

虚幻技术教育解决方案推荐用书

SAMS



Mastering Unreal Technology, Volume II:  
Advanced Level Design Concepts with Unreal Engine 3

# 精通Unreal Engine 3卷II: 3D游戏关卡设计高级篇

(美) Jason Busby Zak Parrish Jeff Wilson 著  
武 侠 孙德元 张青春 张美蓉 译



POWERED BY



UNREAL  
TECHNOLOGY

清华大学出版社

# 精通 Unreal Engine 3 卷 II： 3D 游戏关卡设计高级篇

Jason Busby

(美) Zak Parrish 著

Jeff Wilson

武 侠 孙德元  
张青春 张美蓉 译

清华大学出版社

北 京

Authorized translation from the English language edition, entitled Mastering Unreal Technology, Volume II: Advanced Level Design Concepts with Unreal Engine 3, 1st Edition, 978-0-672-32992-0 by Busby, Jason; Parrish, Zak; Wilson, Jeff, published by Pearson Education, Inc, publishing as Sams Publishing, Copyright © 2010.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc. CHINESE SIMPLIFIED language edition published by Tsinghua University Press, Copyright © 2011.

北京市版权局著作权合同登记号 图字: 01-2010-1111

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

#### 图书在版编目(CIP)数据

精通 Unreal Engine 3 卷II: 3D 游戏关卡设计高级篇/(美) 巴斯比(Busby, J.), (美) 帕里什(Parrish, Z.), (美)威尔逊(Wilson, J.) 著; 武侠, 孙德元, 张青春, 张美蓉 译.—北京: 清华大学出版社, 2011.10

书名原文: Mastering Unreal Technology, Volume II: Advanced Level Design Concepts with Unreal Engine 3

ISBN 978-7-302-26040-0

I. 精… II. ①巴… ②帕… ③威… ④武… ⑤孙… ⑥张… ⑦张… III. ①多媒体—软件工具, Unreal Engine 3 ②游戏—应用程序—程序设计 IV. ①TP311.56 ②G899

中国版本图书馆 CIP 数据核字(2011)第 133064 号

责任编辑: 王 军 王滋润 吴 乐

装帧设计: 牛艳敏

责任校对: 胡雁翎

责任印制: 李红英

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 52.5 字 数: 1409 千字

附 DVD 光盘 1 张

版 次: 2011 年 10 月第 1 版

印 次: 2011 年 10 月第 1 次印刷

印 数: 1~4000

定 价: 98.00 元

---

产品编号: 036203-01

# 中文版序

屈指算来，虚幻引擎技术正式进入中国已经有 5 年时间了。在这 5 年里，国内外的游戏行业都发生了巨大的变化。在国际市场上，整个行业经历了从次世代主机的兴起，到社交游戏和移动终端游戏成为热点的过程。而国内市场则从 2D、2.5D MMO 一统天下，发展到今天各种多样化的 3D 网游受到玩家普遍欢迎，当然还有网页游戏令人瞩目的兴起。伴随着游戏行业的变化，虚幻引擎自身也在迅速地进化。新的全局光照系统、适合制作大型网游的地形系统、更直观便利的关卡编辑器、资源管理器、玩家行为分析系统等，都在这短短几年里整合到了虚幻引擎内。这些新技术与工具也吸引了越来越多的网游开发商加入了虚幻引擎使用者的队伍，第一批虚幻引擎 3 网游大作将会在今明两年陆续上市。在支持中小开发人员和新兴游戏平台方面，虚幻引擎也从不保留。自 2009 年 11 月推出免费版的 UDK 以来，UDK 的独立安装量已经突破 80 万。而且虚幻引擎更是从 2010 年底开始全面支持苹果和安卓等移动设备系统。现在无论在什么平台上，都可以看到众多独立游戏开发人员使用虚幻技术创作的出色项目。

对于这一套虚幻引擎学习书籍，我们非常希望它能给国内的游戏开发人员和爱好者带来必要的帮助，让大家可以顺利地运用虚幻引擎和 UDK 这样国际顶级的创作工具。我们不但希望看到国内有更多采用虚幻技术开发的优秀作品涌现出来，更希望这些作品能获得玩家的喜爱和认可。

