OpenGL 3D Graphics in Liberty BASIC

Lesson Three: Rotation and Scaling

by Robert McAllister

Rotation and scaling are two separate operations but since both are fairly simple they will be combined into the same lesson.

Rotation:

To rotate the entire scene you can use a call to glRotate which has four parameters:

```
CALL glRotatef amt , X , Y , Z
```

The first parameter "amt" is the degrees of rotation you want to apply. The remaining three are used to specify which axis you want to rotate the scene around. Only one axis can be rotated at a time. Typically you will have three variables to keep track of each axis and then use a block of code such as this in your drawing routine:

```
CALL glRotatef xrot , 1 , 0 , 0
CALL glRotatef yrot , 0 , 1 , 0
CALL glRotatef zrot , 0 , 0 , 1
```

In the following snippet uncomment one rotation variable at a time to see the triangle rotate around each of the axis's. Or, uncomment all of them to see it go crazy.

```
'Rotating the scene with glRotate
FOR a = 1 TO 360
  xrot = a
  'yrot = a
  'zrot = a
  CALL
 ClearView eyeX , eyeY , eyeZ , centerX , centerY , centerZ , upX , up
Y , upZ
  CALL glRotatef xrot , 1 , 0 , 0
  CALL glRotatef yrot , 0 , 1 , 0
  CALL glRotatef zrot , 0 , 0 , 1
  CALL glBegin GL.TRIANGLES
```

```
    CALL glColor4fv 1 , 0 , 0 , 1
      CALL glVertex -1 , -1 , 0
    CALL glColor4fv 0 , 1 , 0 , 1
      CALL glVertex 0 , 1 , 0
    CALL glColor4fv 0 , 0 , 1 , 1
      CALL glVertex 1 , -1 , 0
  CALL glEnd
  CALL RefreshView
  CALL Pause 10
NEXT a
WAIT
```

You can also rotate an individual object using a little trigonometry.

```
'Rotating an object
FOR a = 1 TO 360
  angle1 = ((a+90)/57.2957)
  angle2 = ((a+270)/57.2957)
  xAdjustment1 = 0 + (sin(angle1))
  xAdjustment2 = 0 + (sin(angle2))
  CALL
 ClearView eyeX , eyeY , eyeZ , centerX , centerY , centerZ , upX , up
Y , upZ
  CALL glBegin GL.TRIANGLES
    CALL glColor4fv 1 , 0 , 0 , 1
      CALL glVertex 0 + xAdjustment1 , -1 , 0
    CALL glColor4fv 0 , 1 , 0 , 1
      CALL glVertex 0 , 1 , 0
    CALL glColor4fv 0 , 0 , 1 , 1
      CALL glVertex 0 + xAdjustment2 , -1 , 0
  CALL glEnd
  CALL RefreshView
  CALL Pause 10
NEXT a
WAIT
```

Scaling:

OpenGL has a handy function for scaling the entire scene, glScalef.

```
CALL glScalef X , Y , Z
```

With this function you can expand or shrink the entire scene (actually the 3D matrix) along any or all of the three axis's. If a value is set to 1, that axis will not be changed. Here we have the same old triangle but

it has been expanded along the X axis and shrunk along the Y axis.

```
'Scaling with glScalef
CALL
 ClearView eyeX , eyeY , eyeZ , centerX , centerY , centerZ , upX , up
Y , upZ
CALL glScalef 2 , .5 , 1
CALL glBegin GL.TRIANGLES
  CALL glColor4fv 1 , 0 , 0 , 1
    CALL glVertex -1 , -1 , 0
  CALL glColor4fv 0 , 1 , 0 , 1
    CALL glVertex 0 , 1 , 0
  CALL glColor4fv 0 , 0 , 1 , 1
    CALL glVertex 1 , -1 , 0
CALL glEnd
CALL RefreshView
WAIT
```

Scaling can also be accomplished with a scale factor. This can come in handy if you only want to change the size of one object in a scene.

```
'Scaling with a scale factor
FOR a = .5 TO 1.5 step .1
  CALL
 ClearView eyeX , eyeY , eyeZ , centerX , centerY , centerZ , upX , up
Y , upZ
  ScaleFactor = a
  CALL glBegin GL.TRIANGLES
    CALL glColor4fv 1 , 0 , 0 , 1
      CALL glVertex -1 * ScaleFactor , -1 * ScaleFactor , 0
    CALL glColor4fv 0 , 1 , 0 , 1
      CALL glVertex 0 * ScaleFactor , 1 * ScaleFactor , 0
    CALL glColor4fv 0 , 0 , 1 , 1
      CALL glVertex 1 * ScaleFactor , -1 * ScaleFactor , 0
  CALL glEnd
  CALL RefreshView
  CALL Pause 250
NEXT a
WAIT
```

In the next lesson we will learn what "Display lists" are.