PART 1 入门篇- 新手的unity3d漫游指南

第1章 初识神兵- 为什么选择Unity3d

主要内容：常用3D引擎对比分析，说明使用Unity3d进行VR/AR开发的优缺点

1.1 常用3D引擎介绍

1.1.1 游戏引擎的出现

1.1.2 游戏引擎架构基础

1.1.3 主流3D引擎对比分析

1.2 Unity的发展史

1.2.1 Unity引擎的前世今生

1.2.2 Unity引擎的代表作品

1.3 Unity的安装、授权与服务

1.3.1 Unity的授权类型和服务

1.3.2 在Windows下的安装

1.3.2 在Mac 下的安装

1.4 Unity学习资源的获取

1.4.1 Unity的开发者官方认证培训

1.4.2 Unity官方的教学资源

1.4.3 值得推荐的第三方学习资源

1.5本章小结

第2章 蹒跚学步- 和Unity的第一次亲密接触

2.1 Unity编辑器入门

2.1.1 界面布局

2.1.2 工具栏

2.1.3 菜单栏

2.1.4 Console

2.2 Unity中的核心概念和子系统

2.2.1 场景

2.2.2 游戏对象

2.2.3 组件

2.2.4 预设体

2.2.5 Unity的核心子系统

2.3 Unity Asset Store 资源商城简介

2.3.1 Asset Store简介

2.3.2 使用Asset Store获取游戏资源

2.3.2 将游戏资源上传到Asset Store

2.4 本章小结

第3章 无往不利- 在Unity中使用C#进行编程

3.1 C# 语言概述

3.1.1 什么是编程语言

3.1.2 为什么选择C#

3.1.3 C#的开发环境

3.2 C#的基本语法和使用

3.2.1 变量和数据类型

3.2.2 表达式和运算符

3.2.3 流程控制

3.2.4 函数

3.2.5 类和方法

3.3 Unity中的C#脚本开发

3.3.1 创建脚本

3.3.2 使用MonoDevelop编辑器及Visual Studio

3.3.3 实战-如何访问游戏对象

3.3.4 实战：如何访问组件

3.3.5 事件函数

3.4 本章小结

PART 2 进阶篇- Unity3d游戏开发实战

第4章 创建一个新世界-Modelling

4.1 创建基础的游戏场景

4.1.1 Unity中的地形引擎

4.1.2 基本几何体的使用

4.1.3 添加天空盒和Fog

4.2 创建外部的游戏资源

4.2.1 3D模型、材质和动画的创建工具

4.2.2 音效创建工具

4.2.3 其它资源创建工具

4.3 导入外部的游戏资源

4.3.1 2D图形的导入和设置

4.3.2 3D模型的导入和设置

4.3.3 3D动画的导入和设置

4.3.4 音频和视频的导入和设置

4.3.5 Unity资源包的导入和导出

4.4 实战-创建Battlestar项目并准备所需的游戏资源

4.4.1 Battlestar游戏的策划与设计

4.4.2 创建BattleStar项目并添加版本控制

4.4.3 创建并导入BattleStar项目所需的3D美术资源

4.4.4 创建BattleStar的基本游戏场景

4.5 本章小结

第5章 有了光就有了一切- Enlighten

5.1 Unity光照系统（Enlighten）介绍

5.1.1Light组件简介

5.1.2 常见的光源类型

5.1.3 Light inspector简介

5.2 全局光照

5.2.1 全局光照简介

5.2.2 烘焙

5.2.3 Lightmaps的使用

5.2.4 Light Probes和Reflection Probes的使用

5.3 BattleStar游戏实战-给游戏场景添加光照

5.3.1给场景添加光照前的准备工作

5.3.2给场景添加太阳光

5.3.3 添加发光贴图

5.3.4 添加点光源

5.3.5 添加Light Probe Group

5.3.6 添加Reflection Probe

5.4 本章小结

第6章 让游戏画面栩栩如生-粒子系统和shader、Post Processing的使用

6.1 Shuriken粒子系统

6.1.1 什么是粒子系统(particle systems)

