小组成员

刘洋、和昕、陈泽鸿、张航、张芷璇、冯毅伟、张旭、管熙玉

测试计划及结果分析报告

TUP小组

目录

[一、文档说明 2](#_Toc51684499)

[1.1、文档信息 2](#_Toc51684500)

[1.2、文档控制 2](#_Toc51684501)

[1.2.1、变更记录 2](#_Toc51684502)

[1.2.2、评审记录 2](#_Toc51684503)

[二、引言 3](#_Toc51684504)

[2.1、编写目的 3](#_Toc51684505)

[2.2、测试组成员信息 3](#_Toc51684506)

[2.3、参考资料 3](#_Toc51684507)

[2.4、属于和缩略语 3](#_Toc51684508)

[2.5、风险及应急 4](#_Toc51684509)

[三、测试策略 4](#_Toc51684510)

[3.1、整体策略 4](#_Toc51684511)

[3.2、测试目标 4](#_Toc51684512)

[3.3、测试准则 5](#_Toc51684513)

[3.4、测试范围 5](#_Toc51684514)

[3.5、测试交接标准 6](#_Toc51684515)

[3.5.1、单元测试交接标准 6](#_Toc51684516)

[3.5.2、集成测试交接标准 6](#_Toc51684517)

[3.6、测试通过标准 6](#_Toc51684518)

[四、资源需求 6](#_Toc51684519)

[4.1、培训需求 6](#_Toc51684520)

[4.2、运行环境 6](#_Toc51684521)

[五、测试实践安排 7](#_Toc51684522)

[5.1、各阶段时间分配 7](#_Toc51684523)

[5.2、时间要求和人员安排 7](#_Toc51684524)

[六、测试过程管理 7](#_Toc51684525)

[6.1、测试文档管理 7](#_Toc51684526)

[6.2、测试工具 8](#_Toc51684527)

[6.3、游戏测试计划 8](#_Toc51684528)

[6.4、测试模块划分 9](#_Toc51684529)

# 一、文档说明

## 1.1、文档信息

文档编号：

文档基本信息参见下表：

|  |  |
| --- | --- |
| 文档作者 | 张航 |
| 创建日期 | 2020年9月12日 |
| 当前版本 | V3.0 |
|  | V3.0 |

## 1.2、文档控制

### 1.2.1、变更记录

测试计划文档变更记录详见下表：

|  |  |  |  |
| --- | --- | --- | --- |
| 变更日期 | 变更人 | 版本 | 备注 |
| 2020.09.12 | 张航 | V1.0 | 创建文档，添加新的内容 |
| 2020.09.16 | 张航 | V2.0 | 修改文档，划分测试模块 |
| 2020.09.22 | 张航 | V3.0 | 记录测试结果，分析整理形成最后的测试报告。 |

### 1.2.2、评审记录

下表记录详细的评审记录：

|  |  |  |  |
| --- | --- | --- | --- |
| 评审日期 | 评审人 | 版本 | 备注 |
| 2020.09.12 | 张航 | V1.0 | 测试计划第一次评审 |
| 2020.09.22 | 张航 | V2.0 | 测试计划及结果分析报告评审 |
|  |  |  |  |

# 二、引言

## 2.1、编写目的

本文档主要阐述“游戏人工智能算法的设计与实现”测试过程中的一些细节，为该项目的测试工作提供一个框架和规范：

1. 确定项目的测试的策略、范围和方法；
2. 使项目测试工作的所有参与人员对本项目测试的目标、范围、策略、方法、组织、资源等有一个清晰的认识；
3. 从策略角度说明本项目测试的组织和管理，知道测试进行，并作为项目测试工作实施的依据；

预期的读者主要有两类受众：测试管理人员和测试人员。

1. 项目测试负责人根据该测试大纲指定进一步的计划、安排和控制测试过程，主要是对工作任务分配以及时间进度安排；
2. 测试组组员通过测试大纲了解测试过程和相关信息；
3. 测试人员根据该测试大纲中制定的范围、方法确定测试需求、设计测试用例、执行和记录测试过程并记录和报告缺陷。

