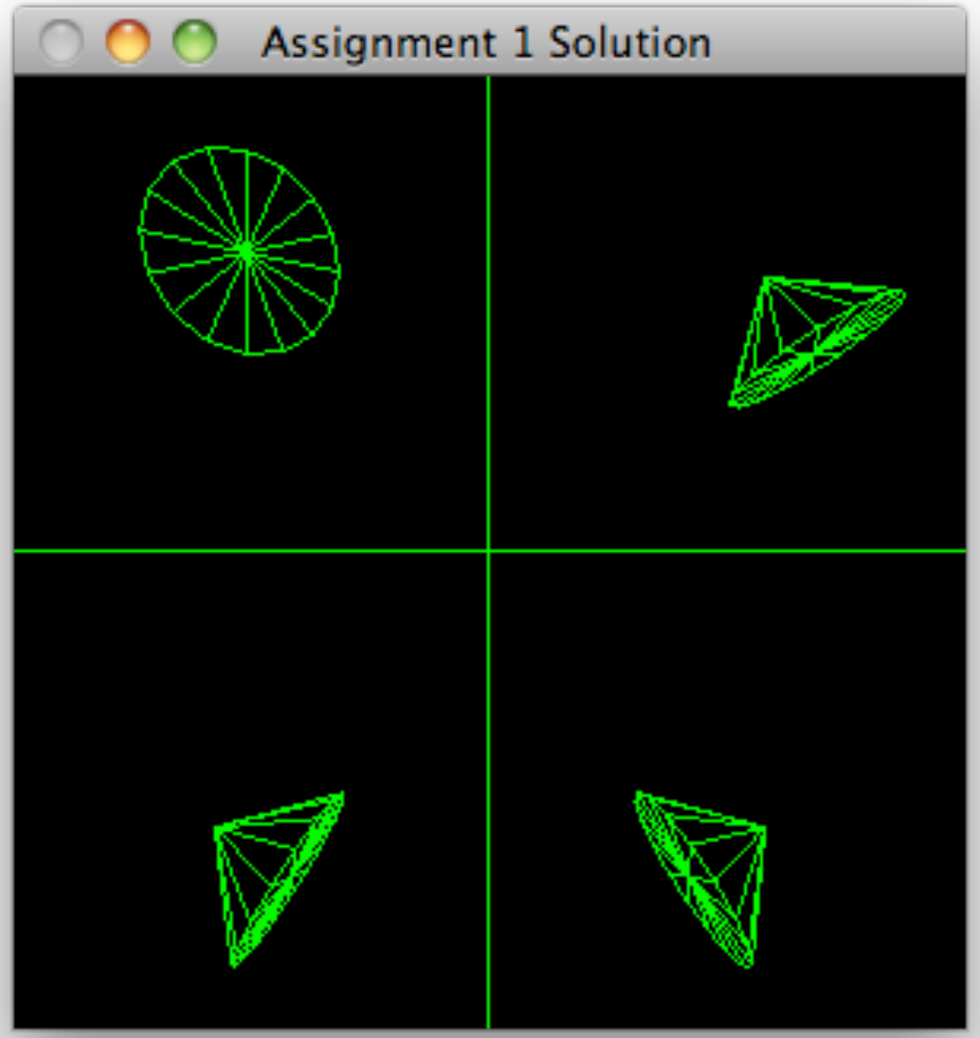
Assignment 1

OpenGL

Specification

- Develop a working OpenGL eclipse project to display 1 or more 3D cones, animated to rotate around one of their axis with single or varying frequency.



Supporting Labs

- Lab 01c introduced a simple timer based animation technique.
- Lab 03a introduced you to a simple modeling framework, which permitted geometry, loaded from a simple text file, to be rendered.
- Lab 04a introduced a you to rendering a simple 3D cone, including suitable techniques to rotate the cone around the x or y axis.