这套专业技术书籍的翻译和编辑是一个漫长和艰巨的过程，在这里也衷心地感谢参与其中的英佩游戏、GA 游戏教育和清华大学出版社的所有同仁和朋友。



英佩游戏 副总裁  
2011 年 6 月 3 日

# 作者简介

**Jason “Buzz” Busby** 是 3D Buzz 公司的主席兼首席执行官，3D Buzz 公司是一家面向全球的、致力于为当今最热门的三维设计领域提供教学服务的公司。这一领域包括三维动画、程序和游戏开发等。Buzz 通过他的网站 [www.3dbuzz.com](http://www.3dbuzz.com) 向全球提供其独具特色的教学内容，即视频教学模块 (Video Training Modules, VTM)，该模块非常专业，信息丰富且寓教于乐。他的网站也在三维动画方面拥有一个健全、友好且实用的在线社区。

**Zak Parrish** 是 3D Buzz 公司的核心讲师。在过去的 6 年中，他与 Jason Busby 和 3D Buzz 公司协同工作，向全球学院提供顶级质量的教学服务。他的成果包括大量的视频教学内容，其中就包括为游戏《虚幻竞技场 2004》和《虚幻竞技场 3》专门制作的教学视频系列。在非常有限的业余时间里，他还就读于 Austin Peay 州立大学，志在获得美术学士学位。

**Jeff Wilson** 是 Advancing Technology 大学的一名学生，他在那里获得了游戏设计学位。他已经研究了 4 年的虚幻引擎技术，同时使用《虚幻竞技场 2003》和《虚幻竞技场 2004》制作了一系列的 mod(mod 是英文单词 modification 的缩写，它是游戏的一种修改或增强程序)。在自学期间，Jeff 运用他在程序和三维美术方面的知识全面掌握了虚幻引擎的内部工作原理，可以创建出类拔萃的游戏应用程序。他已经成为了 3D Buzz 的专业技术顾问，协助制作虚幻引擎技术的教学视频。

# 目 录

第 1 章 高级关卡设计艺术概述	1	3.4.3 GravityVolume	130
1.1 本书读者对象	1	3.4.4 UTKillZVolume	130
1.2 学习前的准备	1	3.4.5 UTScriptedBotVolume	130
1.3 “高级”关卡设计的含义	2	3.4.6 WaterVolume	131
1.4 包含的资源和游戏资源	3	3.4.7 UTWaterVolume	131
1.5 操作 INI 配置文件	3	3.4.8 UTLavaVolume	133
1.6 未来的软件修改	4	3.4.9 UTSlimeVolume	133
第 2 章 高级材质构造	5	3.4.10 UTSpaceVolume	133
2.1 材质实例化和参数	5	3.4.11 LadderVolume	134
2.2 混合	6	3.5 TriggerVolume	136
2.3 UV 操作	7	3.6 PostProcessVolume	148
2.4 法线	8	3.7 LevelStreamingVolume	149
2.5 深度	9	3.8 LightVolume	150
2.6 特殊光照	9	3.9 ColorScaleVolume	150
2.7 反射	10	3.10 ReverbVolume	150
2.8 扭曲	11	3.11 RB_ForceFieldExclude-	
2.9 透明	11	Volume	151
2.10 子表面散射	11	3.12 UTAreaNaming-	
2.11 自定义光照模型	12	Volume	151
2.12 总结	119	3.13 UTAutoCrouchVolume	151
第 3 章 体积的使用	121	3.14 FoliageFactory	151
3.1 可用体积类型	122	3.15 总结	152
3.2 基本的体积属性	123	第 4 章 物理对象	153
3.3 BlockingVolume	124	4.1 物理对象的类型	153
3.3.1 DynamicBlockingVolume	124	4.2 刚体的概念	155
3.3.2 LeviathanBlockingVolume	126	4.3 在 Unreal 中使用刚体	156
3.3.3 TankBlockingVolume	126	4.3.1 碰撞属性	156
3.4 PhysicsVolumes	127	4.3.2 RB_BodySetup	157
3.4.1 DynamicPhysicsVolume	130	4.4 KActors	159
3.4.2 ForcedDirVolume	130	4.5 KAssets	165
		4.6 约束	167