## 2.2、测试组成员信息

人员信息：

|  |  |
| --- | --- |
| 姓名 | 职责 |
| 张航 | 项目测试负责人，负责编写测试计划，执行部分测试，编写初步的测试结果分析报告以及最终的测试结果分析报告。 |
| 冯毅伟 | 测试组组员，负责完成部分测试工作，并撰写初步的测试分析报告。 |

## 2.3、参考资料

|  |  |
| --- | --- |
| 名称 | 备注 |
| 《“游戏人工智能算法的设计与实现”需求分析说明书》 |  |
| 《软件工程专业实训2020》 |  |

## 2.4、属于和缩略语

术语：

|  |  |
| --- | --- |
| 术语/定义 | 说明 |
| UNITY Profiler | UNITY的游戏性能优化工具 |
|  |  |

## 2.5、风险及应急

该项目的测试过程中可能出现如下风险：

|  |  |
| --- | --- |
| 风险 | 应急措施 |
| 不熟悉UNITY Profiler的使用方法导致测试工作无法按计划正常进行。 | 在测试正式开始之前组织测试组组员学习Profiler的使用方法，尽快掌握其使用方法。 |
| 测试组组员无法正常安装和运行Unity导致测试无法正常进行。 | 尝试使用其他组员的电脑完成测试任务。 |

# 三、测试策略

## 3.1、整体策略

本项目特点：

1. 待测项目是《游戏设计》这门课所完成的大作业，并且小组中有三位成员参加过该项目的开发工作，因此对项目有很高的熟悉程度；
2. 参与测试的人员都参加过《软件质量与评测技术》课程，有一定的软件项目测试基础，但是从未使用过UNITY Profiler等相关测试工具；
3. 本次测试工作属于小学期项目当中的一个环节，因此时间比较紧；

根据以上特点，制定本项目的测试过程策略如下：

1. 尽量在有限的时间内发现尽可能多的缺陷，尤其是严重程度为高的缺陷。
2. 测试和开发同步进行；
3. 测试过程要受到控制。根据事前制定的测试计划执行顺序镜子那个测试，并填写测试记录表，保障测试过程是受控的；
4. 确定重点。测试的重点应放在Profiler性能分析、线程性能分析、脚本性能分析这些方面；

## 3.2、测试目标

1. 该项目的测试目标一方面是找出游戏中存在的缺陷并协助开发人员进行修改；另一方面是对游戏的性能开销进行分析，形成初步的测试分析报告，最后汇总得到最终的性能测试分析报告；
2. 在进行性能测试分析过程中，使用UNITY Profiler工具观察CPU、GPU、渲染、内存、声音、视频、物理、UI及全局光照等各类功能模块等各类模块的性能指标；
3. 对游戏运行过程中各个线程进行性能分析，包括主线程、渲染线程和工作线程等；
4. 对游戏人工智能部分的脚本进行性能分析，计算该任务在当前帧内CPU占用时间比例、耗时总时长等性能参数。修改或更换人工智能算法，观察算法的选择对于游戏性能的影响；

## 3.3、测试准则

在独立显卡的计算机上运行时，游戏帧率不小于30FPS。

## 3.4、测试范围

（1）待测项目

UNITY Profiler游戏性能分析

游戏线程性能分析

人工智能脚本性能分析

（2）具体测试内容

|  |  |
| --- | --- |
| 测试内容 | 测试范围 |
| UNITY Profiler游戏性能指标分析 | CPU |
| GPU |
| 渲染（Rendering） |
| 内存（Memory） |
| 声音（Audio） |
| 视频（Video） |
| 物理（Physics） |
| UI |
| 全局光照（Global Illumination） |
| 线程性能分析 | 主线程 |
| 渲染线程 |
| 工作线程 |
| 脚本性能分析 | 当前帧内CPU占用时间比例 |
| 耗时总时长 |
| 修改或更换人工智能算法，观察算法的选择对于游戏性能的影响 |