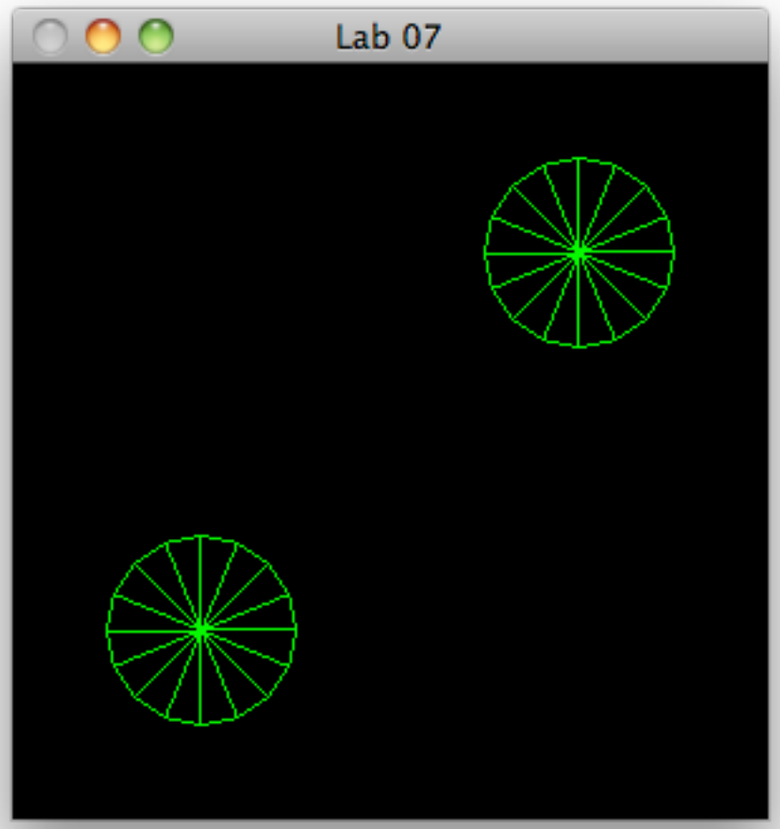
Steps

- Step 1: Refactor lab04a to draw multiple cones at different co-ordinates
- Step 2: Experiment with rotating these cones using the lab04A specialkeys code.
- Step 3: Introduce a Timer (Lab 01c) to trigger the rotations
- Step 4: Download and explore the Assignment 1 Shell code. Complete the implementation of this framework

Step 1: Refactor for lab04a for 2 Cones

```
void drawTriangleFan(float x, float y, float z, float radius)
{
  glBegin(GL_TRIANGLE_FAN);
  //...
  glEnd();
}
```

```
void drawCone(float x, float y, float z, float radius)
{
  //cone sides
  glFrontFace(GL_CW);
  drawTriangleFan(x,y,z,radius);
  //base
  glFrontFace(GL_CCW);
  drawTriangleFan(x,y,0,radius);
}
```




```
void renderScene(void)
{
  glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);

  drawCone(50,50,25, 25);
  drawCone(-50,-50, 25, 25);

  glFlush();
}
```

Step 2: Simple Rotation

- Remove the existing glRotate() call in specialKeys, and merely have the key press increment some global variable (xRot)
- Move the glRotate call into the rendedScene() method, before any calls to render the cones.
- Note carefully the axis around which the cones are rotating. Bear in mind you are trying to have each cone rotate around its own axis.
- Clearly a different axis could be specified, specifically an axis based on the cone's location.

```
void specialKeys(int key, int x, int y)
{
  if (key == GLUT_KEY_UP)
  {
    xRot++;
  }
  //...
  glutPostRedisplay();
}
```

```
void renderScene(void)
{
  glClear(GL_COLOR_BUFFER_BIT |
              GL_DEPTH_BUFFER_BIT);

  glRotatef(xRot, 1f, 0f, 0f);

  drawCone(50,50,25, 25);
  drawCone(-50,-50, 25, 25);

  glFlush();
}
```

glPushMatrix / glPopMatrix

- The call to glRotate will rotate *the entire world.* Ultimately, you wish to just rotate a cone individually.
- Doing this will require a rethink - and specifically each cone's rotation will need to be isolated from the others.
- This is accomplished by glPush/Pop Matrix.
- Bracketing your rotating/rendering with push/pop matrix calls should produce the desired effects.

```
glPushMatrix();
  // rotation and
  // rendering for cone 1
glPopMatrix();
glPushMatrix();
  // rotation and
  // rendering for cone 2
glPopMatrix();
```

Step 2: Timers

- Set a timer and trigger the rotation behavior.

- Based on Lab 01c

Assignment 1 Shell Project

- Download and build a project using the Assignment 1 Shell Archive (on moodle).
- This is a skeleton solution. It *includes:*
  - model loading (triangle and line strips only)
  - specification of cones in mode file
- It does *not include*
  - Cone class implementation.
  - Timer based rotation
- In addition, the sample model file is without any colour specifications.