6.1.2 Unity中的Shuriken粒子系统

6.2 Shader的使用

6.2.1 什么是Shader

6.2.2 Unity的标准Shader

6.2.3 创建自定义的Shader

6.3 Post Processing的使用

6.3.1 Post Processing简介

6.3.2 如何在Unity项目中添加Post Processing

6.4实战-完善BattleStar游戏场景的视觉效果

6.4.1 设置场景材质（标准Shader的使用）

6.4.2 制作添加粒子系统特效1-烟尘

6.4.3 制作添加粒子系统特效2-火花

6.4.4 给游戏添加后处理特效

6.4.5 第三方插件Amplify Shader Editor的使用

6.5 本章小结

第7章 游戏中的UI界面系统

7.1 Unity 中的UI系统简介

7.1.1 NGUI插件简介

7.1.2 UGUI系统简介

7.1.3 Canvas简介

7.1.4 视觉控件简介

7.1.5 交互控件简介

7.2 BattleStar游戏实战-给游戏添加UI

7.2.1 BattleStar游戏中的UI设计

7.2.2 给BattleStar游戏添加UI

7.2 本章小结

第8章 让角色动起来- Unity动画系统

8.1 Unity动画系统概述

8.1.1 Legacy Animation System

8.1.2 Mecanim动画系统

8.2 BattleStar游戏实战- 让游戏中的角色动起来

8.2.1 导入NPC角色资源

8.2.2 创建角色的Prefab

8.2.2 添加Animator组件

8.2.3 设置基础状态机

8.2.4 添加状态机之间的切换

8.2.5 编写控制角色动画的脚本

8.3 实战-更复杂的人物动画机制

8.3.1 如何使用Blend Trees

8.3.2 如何实现人形角色动画的配置和重定向

8.4 本章小结

第9章 有了目标才有方向- Navigation

9.1 Unity中的寻路系统

9.1.1 寻路系统内部工作原理

9.1.2 NavMesh 烘焙设置

9.1.3 NavMesh Obstacle

9.1.4 Off-Mesh Link

9.2 BattleStar游戏实战-在游戏中使用寻路系统

9.2.1 添加地板对象的寻路烘焙

9.2.2 在NPC角色对象上添加并设置NavMeshAgent组件

9.2.3 在脚本中添加寻路状态的动画切换

9.2.4 添加寻路距离

9.3 本章小结

第10章 真实世界的物理法则- 物理系统

10.1 Unity中的物理系统

10.1.1 物理系统概述

10.1.2 Rigidbody组件

10.1.3 Collider组件

10.1.4 Raycast

10.2 BattleStar游戏实战-给游戏添加物理系统

10.2.1 BattleStar游戏中的物理系统设计

10.2.2 在场景中添加物理碰撞系统

10.3 本章小结

第11章 且听风吟-音乐和音效

11.1 Unity中的Audio系统

11.1.1 Audio系统概述

11.1.2 Audio Source

11.1.3 Audio Reverb Zone

11.2 BattleStar游戏实战-给游戏添加背景音乐和音效

11.2.1 将音乐和音效资源导入到项目中

11.2.2 给游戏添加背景音乐

11.2.3 给游戏添加音效

11.3 本章小结

第12章 Unity中的数据存取与游戏性能优化

12.1 Unity中数据存取的常用方法

12.1.1 使用PlayerPrefs

12.1.2 使用Json

12.1.3 使用数据库

12.2 Unity中性能优化的原则和常用方法

12.2.1 Unity中的性能优化简介

