



Unity5.X游戏开发基础

主 编 张 帆 副主编 范义娜



浙江工商大学出版社
ZHEJIANG GONGSHANG UNIVERSITY PRESS

感谢2016年教育部产学合作协同育人项目：大学生创新创业联合基金-教学内容和课程体系改革-《游戏引擎技术》项目和ARM公司提供的大力支持



Unity5.X游戏开发基础

主 编 张 帆 副主编 范义娜



浙江工商大学出版社
ZHEJIANG GONGSHANG UNIVERSITY PRESS

图书在版编目(CIP)数据

Unity5.X 游戏开发基础 / 张帆主编. —杭州: 浙江工商大学出版社, 2017.5
ISBN 978-7-5178-2093-2

I. ①U… II. ①张… III. ①游戏程序—程序设计—高等学校—教材 IV. ①TP317.61

中国版本图书馆 CIP 数据核字(2017)第 071376 号

Unity5.X 游戏开发基础

张 帆 主编 范义娜 副主编

责任编辑 罗丁瑞
责任校对 王文舟
封面设计 林朦朦
责任印制 包建辉
出版发行 浙江工商大学出版社
(杭州市教工路 198 号 邮政编码 310012)
(E-mail: zjgsupress@163.com)
(网址: <http://www.zjgsupress.com>)
电话: 0571-88904980, 88831806(传真)

排 版 杭州朝曦图文设计有限公司
印 刷 杭州五象印务有限公司
开 本 787mm×1092mm 1/16
印 张 37.5
字 数 863.5 千
版 印 次 2017 年 5 月第 1 版 2017 年 5 月第 1 次印刷
书 号 ISBN 978-7-5178-2093-2
定 价 98.00 元

版权所有 翻印必究 印装差错 负责调换

浙江工商大学出版社营销部邮购电话 0571-88904970

目 录

第 1 章 游戏引擎介绍

1.1 游戏引擎简介	3
1.2 游戏引擎的选择	4
1.3 目前流行的游戏引擎	5
1.3.1 虚幻引擎(Unreal4)	5
1.3.2 Cry Engine	6
1.3.3 寒霜引擎(Frosbite Engine)	6
1.3.4 Torque3D/2D 游戏引擎	7
1.3.5 基于 2D 风格的 Cocos2D 游戏引擎	7
1.3.6 多平台发布的 Unity3D 游戏引擎	8
1.4 Unity3D 游戏引擎的下载和安装	8
1.4.1 Unity3D 下载	9
1.4.2 Unity3D 的安装	11
1.4.3 Unity3D 的注册	12
1.4.4 启动 Unity3D	13
1.5 总结	14
1.6 练习题	14

第 2 章 3D 游戏开发所需要的重要概念

2.1 3D 图形学中的重要概念	19
2.1.1 坐标系	19
2.1.2 局部坐标系与世界坐标系(Local & Worldcoordinate System) ...	20
2.1.3 父子物体(Parent-Child)	20
2.1.4 向量(Vector)	21
2.1.5 摄像机(Camera)	22
2.1.6 多边形(Polygons)、边(Edges)、顶点(Vertexes)和面片(Meshes)	22
2.1.7 材质(Materials)、贴图(Textures)和着色器(Shaders)	23
2.1.8 物理引擎	24
2.1.9 碰撞检测(Collision Detection)	24

2.1.10	凸面体与凹面体(Convex and Concave)	25
2.2	Unity3D 中定义的重要概念	25
2.2.1	资源(Assets)	25
2.2.2	工程(Project)	26
2.2.3	场景(Scenes)	26
2.2.4	游戏对象(Game Object)	26
2.2.5	组件(Component)	26
2.2.6	脚本(Scripts)	27
2.2.7	预置(Prefabs)	27
2.3	总结	27
2.4	练习题	28

第3章 Unity3D 界面介绍

3.1	Unity3D 编辑器的布局	31
3.1.1	标题栏	31
3.1.2	主菜单栏	32
3.1.3	Project(项目资源)窗口	40
3.1.4	Hierarchy(层级窗口)	48
3.1.5	Scene 场景窗口	49
3.1.6	Inspector 组件参数编辑窗口	60
3.1.7	Game(游戏预览窗口)	66
3.1.8	Console(控制台)	67
3.2	自定义窗口布局	68
3.2.1	使用 Unity3D 内置的窗口布局功能	68
3.2.2	自定义窗口布局	71
3.3	总结	72
3.4	练习题	72

