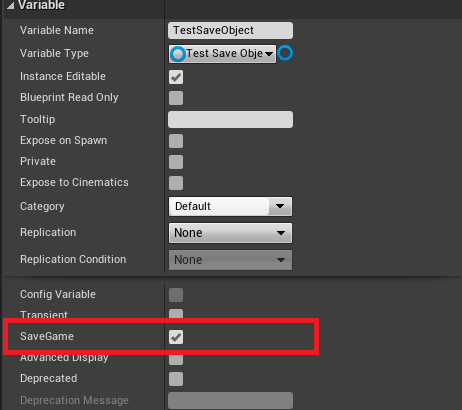
**Save Game使用文档**

**作者：刘连友**

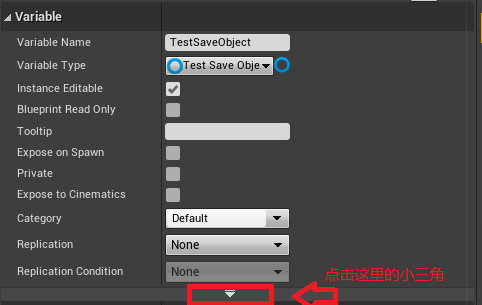
**时间：2021/4/26**

**版本：1.0**

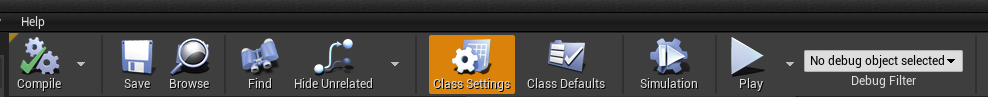
1. **蓝图里使用**
2. 先将需要保存信息的蓝图类里的成员变量的SaveGame 勾选，比如下面的：



