造纸模拟器

测试报告

版本 <0.4>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <8/7/2019> | <0.1> | 第一次迭代 | 戴安东 |
| <15/7/2019> | <0.2> | 第二次迭代 | 李思洋 |
| <22/7/2019> | <0.3> | 第三次迭代 | 李思洋 |
| <29/7/2019> | <0.4> | 第四次迭代 | 戴安东 |

目录

1. 简介 4

1.1 目的 4

1.2 范围 4

1.3 定义、首字母缩写词和缩略语 4

1.4 参考资料 4

1.5 概述 4

2. 测试概要 4

3. 测试环境 4

4. 测试结果及分析 4

4.1 需求覆盖率及缺陷分布 4

4.2 缺陷严重程度 5

5. 缺陷清单 5

5.1 功能性缺陷 5

5.2 非功能性缺陷 6

6. 测试结论与建议 6

测试报告

# 简介

## 目的

发现软件的错误，从而保证软件的质量。

## 范围

造纸模拟器的第四次迭代版本

## 定义、首字母缩写词和缩略语

Unity3d：一款制作游戏的引擎

VR：虚拟现实技术

模型：在unity3d软件中作为可以被操作的对象的存在

C#：一种脚本语言

## 参考资料

《造纸模拟器软件需求规约》，《造纸模拟器迭代报告》

## 概述

这篇文档针对第四次迭代的内容及所完成的功能进行测试，由于游戏软件比较特殊，不能采用常规的测试方法，所以使用人工方法对功能及可能出现的问题进行测试。

# 测试概要

2019年7月29日下午由李徐礼杰，戴安东测试并由戴安东撰写报告。测试方法是直接运行可执行程序，手动移动人物检验是否符合测试报告里的功能，以及对潜在的bug进行排查。

# 测试环境

系统：window 10家庭中文版 version:1809

处理器：Intel®Core™i7-8750H CPU @ 2.20GHz 2.21 GHz

内存:8GB

显卡：NVIDIA RTX 2060

# 测试结果及分析

## 需求覆盖率及缺陷分布

**表1 需求覆盖率及缺陷分布**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **测试内容** | | **测试**  **用例数** | **需求**  **覆盖率** | **缺陷数** | **缺陷率** | **备注** |
| **功能项** | **玩家瞬移** | 1 | 100% | 0 | 0% | 人物可以自由瞬移，可以站在地形上 |
| **碰撞检测** | 5 | 100% | 0 | 0% | 人物可以被地形上的障碍物阻挡 |
| **从仓库取出和捡起功能** | 14 | 100% | 0 | 0% | 可以实现所有仓库物品的加入和取出 |
| **砍下竹子** | 8 | 100% | 1 | 12.5% | 竹子可以被砍下，根据力度判断砍倒需要的次数，但是现阶段竹子不砍倒可以直接捡起来，是一个bug。  在碰撞的过程中会产生振动 |
| **扔出竹子** | 8 | 100% | 0 | 0% | 竹子被扔出去的时候可以和地形产生交互，碰到石灰池能正确触发函数 |
| **蒸煮交互** | 8 | 100% | 2 | 25% | 将泡过的竹子扔向桶，能正确触发函数，但是计时器会因为一些BUG被刷新，导致实际生成的煮过的竹子过多，出现问题。 |
| **舂臼交互** | 10 | 100% | 1 | 10% | 竹子一旦碰到盆会自动定位，被煮过的竹子在敲打时会发生变化，但是用力过猛会发生穿模导致物体沉入地底。。 |
| **荡料入帘** | 2 | 100% | 1 | 50% | 可以正常交互，但是没有加料也能荡出纸来 |
|  |  |  |  |  |  |
| **功能项小计** | 64 | 100% | 2 | 3.1% | 功能测试比较成功，初次迭代所有功能均被覆盖 |
| **非功**  **能项** | **性能** | 20 | 100% | 1 | 5% | 在90%的情况下，按钮响应时间不超过1s，头戴VR设备时目前操纵人物移动会带来眩晕感 |
| **可靠性** | 40 | 100% | 0 | 0% | 每运行5小时至多发生一次故障 |
|  | **非功能项小计** | 60 | 100% | 1 | 1.6% | 程序性能良好可靠 |
|  | **总计** | 124 | 100% | 3 | 2.4% | 程序大致达成了第三次迭代目标 |

## 缺陷严重程度

**表2 缺陷严重程度**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **严重程度** | **致命** | **严重** | **一般** | **细微** | **总数** |
| **缺陷个数** | 0 | 0 | 1 | 2 | 3 |
| **占缺陷百分比** | 0% | 0% | 33.3% | 66.7% | 100% |

# 缺陷清单

## 功能性缺陷

**表3 功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **功能模块** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
| 1 | Bug001 | 细微 | 人物移动 | 后退动画僵硬 | 人物后退时，动画僵硬不自然 | 1 |
| 2 | Bug002 | 一般 | 地形判定 | 地形设置缺陷 | 人物可以进入一些出不去的区域 | 2 |

表中相关项说明：

1. 缺陷编号：为每个缺陷赋予一个唯一的编号，可以通过此编号对缺陷进行跟踪。例如：Bug001。
2. 缺陷严重程度：缺陷可以根据严重程度分为以下几种情况。
3. 致命（fatal）：致命的错误，测试执行直接导致系统死机、蓝屏、挂起、或是程序非法退出；系统的主要功能或需求没有实现。
4. 严重（critical）：严重错误，系统的次要功能点或需求点没有实现；数据丢失或损坏。执行软件主要功能的测试用例导致系统出错，程序无法正常继续执行；程序执行过于缓慢或是占用过大的系统资源。
5. 一般（major）：不太严重的错误，这样的缺陷虽然不影响系统的基本使用，但没有很好地实现功能，没有达到预期的效果。如次要功能丧失，界面错误，打印内容、格式错误，提示信息不太正确，或用户界面太差，简单的输入限制未放在前台进行控制，删除操作未给出提示，操作时间长等。
6. 细微（minor）：一些小问题，对功能几乎没有影响，产品及属性仍可使用。如软件的实际执行过程与需求有较小的差异；程序的提示信息描述容易使用户产生混淆；有个别错别字、 文字排列不整齐；辅助说明描述不清楚，显示格式不规范，长时间操作未给用户进度提示，提示窗口文字未采用行业术语，可输入区域和只读区域没有明显的区分标志，系统处理未优化等。
7. 功能模块：所测试并出现该缺陷的功能模块名称。
8. 缺陷标题：描述缺陷的标题。
9. 缺陷描述：对缺陷的详细描述。描述缺陷的重现步骤和问题所在，描述需清晰明了，使研发人员可以按照描述将缺陷重现出来。必要时可以放上相应的截图，以便于研发人员理解问题所在。

## 非功能性缺陷

**表4 非功能性缺陷列表**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序**  **号** | **缺陷**  **编号** | **严重程度** | **测试类型** | **缺陷标题** | **缺陷描述** | **测试用例编号** |
| **1** | **Bug001** | **细微** | **VR体验** | **VR体验缺陷** | **用户头戴VR设备体验移动时会有不必要的眩晕感** | **1** |

# 测试结论与建议

本次测试结果总体良好。暴露出的问题有3个，均在上方列出，缺陷的严重性并不是很高，建议深入优化地形，避免出现陷阱区域，此外，对后退动作的动画作进一步的调整，使之足够流畅，对移动也要做出调整，使体验更佳。