Open Scene Graph Shadows

In this exercise you will take the basic scene you made in the introductory tutorial and add shadows to specified objects.

- Create a new Win32 console project called *OSG shadows*
- You will need to set the project file for open scene graph. Remember to do this each time you make a new osg project.
    - under *C++> General* you will need to add the include directory for osg
    - under *Linker>General* you will need to add the library directory for osg
    - under *Linker > Input* you will need to add the osgd.lib, osgviewerd.lib, osgdbd.lib, osgUtild.lib, osgGAd.lib, osgShadowd.lib

Create geometry and display in the viewer

- Copy and paste the code from your OSG intro project file (it is available on Moodle as OSG Intro.cpp) (Remember you will need to set up your models folder in your project folder)

- You will need to include additional header files for the shadows

```
#include "stdafx.h"

#include <osgViewer/Viewer>
#include <osgDB/ReadFile>
#include <osg/Node>
#include <osg/ref_ptr>

#include <osg/Geode>
#include <osg/ShapeDrawable>
#include <osg/Material>
#include <osg/Texture2D>
#include <osg/PositionAttitudeTransform>

#include <osgShadow/ShadowedScene>
#include <osgShadow/ShadowMap>
```

- You should run the project to test it before going any further.

Set up the scene to display and receive shadows

Like we did in D-flow, you need to specify which objects in your scene are going to cast and receive shadows – you should limit this to only the geometry which requires dynamic changes. Other lighting and shadows should be baked into your textures.

When using shadows in OSG we need to set up a shadowed scene and then add the shadowed / shadow objects to this scene. So we will need to remove the relevant objects from our root node.

Comment out (or remove) the lines adding the apple, box and ground to the root:

```
 //myRoot->addChild(groundNode.get()); // attach the ground to the root node
..
..
//myRoot->addChild(boxXform.get()); // adds it all to the root node
..
..
//     myRoot->addChild(appleXform.get()); // add the transform node to the root
..
..
```

- Add this code after the code to add the apple to the scene to set up the scene for shadows

```
appleXform->addChild(appleNode.get()); // add the apple to the transform node


    //****************************** set up the scene for casting and receiving
shadows *************************//

const int ReceivesShadowTraversalMask = 0x1;

const int CastsShadowTraversalMask = 0x2;

osg::ref_ptr<osgShadow::ShadowedScene> shadowedScene = new osgShadow::ShadowedScene;

shadowedScene->setReceivesShadowTraversalMask(ReceivesShadowTraversalMask);
shadowedScene->setCastsShadowTraversalMask(CastsShadowTraversalMask);
osg::ref_ptr<osgShadow::ShadowMap> sm = new osgShadow::ShadowMap;
shadowedScene->setShadowTechnique(sm.get());

myRoot->addChild(shadowedScene); // add the shadowed scene to our root (the objects
need to be added to the shadowed scene, NOT the root node)
```

Next we need to add code to define which nodes will cast shadows and which will receive shadows. I want the ground to receive shadows and the box and apple to cast them:

- Add the following code below the code you have just written

```
  myRoot->addChild(shadowedScene); // add the shadowed scene to our root (the
objects need to be added to the shadowed scene, NOT the root node)
```

```
    // *******************define the nodes to cast shadows
****************************************//

    appleXform->setNodeMask(CastsShadowTraversalMask);
    boxXform->setNodeMask(CastsShadowTraversalMask);

     // *******************define the node to receive shadows
****************************************//

    groundNode->setNodeMask(ReceivesShadowTraversalMask);

    //**************** add the shadowed / shadow nodes to the shadowed scene

     shadowedScene->addChild(appleXform);
      shadowedScene->addChild(boxXform);
     shadowedScene->addChild(groundNode);

   //*******set up the viewer to display your scene****************//

   osgViewer::Viewer viewer; // Declare a 'viewer' which will display the scene
```

- Now test your project. You should see shadows cast by the box and the apple onto the ground

You can now create basic shadows. By moving your objects and lights you can change how the shadows appear in the scene.

You should look at the OSG reference documents for more information on setting shadow characteristics.