第4章 Unity3D 脚本程序介绍

4.1	脚本程序初探	77
4.1.1	Mono Develop 脚本编辑器	78
4.1.2	第一个 C# Script 脚本	78
4.2	C# Script 的语法	81
4.2.1	变量声明	81
4.2.2	函数声明	82
4.2.3	类与类的使用	85

4.3 Unity3D 事件函数调用顺序	101
4.3.1 基本事件函数	101
4.3.2 针对游戏对象当前状态的事件函数	104
4.3.3 Unity3D 事件运行顺序	105
4.4 总结	105
4.5 练习题	106
第 5 章 地形编辑器	
5.1 地形编辑范例	109
5.1.1 创建地形	109
5.1.2 为地形绘制贴图纹理	113
5.1.3 为地形放置树木	117
5.1.4 为地形放置花草	121
5.1.5 为地形添加细节模型	122
5.1.6 其他设置	125
5.2 地形编辑的其他设置	126
5.2.1 设置地形的分辨率	126
5.2.2 提高海平面	127
5.2.3 使用地形高度图	133
5.3 为场景添加水体、天空盒与太阳光	137
5.3.1 添加水体	137
5.3.2 添加天空盒	139
5.3.3 添加太阳光	140
5.3.4 为场景添加第一人称角色	141
5.4 树木创建器 (Tree Creator)	142
5.4.1 植物的传统建模方法	142
5.4.2 树木创建器面板	144
5.4.3 制作银杏树	149
5.5 World Machine 制作地形	158
5.5.1 World Machine 制作火山口	158
5.5.2 导入 Unity	182
5.6 Speedtree	184
5.6.1 树的基本结构	185
5.6.2 对柳树的形态结构分析	185
5.6.3 利用 Speed Tree 创建柳树模型	186
5.6.3 Speed Tree 模型导入 Unity5	193
5.6.4 为地形使用笔刷放置 Speed Tree 树模型	194

5.7 总结	195
5.8 练习题	195

第 6 章 3D 模型的导入

6.1 静态模型的导出	199
6.1.1 单位的设置	199
6.1.2 制作一个茶壶模型	200
6.1.3 模型导出之前需要注意的事项	207
6.2 总结	209
6.3 练习题	210

第 7 章 贴图、材质与 Shader 着色器

7.1 贴图 (Texture)	213
7.2 PBS 关键属性介绍	215
7.2.1 固有颜色 (Albedo)	215
7.2.2 金属度 (Metallic)	216
7.2.3 光滑度 (Smoothness)	216
7.3 PBS 其他属性介绍	217
7.3.1 法线贴图 (Normal Map)	217
7.3.2 高度贴图 (Height Map)	218
7.3.3 遮挡贴图 (Occlusion Map)	218
7.3.4 发光属性 (Emission)	219
7.3.5 菲涅尔效果 (Fresnel Effect)	219
7.4 PBS 范例	220
7.5 着色器 (Shader)	226
7.5.1 着色器的作用	226
7.5.2 着色语言	230
7.5.3 Unity3D 中的着色器	230
7.6 总结	235
7.7 练习题	235

第 8 章 光源

8.1 平行光 (Directional Light)	239
8.1.1 太阳光	240
8.1.2 月光	242

8.2 点光源 (Point Light)	244
8.2.1 使用点光源	245
8.3 聚光灯 (Spot Light)	248
8.3.1 使用聚光灯	248
8.4 三种灯光的属性说明	250
8.5 使用程序控制灯光	250
8.6 总结	253
8.7 练习题	253

第 9 章 音 频

9.1 音频剪辑 (Audio Clip)	257
9.2 播放音频	259
9.2.1 音频监听组件 (Audio Listener)	259
9.2.2 音频源组件 (Audio Source)	260
9.2.3 音频混合器 (Audio Mixer)	261
9.2.4 音频过滤器/效果器 (Audio Filters/Audio Effects)	262
9.2.5 回响空间 (Reverb Zones)	263
9.3 Audio Mixer	264
9.3.1 Audio Mixer 原理	264
9.3.2 使用 AudioManager	265
9.4 总结	284
9.5 练习题	284

