OGRE 分析之文件系统（一）

Mythma
http://www.cppblog.com/mythma
Email: mythma@163.com

OGRE 既可以读取普通的文件，又可以读取 Zip 压缩文件。其文件系统可以分目录管理和文件管理两大部分。从应用层次上，OGRE 还提供了配置文件管理、日志文件管理和资源文件管理等。假如再加上 mesh 文件、字体文件、纹理文件等，OGRE 的文件系统可谓十分庞大。配置文件比较常用，因此先从配置文件说起。

一、配置文件

1、配置文件

配置文件为以 key/value 形式保存的文件。利用配置文件可以保存各种自定义数据，如资源文件目录和文件，插件目录和插件文件，以及其他用户数据。

2、Ogre 的 Demo 中有如下几个配置文件：

ogre.cfg：保存 Render System 的设置
Plugins.cfg：存放插件目录及插件名称
resources.cfg：资源文件的位置
terrain.cfg：地形配置文件
media.cfg：媒体文件（如 mesh，图片等）
quake3settings.cfg：例子 Demo_BSP.exe 用到场景配置文件

3、一个有效的 Ogre 配置文件（key/value）有如下几点要求：

1、一个 key 只能对应一个个 value
2、key 与 value 之间的分界符可以是：Tab、冒号(:)、等号(=)
3、设置可以分成段的形式，段名放在中括号内：[SectionName]
4、value 后必须要有回车符
5、可以有注释：用#或@开头的行

4、**Ogre::ConfigFile**

为快速的从文件中加载设置，Ogre 提供了 ConfigFile 类。它用到了迭代器（Iterator）设计模式，有段迭代器和设置项迭代器， 分别用于对段（Section）和设置项（Settings）进行迭代。

注：Ogre 中有一个 Iterator 模板 Ogre::MapIterator< T >。

ConfigFile 中的 settings 都存放于某个 section 下。每一个 ConfigFile 实例都有一个段名为””的段（匿名段）。因此，若配置文件中的配置没有指定段名，设置将存放在该段下。

从如下代码中，可以看出 section 和 settings 之间关系。

```
typedef std::multimap< String,String > SettingsMultiMap
typedef MapIterator< SettingsMultiMap > SettingsIterator
typedef std::map< String,SettingsMultiMap * > SettingsBySection
typedef MapIterator< SettingsBySection > SectionIterator
```

通过如下代码可以分析出如何利用 ConfigFile 来读取配置信息：

```
ConfigFile cf;
cf.load("resources.cfg");
// Go through all sections & settings in the file
ConfigFile::SectionIterator seci = cf.getSectionIterator();
String secName, typeName, archName;
while (seci.hasMoreElements())
{
  secName = seci.peekNextKey();
  ConfigFile::SettingsMultiMap *settings = seci.getNext();
  ConfigFile::SettingsMultiMap::iterator i;
  for (i = settings->begin(); i != settings->end(); ++i)
  {
    typeName = i->first;
    archName = i->second;
    ResourceGroupManager::getSingleton().addResourceLocation(
      archName, typeName, secName);
  }
}
```

ConfigFile 本身并没有直接读取文件，而是通过 Ogre 封装的一套文件操作类来读取数据。