## 3.5、测试交接标准

### 3.5.1、单元测试交接标准

该测试主要由测试组进行，由测试组而根据需要进行约定即可。

### 3.5.2、集成测试交接标准

该测试主要由开发组和测试组共同完成：

1. 约定的测试内容全部完成，并通过了单元测试，由相关人员签字确认；
2. 测试组严格按照测试计划和质量标准中的规定进行测试；

## 3.6、测试通过标准

1. 计划的测试用例已全部执行；
2. 经确定的所有缺陷都已得到了商定的解决结果；

# 四、资源需求

## 4.1、培训需求

该测试项目需要组织培训。参与本次测试的测试人员需要测试组组长对测试人员进行系统的相关培训。培训内容主要包括：

* Unity的安装及基本使用方法；
* UNITY Profiler使用方法；

## 4.2、运行环境

软件环境：

操作系统：Windows10

开发工具：Unity 2019.3.2f1

性能分析工具：UNITY Profiler

编程语言：C#

# 五、测试实践安排

## 5.1、各阶段时间分配

|  |  |  |  |
| --- | --- | --- | --- |
| 测试开始时间： | 2020.09.17 | 测试结束时间： | 2020.09.27 |
| 序号 | 名称 | 完成日期 | 工作量 |
| 1 | 编写测试计划 | 2020.09.13 | 1人/日 |
| 2 | 执行测试，编写初步测试分析报告 | 2020.09.20 | 2人/日 |
| 3 | 完成测试，撰写最终的测试分析报告 | 2020.09.27 | 1人/日 |

## 5.2、时间要求和人员安排

|  |  |  |  |
| --- | --- | --- | --- |
| 任务 | 任务明细 | 具体时间要求 | 人员安排 |
| 测试前业务 | 测试组组员学习使用UNITY Profiler等工具的使用方法。 | 2020.09.07 – 2020.09.11 | 张航、冯毅伟 |
| 编写测试计划 | 根据需求规格说明书编写测试计划。 | 2020.09.12 – 2020.09.13 | 张航 |
| 执行测试 | 根据测试计划执行测试。整个测试过程和开发同步，同时编写初步的测试性能报告。 | 2020.09.14 - 2020.09.20 | 张航、冯毅伟 |
| 完成测试 | 待修改完存在的问题后，汇总全部的初步测试性能报告，基于此编写最终的性能测试报告。 | 2020.09.21 – 2020.09.27 | 张航 |

# 六、测试过程管理

## 6.1、测试文档管理

本项目对测试文档集中管理，文档存放在测试组组长处，每周备份一次。测试文档由不同角色分别创建，各角色创建的文档如下表所示：

|  |  |  |
| --- | --- | --- |
| 文档名称 | 编者 | 其他说明 |
| 《测试计划》 | 张航 | 对整个项目的测试工作做出安排。 |
| 《初步测试性能报告》 | 张航、冯毅伟 | 记录测试工具显示的结果并将其记录下来。 |
| 《最终测试性能报告》 | 张航 | 待解决完问题后汇总全部的初步测试性能报告，编写最终的测试性能报告。 |

## 6.2、测试工具

UNITY Profiler

## 6.3、游戏测试计划

|  |  |  |
| --- | --- | --- |
| UNITY Profiler测试目标 | * 对游戏进行性能分析； * 观察CPU、GPU、渲染、内存、声音、视频、物理、UI及全局光照等各类功能模块的性能指标； | |
| 线程性能分析 | * 对游戏运行过程中各个线程进行性能分析； * 包括主线程、渲染线程和工作线程； | |
| 人工智能脚本性能分析 | * 计算该任务在当前帧内CPU占用时间比例，耗时总时长等性能参数； * 修改或更换人工智能算法，观察算法的选择对于游戏性能的影响； | |
| 测试范围 | 难度调节 | 增加NPC的血条，调整人物以及NPC的伤害值，调整游戏中关卡的难度以及对游戏中NPC的出现概率进行调整 |
| NPC人工智能 | 实现NPC智能避开陷阱； |
| 实现NPC之间的战术配合 |
| boss状态变化 | 增加BOSS，BOSS分为三种状态，血量减少时依次触发状态变化，BOSS在变化状态时会变成狂暴状态（变大变颜色），并且会发射围绕自己一圈的弹幕 |
| 待测AI模型设计 | 运动层 | 自动躲避陷阱 |
| 决策层 | 以决策树为例 |
| 战略层 | 前后夹击，由队长带领 |

