

Near the top of the program:

```
#include "setmaterial.cpp"
#include "setlight.cpp"
#include "osusphere.cpp"
//include "osucone.cpp"
//include "osutorus.cpp"
#include "bmptotexture.cpp"
//include "loadobjfile.cpp"
//include "keytime.cpp"
//include "glsprogram.cpp"
```

In your global variables:

```
GLuint   SphereDL, VenusDL;           // display lists
GLuint   VenusTex;                    // texture object
bool     LightingOn;                  // are we using lighting, true or false?
```

In InitGraphics():

```
int width, height;
char *file = (char *)"venus.bmp";
unsigned char *texture = BmpToTexture( file, &width, &height );
if( texture == NULL )
    fprintf( stderr, "Cannot open texture '%s'\n", file );
else
    fprintf( stderr, "Opened '%s': width = %d ; height = %d\n", file, width, height );

glGenTextures( 1, &VenusTex );
glBindTexture( GL_TEXTURE_2D, VenusTex );
glPixelStorei( GL_UNPACK_ALIGNMENT, 1 );
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT );
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT );
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR );
glTexParameteri( GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR );
glTexImage2D( GL_TEXTURE_2D, 0, 3, width, height, 0, GL_RGB, GL_UNSIGNED_BYTE, texture );
```

In InitLists():

```

SphereDL = glGenLists( 1 );
    glNewList( SphereDL, GL_COMPILE );
        OsuSphere( 1., ??, ?? );           // use enough polygonal detail to look nice
    glEndList( );

VenusDL = glGenLists( 1 );
    glNewList( VenusDL, GL_COMPILE );
    glPushMatrix( );
        glScalef( 0.95f, 0.95f, 0.95f );    // scale of venus sphere, from the table
        glCallList( SphereDL );             // a dl can call another dl that has been
                                            // previously created

    glPopMatrix( );
    glEndList( );

```

In Display():

```

SetPointLight( GL_LIGHT0, ...)
glEnable( GL_LIGHTING );
glEnable( GL_LIGHT0 );
glEnable( GL_TEXTURE_2D );

glBindTexture( GL_TEXTURE_2D, VenusTex );
If( LightingOn )
    glTexEnvf( GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE );
else
    glTexEnvf( GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_REPLACE );
glCallList( VenusDL );

glDisable( GL_TEXTURE_2D );
glDisable( GL_LIGHTING );

```