第 10 章 碰撞盒与触发器

10.1 碰撞盒 (Collider)	287
10.1.1 碰撞盒初探	288
10.1.2 网格碰撞盒 (Mesh Collider)	289
10.1.3 碰撞盒的阻挡作用	291
10.1.4 碰撞事件的运用	301
10.2 触发器	308
10.3 总结	317
10.4 练习题	317

第 11 章 3D 物理模拟

11.1 刚体 (Rigid Body)	321
-----------------------------	------------

11.2 物理材质 (Physics Material)	323
11.2.1 反弹系数	324
11.2.2 摩擦系数	326
11.3 脚本控制刚体 (Control Rigidbody by Script)	328
11.4 布料 (Cloth)	331
11.4.1 飘扬的旗帜	331
11.4.2 软皮球	334
11.5 物理关节 (Physics Joint)	336
11.5.1 铰链关节 (Hinge Joint)	336
11.5.2 固定关节 (Fixed Joint)	343
11.5.3 弹簧关节 (Spring Joint)	346
11.5.4 角色关节 (Character Joint)	352
11.6 总结	362
11.7 练习题	362

第 12 章 2D 物理模拟

12.1 2D 刚体 (Rigidbody 2D)	365
12.2 2D 碰撞盒 (Collider 2D)	366
12.3 2D 物理材质 (Physics Material 2D)	367
12.4 2D 关节 (2D Joints)	368
12.4.1 2D 距离关节 (Distance Joint 2D)	368
12.4.2 2D 弹簧关节 (Spring Joint 2D)	373
12.4.3 2D 固定关节 (Fixed Joint 2D)	375
12.4.4 2D 摩擦关节 (Friction Joint 2D)	377
12.4.5 2D 铰链关节 (Hinge Joint 2D)	378
12.4.6 2D 相对关节 (Relative Joint 2D)	381
12.4.7 2D 滑块关节 (Slider Joint 2D)	383
12.4.8 2D 目标关节 (Target Joint 2D)	386
12.4.9 2D 车轮关节 (Wheel Joint 2D)	387
12.5 2D 恒力 (Constant Force 2D)	391
12.6 2D 区域效应 (Area Effector 2D)	391
12.7 2D 浮力效应 (Buoyancy Effector 2D)	393
12.8 2D 点效应 (Point Effector 2D)	395
12.9 2D 平台效应 (Platform Effector 2D)	397
12.10 2D 表面效应 (Surface Effector 2D)	399
12.11 总结	400
12.12 练习题	401

第 13 章 动画系统

13.1 二足角色动画	406
13.1.1 二足角色资源的制作和导入	406
13.1.2 动画的导出与导入	408
13.1.3 使用 Mecanim 动画系统制作二足角色动画	408
13.1.4 状态机的融合树技术(Blend Tree)	420
13.1.5 状态机的动画层(Animation Layer)与身体蒙版(Body Mask)	427
13.1.6 动画复用(Retargeting)	431
13.1.7 反向运动学(IK)	431
13.2 不规则骨骼动画	434
13.3 无骨骼对象动画	439
13.4 2D 动画	447
13.4.1 帧动画	447
13.4.2 用拆分的方法制作 2D 动画	459
13.5 总结	462
13.6 练习题	462

第 14 章 粒子系统

14.1 粒子系统模块	465
14.1.1 粒子初始化(Initial)与发射器(Emission)模块	466
14.1.2 粒子发射器形状(Shape)模块	467
14.1.3 速度(Velocity)控制模块	468
14.1.4 根据生命周期控制粒子受力(Force Over LifeTime)模块	469
14.1.5 颜色(Color)控制模块	470
14.1.6 大小(Size)控制模块	470
14.1.7 旋转(Rotation)控制模块	471
14.1.8 外力(External Forces)控制模块	471
14.1.9 碰撞(Collision)设置模块	472
14.1.10 子粒子发射(Sub Emitters)模块	473
14.1.11 粒子贴图切片动画(Texture Sheet Animation)	473
14.1.12 粒子渲染(Renderer)控制模块	474
14.2 粒子属性编辑方式	475
14.2.1 传统参数控制方式	475
14.2.2 扩展参数控制方式	477
14.3 粒子系统范例	478
14.3.1 火焰	478

