|  |
| --- |
| Group\_M |
| Marisa Strike 测试文档 |
|  |

|  |
| --- |
| LarryMario  2015/1/3 |

目录

[1 引言 1](#_Toc407971303)

[1.1 编写目的 1](#_Toc407971304)

[1.2 项目信息 1](#_Toc407971305)

[1.3 项目背景 1](#_Toc407971306)

[1.4 参考资料 1](#_Toc407971307)

[2 测试任务概述 2](#_Toc407971308)

[2.1 目标 2](#_Toc407971309)

[2.2 需求与设计简述 2](#_Toc407971310)

[2.3 运行环境 2](#_Toc407971311)

[2.4 测试环境 2](#_Toc407971312)

[3 测试内容说明 3](#_Toc407971313)

[3.1 测试方案 3](#_Toc407971314)

[3.2 测试准备工作 3](#_Toc407971315)

[4 功能测试用例设计 3](#_Toc407971316)

[4.1 游戏逻辑模块测试 3](#_Toc407971317)

[4.2 装备模块测试 4](#_Toc407971318)

[4.3 商店模块测试 4](#_Toc407971319)

[5 性能测试用例设计 4](#_Toc407971320)

[5.1 压力测试 4](#_Toc407971321)

[5.2 负载测试 4](#_Toc407971322)

[5.3 强度测试 4](#_Toc407971323)

[6 评价准则 5](#_Toc407971324)

[6.1 测试范围 5](#_Toc407971325)

[6.2 测试准则 5](#_Toc407971326)

# 引言

## 编写目的

本文档是软件工程课程项目——2D横版过关游戏《Marisa Strike》的测试文档，内容包括对测试方案与测试用例的说明。

本报告撰写完毕后，由软件项目总监进行审查，由软件测试工程师进行阅读。

## 项目信息

* 项目名称：2D横版过关游戏《Marisa Strike》
* 项目提出者：LarryMario
* 项目开发者：LarryMario、ScottFoH、lizhen2013

## 项目背景

近年来，独立游戏越来越受到重视，优秀的独立游戏层出不穷。同时，相关的开发资源，即开发工具、开发教程、图像素材、音乐素材等也越来越多。同人游戏也是独立游戏的一个形式，本项目组即是想在这样一个环境下，尝试进行游戏开发，积累开发经验。

本游戏以《魂斗罗》、《合金弹头》等横版过关游戏为游戏模式原型，遵循《东方Project》的剧情相关设定，加入装备与道具系统，旨在制作出一款具有较强可玩性、重玩性的东方Project同人游戏。

本开发小组旨在通过这次游戏开发，积累一定的游戏开发经验，并为《东方Project》二次同人事业的发展做出自己的贡献。

## 参考资料

* Shari Lawrence Pfleeger, Joanne M.Atlee. *Software Engineering*: *Theory and Practice, Fourth Edition*
* GBT 11457-2006 信息技术 软件工程术语

## 术语表

* 场景：特指Unity引擎中的“Scene”。Unity以1个场景为基本运作范围。
* 游戏对象：特指Unity引擎中的“GameObject”，指使用Unity开发的游戏中的一个独立的对象，例如摄像机、一个方块、角色等。
* 部件：特指Unity引擎中的“Component”，指挂载在游戏对象上的功能性单位，可以赋予游戏对象特定的功能。

# 测试任务概述

## 目标

本项目的测试目标有以下几点：

* 检查游戏系统是否有违背需求之处；
* 找出游戏系统中存在的逻辑错误；
* 找出游戏系统中的性能问题；

## 需求与设计简述

本项目为东方Project同人游戏——《Marisa Strike》，游戏类型为2D横板过关动作游戏。玩家能够控制游戏中主角移动、攻击。游戏的简单人工智能系统能自动控制敌人移动、攻击，游戏相关数据会得到记录，必要的数据会显示在游戏画面上。游戏提供装备系统，能够使主角做出不同的动作、使用不同的攻击。游戏提供商店系统，玩家可购买道具以在游戏中使用。游戏提供自动存档系统，在特定的时间点游戏会自动进行存档，记录当前的游戏进度及道具拥有情况。

本游戏系统分为4个层次：展示层、游戏逻辑层、数据访问层、数据层。

展示层由若干“场景”组成，通过Unity引擎的UI系统与渲染系统展示在游戏窗口中。

游戏逻辑层由若干接口与若干控制器实体类组成，设置接口的目的在于利用Unity的消息系统与事件系统，使类似的实体类（如不同的道具或不同的敌人），需要与其他类型的实体类通行时，可以以一个相同的身份互相交流。

