

入象无形

虚幻引擎程序设计浅析

罗丁力 张三 著

内容简介

大象无形

虚幻引擎程序设计浅析

罗丁力 张三 著



电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以两位作者本人在使用虚幻引擎过程中的实际经历为参考，内容包括三大部分：虚幻引擎 C++ 编程、虚幻引擎浅析和扩展虚幻引擎。本书提供了不同于官方文档内容的虚幻引擎相关细节和有效实践。有助于读者一窥虚幻引擎本身设计的精妙之处，并能学习到定制虚幻引擎所需的基础知识，实现对其按需定制。

本书适合初步了解虚幻引擎编程，并希望学习虚幻引擎架构或者希望定制和扩展虚幻引擎的人士。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

图书在版编目 (CIP) 数据

大象无形：虚幻引擎程序设计浅析 / 罗丁力，张三著. —北京：电子工业出版社，2017.5
ISBN 978-7-121-31349-3

I. ①大…II. ①罗…②张…III. ①游戏程序—程序设计 IV. ①TP317.6

中国版本图书馆 CIP 数据核字 (2017) 第 076264 号

策划编辑：符隆美

责任编辑：徐津平

印 刷：三河市良远印务有限公司

装 订：三河市良远印务有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×980 1/16 印张：19.75 字数：380 千字

版 次：2017 年 5 月第 1 版

印 次：2017 年 5 月第 1 次印刷

定 价：65.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：(010) 51260888-819 faq@phei.com.cn。

前言

建德若偷，质真若渝。大方无隅，大器晚成。
大音希声，大象无形。夫唯道善贷且成。

——老子，《道德经》

虚幻引擎作为业界一流的次时代引擎，开发了无数成功的作品。在短暂的计算机图形学发展历史上，虚幻引擎历经四代，成为游戏引擎界举足轻重的成员之一。

但是虚幻引擎庞大而复杂的设计，阻碍了許多人学习的步伐。尽管有蓝图系统作为图形化编程，降低了虚幻引擎的上手难度，但是当开发者们走入虚幻引擎的 C++ 范畴，依然会感觉到无从下手。

因此，我决定和我的同事一起来撰写本书。希望能够借助我们微薄之力，帮你理解庞大的虚幻引擎是如何工作的。笔者对本书内容的期望是，这是一本笔者在学习虚幻引擎时希望能够获得的书。

同时也请明白，虚幻引擎的代码量为五百万行。本书篇幅不足以分析整个虚幻引擎的所有模块，也无法精确地向读者展示每段代码的意义。相反地，本书立足于：**展示引擎基本结构**，即尽可能告诉读者“它是这么跑起来的”，对于希望精确研究每一段代码过程的读者，本书会告知你如何寻找到对应的代码。

本书主标题为“大象无形”，《道德经》中有言：大器晚成，大音希声，大象无形。本书取“伟大的设计对于使用者来说似乎感觉不到存在”和“优秀的系统设计让开发者不需要过多了解原理即能使用”这样的含义。对于虚幻引擎而言，本书中介绍的很多知识，对于普通开发者来说似乎是“没有感觉到存在”的东西，例如引擎的渲染系统，普通开发者几乎只需要简单地完成导入和摆放就能使用，并不需要实际了解渲染系统的工作原理。能够达到这样的效果，恰恰说明了虚幻引擎设计的优秀：能够让开发者不需要了解系统的机制，就能够快速使用其来完成自己的需求，此即“无形”。然而这样优秀的设计是如何完成的？如何扩展这样的设计来让开发者完成自己独特的需求？这是本书希望探讨的内容。

本书由两位作者共同编撰而成，其中罗丁力先生完成了第一部分（除《引擎系统相关类》章节）和第二部分，以及第三部分中《引擎独立应用程序》《自定义资源和编辑器》章节，张三先生完成了第二部分中《引擎系统相关类》章节与第三部分中《插件开发》章节。

笔者才疏学浅，撰写本书仅仅为个人一家之言。欢迎每一位读者对本书提出建议和指正，也欢迎更多的人去撰写虚幻引擎相关的书籍，共同为虚幻引擎的推广、运用做出努力。你可以发送邮件到 three@sanwu.org。