14.3.2	冰技能特效	483
14.3.3	烟花	501
14.4	使用代码控制粒子系统	519
14.5	总结	523
14.6	练习题	523

第 15 章 UGUI

15.1	UGUI 组件介绍	527
15.1.1	Canvas 相关组件	527
15.1.2	Rect Transform 组件	532
15.1.3	视觉(Visual)相关 UI 控件	534
15.1.4	UI 效果(Effect)相关组件	536
15.1.5	交互(Interaction)相关控件	537
15.2	UGUI 实例	544
15.2.1	创建开始界面	544
15.2.2	游戏菜单界面	546
15.2.3	角色面板和背包系统的背景	550
15.2.4	角色面板	551
15.2.5	创建背包系统	552
15.2.6	关卡选择界面	555
15.3	总结	562
15.4	练习题	562

第 16 章 人工智能

16.1	寻路功能	565
16.2	Nav Mesh Agent	572
16.2.1	Nav Mesh Agent 参数介绍	572
16.2.2	通过脚本修改参数	575
16.3	Off Mesh Link	579
16.3.1	自动添加分离网格链接	580
16.3.2	手动添加分离网格链接	581
16.4	Nav Mesh Obstacle	582
16.5	分层寻路	584
16.6	总结	588
16.7	练习题	588

01

CHAPTER ONE 第 1 章

游戏引擎介绍



本章内容

- 1.1 游戏引擎简介
- 1.2 游戏引擎的选择
- 1.3 目前流行的游戏引擎
- 1.4 Unity3D 游戏引擎的下载和安装
- 1.5 总结
- 1.6 练习题

过去的游戏开发,基本是从零开始一步一步地为游戏添加功能,直到游戏产品完成。这种从零开始的游戏框架搭建,有严重的“车轮再造”嫌疑。此种高成本、低效率的开发方式,已经不适合在当今高速发展、竞争白热化的游戏市场中立足了。

一个游戏团队,应从各个方面节约开发成本,提高开发效率和质量,才能在市场上生存并获得市场的认可,从而获得丰厚的利润。如何站在前人的肩膀上,复用已有的游戏开发技术,防止“车轮再造”所带来的弊端,进而把开发过程更多地集中在游戏内容和游戏可玩性上,是游戏开发人员急需解决的问题之一。

使用过 C/C++、Java 等程序语言创作过游戏的读者都深有体会,每一款游戏都会包括图像、声音、用户输入操作、游戏场景组织、动画控制、游戏逻辑等模块。当你创作了多款游戏之后你会发现,游戏内容可能不同,但是组成游戏的功能模块以及结构却基本差不多。那么是否有一种一劳永逸的办法,把这些基本的功能模块和结构抽象出来,以后在开发游戏时只要往这些功能模块中添加具体的游戏内容便可以了呢?答案是肯定的。“游戏引擎”便是为解决以上问题而诞生的。

1.1 游戏引擎简介

什么是游戏引擎?依据 Google Wiki 上的定义,游戏引擎是指一个可以用于创作和开发视频游戏的软件系统。游戏开发人员可以使用由游戏引擎提供的软件框架和所见即所得的游戏编辑系统来创作不同的游戏。游戏引擎为游戏设计者提供了包括图形图像渲染功能、物理模拟功能、碰撞检测功能、音频控制、程序脚本编写、动画系统、人工智能系统、网络系统、流处理、内存管理、线程管理等游戏开发所必要的功能。

根据百度百科的描述,我们可以把游戏的引擎比作赛车的引擎。引擎是赛车的核心,决定着赛车的性能和稳定性,赛车的速度、操纵感这些直接与车手相关的指标都是建立在引擎的基础上。游戏也是如此,玩家所体验到的剧情、关卡、美术、音乐、操作等内容都是由游戏的引擎直接控制的,它扮演着发动机的角色,把游戏中所有元素捆绑在一起,在后台指挥它们同时、有序地工作。

在游戏引擎面世之前,许多开发者都需要使用各种编程语言和各种开发库(如 OpenGL、DirectX)来搭建整个游戏底层框架,还要时时刻刻注意计算机内部资源的管理,因此没有太多的时间和精力来考虑游戏的创意和玩法。

