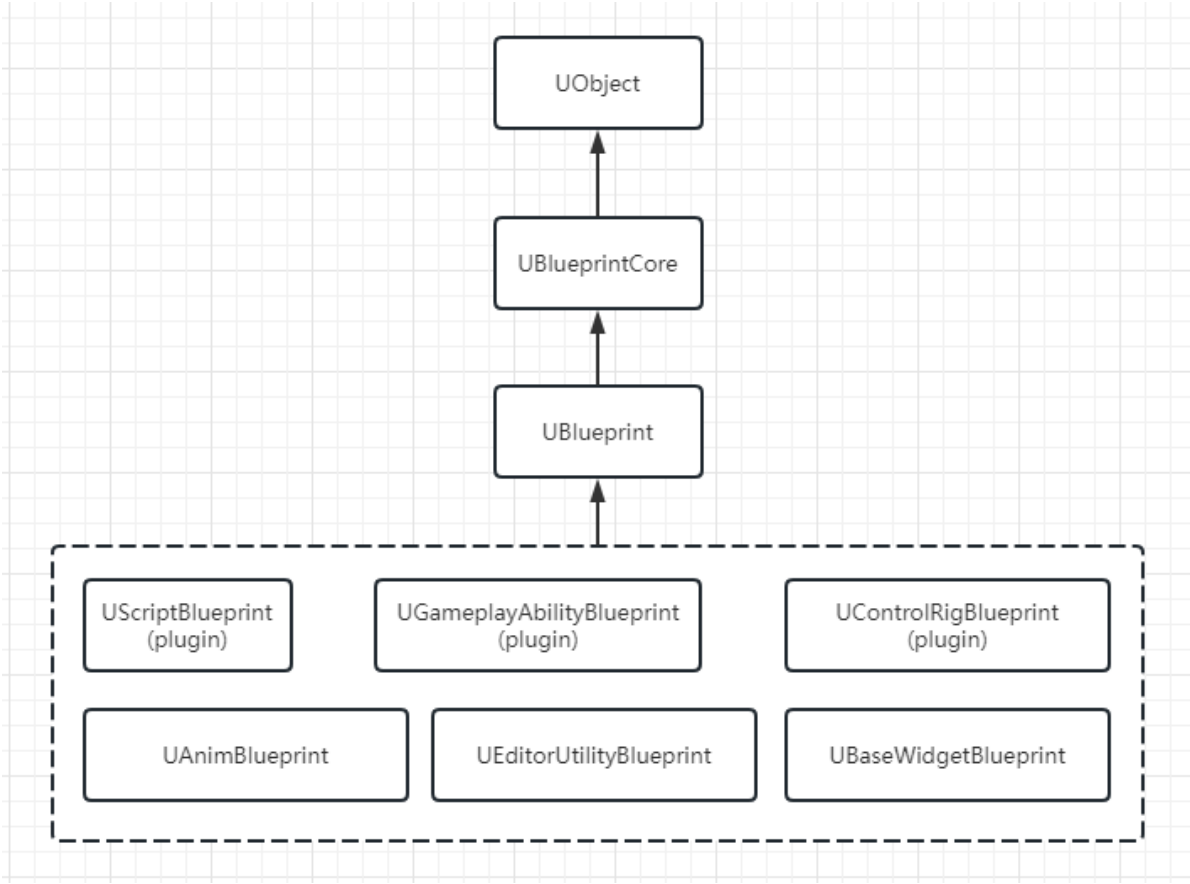
参考文章：

https://zhuanlan.zhihu.com/p/69067129

目录

一、蓝图类结构

下图为蓝图类的继承关系：



1. UBlueprintCore

该类中有三个重要的类变量：

```
TSubclassOf<UObject> SkeletonGenerateClass;
TSubclassOf<UObject> GeneratedClass;
FGuid BlueprintGuid;
```

TSubclassOf SkeletonGeneratedClass：指向skeleton class的指针，每当向蓝图中添加一个变量或者函数时，都会重新生成skeleton class。但是skeleton class只是一个框架类，比如它不包含字节码和自动生成的内部变量。在编译蓝图时，可以设置CompileType为SkeletonOnly来只生成skeleton class。