4.6.1	Constraint Actor 的类型	168	5.2.4	流体	230
4.6.2	RB_ConstraintActor	169	5.3	模块	230
4.6.3	Pulley	170	5.4	Cascade 粒子系统编辑器	236
4.6.4	RB_ConstraintSetup	170	5.4.1	Cascade 界面	236
4.6.5	Linear	171	5.4.2	工具栏	236
4.6.6	Angular	172	5.4.3	Preview 面板	238
4.6.7	Pulley	173	5.4.4	Emitter List 面板	239
4.6.8	RB_ConstraintInstance	173	5.4.5	Properties 面板	240
4.6.9	Linear	173	5.4.6	Curve Editor 面板	241
4.6.10	Angular	174	5.5	粒子专用材质表达式	242
4.6.11	可放置的约束类型	174	5.5.1	Vertex Color	242
4.7	物理资源工具(PhAT)	183	5.5.2	ParticleSubUV	243
4.8	布料	195	5.5.3	MeshSubUV	244
4.8.1	金属布料	196	5.6	粒子模块的类型	252
4.8.2	布料属性	196	5.6.1	加速模块	252
4.9	冲力和力 Actor	202	5.6.2	引力模块	253
4.9.1	RB_LineImpulseActor	202	5.6.3	光束模块	256
4.9.2	RB_RadialImpulseActor	202	5.6.4	颜色模块	258
4.9.3	RB_Thruster	203	5.6.5	销毁模块	259
4.9.4	RB_RadialForceActor	203	5.6.6	生命周期模块	260
4.9.5	RB_CylindricalForceActor	203	5.6.7	位置模块	261
4.10	针对于物理的 Kismet		5.6.8	环绕模块	264
	序列对象	207	5.6.9	朝向模块	266
4.10.1	物理事件	207	5.6.10	旋转模块	266
4.10.2	物理动作	207	5.6.11	旋转速率模块	267
4.11	物理材质	210	5.6.12	尺寸模块	268
4.12	总结	215	5.6.13	产生模块	269
5.6.14			5.6.14	SubUV 模块	270
5.6.15			5.6.15	尾迹模块	271
5.6.16			5.6.16	速度模块	271
5.7	光束发射器	285	5.7	光束发射器	285
5.7.1	Beam TypeData 模块	285	5.7.1	Beam TypeData 模块	285
5.7.2	光束发射器模块	288	5.7.2	光束发射器模块	288
5.8	网格物体发射器	299	5.8	网格物体发射器	299
5.9	尾迹发射器	306	5.9	尾迹发射器	306
5.9.1	Trail TypeData 模块	306	5.9.1	Trail TypeData 模块	306
5.9.2	Trail 模块	307	5.9.2	Trail 模块	307
5.10	流体发射器	314	5.10	流体发射器	314
5.11	细节层次	317	5.11	细节层次	317
<b>第 5 章 使用粒子系统创建特效</b>			<b>217</b>		
5.1	粒子系统的剖析	218			
5.1.1	粒子系统属性	219			
5.1.2	粒子平面实例发射器	224			
5.1.3	平面实例发射器属性	225			
5.1.4	RequiredModule	225			
5.1.5	粒子	228			
5.2	TypeData 模块	228			
5.2.1	光束	228			
5.2.2	网格物体	229			
5.2.3	尾迹	230			



5.12	发射器 actor	324	6.6.3	移动控件	401
5.12.1	LOD	324	6.6.4	旋转控件	402
5.12.2	粒子系统组件	324	6.6.5	调节控件大小	404
5.13	在游戏中修改粒子系统	325	6.6.6	位置编辑器	404
5.13.1	Kismet 修改	325	6.6.7	停靠	406
5.13.2	Matinee 修改	326	6.7	控件状态	410
5.14	特效制作疑难解答	386	6.8	控件导航和焦点链	432
5.15	总结	386	6.9	使用用户输入	433
<b>第 6 章</b>	<b>创建用户界面</b>	<b>387</b>	6.10	数据仓库	437
6.1	用户界面组件	387	6.11	风格和皮肤	438
6.2	用户界面和 HUD	388	6.11.1	风格	438
6.3	用户界面设计 workflow 概述	388	6.11.2	皮肤	441
6.3.1	布局	388	6.11.3	风格覆盖	449
6.3.2	功能	388	6.12	字体	450
6.3.3	实现	388	6.13	添加功能	452
6.4	用户界面页面编辑器	393	6.14	声音	476
6.4.1	页面视口	394	6.15	创建 HUD	478
6.4.2	页面工具条	394	6.16	总结	479
6.4.3	页面 Properties 面板	394	<b>第 7 章</b>	<b>声音系统</b>	<b>481</b>
6.4.4	页面 Positioning 面板	394	7.1	声音类型	481
6.4.5	页面 Docking 面板	394	7.2	USounds 和 SoundCues	482
6.4.6	Scene Tools 面板	394	7.3	AmbientSound Actors	488
6.5	用户界面控件	397	7.3.1	AmbientSound	488
6.5.1	Button 控件	397	7.3.2	AmbientSoundSimple	488
6.5.2	Checkbox 控件	397	7.3.3	AmbientSoundNonLoop	489
6.5.3	Editbox 控件	397	7.3.4	mbientSoundSimple- Toggleable	489
6.5.4	Image 控件	397	7.4	SoundCue Editor	489
6.5.5	Label 控件	397	7.5	SoundCue 节点	490
6.5.6	Label Button 控件	398	7.5.1	SoundNodeWave 节点	490
6.5.7	List 控件	398	7.5.2	Attenuation 节点	490
6.5.8	Numeric EditBox 控件	398	7.5.3	Concatenator 节点	491
6.5.9	Panel 控件	398	7.5.4	Delay 节点	492
6.5.10	Progressbar 控件	398	7.5.5	Distance CrossFade 节点	492
6.5.11	Slider 控件	398	7.5.6	Looping 节点	494
6.5.12	ToggleButton 控件	398	7.5.7	SoundNodeMature 节点	494
6.6	使用控件	398	7.5.8	Mixer 节点	495
6.6.1	创建控件	398	7.5.9	Modulator 节点	495
6.6.2	父层级	400			