数据访问层由若干模型实体类，以及数据访问中间件构成。模型实体类负责承载来自上下两层的离散数据，组成有意义的实体，以供游戏逻辑层使用，或存储入数据层中。

数据层由SQLite数据库构成。数据库均为离线形式，以文件方式存储在游戏玩家的游戏数据目录内。

## 运行环境

* 硬件环境：2014年市场主流中档计算机硬件配置或以上
* 软件环境：安装了.NET Framework 2.0或以上版本的Windows XP，或Windows Vista/7/8。

## 测试环境

* 硬件环境

配置一：

* + CPU：Intel(R) Core(TM) i5-3230M
  + 内存：4.00 GB
  + 显卡：NVIDIA GeForce GT 630M

配置二：

* + CPU：Intel(R) Core(TM) i3-3120M
  + 内存：4.00GB
  + 显卡：NVIDIA GeForce GT 740M

配置三：

* + CPU：Intel(R) Core(TM) i5-2450M
  + 内存：2.00GB
  + 显卡：NVIDIA GeForce GT 610M
* 软件环境
  + 操作系统：Windows 7 Professional 64-bits SP1
  + 开发环境：Unity 4.6.1f1

# 测试内容说明

## 测试方案

测试以黑盒测试为主。计划进行以下类型的测试：

* 功能测试。对每个模块分别进行测试，需要测试的有正常游戏操作、正常游戏流程、游戏对象间通信、边界情况判定、数据库内武器信息读出、数据库内存档信息读出与写入。
* 性能测试。分为三项：压力测试，在正常游戏过程中较大压力情况下，游戏系统的性能；负载测试，计算机在高负载情况下，游戏系统运行的正常性；强度测试：计算机高负载情况下，刻意营造一个极端的游戏过程，测试游戏系统在怎样的情况下会失效，甚至崩溃。

## 测试准备工作

* 所有测试人员在场，包括项目总监、软件工程师、数据库工程师、测试工程师。
* 保证使用的测试工具运行正常。
* 进行非负载测试及强度测试时，测试用计算机不要运行其他大型程序。
* 准备好测试用记录表格，以便对测试结果进行记录。

# 功能测试用例设计

## 游戏逻辑模块测试

测试用例如表4-1。

表格4-1游戏逻辑模块测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **测试编号** | **测试类型** | **测试内容** | **预期结果** |
| AA001 | 基本操作 | 按住A或D键，查看角色动作 | A:角色向左移动  D:角色向右移动 |
| AA002 | 移动状态下按住J键，查看角色动作 | A→J或D→J:角色在移动状态下射击 |
| AA003 | 按住W键，查看角色动作 | W:角色向上看 |
| AA004 | 按住S键，查看角色动作 | S:角色下蹲 |
| AA005 | 按住K+(A或D)，查看角色动作 | K+A:角色向左跳  K+D:角色向右跳 |
| AA006 | 跳起（按K）后，按W，查看角色动作 | K→W:角色向上看 |
| AA007 | 跳起（按K）后，按S，查看角色动作 | K→S:角色向下看 |
| AA008 | 按住J，查看角色动作 | J:角色发射子弹 |
| AA009 | 按住J+W，查看角色动作 | J+W:角色向上发射子弹 |
| AA010 | 按住J+S，查看角色动作 | J+S:角色向下发射子弹 |
| AA011 | 跳起（按K）后， 按住J+W，查看角色动作 | K→J+W:跳跃状态下角色向上发射子弹 |
| AA012 | 跳起（按K）后， 按住J+S，查看角色动作 | K→J+S:跳跃状态下角色向下发射子弹 |
|  |
| **测试编号** | **测试类型** | **测试内容** | **预期结果** |
| AA013 | 基本操作 | 连续快速按下两次A键，查看角色动作 | A→A:角色向左冲刺 |
| AA014 | 连续快速按下两次D键，查看角色动作 | D→D:角色向右冲刺 |
| AA015 | 冲刺状态下，按住K键，查看角色动作 | A→A→K或D→D→K:冲刺状态下起跳，速度与冲刺状态下相同 |
| AA016 | 冲刺状态下，按住J键，  查看角色动作 | A→A→J或D→D→J:速度恢复到正常状态移动射击 |
| AA017 | 冲刺跳状态，按住J键，查看角色动作 | A→A→K→J或D→D→K→J: 速度恢复到正常状态在空中移动射击 |
| AA018 | 正常速度下从平台落下（在平台边缘时按住A或D键），查看角色状态 | 角色在正常下落的同时以正常速度向左或向右偏移 |
| AA019 | 冲刺状态下从平台落下，查看角色状态 | 角色在正常下落的同时以冲刺状态下的速度向左或向右偏移 |
| AB001 | 攻击与受伤相关事件 | 子弹打到“墙壁”（物体） | 子弹消失 |
| AB002 | 子弹打到敌人 | 子弹消失，敌人进行伤害判定，敌人会出现闪烁的效果，并损失相应HP |
| AB003 | 敌人的子弹打到角色 | 子弹消失，角色会出现被击退效果，并出现小段时间的无敌状态 |
| AB004 | 敌人的子弹在角色无敌时间内打到角色身上 | 子弹消失，不对玩家造成伤害 |
| AB005 | 角色承受的一定的攻击次数后 | 角色死亡 |
| AB006 | 攻击与受伤相关事件 | 角色受到地形伤害（如落入深坑） | 角色直接死亡，无视角色无敌状态 |
| AC001 | 道具相关 事件 | 击杀部分特定的敌人 | 敌人死亡后有相应的道具掉落 |
| AC002 | 拾取增益型道具 | 根据类型可能出现情况：   1. 角色恢复一定的血量 2. 角色提升一定的攻击 3. 角色的攻击形式变化 |
| AC003 | 拾取装备型道具 | 将该武器暂时存入临时的列表；过关后，会在角色拥有的武器列表上增加该武器 |
| AC004 | 拾取过关道具 | 出现过关画面 |
| AD001 | 边界判定 | 子弹以极快的速度打向目标 | 子弹击中目标，而非穿过目标 |
| AD002 | 角色的碰撞体与地形边界刚好擦过时 | 动作符合自然逻辑 |
| AE001 | 场景控制 相关 | 按ESC键 | 游戏暂停，出现菜单界面  点击界面中的“继续”按钮  则回到游戏画面继续游戏  点击界面中的“退出”按钮  则回到开始界面 |
| AE002 | 角色刚进入关卡 | 屏幕上回显示开始字样 |
| AE003 | 角色死亡 | 玩家无法控制角色，角色由系统接管，系统结束关卡 |
| AE004 | 角色过关 | 玩家无法控制角色，角色由系统接管，系统结束关卡 |
| AF001 | 自动存档 | 玩家成功通过一关 | 将角色获取的道具存入数据库中“角色拥有的道具”表中。  将关卡信息，角色状态存入数据库的“存档”表中 |

