Proyecto-esqueleto OGRE3D

- **main**

```
#include "IG2App.h"

int main(int argc, char *argv[])  {
    IG2App app;
    try {
       app.initApp();
       app.getRoot()->startRendering();
    }
    catch (Ogre::Exception& e)   {
        Ogre::LogManager::getSingleton().logMessage("An exception has
                                        occured: " + e.getFullDescription() + "\n");
    }
    app.closeApp();
    return 0;
}
```

Ogre initialization

main

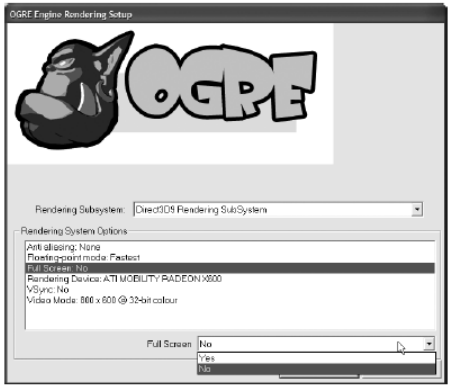
```
#include "IG2App.h"

int main(int argc, char *argv[])
{
    IG2App app;
    app.initApp();

    app.getRoot()->startRendering();
    app.closeApp();
    return 0;
}
```

```
IG2ApplicationContext::initApp()
    mRoot = new Root("...plugins.cfg",
                     "...ogre.cfg", "...ogre.log");
    mOverlaySystem = new OverlaySystem();
    setup() ->
        IG2ApplicationContext::setup();
        ...
        setupScene()
```

Ogre initialization

IG2ApplicationContext::setup()

```
mRoot->showConfigDialog(...);
mRoot->initialise(false);  // autoCreateWindow
createWindow();
initialiseResources();
initialiseRTShaderSystem();
mRoot->addFrameListener(this);
```



IG2App:: setup()

```
IG2ApplicationContext::setup();
addInputListener(this);
mSM = mRoot->createSceneManager();
mShaderGenerator->addSceneManager(mSM);
mSM->addRenderQueueListener(mOverlaySystem);
mTrayMgr = new TrayManager("TrayGUISystem", mWindow);
mTrayMgr->showFrameStats(OgreBites::TL_BOTTOMLEFT);
addInputListener(mTrayMgr);
setupScene();        // ->
```

Application skeleton

- **IG2App**

```
#include "IG2ApplicationContext.h" // Adaptación de OgreBites::ApplicationContext
...
class IG2App : public  OgreBites::IG2ApplicationContext,
               public  OgreBites::InputListener  // observador de
                                                 // IG2ApplicationContext
{
public:
  explicit IG2App() : IG2ApplicationContext("IG2App") { };
  virtual ~IG2App(){ };
protected:
  virtual void setup();
  virtual void shutdown();
  virtual void setupScene();
  virtual bool keyPressed(const OgreBites::KeyboardEvent& evt);  // InputListener
  Ogre::SceneManager* mSM = nullptr;
  OgreBites::TrayManager* mTrayMgr = nullptr;
...
};
```

IG2ApplicationContext

Hereda de **FrameListener** (observador de root) y mantiene una lista de observadores de eventos de entrada (**InputListener ***)

```
class IG2ApplicationContext : public FrameListener  {
   public:
   virtual void createRoot();
   virtual bool frameStarted(const Ogre::FrameEvent& evt) {
                                          pollEvents(); return true; }
   void pollEvents();   // avisa a los observadores
   protected:
   Ogre::Root* mRoot;
   NativeWindowPair mWindow;
   Ogre::FileSystemLayer* mFSLayer;
   Ogre::OverlaySystem* mOverlaySystem;
   Ogre::RTShader::ShaderGenerator * mShaderGenerator;
   ...
   std::set<OgreBites::InputListener*> mInputListeners;
};
```