7.5.10	Continuous Modulator 节点	496
7.5.11	Oscillator 节点	496
7.5.12	Random 节点	497
7.5.13	SoundNodeWaveParam 节点	497
7.6	Kismet 声效动作	506
7.6.1	Play Sound	506
7.6.2	Apply Sound Node	507
7.7	使用音乐	511
7.7.1	MusicTrack	511
7.7.2	MusicTrackBank	512
7.7.3	CrossFadeMusicTracks	512
7.7.4	Set Music Track	513
7.7.5	UT Map Music	514
7.8	使用 MusicTrackBank	515
7.9	总结	533
<b>第 8 章</b>	<b>后期处理特效</b>	<b>535</b>
8.1	控制后期处理的 4 个主要 方法	536
8.2	访问及分配 PostProcess- Effect 对象	537
8.3	后期处理特效的类型	541
8.3.1	光溢出	541
8.3.2	运动模糊	541
8.3.3	景深	542
8.3.4	材质	544
8.3.5	场景效果	544
8.4	后期处理编辑器	546
8.4.1	后期处理特效模块	547
8.4.2	通用模块属性	549
8.5	控制游戏中的后期处理 特效	573
8.5.1	Kismet	573
8.5.2	Matinee	573
8.5.3	PostProcessVolume	574
8.6	总结	574
<b>第 9 章</b>	<b>虚幻引擎中的动画系统</b>	<b>575</b>
9.1	虚幻引擎中的动画系统	576
9.2	骨架动画	580
9.2.1	骨架网格物体	581
9.2.2	小块网格和大段网格	581
9.2.3	细节层次	584
9.2.4	插槽	587
9.2.5	动画集	588
9.3	顶点变形动画	588
9.4	动画集编辑器	591
9.4.1	菜单栏	591
9.4.2	工具栏	592
9.4.3	浏览器	592
9.4.4	预览	593
9.4.5	Properties 区域	594
9.4.6	插槽管理器	597
9.4.7	动画压缩	600
9.4.8	动画压缩类型	600
9.5	动画混合	602
9.6	动画树编辑器	603
9.6.1	菜单	604
9.6.2	工具栏	604
9.6.3	Preview 窗口	604
9.6.4	Properties 窗口	605
9.6.5	工作区	605
9.7	动画树编辑器节点	615
9.7.1	AnimTreeNode 节点	615
9.7.2	AnimTreeNode 输入端	616
9.7.3	AnimTreeNode 属性	616
9.7.4	Animation 节点	616
9.7.5	Animation 节点的 一般属性	617
9.7.6	AnimNodeAimOffset 节点	617
9.7.7	AimOffset Editor	617
9.7.8	AnimNodeBlend 节点	618
9.7.9	AnimNodeBlendBy- Base 节点	618
9.7.10	AnimNodeBlendBy- Physics 节点	619

