Minimal Code

- Root
- RenderWindow
- SceneManager
- Camera
- Viewport
- Game(Render) Loop:
  - Check for event
  - Update render date
  - Draw scene
- LowLevelOgre.h
- LowLevelOgre.cpp
Root is the CEO. It directs other manager classes.

Resource Manager
- Mesh
- Skeleton
- Material
- GPU program
- Texture
- Compositeor
- Font
Camera

- projector
- image plane
- projection of \( p \)
- center of projection
Camera Position, LookAt

Scene Manager, Render Manager

- Scene Manager determines what to render
- Scene Nodes are part of queues within queues.
- Render System (opengl, directx) constructs the frames from the Scene Nodes that are visible relative to the camera.
Simplified Code

- TinyOgre.h
- TinyOgre.cpp
- MinimalOgre.h
- MinimalOgre.cpp
- Ogre Tutorial Framework
  - BaseApplication.h
  - BaseApplication.cpp
  - TutorialApplication.h
  - TutorialApplication.cpp
FrameListener: Called each game loop. This is where one processes events, updates animations, interacts with physic engine.

BaseApplication.h
BaseApplication.cpp
TutorialApplication.h
TutorialApplication.cpp
It is important to be able to trace the execution of Ogre and to print out messages if something goes wrong. Ogre provides an execution log to help with that.

Look for Ogre.log in the directory containing the executable.

Contains a lot of information so flag you output with a more or less unique string, i.e. “------------------------”.

LogManager::getSingleton().logMessage(Ogre::LML_NORMAL,”-----------------------------”);

See Example.