软件测试报告

--基于微信开发者工具开发的飞翔的小鸟游戏



课 程： 软件工程导论

名 称： Flappy Bird

人 员: G17小组

**7.3软件测试报告(STR)**

**说明：**

1.《软件测试报告》(STR)是对计算机软件配置项CSCl,软件系统或子系统，或与软件相关项目执行合格性测试的记录。

2.通过STR，需方能够评估所执行的合格性测试及其测试结果。

**版本记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编号 | 修订日期 | 版本/状态 | 修订人 | 备注 |
| 1 | 2020/12/28 | 0.1 | 陈骁，李以昕 | 初稿 |
| 2 | 2020/12/30 | 0.2 | 陈骁，李以昕 | 初稿改进 |
| 3 | 2020/12/30 | 0.3 | 陈骁，李以昕 | 初稿改进 |

目录

[1引言 5](#_Toc60261715)

[1.1标识 5](#_Toc60261716)

[1.2系统概述 5](#_Toc60261717)

[1.3文档概述 6](#_Toc60261718)

[1.4基线 6](#_Toc60261719)

[2引用文件 6](#_Toc60261720)

[3测试概要 6](#_Toc60261721)

[3.1测试环境 7](#_Toc60261722)

[3.1.1时间与地点 7](#_Toc60261723)

[3.1.2人员与设备 7](#_Toc60261724)

[4测试 8](#_Toc60261725)

[4.1单元测试 8](#_Toc60261726)

[4.1.1测试1（开始游戏前选择的单元测试） 8](#_Toc60261727)

[4.1.1.1开始游戏前选择 8](#_Toc60261728)

[4.1.2测试2（游戏逻辑的单元测试） 9](#_Toc60261729)

[4.1.2.1小鸟获得金币 9](#_Toc60261730)

[4.2.2.2小鸟避障 11](#_Toc60261731)

[4.2.2.3小鸟跳跃 13](#_Toc60261732)

[4.1.3测试3（游戏结算的单元测试） 15](#_Toc60261733)

[4.1.3.1游戏结算 15](#_Toc60261734)

[4.1.4测试4（好友功能的单元测试） 16](#_Toc60261735)

[4.1.4.1发送爱心 16](#_Toc60261736)

[4.1.4.2举报功能 17](#_Toc60261737)

[4.1.5测试5（商店购买功能的单元测试） 18](#_Toc60261738)

[4.1.5.1角色购买 18](#_Toc60261739)

[4.1.5.2皮肤购买 19](#_Toc60261740)

[4.1.5.3地图购买 21](#_Toc60261741)

[4.1.6测试6（排行榜功能的单元测试） 23](#_Toc60261742)

[4.1.6.1排行榜功能 23](#_Toc60261743)

[4.1.7测试7（角色功能的单元测试） 24](#_Toc60261744)

[4.1.7.1角色功能 24](#_Toc60261745)

[4.1.8测试8（收件箱功能的单元测试） 25](#_Toc60261746)

[4.1.8.1点击邮件 25](#_Toc60261747)

[4.1.8.2点击删除已读邮件 25](#_Toc60261748)

[4.1.8.3点击已读全部邮件 26](#_Toc60261749)

[4.1.9测试9（管理员功能的单元测试） 27](#_Toc60261750)

[4.1.9.1账号管理 27](#_Toc60261751)

[4.1.9.2奖励发送 28](#_Toc60261752)

[4.1.9.3举报处理 29](#_Toc60261753)

[4.2集成测试 30](#_Toc60261754)

[4.2.1游戏模块集成测试 30](#_Toc60261755)

[4.2.2好友模块集成测试 31](#_Toc60261756)

[4.2.3商店模块集成测试 32](#_Toc60261757)

[4.2.4好友模块集成测试 32](#_Toc60261758)

[4.2.5好友模块集成测试 32](#_Toc60261759)

[4.2.6好友模块集成测试 32](#_Toc60261760)

[4.3系统测试 33](#_Toc60261761)

[4.4验收测试 33](#_Toc60261762)

[4.5测试用例 33](#_Toc60261763)

[5软件功能概述 33](#_Toc60261764)

[5.1游戏 33](#_Toc60261765)

[5.2商店与角色 33](#_Toc60261766)

[5.3好友以及收件箱 34](#_Toc60261767)

[5.4排行榜 34](#_Toc60261768)

[5.5管理员 34](#_Toc60261769)

[6总体分析 35](#_Toc60261770)

[6.1功能分析 35](#_Toc60261771)

[6.2存在缺陷 35](#_Toc60261772)

[6.3后续补救 35](#_Toc60261773)