9.7.11 AnimNodeBlendBy- Posture 节点	619	9.8.6 SkelControlFoot- Placement 节点	638
9.7.12 AnimNodeBlendBy- Speed 节点	620	9.8.7 SkelControlLookAt 节点	639
9.7.13 AnimNodeBlend- Directional 节点	620	9.8.8 SkelControlSingleBone 节点	640
9.7.14 AnimNodeBlendList 节点	620	9.8.9 SkelControlSpline 节点	641
9.7.15 AnimNodeBlend- MultiBone 节点	621	9.8.10 SkelControlTrail 节点	642
9.7.16 AnimNodeBlendPerBone 节点	621	9.8.11 SkelControlWheel 节点	642
9.7.17 AnimNodeCrossfader 节点	621	9.9 用 FaceFX 制作面部动画	669
9.7.18 AnimNodeMirror 节点	621	9.10 FaceFX Studio	670
9.7.19 AnimNodePlayCustom- Anim 节点	621	9.10.1 音素	671
9.7.20 AnimNodeRandom 节点	622	9.10.2 节点	676
9.7.21 AnimNodeScalePlayRate 节点	622	9.10.3 节点类型	676
9.7.22 AnimNodeScaleRateBy- Speed 节点	622	9.10.4 一般属性	676
9.7.23 AnimNodeSequence- BlendByAim 节点	622	9.10.5 FxBonePoseNode 节点	676
9.7.24 AnimNodeSlot 节点	623	9.10.6 FxCombinerNode 节点	677
9.7.25 AnimNodeSynch 节点	623	9.10.7 FxCurrentTimeNode 节点	677
9.7.26 AnimSequencePlayer 节点	623	9.10.8 FxDeltaNode 节点	677
9.7.27 AnimNodeSequence 节点	624	9.10.9 FUnrealFaceFXMaterial- ParameterNode 节点	678
9.7.28 组	624	9.10.10 FUnrealFaceFXMorph- Node 节点	678
9.7.29 显示	625	9.10.11 链接	678
9.8 顶点变形节点	635	9.10.12 曲线	692
9.8.1 顶点变形节点的一般 属性	635	9.11 FaceFX Studio 用户界面	697
9.8.2 MorphPose 节点	635	9.11.1 菜单栏	697
9.8.3 MorphNodeWeight 节点	636	9.11.2 工具栏	698
9.8.4 Skeletal Controller 节点	636	9.11.3 Actor 面板	698
9.8.5 SkelControlLimb 节点	637	9.11.4 标签栏	701
		9.12 物理动画	718
		9.13 物理动画的创建过程	723
		9.14 骨架网格物体 Actor	723
		9.15 骨架网格物体组件	723
		9.16 总结	724
		第 10 章 创建过场动画	725
		10.1 过场动画	725
		10.1.1 移动 actor	725

10.1.2	移动相机 .....	726
10.1.3	镜头切换 .....	727
10.1.4	游戏中渲染与预渲染 .....	727
10.2	相机 Actor .....	728
10.3	相机效果 .....	729
10.3.1	景深 .....	729
10.3.2	运动模糊 .....	730
10.3.3	视野 .....	730
10.3.4	屏面特效 .....	731

10.4	利用光照进行图像分离 .....	731
10.5	DumpMovie 命令 .....	732
10.6	小结 .....	732
10.7	总结 .....	797
附录 A	分布 .....	799
附录 B	曲线编辑器 .....	815

# 第 1 章



## 高级关卡设计艺术概述

欢迎阅读 Unreal Engine 3 高级关卡设计艺术！无论您是阅读过前两卷书籍、创建过自己的游戏及关卡，还是仅玩过由 Unreal Engine 3 构建的游戏，或许您都会意识到这款游戏引擎的强大功能。您已经见识过由 Unreal Engine 3 创建的具有令玩家震惊的美妙环境的广阔天地。您也已经了解了 Unreal Editor 赋予游戏美术工作人员的强大功能。

现在，是时候提高技术水平了，不仅是提高一点点，而是使 Unreal Engine 3 技术的功能发挥到极限，从而使创建的游戏性体验更加令人难以置信。

如果想要深深地震撼玩家，您就要掌握一些高难度技术：粒子系统、自定义动画、可以进行物理反应的动态对象、高级材质等。所有这些技术都内置在 Unreal Engine 3 中——接下来将向您展示如何运用它们。不仅仅只是做个概述，而是尽可能完整地进行介绍。

### 1.1 本书读者对象

首先澄清一下，如您所见，本书不是面向初学者的。

现在，如果您还是初学者或者仅对引擎有甚少的了解和经验，这里申明一点：我们绝对没有抛弃您。学习能力强的人应该能跟上我们的脚步：仅需要慢慢地、专心地花时间学习。本书是按照易跟学的、循序渐进的方式编写的，所以任何经验层次的用户都可以使用本书——这是根据多年培训虚幻引擎开发人员的经验。

如果您已经熟悉了 Unreal Editor，并且编写了一些基本的虚幻程序，那么您的阅读速度一定可以更快。

### 1.2 学习前的准备

本书将使用游戏《虚幻竞技场 3》作为演示示例。

尽管这里教授的许多概念也同样适用于其他的基于 Unreal Engine 3 的游戏，这些游戏也提供了到 Unreal Editor 的访问。但是，大部分的指南都是基于《虚幻竞技场 3》的游戏资源。另外，其他游戏有时会修改引擎和编辑器，这会使它们的工作方式变得不同，或者至少让您混淆一些概念，这不是我们所希望的！

