Broadview
www.broadview.com.cn

这是一本不一样的Unity实战书，
是多年一线开发经验和实际运用技术的分享

Unity 3D 实战核心技术详解

姜雪伟 著

从数学及Shader出发，分享实战项目，剖析架构，系统讲解游戏开发与制作

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Unity 3D 实战核心技术详解

姜雪伟 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

《Unity 3D 实战核心技术详解》详细介绍了实际游戏开发中使用的核心技术，每一章都结合了游戏开发的实战案例。首先，介绍了 3D 数学在 Unity 中的运用，3D 数学知识包括：Unity 坐标系统、向量、矩阵、四元数、欧拉角等基础知识。其次，介绍了游戏开发中常用的核心技术：Avatar 换装系统、消息事件系统、Protobuf 在游戏中的运用，以及游戏中的文本文件加密算法等。再次，介绍了游戏中的 AI 行为树算法、残影算法、移动端实时阴影绘制、移动端海水的绘制等技术。然后，在游戏架构设计方面，介绍了最经典的针对 UI 的 MVC 架构设计和对于角色动作和技能的 FSM 有限状态机架构，以及游戏版本迭代使用的热更新技术方案。最后，介绍了移动端 GPU 编程和游戏开发的一些经验。

《Unity 3D 实战核心技术详解》适合具备一定 Unity 开发经验的初学者和有一定 Unity 项目开发经验的游戏开发者阅读。

图书在版编目（CIP）数据

Unity 3D 实战核心技术详解 / 姜雪伟著. —北京：电子工业出版社，2017.1
ISBN 978-7-121-30432-3

Ⅰ. ①U… Ⅱ. ①姜… 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（2016）第 284517 号

策划编辑：董　英
责任编辑：徐津平
印　　刷：三河市双峰印刷装订有限公司
装　　订：三河市双峰印刷装订有限公司
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：21　字数：455 千字　彩插：4
版　　次：2017 年 1 月第 1 版
印　　次：2017 年 1 月第 1 次印刷
印　　数：3500 册　定价：79.00 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 zlts@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。

推荐序

Unity跨平台引擎这几年在国内的发展势头非常迅猛，截至目前已经横跨27个主流游戏平台，尤其是在手游开发领域发展势头尤为凸显。大量的中小型公司几乎都使用Unity开发游戏。所以在App Store和Android Market上有大量用Unity开发的优质游戏或应用。开发者青睐Unity主要有两个方面原因，一方面是Unity很方便就能跨平台，另一方面是Unity上手快，但是在我看来它受欢迎最主要的原因是上手快。

跨平台。Unity能一键发布在27个平台上，但是每个平台的特性是不同的，甚至可以说完全不一样。比如代码层面，接入移动SDK，不是所有厂商都能提供Unity的SDK接口。一些平台特定的API，Unity可能没有提供，或者说还没来得及提供，在这些复杂情况下我们只能通过原生方法去调用它。此外，硬件层面内存CPU性能的不同，操作层面触屏、手柄、PC实体按键也都不同，游戏玩法从设计上不可能都一样，所以真正的跨平台并不是随随便便打个包就能完事的。

上手快。Unity底层全部使用C/C++与OpenGL ES 3D渲染引擎交互，C#端封装了UnityEditor.DLL和UnityEngine.DLL，通过与Unity内部C/C++代码交互从而与底层代码相互调用。从开发层面看，开发者不用考虑怎么实现复杂的图形学、3D数学算法，甚至完全不懂OpenGL ES的同学也能用Unity开发游戏。因为一切复杂的东西Unity在底层都封装好了，开发者只需要编写C#代码访问Unity提供的UnityEditor.DLL及其提供的API接口即可从容地开发3D游戏。听起来是不是很酷炫呢？可是Unity依然存在很多隐患。

目前看来Unity确实很强大，能做的东西确实很多。但是它却有点“臃肿”了。做过Unity手游开发的人应该都知道，Unity的效率并不高。在UI层面，无论是NGUI，还是UGUI，界面

元素稍微多一点打开界面的时候就会出现卡顿。在 3D 层面，同屏人数多一点或者 Shader 复杂一些，帧率马上就掉下去了。Unity 引擎“大而全”，它为了各个平台的兼容性，性能问题必然会显现出来，因为没办法对某个游戏平台做针对性的优化，加上 Unity 没有开放源码，开发者更多时候只能靠“猜”，所以有些更大的公司会使用自研引擎。