## 装备模块测试

测试用例如表4-2。

表格4-2装备模块测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **测试编号** | **测试类型** | **测试内容** | **预期结果** |
| BA001 | 浏览装备栏 | 显示内容 | 在界面内正常显示装备和信息 |
| BA002 | 按A、D（左右键）翻页 | 装备栏显示前一页或后一页装备 |
| BA003 | 按W、S（上下键）选择装备 | 在当前显示的页内选择装备 |
| BA004 | 选中装备后按确认键（Z键） | 选中装备并在相应位置显示该装备信息 |
| BB001 | 穿戴装备 | 选择穿戴装备 | 显示玩家装备后属性变化 |
| BB002 | 取消穿戴 | 取消选中可按W、S（上下键）继续选择 |
| BC001 | 退出装备栏 | 按X键 | 退回主菜单 |

## 商店模块测试

测试用例如表4-3。

表格4-3商店模块测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **测试编号** | **测试类型** | **测试内容** | **预期结果** |
| CA001 | 浏览商店 | 显示商店内容 | 在界面内正常显示商品和信息 |
| CA002 | 按A、D（左右键）翻页 | 商店显示前或后一页商品 |
| CA003 | 按W、S（上下键）选商品 | 在当前显示的页内选择商品 |
| CA004 | 选中商品后按确认键（Z键） | 选中商品并在相应位置显示该商品信息 |
| CB001 | 购买商品 | 选中商品后按确认键（Z键）后选择购买操作 | 提示购买商品成功，商品信息添加进数据库，装备页中显示购买的新商品 |
| CB002 | 选中商品后按确认键（Z键）后选择取消操作 | 取消选中可按W、S（上下键）继续选择 |
| CC001 | 退出商店 | 按X键 | 退回主菜单 |

# 性能测试用例设计

## 压力测试

测试用例如表4-4。

表格4-4压力测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **测试编号** | **测试类型** | **测试内容** | **预期结果** |
| DA001 | 玩家射击大量子弹的时候 | 同屏同时存在200发子弹 | 帧数（FPS）维持在60 |
| DA002 | 同屏同时存在500发子弹 | 帧数（FPS）维持在45 |
| DA003 | 同屏同时存在2000发子弹 | 帧数（FPS）维持在20 |
| DB001 | 同屏存在大量的敌人 | 同屏同时存在20个敌人 | 帧数（FPS）维持在60 |
| DB002 | 同屏同时存在50个敌人 | 帧数（FPS）维持在45 |
| DB003 | 同屏同时存在200个敌人 | 帧数（FPS）维持在20 |
| DC001 | 同屏存在大量的敌人并向玩家射击（一秒6发子弹） | 同屏同时存在20个敌人并攻击 | 帧数（FPS）维持在60 |
| DC002 | 同屏同时存在50个敌人并攻击 | 帧数（FPS）维持在30 |
| DC003 | 同屏同时存在200个敌人并攻击 | 帧数（FPS）维持在15 |