12.2.2 Unity中的遮挡剔除

12.2.3 使用Profiler和Frame Debugger

12.2.4 灵活使用AssetBundle加载资源

12.3 BattleStar游戏实战-添加数据存取机制并优化游戏

12.3.1 游戏数据存取机制的设计

12.3.2 实现游戏数据存取机制

12.3.2 实现游戏的核心逻辑机制

12.3.4 对游戏性能进行优化

12.3.5 完善BattleStar项目

12.3.6 BattleStar项目的配置和发布

12.4 本章小结

第13章 一个人的世界很孤单- Unity网络编程

13.1 UNET简介

13.1.1 UNET中的服务器

13.1.2 High Level API

13.1.3 Transport Layer API

13.2 Unity中的第三方网络插件-Photon

13.2.1 什么是Photon

13.2.2 Photon的主要功能特性

13.2.3 Photon Cloud vs Photon Server

13.2.4 PUN vs PUN+ vs UNET

13.3 虚拟聊天室 - 使用Unity和Photon创建一个简单的多人在线游戏

13.3.1 多人在线FPS游戏的产品策划

13.3.2 创建Unity项目和基础场景

13.3.3 测试游戏的主要场景

13.3.4 添加多人在线功能

13.3.5 优化和完善游戏

13.3.6 添加语音对话功能

13.4 本章小结

PART 3 高级篇- VR/AR开发实战

第14 章 VR/AR开发技术入门

14.1 虚拟现实相关技术介绍

14.1.1 立体显示技术

14.1.2 场景建模技术

14.1.3 自然交互技术

14.2 主流的虚拟现实设备

14.2.1 Oculus Rift

14.2.2 HTC Vive

14.2.3 Sony Playstation

14.2.4 Samsung Gear VR

14.2.5 Google Daydream

14.2.6 Microsoft HoloLens

14.2.7 Magic Leap

14.3 主流的虚拟现实开发工具

14.3.1 虚拟现实开发平台简介

14.3.2 第三方工具和SDK

14.4 虚拟现实应用开发的基本流程和注意事项

14.4.1 虚拟现实应用开发的基本流程

14.4.2 虚拟现实应用开发的注意事项。

14.5 本章小结

第15 章 实战开发HTC Vive和Oculus Rift上的跨平台对战游戏SpaceCraft

15.1 HTC Vive平台开发概述

15.1.1 HTC Vive设备及平台简介

15.1.2 Lighthouse技术原理

15.1.3 HTC Vive手柄交互详解

15.1.4 SteamVR插件简介

15.1.5 VRTK插件简介

15.2 配置HTC Vive的开发测试环境

15.2.1 安装HTC Vive硬件和SteamVR 软件

15.2.2 设置并打开SteamVR

15.2.3 运行SteamVR的测试场景

15.2.4 TPCast无线模块

15.2.5 Vive 追踪器简介

15.3 跨平台VR联网对战塔防游戏的策划

15.4 搭建游戏场景

15.4.1 创建项目并进行基本设置

15.4.2 新建塔防场景

15.4.3 实现敌人的自动寻路功能

15.4.4 添加其它游戏逻辑组件

15.5 将游戏变成VR版

15.5.1 使用手柄控制游戏开始

15.5.2 使用手柄在游戏中进行传送

15.5.3 使用手柄与场景中的物体互动

15.5.4 给玩家装配武器

15.6 完成游戏的联网功能

15.6.1 导入PUN插件并实现联网功能

15.6.2 添加玩家的“化身”

