论文初稿

封面

- 题目：基于Unity3D的跨平台回合制游戏App应用设计与开发

- 姓名：阮复伸

- 学号：181713600007

- 专业：计算机科学与技术

- 指导老师：王洋

- 论文提交日期：

目录

摘要

随着移动设备的不断普及，移动设备中的各类应用与游戏也变得丰富多彩。然而随着玩家对游戏的画面、音效、粒子特效等要求不断增高，越发凸显出一个游戏的架构设计的好坏。当一个游戏的架构设计不合理时，会发生掉帧、卡顿、程序崩溃等各种问题，严重影响了玩家在游戏中的体验。尤其是在移动设备平台中，如果在屏幕中突然出现大量粒子效果或面数很高的模型，将对游戏的架构设计将有很高的要求。而Unity ECS架构则正是为了解决这个问题诞生的，ECS相较于传统的Unity设计思路而言，使用了C# Job system、多线程、协同处理等多种方式。在渲染大量粒子、多边形等方面更加有优势。

本文主要工作如下

1. 参照官方文档，了解ECS的概念及实现方式。

2. 参照官方文档，了解LWRP中Post Processing Volume的使用方法，与贴图的基本制作流程。

3. 编写独立游戏的Game Design Document。

4. 根据Game Design Document，在ECS框架内编写游戏代码，使用LWRP渲染模式组织美术资源。

5. 测试实机运行效率。

6. 发布应用。

关键词： Unity3D；游戏引擎；移动设备；架构设计

## Abstract

插 图 索 引

附 表 索 引

4

第一章 绪论

1.1 研究背景及意义

研究背景：

Unity3D是由Unity Technologies开发的一个让玩家轻松创建诸如三维视频游戏、建筑可视化、实时三维动画等类型互动内容的多平台的综合型游戏开发工具，是一个全面整合的专业游戏引擎。Unity类似于Director,Blender game engine, Virtools 或 Torque Game Builder等利用交互的图型化开发环境为首要方式的软件。其编辑器可运行在Windows、Linux、Mac OS X下，可发布游戏至Windows、Mac、Wii、iPhone、WebGL、Android等多种类型的平台上。

Unity3D在2018年引入了可编程渲染管线(Scriptable Render Pipeline)的概念，SRP主要分为两个方向，针对于PC、PS4、XBOX等高性能设备的High Definition Render Pipeline(HDRP)，和针对于移动设备、网页平台的Light Weight Render Pipeline(LWRP)。SRP的引入，使得开发者可以使用C#代码来控制渲染过程中，各个物体的渲染顺序和效果，让程序和渲染更加灵活、效率更高。

同样，Unity3D在2018年还引入了全新的ECS开发理念。传统的GameObject-MonoBehaviour机制适合无基础的新手快速入门，但是随着项目的复杂度不断增加，这种传统的架构会让代码变得难以阅读、维护和优化。而Unity3D新的ECS(Entity-Component-System)开发理念，因Blizzard开发的射击游戏Over Watch被开发者所熟知。ECS开发理念主要关注如何组建并处理游戏中的数据和行为。

使用了LWRP与ECS流程制作的Unity3D移动端游戏，相较于传统MonoBehavior 与PBR相结合制作游戏的流程，有以下优势：

1. 程序结构更加清晰，代码易于维护。

2. 在较低性能的移动端运行时，也可以启用更多的动态光照，保证画面的渲染效果。

3. LWRP以更少的性能消耗实现相同甚至更好的粒子特效、环境光照、Post Processing。

研究意义

LWRP与ECS相结合的项目流程，目前多应用在大型多人在线MMORPG或者3A单机游戏中，在中小型独立游戏中却很少有此类流程的应用。而且LWRP与传统BPR项目在Substance Designer贴图生成、材质制作、后期效果表现等环节内也有很大的不同。 本论文尝试使用LWRP、ECS等Unity3D的新特性，以一款简单的独立游戏为载体，实现在移动平台(iPhone6s至iPhone11 Pro Max或同代Android设备)中，运行时下更好的画面渲染效果、更多动态光照、在创建或销毁GameObject和粒子时帧率相对稳定等效果。

研究现状

目前，各大应用市场中，使用Unity3D ECS框架与LWRP渲染管线的三维项目非常少，多数移动端游戏中，还是以传统的BPR渲染方式为主。Post Processing大多依赖第三方组件，可靠性与稳定性较差。

