertex );
}
```

Report:

I finished all parts of the midterm, but not 100% successful. There is not enough time to calculate the coordinates of the vertices of the pentagon. I'm giving myself 90 points.