感谢 Unreal Engine，陪伴我度过了最美好的青春。

阅读之前

你好，欢迎你阅读本书。在这里我希望能向你讲述一些关于阅读本书的约定。首先，这不是一本“虚幻引擎入门宝典”或是“虚幻引擎从入门到精通”。本书的作者们希望把视角集中到那些市面上的教程没有涉及的领域，所以我们不会教你：

- (1) 如何下载引擎
- (2) 如何安装引擎和 Visual Studio
- (3) 如何更新引擎
- (4) 如何申请虚幻引擎账户

我们假定你已经掌握这些知识。并且我们也不会教你：

- (1) C++ 语法
- (2) C 语言语法

我们认为你在使用虚幻引擎的 C++ 语言进行编程之前，已经掌握了 C++ 的基础语法，包括函数、变量、类、指针与模板。当然，我们会向你解释虚幻引擎中的独有的 C++ 成分，包括 C++11 标准的一些内容。

如果你已经做好了准备，欢迎开始你的阅读之旅。本书分为以下三个部分：

虚幻引擎 C++ 编程 这个部分简单介绍虚幻引擎的 C++ 编程方式，你可以通过这个部分回顾、整理你从官方文档学习到的有关使用虚幻引擎进行编程的知识，并给出了一部分官方文档尚未介绍但可以被使用的库、API 与技巧。

虚幻引擎浅析 这个部分将会引导读者去研究虚幻引擎源码，并给出笔者认为在深入使用虚幻引擎进行游戏开发的过程中，可能会需要具备的引擎架构、模块如何工作的知识。换句话说，这个部分介绍虚幻引擎是如何工作的。

扩展虚幻引擎 这个部分则是通过介绍虚幻引擎的插件编写，将第二部分的知识运用起来，让读者不至于觉得这是“屠龙之技”，虽有思辨的乐趣，却没有用武之地。进而赋予读者**定制虚幻引擎**以符合自己游戏实际情况的能力。笔者认为这是专业游戏开发者所需要具备的技能。

在每一小节开头，笔者会提供一个常常被问及的问题，然后根据这个问题来阐述接下来的内容，就像这样：

问题

我该如何学习虚幻引擎？

读者可以在阅读完每一个小节后，回顾小节开头的问题，以检验自己是否已经理解了本节的内容。

笔者在这里衷心地祝愿你找到你希望学习的知识，祝你一切顺利！

鸣谢

本书在撰写过程中受到了大量同行、朋友及亲人的帮助，有许多同行无私地贡献了自己的想法、意见及自己宝贵的经验，在此对他们表示真挚的感谢：

非常感谢 NetFly 和秦春林先生对本书的支持，他们不仅帮助笔者联系了本书的出版社，也非常认真地审阅本书的稿件，并给出了中肯有效的意见，没有他们的帮助，本书不可能出版。

非常感谢傅建钊先生对本书的帮助，提出了大量有效的意见，并组织了相当多的业内人士共同讨论本书的主题，他的知乎专栏《Inside UE4》对虚幻引擎的剖析同样非常精彩，建议读者可以参考。

同时，也有不少同行针对书中许多主题给出了自己独到的见解，并被整理到书中。LSFW 先生给笔者多次反复讲解渲染框架设计，贡献出了自己对渲染系统的研究成果；黄河水先生、Dest1ny 先生撰写了大量博客来分析虚幻引擎的底层架构，给笔者启发颇多；王德立先生帮助本书绘制了插图。还有许许多多同行，在此恕无法一一举名。

感谢三巫社区和 Epic Games 对本书的出版过程的支持与帮助。

最后，作者之一罗丁力希望感谢 Black Rock Shooter，感谢她在撰写本书的过程中，对其鼓励与陪伴。

读者服务

轻松注册成为博文视点社区用户 (www.broadview.com.cn), 您即可享受以下服务。

- **提交勘误:** 您对书中内容的修改意见可在【提交勘误】处提交, 若被采纳, 将获赠博文视点社区积分 (在您购买电子书时, 积分可用来抵扣相应金额)。
- **与作者交流:** 在页面下方【读者评论】处留下您的疑问或观点, 与作者和其他读者一同学习交流。