关于热更新，我觉得它更多是用来修正游戏 Bug 的。如果没有热更新，修改一个 Bug 后需要重新提交 App Store 评审，这样来来回回可能要耽误好几天时间，不是很合理。由于苹果禁止了 JIT，Unity 官方没有提供热更新的方案，因此我们都采用 Lua 进行热更新的开发。虽然 Lua 的解释器可以用 C++代码解释，但是在处理 Lua 与 C#之间穿透的时候依然会很慢，程序设计不合理就会产生性能问题。

所以 Unity 最大的痛点就是性能问题，它上手确实很快，但是想学好却是一件很难的事情。脚本开发非常地灵活，但是太灵活就可能会被滥用。现在的互联网上已经充满了各式各样的 Unity 教程，但是大多数都比较零散，我们太需要一本从实战项目总结出来的好书。姜雪伟的这本新书，从数学以及 Shader 层面出发，通过实战项目经验的分享以及框架层面的剖析为大家揭露开发的机密，不可不读。

宣雨松 MOMO
资深 Unity 3D 开发者

推荐语

Unity 3D 游戏引擎在如火如荼的 AR、VR 技术方向下正在大放异彩。Google 的 Daydream 引导的新一波移动 VR 浪潮，将会使广大 Unity 3D 手游程序员继续在 VR 移动大势中弄潮。如果你对手游感兴趣，如果你不想错过 AR、VR 乃至 MR 技术浪潮，那么姜雪伟的这本《Unity 3D 实战核心技术详解》，你是不应该错过的。本书不同于以往从 hello world 开始介绍的 Unity 图书，而是从 3D 数学和 Unity 3D 坐标系说起，继而就 Unity 3D 的换装技术进行讲解，然后就 MVC 架构设计、Protobuf、移动端 Shader、移动端海水等 Unity 3D 常备技能一一道来。无论是 Unity 3D 的新手，还是 Unity 3D 技术熟手，都能从姜雪伟有底蕴的文字中获得力量。

——王文刚

微软.NET MVP，原创技术博客 www.xifarm.com 的博主

爱工作，爱编程，爱分享，是我对姜雪伟的第一印象。姜雪伟在业余时间写了这本书，很不容易，也很实用，这本书里蕴藏着他的心血和他对游戏开发的热爱。这本书由浅入深，从基础 3D 数学到高级的 GPU 编程。如果你对游戏开发比较感兴趣，或者正在通往游戏开发的路上，想系统地了解游戏开发和制作均可以阅读本书。从架构的角度看，本书介绍了游戏的 UI 模块、IO 模块、Sound 模块、Net 模块等。从游戏制作的角度看，本书介绍了 UI 的制作、角色的加载及使用、场景的制作（地形、树木、水、阴影等）、粒子和物理系统的使用。恭喜买到本书的你可以系统而详细地了解游戏的开发和制作，祝愿姜雪伟的这本书可以大卖！

——张泽瑞（小阿哥）

掌趣科技主程，一个从业 8 年不忘初心的游戏开发者

Unity 3D 是一个富有生命力的游戏引擎，从最初的支持移动平台、多平台，到现在的支持AR/VR，它一直紧跟着发展的趋势。市场上众多用 Unity 3D 开发的作品也证实了 Unity 3D 用于游戏开发是可靠的。那么怎样利用 Unity 3D 更好地进行开发，怎样充分挖掘 Unity 3D 的特性，怎样在 Unity 3D 中做出和其他大作一样的游戏产品，这些才是我们开发者应该关注的。

姜雪伟这本《Unity 3D 实战核心技术详解》正是为了尝试解决上述问题而生的。这本书从 Unity 3D 乃至 3D 游戏的一些基本原理着手，让读者在编程时更清楚为什么这么做。同时辅以大量的实例，这些实例能让读者接触到一个真正的商业项目中遇到的一些难题和解决方案。这些经验对于开发者而言才是最宝贵的。

——蔡俊鸿

《Unity 5 实战：使用 C#和 Unity 开发多平台游戏》译者，

广州西姆雷娱乐有限公司 Technical Lead，kakashi01.com 博主

当前 VR、AR 技术得到了普遍关注，成为了当下最热门的技术之一，这使得 Unity 的使用领域不再局限于游戏开发，在虚拟现实、3D 空间展示（如房地产楼盘展示）等领域都有良好的表现，已经面世的 Unity 游戏充分显示了 Unity 的延展性。大到数百人进行研发的 MMORPG，小到一两个人开发的独立游戏，都可以使用 Unity 进行开发。

