

OpenGL 3D Graphics in Liberty BASIC

Lesson Eight: Transparent Surfaces and Fog

by Robert McAllister

Transparent surfaces:

You can use a transparent surface to simulate a window or maybe the surface of some water. Back in Lesson One the alpha value for glColor4fv was briefly mentioned. Here we will use it to control the transparency of a surface. The acceptable values are between 0 & 1, with 0 being completely transparent and 1 being solid.

Before this will work a few OpenGL states need to be set. The first is to enable blending. This tells OpenGL that if there is a transparent surface in the scene that it needs to blend the colors wherever it interacts with another surface.

The second call needed is to glBlendFunc and it tells OpenGL which method to use when blending the colors.

The only thing to keep in mind is that the transparent surface needs to be created last in order to work.

```
'transparent triangle
GL.BLEND = 3042
GL.SRC.ALPHA = 770
GL.ONE.MINUS.SRC.ALPHA = 771

CALL glEnable GL.BLEND

CALLDLL #gl , "glBlendFunc" , _
    GL.SRC.ALPHA AS long , _
    GL.ONE.MINUS.SRC.ALPHA AS long , _
    ret AS long

FOR a = 1 TO 360
    CALL
ClearView eyeX , eyeY , eyeZ , centerX , centerY , centerZ , upX , up
Y , upZ
    CALL glRotatef a , 0 , 1 , 0

'solid black line
CALL glBegin GL.LINES
```

```
CALL glColor4fv 0 , 0 , 0 , 1
CALL glVertex -1 , 0 , -.5
CALL glVertex 1 , 0 , -.5
CALL glEnd
```

```
'transparent triangle
CALL glBegin GL.TRIANGLES
CALL glColor4fv 0 , 0 , 1 , .3
' make the surface 30 percent solid
CALL glVertex -1 , -1 , 0
CALL glVertex 0 , 1 , 0
CALL glVertex 1 , -1 , 0
CALL glEnd
```

```
CALL RefreshView
CALL Pause 10
NEXT a

WAIT
```

Fog:

Fog is an easy to add special effect with OpenGL. It can be used to add depth perception to a scene. The first step is to make a call to glEnable with GL.FOG as the argument. As with all other enabled functions, it can be turned off with a call to glDisable.

Next is to set the color of the fog. Again, the values for the individual color should be between 0 & 1. The default values are 0,0,0,0.

Then we need to set the density of the fog. This value should also be between 0 & 1, with the default value being 1.

```
'fog
GL.FOG = 2912
GL.FOG.DENSITY = 2914
GL.FOG.COLOR = 2918

CALL glEnable GL.FOG

STRUCT fogColor , red AS ulong , green AS ulong , blue AS
ulong , alpha AS ulong
fogColor.red.struct = R4( .5 )
fogColor.green.struct = R4( .5 )
fogColor.blue.struct = R4( .5 )
fogColor.alpha.struct = R4( 1 )
```

```
CALLDLL #gl , "glFogfv" ,_ ' set fog color to gray
    GL.FOG.COLOR AS long,_
    fogColor AS STRUCT ,_
    ret AS void

density = R4( .25 )
CALLDLL #gl , "glFogf" ,_ ' set the density of the fog
    GL.FOG.DENSITY AS long,_
    density AS ulong ,_
    ret AS void

FOR a  = 1 TO 345
    CALL
ClearView eyeX , eyeY , eyeZ , centerX , centerY , centerZ , upX , up
Y , upZ
    CALL glColor4fv 0 , 0 , 0 , 1
    CALL glBegin GL.QUADS
        CALL glVertex -2 , -1 , -2
        CALL glVertex -2 , -1 , 2
        CALL glVertex 2 , -1 , 2
        CALL glVertex 2 , -1 , -2
    CALL glEnd
    CALL glRotatef a , 0 , 1 , 0
    CALL glColor4fv 0 , 0 , 1 , 1
    CALL glBegin GL.TRIANGLES
        CALL glVertex -1 , -1 , -2
        CALL glVertex 0 , 1 , -2
        CALL glVertex 1 , -1 , -2
    CALL glEnd
    CALL glBegin GL.TRIANGLES
        CALL glVertex -1 , -1 , 0
        CALL glVertex 0 , 1 , 0
        CALL glVertex 1 , -1 , 0
    CALL glEnd
    CALL glBegin GL.TRIANGLES
        CALL glVertex -1 , -1 , 2
        CALL glVertex 0 , 1 , 2
        CALL glVertex 1 , -1 , 2
    CALL glEnd
    CALL RefreshView
    CALL Pause 15
NEXT a

WAIT
```

In the next lesson we will dig into "[OpenGL calls and argument types](#)"