

Definir elementos de la escena mediante una clase Obj

- ❑ El diseño de las clases que se ve ahora se refactoriza más adelante con la introducción de la superclase **EntityIG**
- ❑ Las clases que defináis heredan de **OgreBites::InputListener** de forma que sus objetos puedan responder a eventos
- ❑ La definición de las clases se hace usando el espacio de nombres:

using namespace Ogre;

- ❑ Las clases tienen un atributo **SceneNode* mNode** que permite asociar el elemento que va a la escena con este nodo e introducirlo así en el grafo de la escena para que forme parte de ella
- ❑ Las clases suelen tener un atributo **SceneManager* mSM** para poder crear entidades
- ❑ Los elementos que se construyan con una clase dentro del **setupScene()** se destruyen en **shutdown()**

La clase Obj sin heredar de EntityIG

```
❑ class Obj : public OgreBites::InputListener { // #include <OgreInput.h>
    public:
        Objt(Ogre::SceneNode* node); // #include <OgreSceneNode.h>
        ~Objt();

        // Métodos de InputListener que pueden redefinirse
        virtual void frameRendered(const Ogre::FrameEvent & evt);
        virtual bool keyPressed(const OgreBites::KeyboardEvent& evt);
        virtual bool keyReleased(const OgreBites::KeyboardEvent& evt);
        virtual bool mousePressed(const OgreBites::MouseButtonEvent& evt);
        virtual bool mouseRelease(const OgreBites::MouseButtonEvent& evt);
        virtual bool mouseMoved(const OgreBites::MouseMotionEvent& evt);
        virtual bool mouseWheelRolled(const OgreBites::MouseWheelEvent& evt);

        ...
    }
```

La clase Obj sin heredar de EntityIG

❑ Constructora

```
class Obj : public OgreBites::InputListener {  
    protected:  
  
        Ogre::SceneNode* mNode;  
        Ogre::SceneManager* mSM; //No siempre necesario  
        // #include <OgreSceneManager.h>  
        // #include <OgreEntity.h>  
  
        ...  
}  
Obj::Obj(Ogre::SceneNode* node) {  
    //La constructora pasa el nodo asociado al objeto  
    //creado, como parámetro  
    mNode = node;  
    Ogre::SceneManager* mSM = mNode->getCreator();  
  
    ...  
}
```

Grafo de la escena

InputListener de la clase Obj

❑ **virtual bool keyPressed(const OgreBites::KeyboardEvent& evt);**

```
bool Obj::keyPressed(const OgreBites::KeyboardEvent& evt) {  
    if (evt.keysym.sym == SDLK_??) // #include <SDL_keycode.h>  
    {  
        ....  
    }  
    return true;  
}
```

```
void Obj::frameRendered(const Ogre::FrameEvent & evt) {  
    // evt.timeSinceLastFrame -> en segundos (float/double)  
    // evt.timeSinceLastEvent  -> en segundos (float/double)
```

...

}

Uso de la clase Obj en IG2App

```
void IG2App::setupScene(void) {  
    ...  
    nodoObjeto = ...;  
    //Por ejemplo, nodoObjeto =  
    //      mSM->getRootSceneNode()->createChildSceneNode("...");  
    //      nodoPadre->createChildSceneNode("...");  
    obj1 = new Obj(nodoObjeto);  
    addInputListener(obj1);  
    ...  
}  
void IG2App::shutdown() { ...  
    delete obj1;  
}
```