姜雪伟老师的这本《Unity 3D 实战核心技术详解》与目前市面上的同类书籍有很大不同，它不是一些入门的基础知识，而是姜雪伟多年一线开发经验的技术分享，这本书从数学知识出发，剖析架构层面，以实例详细介绍游戏开发实际过程中常见的一些难点和重点，无论是 Unity 3D 的初学者还是已经入行的开发者，这本书对你们的能力提升都有很大的帮助，实用性很强。

——张文（优弧）

泰课在线产品经理，极客，独立开发者

前　言

在 IT 行业中游戏程序员的薪资相对于其他领域来说还是比较高的，这也导致了许许多多的软件开发者转入游戏行业，从事 Unity 3D 游戏开发。Unity 引擎因为具有上手快并且能跨平台开发的优点，被众多开发者所喜爱。虽然游戏研发公司对开发人员的需求越来越大，但是初次踏入游戏行业的开发者由于游戏产品项目经验不足，并不被游戏公司所认可。我在泰课在线教育做讲座时发现，有的学员水平还是不错的，但由于他们在游戏行业的工作经验不足，最终很难被游戏公司录用。如今，政府和公司支持的各种创客空间的兴起，为开发者提供了非常好的硬件条件，极大地促进了 IT 行业的繁荣发展，特别是游戏和软件开发的发展。

但是游戏行业也是一种技术浓缩型行业，作为一名从业者，要想在这个行业生存和发展，没有过硬的技术是不行的。可是，多年的从业经历让我见到许许多多的年轻人，他们在公司里只能写一些简单的逻辑，长期被边缘化，随时都有被辞退的危险。究其原因，还是他们缺乏解决实际问题的经验，在公司的产品开发中无法提供价值。在和他们接触的过程中，我发现了一个很严重的问题：这些能力欠佳的年轻人也有强烈的欲望去学习专业知识，但是市场上的 Unity 书籍讲述的都是一些入门的基础知识，已经远远不能满足手游开发工作的需要，他们急需能够与实际开发相结合并且能够帮助他们解决实际问题的书籍。作为他们的前辈，我也有过相似的经历，花费了很多时间去摸索，我一直在思考如何才能让后来者少走弯路。经过多年的经验积累，我终于奉上了这本呕心沥血之作。本书讲述的所有知识点都是与实际开发密切相关的，很多代码可以直接拿过来运用到项目中，实用性超强！

本书致力于用最通俗的语言讲述初学者最需要的知识。在写作形式上，本书采用图文并茂的方式，让读者更容易掌握知识，为开发能力的提升提供了巨大的帮助。在技术知识的安排上，本书并没有把核心技术一一讲解，而是选择了一些游戏开发中经常使用的技术。

本书的编写不同于只写对 Unity 编辑器的一些基本操作的书籍。本书重点讲解游戏开发中实际运用的技术，分享游戏开发经验，阅读完本书，游戏开发者在技术方面和开发经验方面都会有一个质的提升。本书并不是介绍一个游戏是如何制作的，而是介绍游戏开发中的各个核心技术点。因为对于现在的游戏公司来说，核心技术只掌握在少数几个人手里，其他人很难接触到，他们或者是在已有框架的基础上开发，或者是使用已经封装好的技术模块开发。这样对于开发者技术的提升是非常不利的，也不利于增长实际项目开发经验。本书的宗旨就是把游戏中使用的核心技术无偿奉献给开发者，让开发者在最短的时间内提高自己的技能。

姜雪伟

2016 年 10 月

提示

登录博文视点官方网站（www.broadview.com.cn）进入本书页面，在“资源文件”栏目中可以免费下载本书的源代码。

目　录

1

第 1 章

3D 数学与 Unity

任何游戏开发都离不开数学知识，数学知识在 IT 领域一直是非常重要的，没有数学知识的支撑，程序就失去了灵魂。所以在 IT 开发，特别是游戏开发或者 VR/AR 开发领域，运用数学知识解决问题是程序员必须要掌握的技能之一，并且还要能将它们灵活运用到实际项目中解决实际问题。

我们就从游戏中最基本的也是应用最广泛的数学知识点：坐标系、向量、矩阵、四元数、欧拉角等讲起，逐一给大家介绍其在手游开发中的运用。在介绍的过程中会结合实际项目案例，帮助读者真正地做到学以致用。本章的知识点并没有先后顺序，可以采用跳跃式阅读，希望读者学过以后可以在游戏开发时尝试着运用所学的知识。

