

OpenGL 3D Graphics in Liberty BASIC

Lesson Two: Moving objects around in 3D

by Robert McAllister

There are two ways to move an object on the screen in 3D, change the location of the object or change your relative view of the object. Both methods require the scene to be redrawn for each frame.

Moving an object can be done by setting up a for/next loop and changing one or more of the X Y or Z values with each pass. You'll notice there is a pause added to each step so we can see what happens.

```
'Moving an object
FOR a = 1 TO 20
CALL
ClearView eyeX , eyeY , eyeZ , centerX , centerY , centerZ , upX , up
Y , upZ
CALL glBegin GL.TRIANGLES
    CALL glColor4fv 1 , 0 , 0 , 1
    CALL glVertex (-2 + xAdjustment) , -1 , 0
    CALL glColor4fv 0 , 1 , 0 , 1
    CALL glVertex (-1 + xAdjustment) , 1 , 0
    CALL glColor4fv 0 , 0 , 1 , 1
    CALL glVertex (0 + xAdjustment) , -1 , 0
CALL glEnd
CALL RefreshView
CALL Pause 150
xAdjustment = xAdjustment + .1
NEXT a
WAIT
```

Moving the camera can accomplish the same effect. The following is a line you will see used often in these programs.

```
CALL
ClearView eyeX , eyeY , eyeZ , centerX , centerY , centerZ , upX , up
Y , upZ
```

The values eyeX , eyeY , eyeZ can be thought of as the location of a camera taking a snapshot of the scene. And centerX , centerY , centerZ represents the location the camera is aimed at in the scene.

Setting up a for/next loop that changes the value of eyeX and centerX gives the same result as the program above.

```
'Moving the camera
FOR a = 1 TO 20
CALL
ClearView eyeX , eyeY , eyeZ , centerX , centerY , centerZ , upX , up
Y , upZ
CALL glBegin GL.TRIANGLES
    CALL glColor4fv 1 , 0 , 0 , 1
        CALL glVertex -2 , -1 , 0
    CALL glColor4fv 0 , 1 , 0 , 1
        CALL glVertex -1 , 1 , 0
    CALL glColor4fv 0 , 0 , 1 , 1
        CALL glVertex 0 , -1 , 0
CALL glEnd
CALL RefreshView
CALL Pause 150
eyeX = eyeX - .1
centerX = centerX - .1
NEXT a
WAIT
```

Here's a little program that moves a triangle around in a circle. What would happen if you put xAdjustment or yAdjustment in the Z value for the vertex calls?

```
'Moving an object in a circle
FOR a = 1 TO 360
angle=(a/57.2957)
xAdjustment=0+(sin(angle))
yAdjustment=0-(cos(angle))
CALL
ClearView eyeX , eyeY , eyeZ , centerX , centerY , centerZ , upX , up
Y , upZ
CALL glBegin GL.TRIANGLES
    CALL glColor4fv 1 , 0 , 0 , 1
        CALL glVertex (-1 + xAdjustment) , (-1 + yAdjustment) , 0
    CALL glColor4fv 0 , 1 , 0 , 1
        CALL glVertex (0 + xAdjustment) , (1 + yAdjustment) , 0
    CALL glColor4fv 0 , 0 , 1 , 1
        CALL glVertex (1 + xAdjustment) , (-1 + yAdjustment) , 0
CALL glEnd
CALL RefreshView
CALL Pause 10
NEXT a
WAIT
```

In the next lesson we will learn about [Rotation and scaling](#)