TSubclassOf GeneratedClass：指向全编译class的指针，通常使用的都是这个class，其内部的属性都被填充，可以完整的描述一个蓝图。由于蓝图类class是就地编译的，因此不管编译几次，这个指针都不需要改变。

FGuid BlueprintGuid：蓝图的Guid，根据蓝图的资源路径hash生成。

2. UBlueprint

负责实现蓝图的主要功能，包括蓝图属性、蓝图函数、节点连接关系等。

主要成员变量：

TSubclassOf ParentClass：指向蓝图类父类的UClass。会注册到AssetRegistryTags中，以支持蓝图预览。

TArray<UEdGraph*> UbergraphPages：所有ubergraphpages，一个ubergraph就是编辑器左侧显示的一个eventgraph，里面会包含所有event的实现，在蓝图编译时，它们会被复制到一个大的ubergraph中。

TArray<UEdGraph*> FunctionGraphs：所有函数graph，ConstructionScript也是一个函数。

TArray<UEdGraph*> DelegateSignatureGraphs：所有事件代理Graph。

TArray<UEdGraph*> MacroGraphs：所有这个蓝图实现的宏。

TArray<UEdGraph*> EventGraphs：所有的Event声明。

TArray<class UTimelineTemplate*> Timelines：蓝图中的Timeline，timeline需要被特别处理。

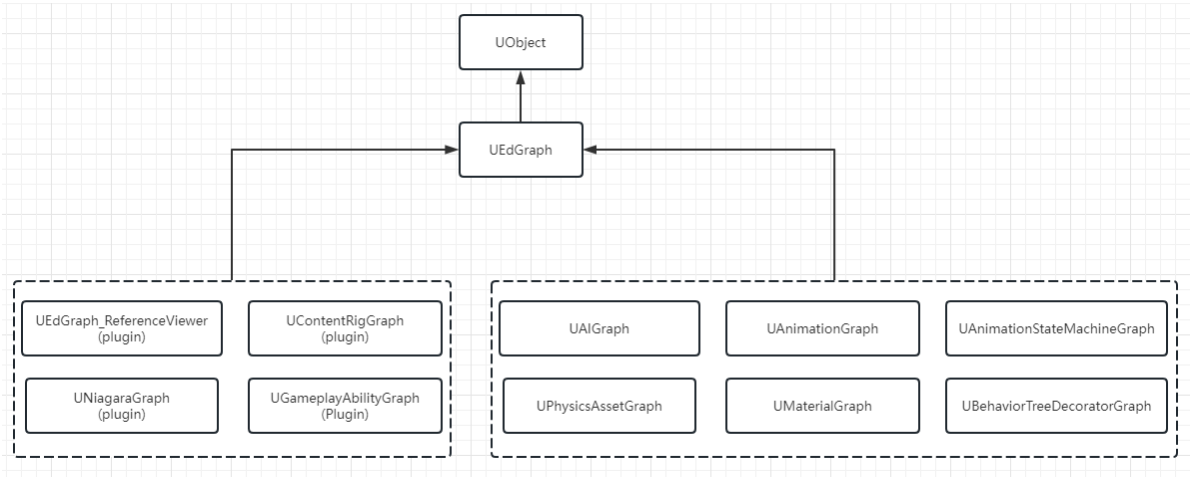
TArray NewVariables：这个蓝图自己创建的变量，不包含从父类继承的，在编译时会添加到BlueprintGeneratedClass中。

