

Unreal Blueprint 学习笔记

郑华

2018 年 5 月 25 日

目录

第一章 概述	7
1.1 蓝图简介	7
1.2 蓝图工作原理	7
1.3 蓝图与 C++	8
1.3.1 蓝图	8
1.3.2 C++	8
1.3.3 蓝图 Vs C++	8
1.4 参考	9
第二章 蓝图类型	11
2.1 关卡蓝图	11
2.2 类蓝图	11
2.3 仅包含数据的蓝图	11
2.4 蓝图接口	11
2.5 蓝图宏库	11
2.6 参考	11
第三章 蓝图变量与流程控制	13
3.1 变量类型	14
3.1.1 变量概述	14

3.1.2	基本类型	14
3.1.3	蓝图数组	14
3.2	流程控制	14
3.2.1	概述	14
3.2.2	开关节点	14
3.2.3	Branch	14
3.2.4	DoN	14
3.2.5	DoOnce	14
3.2.6	FlipFlop	14
3.2.7	ForLoop	14
3.2.8	ForLoopWithBreak	14
3.2.9	Gate	14
3.2.10	MultiGate	14
3.2.11	Sequunce	14
3.2.12	WhileLoop	14
3.3	函数	14
3.3.1	访问修饰符	14
3.3.2	纯函数和非纯函数	14
3.3.3	如何创建函数	14
3.3.4	编辑函数	15
3.3.5	调用函数	15
3.4	参考	15

第四章 蓝图与 C++ 交互 17

4.1	参考	17
-----	----	----

第一章 概述

1.1 蓝图简介

代码控制 转化为 -> 可视数据流

虚幻引擎中的蓝图可视化系统是一个完整的游戏脚本系统，其理念是使用基于节点的界面从虚幻编辑器中创建游戏可玩性元素，该系统非常灵活且非常强大，因为它为设计人员提供了一般仅供程序员使用的所有概念及工具。它是一种特殊类型的资源，为关卡设计师和游戏开发人员提供了一种在编辑器中快速创建及迭代游戏可玩性的工具。

通过使用蓝图，设计人员几乎可以创作任何游戏元素的原型，以及实现或修改这些元素。

- Games（游戏）创建游戏规则，调整游戏条件等。
- Players（玩家）使用不同的网格物体、材质或角色自定义来创建变种
- Cameras（相机）创建新相机视角的原型或者在游戏运行过程中动态地改变相机。
- Input（输入）修改玩家操作，或允许玩家向道具传入输入
- Items（道具）武器、法术、掉落物、触发器等。
- Environments（环境）创建随机的装置或者程序化地生成道具。

对于程序员来说，我们可以把它理解为一种可视化的高级语言（C# 等），它有基本的变量、函数、类型转换，支持继承、多态等

1.2 蓝图工作原理

从蓝图的基本形式上讲，蓝图是针对您游戏添加的**可视化脚本**。通过使用连线把节点、事件、函数及变量连接到一起，这样就可以创建复杂的游戏性元素。蓝图通过各种用途的节点构成图来进行工作，这些节点包括针对蓝图每个实例的对象构建、独立的函数、一般的游戏性事件，从而

实现各种行为及其它功能。

1.3 蓝图与 C++

在理论上，蓝图与 C++ 之间是可以相互调用的，如图1.1所示。

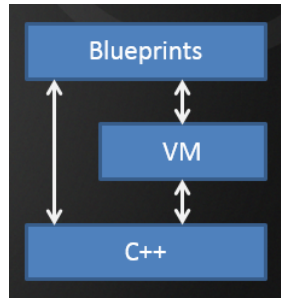


图 1.1: 结构图

1.3.1 蓝图

- 一种面向美术人员和设计师友好的可视化脚本系统
- 跟 UnrealScript 使用的相同的虚拟机
- 几乎跟 UnrealScript 一样强，在某些方面甚至更强

1.3.2 C++

- 一直是 UE 游戏编程中的一部分
- 跟虚拟机相互紧密地结合在一起
- 在 UE4 里面得到巨大的改进用来让编程人员替代 UnrealScript

1.3.3 蓝图 Vs C++

- 蓝图适合快速迭代
- 蓝图比原生 C++ 消耗更多的 CPU 性能
 - 运行在虚拟机上

- 大约比原生 C++ 代码慢 8 10 倍
 - 对于大多数简单的事件驱动的任务，你可能并不会发现性能有什么大的消耗
 - 许多发行的 AAA 游戏都在使用蓝图（包括 Epic 自己）
 - 每帧需要更新的（tick）应用使用原生 C++（物理模拟）
- 但是，蓝图仍然很合适做原型开发

总结

- 蓝图可以让设计师快速简单得实现想法
- 图状结构可以让程序员快速简单得用 C++ 代码实现并优化
- 代码可以在编辑器中添加到任意工程中
- 快速协作、畅通沟通，不会出现模糊不清的玩法设计。

个人理解，蓝图系统非常强大，可用于快速原型设计、简单事件驱动玩法的开发，但是如果玩法复杂需要耗费非常多的 CPU 性能，那么最好把设计师使用蓝图设计的玩法翻译成原生的 C++ 代码，如果本身一些像物理模拟等需要每帧更新又特别耗时的那么需要使用 C++ 来进行开发。

1.4 参考

蛮牛课程: <http://edu.manew.com/course/101/reviews>

<https://blog.csdn.net/kitok/article/details/80267273>

蓝图入门: https://www.cnblogs.com/ghl_carmack/p/5922131.html

UE4 笔记:http://www.cnblogs.com/ghl_carmack/category/816351.html

第二章 蓝图类型

2.1 关卡蓝图

2.2 类蓝图

2.3 仅包含数据的蓝图

2.4 蓝图接口

2.5 蓝图宏库

2.6 参考

UE4-2: http://www.cnblogs.com/ghl_carmack/p/5922190.html

第三章 蓝图变量与流程控制

3.1 变量类型

3.1.1 变量概述

3.1.2 基本类型

3.1.3 蓝图数组

3.2 流程控制

3.2.1 概述

3.2.2 开关节点

3.2.3 Branch

3.2.4 DoN

3.2.5 DoOnce

3.2.6 FlipFlop

3.2.7 ForLoop

3.2.8 ForLoopWithBreak

3.2.9 Gate

在蓝图接口中创建

3.3.4 编辑函数

3.3.5 调用函数

3.4 参考

UE4-3: http://www.cnblogs.com/ghl_carmack/p/5924965.html

第四章 蓝图与 C++ 交互

4.1 参考

UE4-4: http://www.cnblogs.com/ghl_carmack/p/5925405.html

第五章 玻璃红外门