我在游戏行业从事研发工作十五年了，也算是行业中的一名“老兵”，参与或主导过十多款大型网络游戏产品的开发，月流水最多的游戏产品达到五千万以上。当然我也经历过失败的产品，受到过惨痛的教训，对于成功的产品和失败的产品我们在这里不做过多讨论。言归正传，不论成功的产品还是失败的产品，在项目研发过程中都要运用数学知识来解决产品的玩法问题。尤其对于新人，知道如何把大学里面学的数学知识运用到实际项目开发中尤其重要。当前关于 Unity 开发的书籍大部分是偏重于编辑器功能介绍的。当然也有书籍介绍数学知识，但是偏重于

纯理论的数学知识，相对来说比较枯燥，而且没有与实际的开发结合起来，这样就导致了理论与实际的脱钩。本章在介绍理论的同时也结合实际开发操作，真正做到实践与理论结合。

Unity 引擎是非常成熟的，引擎内部运用了很多数学知识，它对开发者来说是不可见的，而且它已封装好的算法也不是很全面。此外，要使用引擎封装的算法也要明白其实现原理。用 Unity 引擎开发游戏就是在引擎的上层再封装一层游戏架构，在游戏开发的逻辑中需要根据策划需求重新封装一些数学算法以便于逻辑调用。我在业余时间担任多家教育网站的高级讲师，比如，51CTO 教育在线、CSDN 教育在线、泰课教育等，就是想把自己的一些项目经验分享给开发者，让开发者在比较短的时间内能够快速提升自己，这也是本书的写作目的之所在，接下来首先讲一下 Unity 坐标系。

1.1 Unity 坐标系

在学习使用 Unity 时，有个问题值得大家思考一下，Unity 使用的是哪种坐标系，左手坐标系还是右手坐标系？在介绍 Unity 坐标系之前先给大家解释一下 3D 坐标系。3D 坐标系表示的是三维空间，3D 坐标系存在三个坐标轴，分别为 x 轴、y 轴、z 轴，如图 1-1 所示。

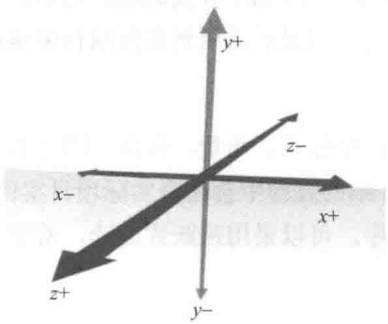

图 1-1　3D 坐标系

3D 坐标系分为左手坐标系和右手坐标系。大家可以想象一下：在计算机屏幕上，有三个轴，一个是 x 轴，它是由左指向右的；另一个是 y 轴，它是由下指向上的；剩下的一个是 z 轴，z 轴的指向就决定了 3D 引擎使用的是左手坐标系还是右手坐标系。z 轴指向屏幕里面的是左手坐标系，z 轴指向屏幕外面的是右手坐标系。对于 3D 引擎来说，大部分都使用左手坐标系，以 Unity 引擎为例，它使用的是左手坐标系。明白了坐标系后，在场景中操作物体对象时就会有针对性地通过 x 轴、y 轴、z 轴实现物体对象位置的调整。为什么要告诉大家坐标系呢？因为它其实和我们现实生活一样，现实世界由东、西、南、北四个方向组成，这样大家就可以根据这四个方向建房屋、修路等。虚拟世界也需要坐标系，只是它表示的比较简单而已。

Unity 引擎的左手坐标系也被称为世界坐标系，做游戏开发时需要美工制作美术模型，运用

MAX 工具把建好的模型放到游戏场景中。在默认情况下，局部坐标和世界坐标系的原点是重合的，不能把所有的模型都叠加在世界坐标系的原点上，因此需要移动模型。模型移动时就会发生模型的局部坐标到世界坐标的转换，这个移动过程就是把模型的局部坐标转化成世界坐标。只是这个转化过程是在引擎编辑器内部实现的，实际上它就是将模型的各个点与世界矩阵相乘得到的。Unity 编辑器中的物体都在世界坐标系里面，比如我们通常使用的函数 transform.position，它就是获取到当前物体的世界坐标位置，用户无须自己去计算，因为引擎内部已经计算好了。明白了原理后，再使用编辑器解决问题更有助于理解，做到知其然且知其所以然。如果要获取物体自身的坐标，也就是局部坐标，可以使用函数 transform.localPosition 获取当前模型的局部坐标。

