Virtuos - Development

**Technical Design Document**

(Version 2)

Table of Contents

[1. Overview 3](#_Toc260387709)

[2. Requirements 3](#_Toc260387710)

[2.1. Brief 3](#_Toc260387711)

[2.2. Reference 3](#_Toc260387712)

[3. Implementation details 4](#_Toc260387717)

[4. Risks 6](#_Toc260387720)

[5. Estimates 6](#_Toc260387721)

# Overview

AssetManager需要管理关卡文件和类型的默认资源文件，关卡文件中存储了关卡一开始时存在的Actor，类型的默认资源文件管理每个类型的场景对象需要的预设资源，比如模型、纹理和材质等，不包含变换，因为该文件管理的是关卡中动态加载的资源的一个模板。这两种文件中存储的信息使用json字符串存储。通过AssetManager可以实现数据的解析和存储，并且可以生成设定的默认类对象。

# Requirements

## Brief

**Json字符串解析与存储：**

* 资源由json字符串存储

**存档：**

* 支持多个存档槽位，存档会存在本地xml
* Editor模式下编辑内容保存到关卡存档，并在PIE开始时快速存档保存当前状态以恢复
* Game模式下存快速存档到存档槽中

**读档：**

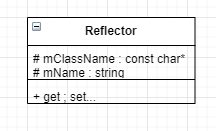
* Game模式下玩家可以选择在游戏进行中或主菜单中即时读取
* Editor模式下结束PIE时读取快速存档恢复编辑状态

## Reference

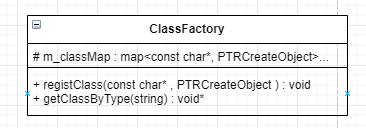
* 无

# Implementation details

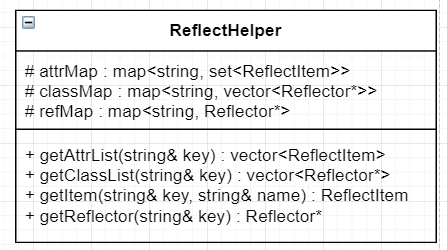
<**Reflection**>



所有需要加入反射的类，如Object、Component、UIControl类都需要继承自Reflector类集中管理。

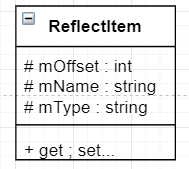


ClassFactory是生成新类的工厂，新建Reflector的派生类时，将此类的生成函数和类名利用宏进行包装，在Engine里手动注册。New新类时可以根据类名的string生成新此类实例。

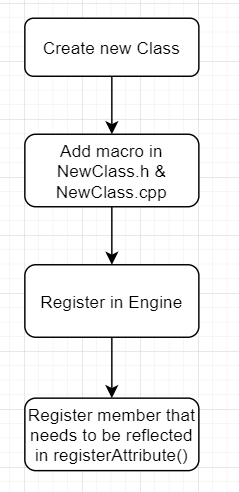


ReflectHelper保存反射类的其他信息，如类成员。类的所有需要序列化的属性，加入到Registe Attribute方法注册到ReflectHelper中。

AttrMap存储类的实例与其属性的映射；ClassMap存储类名与类实例的映射；RefMap存储Reflector名称及其实例指针的映射。

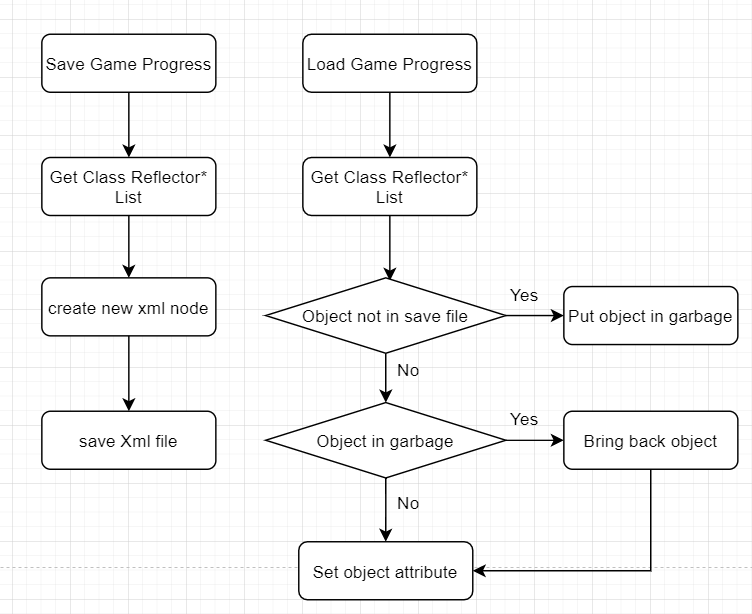


ReflectItem对应类加入反射的成员属性，通过字节偏移量找到成员属性的值，可实时获取并修改。根据成员属性的类型有对应的get与set方法。



反射注册流程

<**Serializer**>



游戏快速存档的序列化/反序列化

游戏关卡存档与快速存档流程相近，区别在于快速存档是选取部分类保存，读档时set调整属性值；关卡存档是整个关卡所有内容保存，读档时为重新生成。

引入垃圾桶，减少对象的重复new与delete。

# Risks

* 反射系统加大了内存的占用，且当删除一个物体时反射没有保持一致性，存在野指针，否则会导致效率大幅下降。考虑移除ClassMap转而用reflector遍历。

# Estimates

|  |  |
| --- | --- |
| Tasks | Estimate in days/person |
| Research and design | 2 |
| Implement Reflection | 3 |
| Implement Save & Load | 3 |
| Implement Input | 2 |
| Total | 10 |