TArray<class UBreakpoint*> Breakpoints：断点集合，用于调试

其实，蓝图编译过程就是把UBlueprint描述的信息转换为BlueprintGeneratedClass的过程。

3. UEdGraph

图的基类是UEdGraph，它有许多派生类，比如UAIGraph、UAnimationGraph等，用于实现不同模块的功能。图的继承关系如下：



UEdGraph

主要成员变量：

TSubclassOf Schema：这个图要遵循的schema。

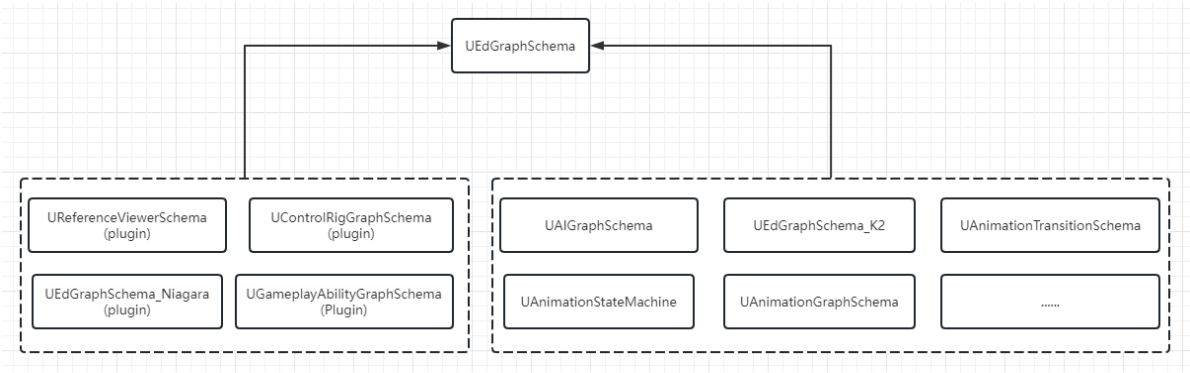
TArray<class UEdGraphNode*> Nodes：图中的所有节点。

TArray<class UEdGraph*> SubGraphs：这个图包含的子图，一个例子就是动画蓝图的状态机，状态机就作为子图存在。

UEdGraph可以理解为节点数据的容器。

模式（UEdGraphSchema）

模式是与Graph对应的，约定了当前Graph能创建什么样的节点、两个引脚能否产生连接等等。类的继承关系如下：



主要方法：

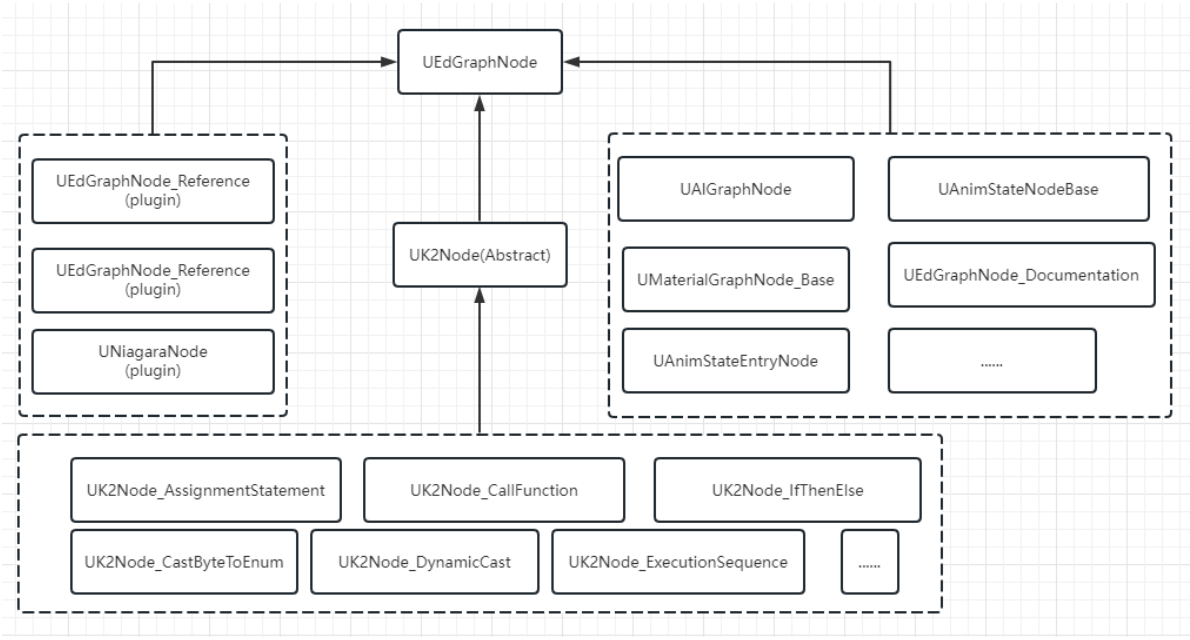
GetGraphContextActions()：在当前蓝图的空白处单击右键出现的节点菜单栏，列出了当前Graph能添加的所有节点