但是，目前Unity3D ECS框架中所依赖的主要类库，均为Beta版本，有很多的不确定性，在各类主流移动端设备上未进行充分测试。并且在项目编码过程中，Unity3D官方可能会更新ECS框架类库，导致项目部分功能需要重写。

预期目标

通过一个简单的独立游戏项目，了解并熟悉以下流程：

1. 明确LWRP中贴图美术流程。模型贴图制作、Substance Designer中LWRP节点生成、各类贴图对应LWRP LitShader中各类通道的表现效果。
2. Unity3D中，使用Pro Builder创建简单的游戏原型，并在项目中使用。
3. 实现LWRP中水面的反射与折射的表现效果。
4. 基于Unity Job System和ECS的游戏框架设计
5. 在移动设备中测试游戏性能

论文结构安排

1. 通过官方文档及相关资料，了解Unity Job System、ECS设计思路与传统GameObject-MonoBehaviour设计思路的差别。
2. 使用LWRP LitShader时，各个贴图的作用，以及各类参数所影响的表现效果。
3. 编写游戏设计文档。
4. 根据游戏设计文档编写项目代码
5. 在unity中，编写在LWRP下可以使用的水面着色器。
6. 测试项目在移动设备端的运行效率。

第二章 Unity3D Job System、 ECS架构设计目的和原理

第三章 独立游戏的游戏设计文档编写

在程序开发中，每一种文档在不同阶段都有不用的作用，最后这些文档都会汇总至同一份文档中——游戏设计文档(Game Design Document)，或者简写为GDD。

GDD并没有严格的格式或长度要求，但是要保证其可以精确第描述出游戏。

一般情况下，一份合格的GDD，要具有以下信息：

游戏名称

游戏系统

目标玩家年龄

ESRB分级

游戏概要、可玩性

游戏玩法的独特性

与众不同的卖点

竞品

此独立游戏的基本GDD内容如下表

|  |  |
| --- | --- |
|  | |
| 游戏名称 |  |
| 游戏系统 |  |
| 目标玩家年龄 |  |
| ESRB分级 |  |
| 游戏概要、可玩性 |  |
| 游戏玩法的独特性 |  |
| 与众不同的卖点 |  |
| 竞品 |  |

第四章 独立游戏架构设计

第五章 项目编码、测试

结论

参考文献

[1] 陈嘉栋 [M] 《Unity3D 脚本编程 使用C#语言开发跨平台游戏》 北京 电子工业出版社 2019.6

[2] Scott Rogers [M] 《通关 游戏设计之道》 北京 人民邮电出版社 2013.11

[3] Erich Gamma [M] 《Design Patterns》 北京 机械工业出版社 2019.5

[4] 冯乐乐 [M] 《Unity Shader入门精要》 北京 人民邮电出版社 2016.6

[5] 加藤政村 [M] 《Unity游戏设计与实现 南梦宫一线程序员的开发实例》 北京 人民邮电出版社 2015.2

[6] Unity3D官方LWRP文档 [DB/OL] https://docs.unity3d.com/Packages/com.unity.render-pipelines.lightweight@6.9/manual/index.html

[7] Unity3D ECS官方介绍 [DB/OL] https://connect.unity.com/p/part-1-unity-ecs-briefly-about-ecs

[8] Unity官方ECS项目Github源码 [DB/OL] https://github.com/Unity-Technologies/EntityComponentSystemSamples

致谢

紧张、充实而又难忘的大学学习生涯即将结束，在大学学习和撰写论文期间得到了许多人的帮助，使我终身难以忘怀。轻风系不住流云，流云却带走了岁月，打开尘封的记忆，往事如风却又历历在目，大学的学习生活即将结束。在这里我首先要感谢这四年来为我授课的各位老师，真心地说一句：你们辛苦了！感谢××老师在我的论文选题、定稿以及中期检查等方面都给了我精心的指导。您提出的宝贵意见使我在论文选题、撰写以及修改的过程中，不再像当初那样茫然无措，而是知道自己论文的不足和修改的方向。您正直、严谨的治学态度对我影响颇深，受益匪浅，无论在今后的学习还是工作当中，我都铭记于心。