用 Unity 引擎开发移动端手游会经常用到屏幕坐标系，屏幕坐标系就是通常使用的电脑屏幕，它是以像素为单位的，屏幕左下角为（0,0）点，右上角为（Screen.Width, Screen.Height）点，Z 的位置是根据相机的 Z 缓存值确定的。通常使用鼠标在屏幕上单击物体，它就是屏幕坐标。通过函数 Input.mousePosition 可以获得鼠标位置的坐标。我们使用的虚拟摇杆可以在屏幕上滑动，它也是屏幕坐标，可以通过函数 Input.GetTouch(0).position 获取到手指触摸屏幕坐标。在游戏开发中，比如单击场景中的 3D 物体就需要从屏幕上发射一条射线与物体的包围盒相交，用于判断是否选中物体，对于 UI 的操作也都是基于屏幕坐标系的。

通过相机才能看到虚拟世界的物体。相机有自己的视口坐标，物体要转换到视口坐标才能被看到。相机的视口左下角为（0,0）点，右上角为（1,1）点，Z 的位置是以相机的世界单位来衡量的。（0,0）点和（1,1）点是通过公式进行缩放计算的，这里面存在一个变换，读者了解就可以了。这也是为什么视口的大小通常都是（0,0）和（1,1），效果如图 1-2 所示，图的中心点是摄像机。

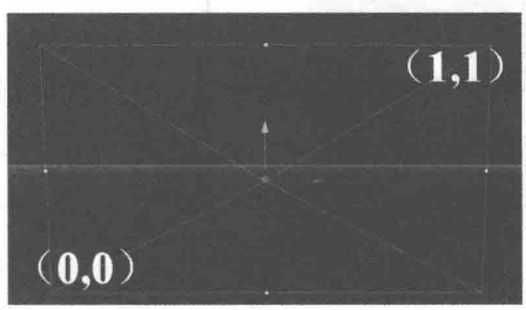

图 1-2　视口大小

下面介绍世界坐标、屏幕坐标、相机坐标之间的转换方式。举一个简单的例子，在一个空

场景里面放置一个立方体，物体在编辑器中也就是世界坐标系中的摆放如图 1-3 所示。

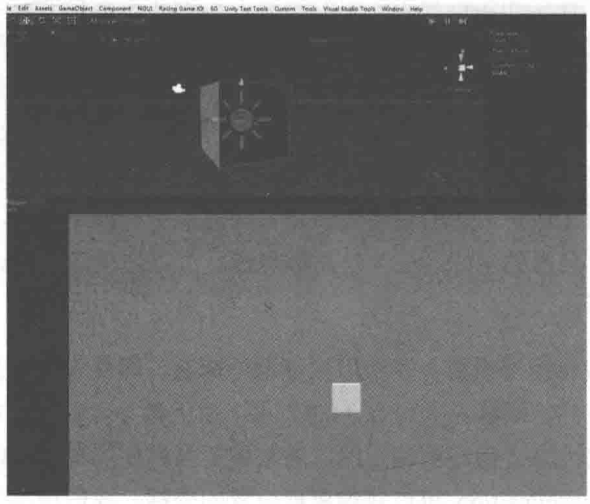

图 1-3　物体在编辑器中的摆放

获取物体位置的通常写法是 transform.position，它表示的是立方体在 3D 世界中的世界坐标的位置。如果使用的是触摸屏幕，那么可以通过函数 Input.GetTouch(0).position 获取到屏幕坐标。它们之间的转换方式如下。

- 世界坐标到屏幕坐标的转化函数：camera.WorldToScreenPoint(transform.position)。
- 屏幕坐标到视口坐标的转化函数：camera.ScreenToViewportPoint(Input.GetTouch(0).position)。
- 世界坐标到视口坐标的转化函数：camera.WorldToViewportPoint(obj.transform.position)。

这些转换也是固定流水线的矩阵变换，只是 Unity 将其封装好了而已。如果想学习固定流水线，可以参考《手把手教你架构 3D 游戏引擎》一书，里面有固定流水线的详细讲解，下面介绍向量运算。

1.2　向量

向量的基本运算包括加法、减法、点乘、叉乘、单位化运算等，其中减法、点乘、叉乘、单位化运算在游戏开发中使用得最为广泛。首先介绍一下向量，向量的表示如图 1-4 所示。