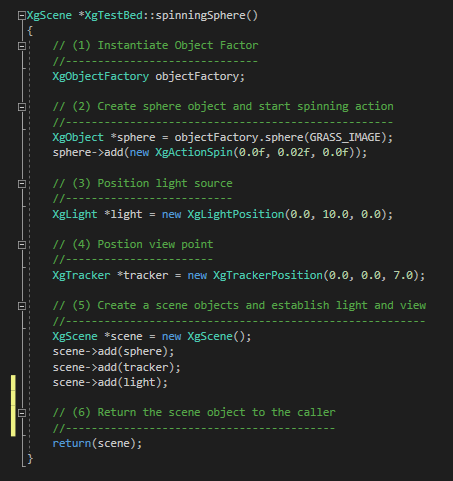
**Creating a Basic 2D Animation – Idle**

To illustrate the very basics of the Xg Engine we will use the API to create a spinning sphere. This is similar to the traditional “Hello World” triangle as seen in most OpenGl tutorials. In this example, a textured spinning sphere is placed in the middle of the animation, with a point light source hovering above. The position of the camera is placed on the same plane as the origin of the sphere. The light source and the camera are set at a fixed position. In future examples we will see how the light source and camera can be animated.

The creation of the spinning sphere can be accomplished in the six steps outlined in the sample code.

1. Is the instantiation of the **XgObjectFactory** object factory. The object factory provides a simple means of creating basic 3d shapes.

2. Is the creation of the **XgObject** sphere. This object is created by the object factory. A pathname is provided as the grass texture displayed. The add() function comes with two options. In this option, a “spin” action is added to the sphere. Parameters in the constructor allow for control of the spin direction and speed. In this case, the spin is applied to the Y-axis. Note, this spin action can be applied to any object.

3. Is the creation of the light source. The **XgLightPosition** creates a point light source. Later we will look at means of creating multi-lights, directional lights and spotlights. This light has no animation associated with it, since it resides at a fixed position.

4. Is the create of the camera. The **XgTrackerPosition** creates a camera position at a fixed point. The position of the camera is defined by the x, y, z coordinates in the constructor. There are other constructors that create arena, walk-through and fly-through cameras.

5. Is the creation of the scene. The **XgScene** object is created and each of the assets are added. The scene object will manage the animation of the assets. There is no predefined order that the assets must be added. If an asset is not added to the scene, it will not be animated.

6. Return the scene object.