页面入口: <http://www.broadview.com.cn/31349>



目录

第一部分 虚幻引擎 C++ 编程	1
第 1 章 开发之前——五个最常见基类	2
1.1 简述	2
1.2 本立道生：虚幻引擎的 UObject 和 Actor	2
1.2.1 UObject 类	2
1.2.2 Actor 类	5
1.3 灵魂与肉体：Pawn、Character 和 Controller	6
1.3.1 Pawn	6
1.3.2 Character	7
1.3.3 Controller	7
第 2 章 需求到实现	9
2.1 分析需求	9
2.2 转化需求为设计	10
第 3 章 创建自己的 C++ 类	12
3.1 使用 Unreal Editor 创建 C++ 类	12
3.2 手工创建 C++ 类	14
3.3 虚幻引擎类命名规则	15

第 4 章 对象	16
4.1 类对象的产生	16
4.2 类对象的获取	18
4.3 类对象的销毁	18
第 5 章 从 C++ 到蓝图	20
5.1 UPROPERTY 宏	20
5.2 UFUNCTION 宏	20
第 6 章 游戏性框架概述	22
6.1 行为树：概念与原理	22
6.1.1 为什么选择行为树	22
6.1.2 行为树原理	22
6.2 虚幻引擎网络架构	26
6.2.1 同步	26
6.2.2 广义的客户端-服务端模型	27
第 7 章 引擎系统相关类	30
7.1 在虚幻引擎 4 中使用正则表达式	30
7.2 FPaths 类的使用	31
7.3 XML 与 JSON	32
7.4 文件读写与访问	33
7.5 GConfig 类的使用	35
7.5.1 写配置	36
7.5.2 读配置	36
7.6 UE_LOG	37
7.6.1 简介	37
7.6.2 查看 Log	37
7.6.3 使用 Log	37

7.6.4	自定义 Category	38
7.7	字符串处理	38
7.8	编译器相关技巧	39
7.8.1	“废弃”函数的标记	39
7.8.2	编译器指令实现跨平台	39
7.9	Images	40

第二部分 虚幻引擎浅析 45

第 8 章	模块机制	46
8.1	模块简介	46
8.2	创建自己的模块	47
8.2.1	快速完成模块创建	47
8.2.2	创建模块文件夹结构	48
8.2.3	创建模块构建文件	49
8.2.4	创建模块头文件与定义文件	49
8.2.5	创建模块预编译头文件	50
8.2.6	引入模块	51
8.3	虚幻引擎初始化模块加载顺序	52
8.4	道常无名：UBT 和 UHT 简介	55
8.4.1	UBT	55
8.4.2	UHT	57
第 9 章	重要核心系统简介	62
9.1	内存分配	62
9.1.1	Windows 操作系统下的内存分配方案	62
9.1.2	Intel TBB 内存分配器	63
9.2	引擎初始化过程	65

9.3	并行与并发	67
9.3.1	从实验开始	68
9.3.2	线程	71
9.3.3	Task Graph 系统	73
9.3.4	Std::Thread	77
9.3.5	线程同步	78
9.3.6	多进程	80
第 10 章	对象模型	81
10.1	UObject 对象	81
10.1.1	来源	82
10.1.2	重生：序列化	83
10.1.3	释放与消亡	92
10.1.4	垃圾回收	94
10.2	Actor 对象	99
10.2.1	来源	101
10.2.2	加载	104
10.2.3	释放与消亡	105
第 11 章	虚幻引擎的渲染系统	107
11.1	渲染线程	107
11.1.1	渲染线程的启动	108
11.1.2	渲染线程的运行	108
11.2	渲染架构	109
11.2.1	延迟渲染	109
11.2.2	延迟渲染在 Post Process 中的运用	110
11.3	渲染过程	111
11.3.1	延迟渲染到最终结果	112
11.3.2	渲染着色器数据提供	123

11.4	场景代理 SceneProxy	126
11.4.1	逻辑的世界与渲染的世界	126
11.4.2	渲染代理的创建	127
11.4.3	渲染代理的更新	128
11.4.4	实战：创建新的渲染代理	128
11.4.5	进阶：创建静态渲染代理	132
11.4.6	静态网格物体渲染代理排序	148
11.5	Shader	150
11.5.1	测试工程	150
11.5.2	定义 Shader	152
11.5.3	定义 Shader 对应的 C++ 类	153
11.5.4	我们做了什么	162
11.6	材质	163
11.6.1	概述	164
11.6.2	材质相关 C++ 类关系	166
11.6.3	编译	167
11.6.4	ShaderMap 产生	168
第 12 章	Slate 界面系统	170
12.1	Slate 的两次排布	170
12.2	Slate 的更新	170
12.3	Slate 的渲染	171
第 13 章	蓝图	173
13.1	蓝图架构简述	173
13.2	前端：蓝图存储与编辑	174
13.2.1	Schema	174
13.2.2	编辑器	175
13.3	后端：蓝图的编译	176