如前所述，我们将始终围绕指南展开教学，所以您最好至少对 Unreal Editor 的关键概念和操作有一些基本的认识。最理想的情况是，您已经对虚幻引擎有如下这些基本的认识和研究：

- 导航透视图与正交视图。
- 使用 BSP 画刷创建添加型环境。
- 在 Generic 浏览器中导航及创建资源。
- 在 Generic 浏览器中加载及卸载资源包。
- 把 Generic 浏览器和 Actor Classes 浏览器中的对象放置到关卡中。
- 在 Actor Properties 窗口中调整 actor 属性。
- 建立基本的 Kismet 序列。
- 在 Material Editor(材质编辑器)中创建基本的材质。
- 使用 Matinee 制作简单的关键帧动画。

您已经掌握所有这些内容了吗？请不要慌张，就像我们一直所说的，您将能跟上我们的脚步，慢慢来。但是如果您已经很熟悉 Unreal Engine 3 和 Unreal Editor，就会发现学习过程是非常地顺利并快速。为了达到这个目的，您应该找到以下的培训资源：

- 《精通 Unreal Engine 3 卷 I：3D 游戏关卡设计基础篇》——清华大学出版社引进并出版的这个系列的第 I 卷，这本书将会使您开始认识并初步掌握 Unreal Editor。这本书特别适用于所有刚涉及 Unreal Engine 3 游戏 MOD 制作的人员。
- 《虚幻竞技场 3: 珍藏版》——《虚幻竞技场 3》珍藏版 DVD 中包含了数小时的 Unreal Editor 培训——包括专门为初学者准备的一些视频。
- [www.3dbuzz.com](http://www.3dbuzz.com)——这是 3D Buzz 公司的网站，发布包括虚幻引擎技术、编程、3D 图形等专业的视频培训内容。现在就可以通过观看 Video Training Modules 进行入门学习，或者查看论坛来获得大量有价值的信息。并且，因为这个网站是由本书的作者管理并运行的，所以这是与作者进行交流的一种非常好的方式。

## 1.3 “高级”关卡设计的含义

Unreal Editor 把大量的工具集和有用的功能汇集到一起。其中有些工具非常简单直观并且这些年来都没怎么修改过。而另外一些工具在 Unreal Engine 3 中发生了巨大的改变，为我们带来了比以往更强大的功能和灵活性。还有一些全新的游戏编辑功能：它们第一次把高端的影片和特效带到了游戏制作中。

本书主要集中讲解了 Unreal Engine 3 中的全新工具和已经进行了很大改进的工具。但也会向您展示一些常用工具(如 Material Editor)的突破性技术，从而使您可以比其他正在使用此工具的 99% 人员创建出更强大的资源(现在是一个充满竞争的世界，技术达到平均水平已经不足以成功了)。

## 1.4 包含的资源 and 游戏资源

除了卓越的天赋和个性外，您还需要大量其他的资源帮助您完成本书的指南。比如类似于纹理、声音、图像的资源。

幸运的是，您将会在本书配套的 DVD 中找到所需的大部分资源。这些资源存放在每一章对应的文件夹中。

所需的其他资源是随同《虚幻竞技场 3》游戏安装的。您将会在包(UPK 文件)中找到它们。在默认安装后，可以在如下路径的安装文件夹中找到这些包：

C:\Program Files\UT3\UTGame\CookedPC

如果您已经把游戏安装在不同的磁盘或文件夹，那么起始路径将是不同的，但是一旦您进入 UT3 文件夹，接下来的文件夹结构就与给出的路径相同了。

在 CookedPC 文件夹中有很多子文件夹，每一个子文件夹都包含许多包。您最需要注意的子文件夹如下：

- ▶ **Environments**——此子文件夹包含一些用来构建《虚幻竞技场 3》关卡的资源，如静态网格物体、纹理和材质。这些包按主题进行了分类存放，这样使得您能够基于想要创建的关卡，方便地查找所需的资源。
- ▶ **Characters**——需要使用 UT3 角色创建一个过场动画序列？这些角色资源就位于此子文件夹中，每种类型的角色被放置于不同的包里。
- ▶ **Maps**——需要修改现有的地图？想要知道《虚幻竞技场 3》中内置的关卡地图的某个效果是如何构建的？这些地图就在此子文件夹中。
- ▶ **Effects**——这个子文件夹提供了创建如下特效所需的资源：粒子特效(爆炸，火焰等)、镜头眩光、车辆和武器特效，以及其他的一些特效。

