

Near the top of the program:

```
#include "setmaterial.cpp"
#include "setlight.cpp"
#include "osusphere.cpp"
#include "osucube.cpp"
#include "osucylindercone.cpp"
#include "osutorus.cpp"
#include "bmptotexture.cpp"
#include "loadobjmtlfiles.cpp"
// #include "keytime.cpp"
// #include "glslprogram.cpp"
```

In your global variables:

```
GLuint  SphereDL, ...           // display lists
GLuint  SphereTex, ...         // texture objects
```

In InitGraphics():

```
int width, height;
char *file = (char *) "spheretex.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, &SphereTex );
glBindTexture( GL_TEXTURE_2D, SphereTex );
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 );
        SetMaterial( ????? );
        glBindTexture( GL_TEXTURE_2D, SphereTex );
        OsuSphere( ??, 64, 64 );    // use enough polygonal detail to look nice
    glEndList( );

...

// THE OBJ OBJECT IS TREATED DIFFERENTLY:
GLuint dl = LoadObjMtlFiles( (char *)"dino.obj";
ObjDL = glGenLists( 1 );
glNewList( ObjDL, GL_COMPILE );
    SetMaterial( ????? );
    glBindTexture( GL_TEXTURE_2D, ObjTex );
    glCallList( dl );
glEndList( );

```

In Display():

```

if( <<we-are-in-texture-mode>> )
    glEnable( GL_TEXTURE_2D );
else
    glDisable( GL_TEXTURE_2D );

glEnable( GL_LIGHTING );
glEnable( GL_LIGHT0 );
SetPointLight( GL_LIGHT0, ... );
glTexEnvf( GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE );

if( <<we-want-to-display-the-sphere>> )
{
    glCallList( SphereDL );
}

...

glDisable( GL_TEXTURE_2D );
glDisable( GL_LIGHTING );

```