## 6.4、测试模块划分

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 脚本基本信息 | | | | | |
| 脚本名称 | Orc | | | | |
| 脚本功能概述 | 控制兽人行为逻辑 | | | | |
| 使用该脚本的NPC | 兽人 | | 控制兽人狂暴、攻击玩家、移动、逃跑行为 | | |
|  | |  | | |
| 脚本内容及功能简介 |  | | | | 怪物根据玩家所在位置逃向相反方向。 |
|  | | | | 怪物的状态转化为狂暴。 |
|  | | | |  |
| 脚本性能分析 | | | | | |
| CPU使用率 | Scripts | |  | | |
| Rendering | |
| Physics | |
| 渲染 | Batches | |  | | |
| Triangles | |
| 内存 | Total Allocated | |  | | |
| Texture Memory | |
| Mesh Memory | |
| 音频 | Playing Audio Sources | |  | | |
| Audio Voices | |
| Total Audio CPU | |
| Total Audio Memory | |
| 视频 | Total Video Sources | | 无视频 | | |
| Playing Video Sources | |
| Total Video Memory | |
| 物理 | Active Dynamic | |  | | |
| Static Colliders | |
| Rigidbody | |
| Trigger Overlaps | |
| UI | Layout | |  | | |
| Render | |
| 全局光照 | Total CPU | |  | | |
| Input Lighting | |
| Systems | |
| 线程性能分析 | | | | | |
| 主线程 |  | | | | |
| 渲染线程 |  | | | | |
| 工作线程 |  | | | | |
| 修改后的脚本性能分析 | | | | | |
| 改变兽人数量（初始值4） | 100 | CPU | | 生成100只兽人怪物。    此时性能有较大的消耗，但是电脑仍然能正常运行。    两个生成点一个生成50个。 | |
| 主线程 | |
|  | 耗时时长 | |
| 1000 | CPU | | 生成1000只兽人。    此时性能开销较大，但游戏仍然可以运行。    两个生成点一个生成500个。 | |
| 主线程 | |
| 耗时时长 | |
| 改变兽人伤害值（初始值5） | 20 | CPU | | 伤害值修改成20。    每一次受到伤害减少20点血量，该图片显示的是受到3次伤害后的剩余血量。    大部分性能都消耗在渲染方面。 | |
| 主线程 | |
| 耗时时长 | |
| 50 | CPU | | 伤害值修改为50。    受到一次伤害以后减少50点体力。    同样，性能开销主要集中在渲染方面，总体上对电脑性能消耗不大。 | |
| 主线程 | |
| 耗时时长 | |
| 结果分析 | | | | | |
| 经过上述测试，我们发现兽人的脚本对于电脑性能要求并不是很高，且大部分都消耗在渲染方面。当兽人数量变得非常大时，游戏对电脑性能的消耗比较大，但仍然在可接受的范围内。同时，改变兽人预制体的属性时，性能开销并没有明显的改变。总体来说达到了预期的要求。 | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 脚本基本信息 | | | | | |
| 脚本名称 | Skeleton | | | | |
| 脚本功能概述 | 控制骷髅兵行为脚本 | | | | |
| 使用该脚本的NPC | 骷髅兵 | | 控制骷髅兵群居性行为、移动、攻击、狂暴、逃跑 | | |
|  | |  | | |
| 脚本内容及功能简介 |  | | | | 怪物的群居性行为。 |
|  | | | | 怪物监视状态。 |
|  | | | |  |
| 脚本性能分析 | | | | | |
| CPU使用率 | Scripts | |  | | |
| Rendering | |
| Physics | |
| 渲染 | Batches | |  | | |
| Triangles | |
| 内存 | Total Allocated | |  | | |
| Texture Memory | |
| Mesh Memory | |
| 音频 | Playing Audio Sources | |  | | |
| Audio Voices | |
| Total Audio CPU | |
| Total Audio Memory | |
| 视频 | Total Video Sources | | 无视频 | | |
| Playing Video Sources | |