### 注意

虚幻引擎的文件夹及包结构设计不仅是为了保持简单的操作，还有一个同等重要的原因是内存使用。当加载一个包时，它的所有内容都存放在内存中。因为不可能有无限的内存，所以您不想加载不必要的包。文件夹和包结构使得您可以仅加载现在需要的资源(加载多少包才算过多呢？这取决于您的系统空间。但是如果您察觉到运行速度减慢或者偶尔不稳定的情况，那么有可能是内存中载入了过多的包，这时需要卸载一些包)。

## 1.5 操作 INI 配置文件

本书中的一些指南会需要您更改《虚幻竞技场 3》的 INI 配置文件以在实际游戏中查看效果。当出现这种情况时，我们将准确地告诉您更改哪个文件及如何操作。

这些 INI 文件在您本地 Documents(文档)文件夹中。如果在 Windows 系统上执行了《虚幻竞技场 3》的默认安装，那么路径如下所示。

Documents\My Games\Unreal Tournament 3\UTGame\Config\

和注册表键值不同，删除对 INI 配置文件的修改非常简单。只需要删除更改的 INI 文件。当下次启动《虚幻竞技场 3》或者 Unreal Editor 时，它会自动创建一个使用默认设置的新 INI 文件。

顺便说一下，当创建了一个完整的《虚幻竞技场 3》MOD 时，玩家可以包含仅针对他们的 mod 的 INI 文件，这样，其他玩家也可以欣赏到他们的创作。

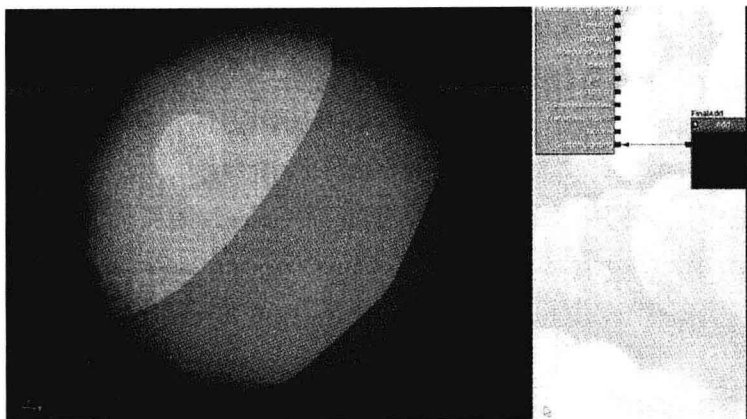
## 1.6 未来的软件修改

最后，在此声明：我们已经尽所能来保证在编写本书时，其指南和概念是最新、最精确的。它反映了所有已知的用于编写精确的、稳定的 Unreal 代码所需要的一切知识。我们已经做得非常认真、仔细，但是我们不能预测未来。总之：如果有一天，某些补丁或者软件更新改变了本书指南中的工作方式，那么这不是我们所能控制的。

接下来，让我们充满热情地开始学习吧。



## 第 2 章



# 高级材质构造

下面的几个教程将会探讨一些特效，这些特效可以通过创建更复杂的表达式网络或通过使用参数创建可自定义的材质来实现。这正是 Unreal Engine 3 中提供的材质系统功能的真正表现突出之处。通过学习本章，您就可以创建一些材质，同时将这些特效和法则付诸实践。创建的许多材质将会使用前面提及的几种方法以达到预期的结果。

### 2.1 材质实例化和参数

Unreal Engine 3 的材质系统中作用最大的部分可能是使用材质内部参数的功能，这些参数可以在运行时进行修改，或结合材质实例化进行使用。在运行时修改参数，可以根据游戏内置事件控制材质的外观或制作动画效果。可以使用材质实例化通过单一模板创建多种不同的材质，这样在需要很多具有相似基础结构或目的的材质时，可以节省大量时间。由于材质实例可以根据实时的更改进行更新，因此这使美工人员的最后调整工作变得异常简单。

通常在需要使用某种材质的几种不同颜色的变化时使用材质实例化。可以通过使用颜色的某个参数创建一种材质并将其实例化，而不是创建多种材质。也可以使用这项技术为游戏中的角色处理材质，不过使用方法更加复杂。创建一个包含节点网络的基础材质，其中节点网络可以用于游戏中的所有角色(或者至少是大多数角色)。使用与属性和纹理相关的参数，这样才可以调整这些值，或者在每个实例(或者是每个角色，视情况而定)的基础上替换出的纹理。我们可以更进一步深化使用实例化的方法，因为可以为整个游戏创建少量的基础材质，根据它们得出大部分其他材质。除了可以更加有效地为不计其数的网格物体创建材质外，还可以使游戏保持非常统一的外观和感觉。