GetContextMenuActions()：在节点或引脚上右键出现的操作菜单栏，比如deletelink、break等

CanCreateConnection()：传入两个引脚，判断两个引脚间能否建立连线。返回的并不是一个boo值，而是枚举ECanCreateConnectionResponse，包含了更多信息。

举个例子，CONNECT_RESPONSE_BREAK_OTHERS_A，断开引脚A上的其他连接，然后创建AB间的连接，连接两个执行引脚就是这样的。

4. UEdGraphNode



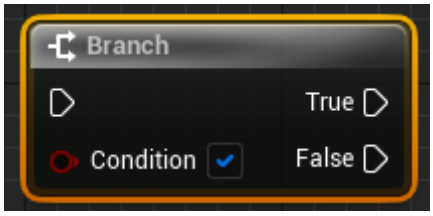
UEdGraphNode

主要成员变量：

TArray<UEdGraphPin*> Pins：节点上的所有引脚。

int32 NodePosX,NodePosY,NodeWidth,NodeHeight：节点的位置，长宽，在展开时会用到，避免节点间产生重叠。

可以看到，UEdGraphNode只是提供了一个基本的节点描述功能，不同类型节点的逻辑都需要由对应的子类进行约定。比如一个IfThenElse节点：

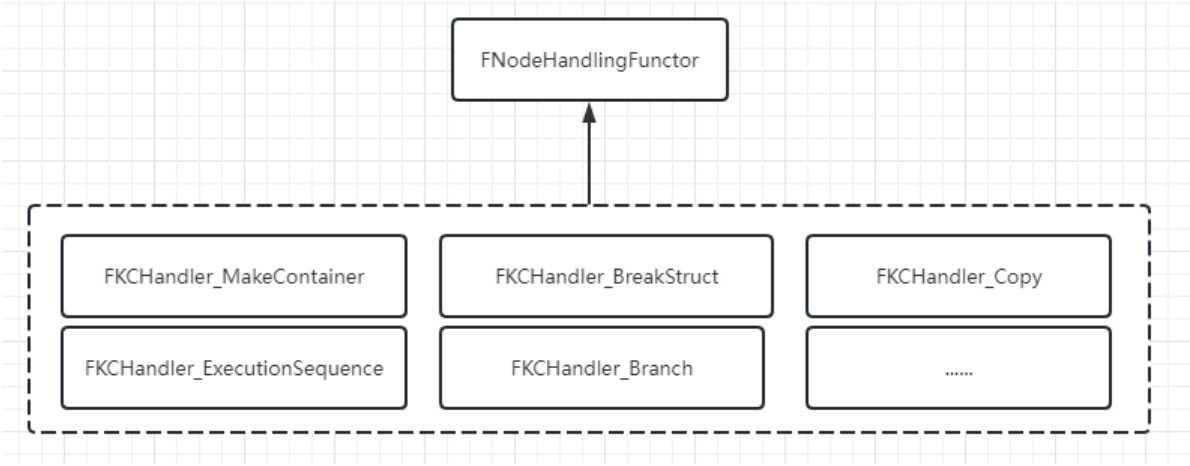


执行input节点，数据input节点，两个输出节点都由UK2Node_IfThenElse创建，数据input节点的默认值也由UK2Node_IfThenElse设置，这些都可以理解为一个节点的外在表现。

而一个节点的内在含义，或者说逻辑，由FNodeHandlingFunctor类实现。比如IfThenElse节点，Condtion为true时走then输出引脚，为false时走else输出引脚，这个逻辑就由继承FNodeHandlingFunctor的FKCHandler_Branch实现。

FNodeHandlingFunctor

是处理一类节点编译的帮助类，类图如下：



主要方法：

RegisterNets()：用于注册引脚和创建Terminal。例如，K2Node_Self会把节点的self pin与一个Terminal对应起来，FKCHandler_Switch会创建一个中间BoolTerm用于存储输入变量与各个值的比较结果。

Compile()：用于在编译时生成FBlueprintCompiledStatement，这可以理解为抽象语法树的节点，在下文中会有介绍。

RegisterLiteral（）：用于注册字面Terminal。

二、蓝图虚拟机