| Total Video Memory | |
| 物理 | Active Dynamic | |  | | |
| Static Colliders | |
| Rigidbody | |
| Trigger Overlaps | |
| UI | Layout | |  | | |
| Render | |
| 全局光照 | Total CPU | |  | | |
| Input Lighting | |
| Systems | |
| 线程性能分析 | | | | | |
| 主线程 |  | | | | |
| 渲染线程 |  | | | | |
| 工作线程 |  | | | | |
| 修改后的脚本性能分析 | | | | | |
| 改变骷髅兵数量（初始值3） | 1 | CPU | | 骷髅兵生成数量修改为1。    此时只有一只骷髅兵，当前为逃跑状态。    此时大部分性能都消耗在渲染方面，物理方面的性能消耗较少。 | |
| 主线程 | |
| 耗时时长 | |
| 100 | CPU | | 骷髅兵生成数量修改为100    此时性能开销稍有增大，其中物理方面的开销增大较为明显。但仍然在可接受范围内，并且游戏仍然能够正常运行。 | |
| 主线程 | |
| 耗时时长 | |
| 1000 | CPU | | 生成的骷髅兵数量修改为1000      此时物理方面的性能开销较大，游戏有些许卡顿，但仍然在可接受的范围内。 | |
| 主线程 | |
| 耗时时长 | |
| 改变骷髅兵的视野大小（初始值为20） | 5 | CPU | | 将骷髅兵的视野范围改为5。    此时玩家离骷髅兵很近，但是骷髅兵仍然处于巡逻状态。    性能开销变化不明显。 | |
| 主线程 | |
| 耗时时长 | |
| 100 | CPU | | 将骷髅兵的视野范围修改为100    刚进入场景时骷髅兵就发现了玩家，并从巡逻状态转换为攻击状态。      此时性能消耗变化并不是很明显。 | |
| 主线程 | |
| 耗时时长 | |
| 结果分析 | | | | | |
| 经过上述测试，我们发现骷髅兵数量的改变会对游戏性能有较大的影响，而改变其自身属性并不会对游戏有太大的影响。总的来说该脚本对电脑性能的要求并不高，总的来说是比较适合的。 | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 脚本基本信息 | | | | | |
| 脚本名称 | EnemyFar | | | | |
| 脚本功能概述 | 控制法师行为逻辑脚本 | | | | |
| 使用该脚本的NPC | 法师 | | 控制法师的移动、攻击以及与兽人之间的配合 | | |
|  | |  | | |
| 脚本内容及功能简介 |  | | | | 与兽人配合行为 |
|  | | | |  |
|  | | | |  |
| 脚本性能分析 | | | | | |
| CPU使用率 | Scripts | |  | | |
| Rendering | |
| Physics | |
| 渲染 | Batches | |  | | |
| Triangles | |
| 内存 | Total Allocated | |  | | |
| Texture Memory | |
| Mesh Memory | |
| 音频 | Playing Audio Sources | |  | | |
| Audio Voices | |
| Total Audio CPU | |
| Total Audio Memory | |
| 视频 | Total Video Sources | | 无视频 | | |
| Playing Video Sources | |
| Total Video Memory | |
| 物理 | Active Dynamic | |  | | |
| Static Colliders | |
| Rigidbody | |
| Trigger Overlaps | |
| UI | Layout | |  | | |
| Render | |
| 全局光照 | Total CPU | |  | | |
| Input Lighting | |
| Systems | |
| 线程性能分析 | | | | | |
| 主线程 |  | | | | |
| 渲染线程 |  | | | | |
| 工作线程 |  | | | | |
| 修改后的脚本性能分析 | | | | | |
| 改变敌人数量（初始值3） | 1 | CPU | | 性能开销不大    法师的数量设为1，另一侧的生成点设为0，总数为1。 | |
| 主线程 | |
| 耗时时长 | |
| 100 | CPU | | 法师的数量变为100，图中小地图紫色区域就是法师的位置。      性能开销有所增加，但是增加幅度并不是很大。    两个生成点每个生成50个。 | |
| 主线程 | |
| 耗时时长 | |
| 1000 | CPU | | 法师数量变为1000，图中小地图紫色区域就是法师的位置。      当脚本运行以后对电脑性能有较大的开销，但总体来说还是可以承受的。    两个生成点各生成500个法师。 | |