一款完整的游戏引擎,能够提供给游戏开发者稳定的底层框架和完善的创作工具,使得开发者更加关心游戏的内容和可玩性,不用再去过多地担心游戏程序的底层细节,而且能够让开发出来的游戏产品更加稳定可靠。有了游戏引擎,游戏的开发过程更加简单,同时大大降低开发成本、缩短开发周期,进而降低研发的风险。出于以上原因,越来

越多的开发者倾向于使用健全的游戏引擎来制作自己的游戏。正因为有庞大的市场需求,游戏引擎的市场才逐渐形成。

简单地说,引擎就是“用于控制所有游戏功能的主程序,从计算图形图像、碰撞、物理系统和物体的相对位置,到接受玩家的输入,以及按照正确的音量输出声音等等。”无论是2D游戏还是3D游戏,无论是角色扮演游戏、即时策略游戏、冒险解谜游戏或是动作射击游戏,哪怕是一个只有1兆的小游戏,都有这样一段起控制作用的代码。

经过不断的进化,如今的游戏引擎已经发展为一套由多个子系统共同构成的复杂系统,从建模渲染、动画到光影、粒子特效,从物理系统、碰撞检测到文件管理、网络特性,还有专业的所见即所得编辑工具和插件,几乎涵盖了开发过程中的所有重要环节。

总而言之,游戏引擎是一种可复用的、适应性较强的游戏开发中间件,游戏开发人员可以使用同一套游戏引擎来开发出不同的游戏产品,甚至可以很方便地把同一款游戏产品发布到不同的运行平台上。

更多对游戏引擎的描述,可以参考百度百科(<http://baike.baidu.com/view/33343.htm>)和维基白科(http://en.wikipedia.org/wiki/Game_engine)的相关描述。

1.2 游戏引擎的选择

随着游戏开发市场的日益庞大,游戏引擎产品层出不穷。游戏引擎根据其使用范围,可分为公司内部使用与针对游戏开发市场两种。公司内部使用的游戏引擎一般不会发布到市面上,此类游戏引擎是由公司内部开发,并在内部使用,一般只有加入该公司的开发者才能使用到它。而另外一种则是发布到市场上的游戏引擎产品,要使用这类引擎,可以通过免费下载、付费授权等方式获得,具体根据不同引擎开发商的商业策略来定。

目前,面向游戏开发市场流行的游戏引擎有很多种。根据是否开源可以分为开源游戏引擎与非开源引擎;根据是否付费可以分为付费引擎与免费引擎;根据画面空间维数可分为2D游戏引擎和3D游戏引擎;根据针对的游戏类型划分,可分为针对RPG游戏的引擎、针对第一人称射击的引擎、针对第三人称射击游戏的引擎等等。当然,随着游戏引擎市场的发展,其分类也越来越多,界限却越来越模糊。目前,使用比较多的游戏引擎有:Unreal 虚幻引擎、Cry Engine、寒霜、Unity3D、Cocos2D 等等(更多的游戏引擎划分可以参考维基百科的“游戏引擎分类”)。

在琳琅满目的游戏引擎产品中,挑选合适的游戏引擎可以使开发事半功倍。那么如何挑选合适的引擎,此处给出一些建议,以供读者参考。

在考虑使用什么游戏引擎时,需要考察以下几个方面的内容:

- 开发的是什么类型的游戏。每种不同游戏引擎产品,对游戏类型都有所侧重。例如 Unreal 虚幻引擎和 Cry Engine 比较擅长开发第一人称和第三人称的次时代游戏,Game Maker 用于2D的RPG游戏制作,Big World 引擎适合制作大规模大地图的网络多人游戏。
- 开发的游戏成本预算大概是多少。针对目前游戏引擎市场的价格不一,有的引擎授权需要几十万元美金,甚至百万元美金以上,而有的只要几百元美金,还有的是

