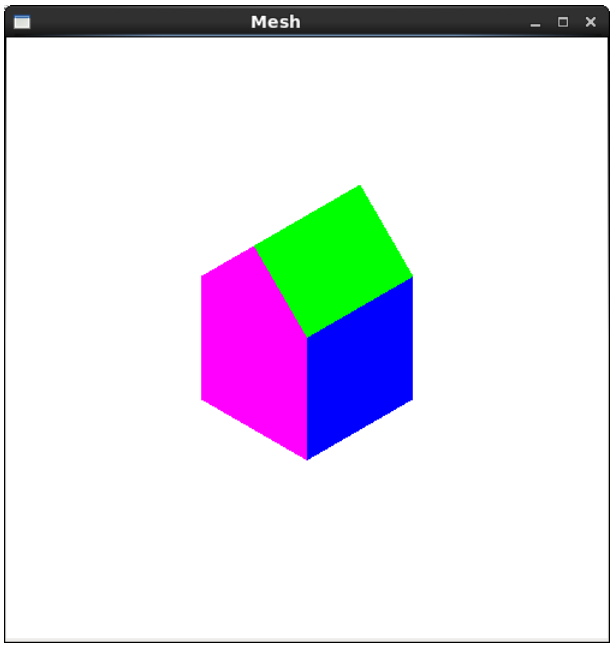
Yazhuo Liu
Lab 9

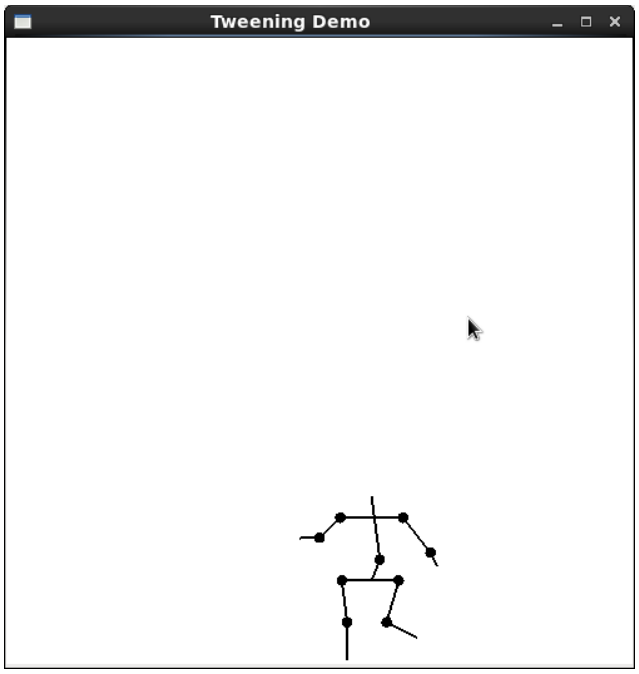
Exercise 1




```
void Mesh::drawMesh()        // use OpenGL to draw this mesh
{
  // draw each face of this mesh using OpenGL: draw each polygon.
  if( isEmpty() ) return; // mesh is empty

  glEnable ( GL_CULL_FACE );
  glCullFace ( GL_BACK );
  for(int f = 0; f < numFaces; f++) // draw each face
  {
    glBegin(GL_POLYGON);
    cout << endl;
    setColor( f );
    for(int v = 0; v < face[f].nVerts; v++) // for each vertex
    {
          int in = face[f].vert[v].normIndex ; // index of this normal
       int iv =  face[f].vert[v].vertIndex ; // index of this vertex
       glNormal3f(norm[in].x, norm[in].y, norm[in].z);
          cout << "[" << norm[in].x << "," << norm[in].y << "," <<
                    norm[in].z << "]" << "    ";
       glVertex3f(pt[iv].x, pt[iv].y, pt[iv].z);
          cout << "(" << pt[iv].x << "," << pt[iv].y << "," <<
                    pt[iv].z << ")" << "    ";
    }
    glEnd();
    cout << endl;
  }
} //drawMesh
```

Exercise 2




```
...
Point2 A[10], B[10];
Point2 A1[10], B1[10];
Point2 A2[10], B2[10];
Point2 center( 0, 0 );
float t = 0, deltat = 0.05;
float deltax = 1, deltay = 0;
...
void animate()
{
  t += deltat;
  center.x += deltax;    center.y += deltay;      //move center for clarity of display
  if ( t > 1 ){
          t = 1.0;
          deltat = -deltat;                       //reverse direction
       deltax = -deltax;
       deltay = -deltay;
  } else if ( t < 0 ) {
          t = 0;
          deltat = -deltat;                       //reverse direction
       deltax = -deltax;
       deltay = -deltay;
  }
  glutPostRedisplay ();
}
...
```

Report:

This lab is fairly easy. I think I successfully finished it.