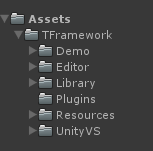
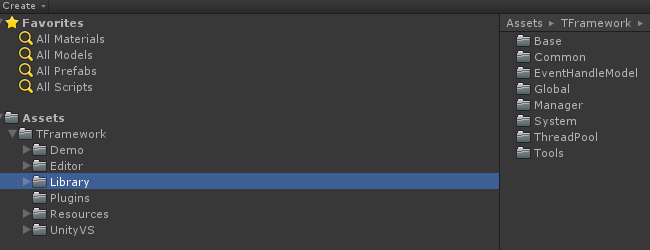
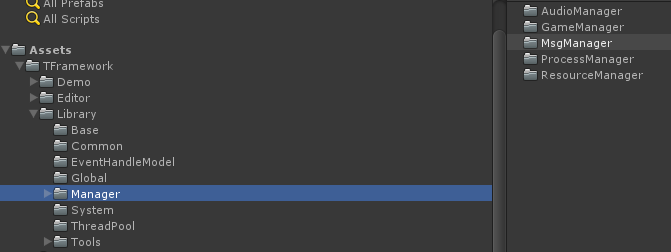
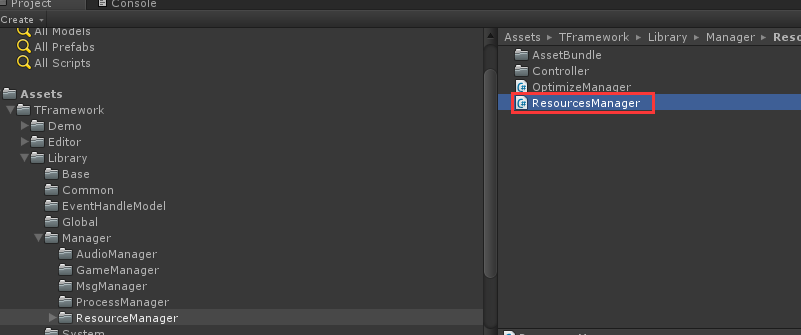
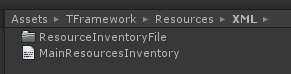
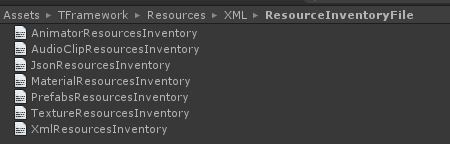
# Tframework框架文档

1. 文件结构
   1. 
      1. Demo – 框架功能样例
      2. Editor – 框架的UnityEditor功能扩展
      3. Library – 框架基础源码
      4. Plugins – 框架插件库
      5. Resources – 框架资源文件
      6. UnityVS – Unity的VS联调插件
   2. 
      1. Base – 基础类
      2. Common – 接口
      3. EventHandleModel – 事件模块
      4. Global – 通用类
      5. Manager – 管理类
      6. System – 系统类
      7. ThreadPool – 线程池
      8. Tools – 工具类
   3. 
      1. AudioManager – 音频管理
      2. GameManager – 游戏机制管理
      3. MsgManager – 消息机制管理
      4. ProcessManager – 流程管理
      5. ResourceManager – 资源管理
2. 框架使用流程
   1. MainConrol -- 框架有整体的程序入口
      1. 继承MainConrol,次类中的Awake()一般不做修改，会初始化框架必须的功能模块。
      2. 重写IniMainModules和IniMarginalModules调用base，然后在后面添加需要的其它模块并进行初始化。
      3. 新建一个空场景，在场景中创建空物体命名为对应程序入口类的名字，例如:现在直接用MainControl，则空物体名字为MainControl；将继承完成的新类挂载空物体上，保存场景。这个场镜就是程序的初始场景。
3. 模块使用介绍
   1. ResourcesManager – 资源管理器
      1. 
      2. 资源管理器支持网络下载资源和本地加载资源
      3. 在框架资源文件夹有对应的配置文件
         1. 
            1. 主配置文件为MainResourcesInventory – 用来加载其它的配置文件
         2. 
            1. 子配置文件中进行对应类型文件的配置，配置完成，即可使用框架资源管理器进行加载。
      4. 流程管理器 – 未完成
         1. 整个游戏的流程管理，大致有两种，单一场景的和多场景切换的。
         2. 在使用单一场景时，推荐使用流程管理器进行开发，可保证流程的顺序性。
         3. 多场景切换，使用对应的场景切换逻辑就可以实现流程的顺序性，无需使用流程管理器。