15.6.3 修复游戏中的小问题

15.7 适配到Oculus平台

15.7.1 导入Oculus SDK

15.7.2 使用Oculus的CameraRig

15.7.3 功能逻辑的优化

15.8 添加背景音乐和音效

15.9.1 添加背景音乐

15.9.2 添加交互时的音效

15.9 将游戏发布到VivePort平台

15.9.1 VivePort 平台介绍

15.9.2 将产品上传到Viveport商城

15.9.3 Oculus Store平台简介

15.9.4 将产品上传到Oculus Store平台

15.9.5 Steam平台简介

15.9.6 将产品上传到Steam商城

15.10 本章小结

第16章 实战-开发Google Daydream VR平台上的《VSpaceCraft》

16.1 Google Daydream VR平台开发概述

16.1.1 Google Daydream VR设备及平台简介

16.1.2 Google Daydream VR中的交互

16.2 《VSpaceCraft》的产品策划

16.3 配置Google Daydream VR的开发测试环境

16.3.1 准备好相关的开发硬件  
16.3.2 系统和软件的安装  
16.3.3 创建新的Daydream项目

16.4 游戏基本结构的设计和实现

16.5 游戏场景的搭建和设置

16.6 深入了解Daydream Controller

16.7 将游戏适配到Daydream VR平台

16.7.1 实现Daydream Controller的按钮事件

16.7.2 使场景支持Daydream

16.7.3 使用Daydream Controller的射线

16.7.4 隐藏控制器模型

16.8 将产品发布到 Google Play VR和Daydream

16.8.1 Google Play VR和Daydream介绍

16.8.2 将Daydream VR应用发布到Google Play VR和Daydream

16.9 本章小结

第17章 实战-使用Unity和Vuforia SDK开发AR游戏iDancer

17.1 Vuforia SDK简介

17.1.1 什么是Vuforia

17.1.2 Vuforia的功能模块简介

17.1.3 Vuforia的授权类型

17.1.4 Vuforia的工具和资源

17.1.5 Vuforia支持的设备及平台

17.2 iDancer的产品策划

17.3 配置开发测试环境

17.3.1 在 Unity 中导入 Vuforia SDK

17.3.2 开发前的准备工作

17.4 导入项目所需的游戏资源

17.5添加游戏交互机制

17.5.1 利用 Image Target实现图像识别

17.5.2 在 iOS 设备上编译运行

17.5.3 让用户来定义Image Target

17.6 iDancer游戏的UI设计与实现

17.6.1 iDancer的UI设计思路

17.6.2 设计《iDancer》的 UI

17.7 实现 Unity 与 iOS 的整合

17.7.1 实现思路及原理

17.7.2 实现 Unity中的接口

17.8 将项目编译到 Android 平台

17.9 将游戏发布到 AppStore和安卓应用商城

17.9.1 将游戏发布到AppStore

17.9.2 将游戏发布到安卓应用商城

17.10 本章小结

第18章 实战-使用Unity和Wikitude SDK开发AR游戏

18.1 Wikitude及其它AR SDK简介

18.1.1 什么是Wikitude SDK

18.1.2 Wikitude的功能特性

18.1.3 Wikitude所支持的设备及平台

18.1.4 Wikitude的授权费

18.1.5 Wikitude vs Vuforia vs Kudan

18.1.6 国内的AR SDK简介

18.2 口袋动物园游戏的策划

18.3 配置开发测试环境

18.3.1 创建 Wikitude 项目

18.3.2 创建 Target

18.3.3导入 Wikitude Unity SDK

18.4 使用Wikitude SDK开发口袋动物园应用

18.4.1 创建 ImageTracker

18.4.2 添加模型

18.5 添加游戏互动机制

18.5.1 实现缩放功能

18.5.2 实现旋转功能

18.5.3 添加点击手势

18.6 将产品编译运行在iOS和Android平台

18.6.1 将产品在iOS平台中编译

18.6.2 将产品在Android平台中编译

18.7 将产品发布到AppStore和应用商城

18.7.1 将游戏发布到AppStore

18.7.2 将游戏发布到安卓应用商城

18.8 本章小结

第19章 实战开发HoloLens平台的HoloSpace游戏

19.1 HoloLens平台开发概述

19.1.1 HoloLens设备及平台简介

19.1.2 HoloLens中的语音交互

19.1.3 HoloLens中的手势动作

19.1.4 Gaze

19.1.5 Spatial Sound

19.1.6 SpatialMapping

19.1.7 Sharing holograms

19.2 HoloSpace产品策划

19.3 配置HoloLens的开发测试环境

19.3.1 准备好HoloLens硬件

19.3.2 电脑及操作系统配置要求

19.3.3 安装Visual Studio和其它软件

19.3.4 设置并运行HoloLens的测试场景

19.4 在Unity中导入所需的美术资源

19.5 实现游戏的核心机制

19.5.1 模拟火箭升空的效果

19.5.2 将火箭部署到HoloLens世界中

19.5.3 调整火箭的初始位置

19.5.4 丰富对火箭的交互

19.6 游戏的UI设计与开发

19.6.1 HoloLens应用中UI设计的注意事项

19.6.2 HoloSpace的UI设计思路

19.6.3 添加场景中的菜单

19.7 添加背景音乐和音效

19.7.1 添加背景音乐

19.7.2 添加交互时的音效

19.8 HoloLens与Vuforia

19.9 将产品发布到Windows Store平台

19.9.1 微软Windows Store平台介绍

19.9.2 将产品发布到Windows Store平台

19.10 本章小结

第20章 实战-使用苹果ARKit和Unity开发AR应用iGirl

20.1 苹果ARKit简介

20.1.1 什么是ARKit

20.1.2 ARKit的功能特性

20.1.3 ARKit所支持的设备及平台

20.2 iGirl小游戏的策划

20.3 使用Unity ARKit 开发iGirl

20.3.1 开发前的准备工作

20.3.2 Unity ARKit插件介绍

20.3.3创建项目并导入ARKit插件

20.3.4 运行ARKit插件的示例场景

20.3.5 创建自己的场景

20.3.6 配置AR Camera

20.3.7 放置模型

20.3.8 将产品编译运行在iOS 平台

20.4 本章小结

附录

Unity3d中的快捷方式

本书中涉及的计算机术语