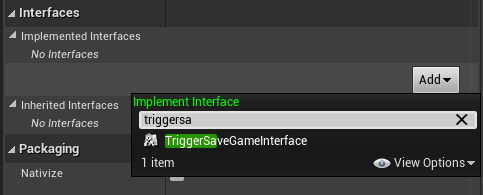
这个界面就是在选中变量的属性矿体里面，如果看不到，请点击下拉三角：



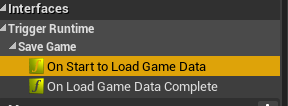
1. 然后点击ClassSetting按钮：



1. 此时Details Panel 会显示Interface，点击Add按钮，然后添加TriggerSaveGameInterface



1. 添加完之后，在MyBlueprint会有新的可实现的函数：

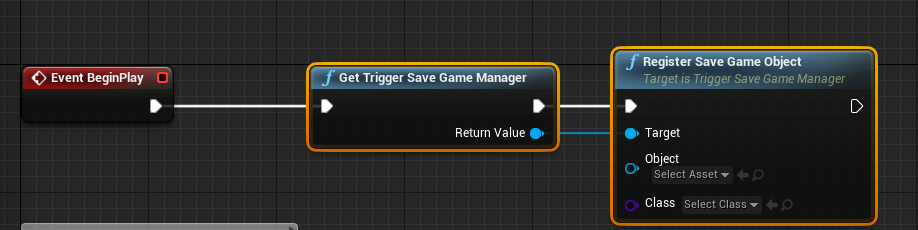


这两个函数的作用的调用时间是：

OnStartToLoadGameData: 当开始读取这个类对象里面的所有的勾选SaveGame的数据。

OnLoadGameDataComplete: 这个对象的所有的勾选SaveGame的数据已经读完档

1. 然后在某一个地方调用GetTriggerSaveGameManager->RegisterSaveGameObject



在这里我是在Beginplay里面做的，其实这个是可以在其他的任何地方，调用的。

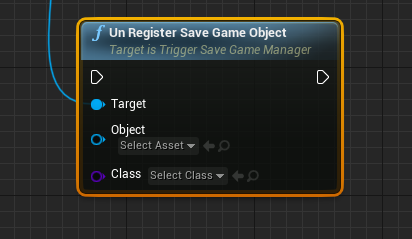
在节点RegisterSaveGameObject里面有3个参数：

Target: 是TriggerSaveGameManager

Object: 是要注册的Object,如果是上图的话，那么这个链接的是Self 节点。

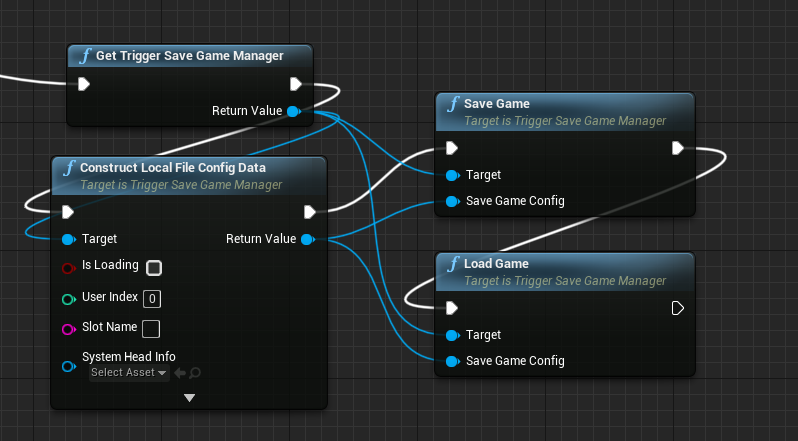
Class: 是要注册到哪一个SaveGame的策略，这里提到的策略是指，可能在一个游戏中会有很多种存储的方式，每一个存储的方式都是一个策略。目前只有SGS\_SaveGameFlag 这一种策略，(SGS是SaveGameSystem 的缩写)。

当然对应的还有一个函数节点：



这个是将对应的Object从指定的策略中去掉。

1. 配置完成。
2. 如果需要调用SaveGame或者LoadGame的话，可以使用节点：LoadGame/SaveGame



在LoadGame/SaveGame都有一个参数：SaveGameConfig。

目前只有保存到本地的.sav 文件，所以只有一个函数去构造ConstructLocalFileConfigData.

这个函数里面有5个可改的参数：

IsLoading: 用来确定这个config是否用来载入数据的。

UserIndex, SlotName:这个和UE4自带的SaveGameToSlot是一样的。

SystemHeadInfo: 这个是用来保存SaveGame的数据的头信息，可以是空的，如果是空的那么就用当前最新的头信息

Class： 这个是用来指定当前应该用哪种策略来Save/Load

1. 补充说明：上面所有用的UObject的都可以通过在资源里继承，来进行提前配置的。
2. **代码里使用**
3. 在所有的需要使用SaveGameSystem来存储信息的类继承ITriggerSaveGameInterface.

请包含头文件：

#include "SaveGame/TriggerSaveGameInterface.h"

1. 在类里面实现两个函数（如果不关心，可以不是实现的）：

StartToLoadGameData, LoadGameDataComplete**。**

一般这两个函数是用来自定义如何根据存储的数据恢复现场的。请注意这里恢复现场如果使用了另外一个Object的信息，而那个信息也是保存起来的，这个时候那个对象的信息是不可靠的，这里只能保证本地的数据是可靠的。针对这种情况可以继续阅读，下面会有此类解决方案的

1. 在这个对象的需要保存的信息成员变量的UPRPERTY(SaveGame)
2. 将要保存的对象在莫一个地方进行注册，在注册的地方需要包含头文件：

#include "SaveGame/TriggerSaveGameManager.h"

#include "TriggerBlueprintLib.h"

然后调用：

UTriggerBlueprintLib::GetTriggerSaveGameManager()->RegisterSaveGameObject(**XXObject**, USGS\_SaveGameFlag::StaticClass());

XXObject：是要保存的对象

Class: 因为目前只有USGS\_SaveGamFlag可用，所以这里显示的是这个的

1. 如果需要Save/Load的话，可以调用

UTriggerBlueprintLib::GetTriggerSaveGameManager()->LoadGame(UTriggerBlueprintLib::GetTriggerSaveGameManager()->ConstructLocalFileConfigData(...));

1. 有一些事件是可以注册使用的，这些在

#include "SaveGame/TriggerSaveGameDefinition.h"

这里面的LoadGameComplete调用的时候，所有的Object都是序列化好的，因此这个时候所有的信息都是可靠的