| 主线程 | |
| 耗时时长 | |
| 改变法师速度（初始值3.5） | 10.0 | CPU | | 法师的数量为2      性能开销仍然在可接受的范围内。  法师的速度明显感觉到有所提高，玩家已经很难逃离法师的追击。 | |
| 主线程 | |
| 耗时时长 | |
| 100.0 | CPU | | 法师数量为2      性能开销仍然在可接受的范围内，并没有太大的性能开销。  法师此时的速度已经变得非常快，玩家几乎不能摆脱其追击。 | |
| 主线程 | |
| 耗时时长 | |
| 结果分析 | | | | | |
| 经过测试，可以得出在正常情况下该脚本对电脑的性能开销还在可以接受的范围内，只有在非常极端的情况下才会出现较大的性能开销。总体上来说是可以接受的。 | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 脚本基本信息 | | | | | | | |
| 脚本名称 | BossAction\RoadRoller\Paotai | | | | | | |
| 脚本功能概述 | Boss的人工智能，控制Boss的移动、状态转换。 | | | | | | |
| 使用该脚本的NPC | Boss | | | | 控制Boss的移动、状态转换。 | | |
| 脚本内容及功能简介 |  | | | | | | 召唤炮台/小怪 |
|  | | | | | | 转换为状态二 |
|  | | | | | | 转换为状态三 |
| 脚本性能分析（第二阶段） | | | | | | | |
| CPU使用率 | Scripts | | | |  | | |
| Rendering | | | |  | | |
| Physics | | | |  | | |
| GPU使用率 |  | | | | | | |
| 渲染 | Batches | | | |  | | |
| Triangles | | | |  | | |
| 内存 | Total Allocated | | | |  | | |
| Texture Memory | | | |  | | |
| Mesh Memory | | | |  | | |
| 音频 | Playing Audio Sources | | | |  | | |
| Total Audio CPU | | | |
| Total Audio Memory | | | |
| 视频 | Total Video Sources | | | | 无视频 | | |
| Playing Video Sources | | | |
| Total Video Memory | | | |
| 物理 | Active Dynamic | | | |  | | |
| Static Colliders | | | |
| Rigidbody | | | |
| Trigger Overlaps | | | |
| UI | Layout | | | |  | | |
| Render | | | |
| 全局光照 | Total CPU | | | |  | | |
| Input Lighting | | | |
| Systems | | | |
| 线程性能分析 | | | | | | | |
| 主线程 |  | | | | | | |
| 渲染线程 |  | | | | | | |
| 工作线程 |  | | | | | | |
| 修改后的脚本性能分析 | | | | | | | |
| 改变Boss速度（初始2.5） | 1.0 | | CPU | | |  | |
| 主线程 | | |
| 耗时时长 | | |
| 100.0 | | CPU | | |  | |
| 主线程 | | |
| 耗时时长 | | |
| 1000.0 | | CPU | | |  | |
| 主线程 | | |
| 耗时时长 | | |
| 改变伤害值（初始值10） | | 21 | | CPU | | 每一次造成8点伤害。 | |
| 主线程 | |
| 耗时时长 | |
| 1.0 | | CPU | | 受到3次Boss的攻击，每次伤害造成6点伤害。 | |
| 主线程 | |
| 耗时时长 | |
| 改变攻击范围（初始值100） | | 1.0 | | CPU | | 当受到攻击时会有瞬间的较大性能开销，只有当靠近Boss时才会受到伤害。 | |
| 主线程 | |
| 耗时时长 | |
| 1000.0 | | CPU | | 当受到攻击时可以看到有一个瞬间的脚本性能开销，此时即使我们离Boss很远也会受到伤害。 | |
| 主线程 | |
| 耗时时长 | |
| 结果分析 | | | | | | | |
| 经过以上测试，得出Boss的脚本仅仅只会在一瞬间对性能有较大的开销，但总体上来说没有对计算机产生很大的影响，总体性能开销并不是很大。但我们需要的效果能够很好地展现出来，因此该脚本是比较适合运用在该款游戏中的。 | | | | | | | |