## 负载测试

测试用例如表4-5。

表格4-4压力测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **测试编号** | **测试类型** | **测试内容** | **预期结果** |
| EA001 | CPU高负载 | CPU负载50% | 帧数（FPS）维持在60左右 |
| EA002 | CPU负载75% | 帧数（FPS）维持在45左右 |
| EA003 | CPU负载95% | 帧数（FPS）维持在20左右 |
| EB001 | 内存高负载 | 内存负载70% | 帧数（FPS）维持在60左右 |
| EB002 | 内存负载90% | 帧数（FPS）维持在30左右 |
| EB003 | 内存负载99% | 游戏无法运行 |
| EC001 | CPU和内存同时高负载 | CPU负载50%，内存负载70% | 帧数（FPS）维持在45左右 |
| EC002 | CPU负载50%，内存负载90% | 帧数（FPS）维持在20左右 |
| EC003 | CPU负载50%，内存负载99% | 游戏无法运行 |
| EC004 | CPU负载75%，内存负载70% | 帧数（FPS）维持在30左右 |
| EC005 | CPU负载75%，内存负载90% | 帧数（FPS）维持在15左右 |
| EC006 | CPU负载75%，内存负载99% | 游戏无法运行 |
| EC007 | CPU负载95%，内存负载70% | 帧数（FPS）维持在25左右 |
| EC008 | CPU负载95%，内存负载90% | 帧数（FPS）维持在10左右 |
| EC009 | CPU负载95%，内存负载99% | 游戏无法运行 |

## 强度测试

测试用例如表4-5。

表格4-5压力测试用例

|  |  |  |  |
| --- | --- | --- | --- |
| **测试编号** | **测试类型** | **测试内容** | **预期结果** |
| FA001 | CPU高负载时玩家射击大量子弹 | CPU负载75%同屏同时存在500发子弹 | 帧数（FPS）维持在30左右 |
| FA002 | CPU负载75%同屏同时存在2000发子弹 | 帧数（FPS）维持在10左右 |
| FA003 | CPU负载95%同屏同时存在500发子弹 | 帧数（FPS）维持在10左右 |
| FA004 | CPU负载95%同屏同时存在2000发子弹 | 游戏无法运行 |
| FB001 | 内存高负载时玩家射击大量子弹 | 内存负载90%同时存在500发子弹 | 帧数（FPS）维持在20左右 |
| FB002 | 内存负载99%同时存在500发子弹 | 游戏无法运行 |
| FB003 | 内存负载90%同屏同时存在2000发子弹 | 帧数（FPS）维持在10左右 |
| FB004 | 内存负载99%同屏同时存在2000发子弹 | 游戏无法运行 |
| FC001 | CPU和内存同时高负载，同时出现敌人 | CPU负载75%，内存负载90%同屏同时存在50个敌人 | 帧数（FPS）维持在20左右 |
| FC002 | CPU负载75%，内存负载99%同屏同时存在50个敌人 | 游戏无法运行 |
| FC003 | CPU负载75%，内存负载90%同屏同时存在200个敌人 | 游戏无法运行 |
| FC004 | CPU负载75%，内存负载99%同屏同时存在200个敌人 | 游戏无法运行 |
| FC005 | CPU负载95%，内存负载90%同屏同时存在50个敌人 | 帧数（FPS）维持在10左右 |
| FC006 | CPU负载95%，内存负载99%同屏同时存在50个敌人 | 游戏无法运行 |
| FC007 | CPU负载95%，内存负载90%同屏同时存在200个敌人 | 游戏无法运行 |
| FC008 | CPU负载95%，内存负载99%同屏同时存在200个敌人 | 游戏无法运行 |

# 评价准则

* 游戏中测试的按键可修改，测试时以实际映射的按键为准。标准按键如下：
  + A：左键。角色面向左方时同时代表前键，右方时代表后键。
  + D：右键。角色面向左方时同时代表后键，右方时代表前键。
  + W：上键。
  + S：下键
  + J：攻击键
  + K：跳跃键
  + Escape：暂停键或返回键
* 功能性测试结果分为四档：
  + 通过：运行正常，结果完全如预期。
  + 部分通过：运行基本正常，但结果稍有偏差。
  + 不通过：运行出现问题，结果有较大偏差
  + 失败：出现严重失效现象，与结果完全不符。
* 游戏性能测试一帧数作为主要标准，数值越高代表运行越流畅。
  + 游戏运行帧数低于5帧时，视为游戏无法运行。