如前所述，参数可以被用于使材质响应游戏中的事件。这些参数是动态的，可以通过 Kismet、Matinee 或使用代码对这些参数进行修改。关闭或开启材质的发光区域、替换纹理或控制纹理坐标平移或平铺等效果，都可以通过使用参数实现。

另一类参数是静态参数，仅在材质实例化时使用，因为它们不能在运行时修改，所以得名静态。这些名称中带有 Static 前缀的参数可以在 UnrealEd 的实例中进行修改，但是只要游戏开始，它们就会被设为固定值。这两个参数(即 StaticComponentMaskParameter 和 StaticSwitchParameter)允许遮挡纹理的某些特定通道或者可能遮挡材质内的整个网络。在本章后续部分构造毛皮着色器中应用这项技术，如图 2-1 所示。

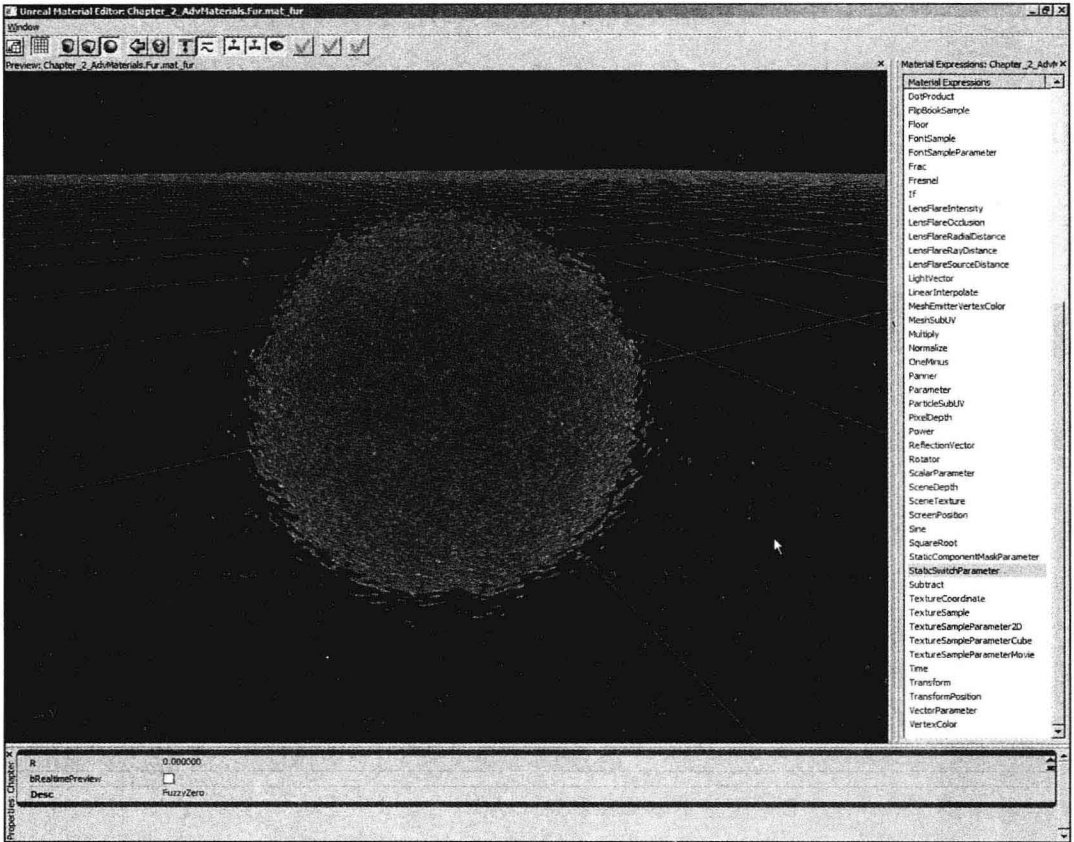


图 2-1 在本章后续内容中将创建一个这样的毛皮着色器

## 2.2 混合

通过使用 LinearInterpolate 表达式或纹理蒙版，材质可以在不同的值、颜色或纹理之间进行混合，如图 2-2 所示。这个效果可用于对角色的皮肤采用较低的高光强度值，而对角色的眼睛采用一个更高的值，这些都在同一材质内进行。另一种可行的方法是将渲染(使用 3D 应用程序)后网格物体的顶