免费的。当然,在游戏引擎市场中,价格的高低很大程度上影响了该引擎的质量、效率、功能和技术支持等方面的内容。

- 开发针对何种平台上的游戏。例如 Cocos2D 适合开发智能手机上的游戏,Unity3D 适合开发多平台的 3D 和 2D 的游戏等等。
- 引擎所提供的功能。一个游戏引擎提供的功能越多,越完善,我们能够实现的效果就越丰富,就有越多的想法能够表达出来。比如是否提供次时代画质、是否提供物理模拟、是否拥有网络开发功能等等。
- 帮助文档是否完整,论坛是否活跃,技术支持是否及时准确。这些是使用游戏引擎的重要技术保障。如果一个游戏引擎的帮助文档不完整,技术支持不够健全,建议不要使用。
- 是否具有扩展性。一些游戏引擎是完全封闭的,为它添加功能是非常困难的事情;有的游戏引擎是提供源代码的,虽然这样自由度非常高,能修改引擎的源代码,但是对于一个游戏开发团队来说,维护代码是一件非常麻烦的事情;如果游戏引擎能够提供一种接口,使得用户能够根据自己的需要编写功能插件,那么是最好的事情了。
- 是否简单易用,界面是否友好。有的游戏引擎虽然提供了非常出色的画面效果和效率管理,但是入门难、界面复杂等问题使得很多用户望而却步,同时也降低了开发效率,所以在选择的时候需要衡量。
- 了解该引擎的“坑”。“坑”是游戏开发界中的行话,指的是在使用某款游戏引擎时可能会遇到的技术难题或者是该引擎潜在的缺点。由于目前的软件框架都存在稳中有降自的优缺点和适用领域,因此引擎带来的“坑”是不可避免的。面对这些“坑”,开发者只能选择“填坑”(修复或解决问题)或者“绕道”(利用其他途径实现功能)的方法。了解它可以为我们选择合适的引擎提供更多的参考。

1.3 目前流行的游戏引擎

列举几款流行的游戏引擎,以供读者参考。

1.3.1 虚幻引擎(Unreal4)

读者如果玩过《战争机器》《虚幻竞技场》《细胞分裂》等重量级次时代游戏,定会被它们壮观的游戏场景和华丽的画面所吸引,如图 1-1 所示。



(a)《战争机器》



(b)《虚幻竞技场》



(c)《细胞分裂》



(d) Unreal Logo

图 1-1 Unreal Logo 以及 Unreal 引擎开发的游戏

这些游戏都是由全球顶级游戏 Epic 公司开发的 Unreal 游戏引擎制作出来的。目前已经发布的版本为 Unreal4。

Unreal4 又名虚幻引擎 4,是一套针对 PC、Xbox、PlayStation、IOS、Android、Oculus 平台的完整的游戏开发框架。它的所有功能编写理念都是为了更加容易的内容制作和编程开发,为了让所有的美术开发人员能够在牵扯到最少程序开发内容的情况下使用抽象程序助手来自由创建虚拟场景,以及提供程序等高效率的模块和可扩展的开发构架来创建、测试和完成各种类型的游戏制作。

虚幻引擎 4 目前已经完全免费,同时包括了整个引擎的源代码。如果需要深入了解虚幻引擎,可以登录官网网址:<http://www.unrealengine.com/>。

1.3.2 Cry Engine

科幻题材的第一人称射击游戏《孤岛危机》系列,是 Cry Engine 的代表作之一,它以细腻的画面、逼真的场景和高效的物理模拟等效果征服了很多玩家。该游戏是由 Crytek 自主研发的 Cry Engine 游戏引擎制作而成,如图 1-2 所示。该引擎目前已经发布到 Cry Engine V,也是以免费的方式发布。官网网址:<https://www.cryengine.com/>。



(a)《孤岛危机》



(b)Cry Engine Logo

图 1-2 Cry Engine Logo 以及由 Cry Engine 开发的游戏

1.3.3 寒霜引擎(Frosbite Engine)

寒霜引擎是瑞典 DICE 游戏工作室为著名电子游戏产品《战地》系列设计的一款 3D 游戏引擎。该引擎从 2006 年起开始研发,第一款使用寒霜引擎的游戏在 2008 年问世。寒霜引擎的特色是可以运作庞大而又有着丰富细节的游戏地图,同时可以利用较低的系统资源渲染地面、建筑、杂物的全破坏效果。使用寒霜引擎可以轻松地运行大规模的、所有物体都可被破坏的游戏,如图 1-3 所示。官网网址:<http://store.dice.se/>。