13.4	蓝图虚拟机	187
13.4.1	便笺纸与白领的故事	187
13.4.2	虚幻引擎的实现	189
13.4.3	C++ 函数注册到蓝图	193
13.5	蓝图系统小结	194
第三部分 扩展虚幻引擎		197
第 14 章	引擎独立应用程序	198
14.1	简介	198
14.2	如何开始	198
14.3	BlankProgram	199
14.4	走得更远	202
14.4.1	预先准备	203
14.4.2	增加模块引用	203
14.4.3	添加头文件引用	203
14.4.4	修改 Main 函数为 WinMain	204
14.4.5	添加 LOCTEXT_NAMESPACE 定义	204
14.4.6	添加 SlateStandaloneApplication	204
14.4.7	链接 CoreUObject	205
14.4.8	添加一个 Window	205
14.4.9	最终代码	205
14.5	剥离引擎独立应用程序	207
第 15 章	插件开发	208
15.1	简介	208
15.2	开始之前	208
15.3	创建插件	210
15.3.1	引擎插件与项目插件	210

15.3.2	插件结构	210
15.3.3	模块入口	213
15.4	基于 Slate 的界面	213
15.4.1	Slate 简介	213
15.4.2	Slate 基础概念	214
15.4.3	最基础的界面	214
15.4.4	SNew 与 SAssignNew	219
15.4.5	Slate 控件的三种类型	220
15.4.6	创建自定义控件	222
15.4.7	布局控件	225
15.4.8	控件参数与属性	227
15.4.9	Delegate	228
15.4.10	自定义皮肤	234
15.4.11	图标字体	239
15.4.12	组件继承	241
15.4.13	动态控制 Slot	244
15.4.14	自定义容器布局	248
15.5	UMG 扩展	255
15.6	蓝图扩展	261
15.6.1	蓝图函数库扩展	261
15.6.2	异步节点	264
15.7	第三方库引用	270
15.7.1	lib 静态链接库的使用	270
15.7.2	dll 动态链接库的使用	273
第 16 章	自定义资源和编辑器	276
16.1	简易版自定义资源类型	276
16.2	自定义资源类型	278
16.2.1	切分两个模块	278

16.2.2	创建资源类	281
16.2.3	在 Editor 模块中创建工厂类	281
16.2.4	引入 Editor 模块	284
16.3	自定义资源编辑器	286
16.3.1	资源操作类	287
16.3.2	资源编辑器类	290
16.3.3	增加 3D 预览窗口	295

第一部分

虚幻引擎 C++ 编程

本文假定你已经了解 C++ 的语法，包括变量、函数、类与指针。

我们不会在这里教你如何书写函数声明，如何调用一个函数等。本文并不需要你了解虚幻引擎的 C++ 编程，因为我们会讲述到。但是，不同于官方教程的简单明了，我们将会向你介绍更多关于“为什么这样设计”的内容。如果你希望在几分钟之内上手虚幻引擎的 C++ 编程，可能你更应该直接阅读虚幻引擎的 C++ Quick Start Guide。

如果你发现，在阅读了官方的文档之后，你依然感到无所适从，不知道如何下手的话，我想接下来的章节能够让你对虚幻引擎的开发，有更加全面的认识。

本部分希望用一组相互之间比较独立的短文，来向你陈述虚幻引擎 C++ 编程的一些技巧，并帮助你准备后文需要的一些知识。

第 1 章

开发之前——五个最常见基类

1.1 简述

问题

如何最快上手虚幻引擎的 C++ 编程？

许多人都询问过这样的问题。学习虚幻引擎的编程技术有许多的道路，但是笔者认为，抓住最核心的五个类，提纲挈领地学习，能够更好地理解。这五个类就是：

UObject

Actor

Pawn

Controller

Character

1.2 本立道生：虚幻引擎的 UObject 和 Actor

1.2.1 UObject 类

问题

什么时候该继承自 UObject 类？什么时候应该声明一个纯 C++ 类？

任何一个 C++ 程序员都知道，不同于 Java，C++ 的类可以没有父类。那么，什么样的类对象应该继承自 UObject 类？