[6.4评价与结论 35](#_Toc60261774)

[7注解 35](#_Toc60261775)

[附录 37](#_Toc60261776)

**软件测试计划的正文的格式如下：**

## 1引言

本章应分成以下几条。

### 1.1标识

|  |  |  |
| --- | --- | --- |
| 文件状态：  [ ] 草稿  [ ] 正式发布  [√] 正在修改 | 文件标识： | SE2020-G17-软件测试计划(STP) |
| 当前版本： | 0.2 |
| 作者： | 陈骁 |
| 编写日期： | 2020-12-08 |

### 1.2系统概述

用途:

本产品是一款主要面向20-30岁有许多碎片时间的游戏爱好者的微信小程序，玩家可以在游戏中搜集拥有不同属性和技能的角色以及相应的皮肤，同时还可以体会到不同难度的游戏内容和地图。

一般特性:

基于微信小程序的休闲类的角色搜集手游

开发、运行和维护的历史:

开发历史:

|  |  |
| --- | --- |
| 时间 | 内容 |
| 2020/10/1—2020/10/15 | 项目开发方向及形式确定 |
| 2020/10/19—2020/10/27 | 项目计划书 |
| 2020/10/28—2020/11/6 | 可行分析报告 |
| 2020/11/8—2020/11/29 | 软件需求说明书 |
| 2020/11/29—2020/12/6 | 用户手册 |
| 2020/12/6—2020/12/16 | 项目设计 |
| 2020/12/16-2020/12/28 | 项目实现 |

暂无运行和维护的历史

投资方:G17组全体成员

需方:杨枨老师

用户: 20-30岁有许多碎片时间的游戏爱好者

开发方:G17组全体成员

### 1.3文档概述

本文档主要用于对本项目的软件设计说明，明确开发时应注意的要点，以及作为设计的说明书供需方和开发人员参考。

本文档在开发期间由SE2020-G17组全体成员编写，仅供SE2020-G17小组使用，不应进行商业性传播或为其他人员使用。

该软件设计说明文档对《基于微信开发者工具开发的飞翔的小鸟游戏》软件做了全面细致的设计说明文档。

本书的预期读者为评审组成员，项目组成员，当然用户和关键用户组

### 1.4基线

项目计划书 -基于微信开发者工具开发的飞翔的小鸟游戏1.2

可行性报告-基于微信开发者工具开发的飞翔的小鸟游戏1.2

软件需求说明-基于微信开发者工具开发的飞翔的小鸟游戏1.0

项目设计-用户手册1.0

项目设计-软件(结构)设计说明1.0

项目设计-数据库(顶层)设计说明1.1.

项目设计-系统设计(结构设计)说明1.0

## 2引用文件

本章应列出本文档引用的所有文档的编号、标题、修订版本和日期。本章也应标识不能通过正常的供货渠道获得的所有文档的来源。

[1]GB+T-8567-2006计算机软件文档编制规范 11 - 软件测试计划(STP)

[2]SE2020-G17-项目介绍正式版1.0.docx

[3]SE2020-G17-项目计划书1.2.docx

[4]SE2020-G17-可行性分析1.2.docx

[5]SE2020-G17-软件需求说明1.0.docx

[6]SE2020-G17-数据字典.docx

[7]SE2020-G17-用户手册1.0

[8]SE2020-G17-软件(结构)设计说明1.0

[9]SE2020-G17-数据库(顶层)设计说明1.1

[10]SE2020-G17-系统设计(结构设计)说明1.0

[11] 张海藩编著 《软件工程导论》（第六版） 清华大学出版社

## 3测试概要

### 3.1测试环境

硬件环境：三台Windows10操作系统的电脑，两台安卓手机，一台苹果手机

软件环境：Windows 10 操作系统 ，Android操作系统 ，IOS操作系统WeTest，postman，微信开发者工具，

#### 3.1.1时间与地点

|  |  |  |
| --- | --- | --- |
| 测试时间 | 测试项目 | 测试地点 |
| 12.30 | 能否搜索到小程序，正常登陆，正常邀请好友 | 理四508 |
| 12.31 | 游戏有无BUG， | 理四508 |
| 12.31 | 角色，商店功能是否正常使用 | 理四508 |
| 1.1 | 分数，排行榜是否正常统计 | 理四508 |
| 1.1 | 管理员部分功能测试 | 理四508 |
| 1.2 | 好友及邮箱功能 | 理四508 |
| 1.3 | 整体检测 | 理四508 |

#### 3.1.2人员与设备

**计算机硬件**：CPU为(英特尔)Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz(2201 MHz)；内存为16.00 GB (3300 MHz)；固态硬盘130G；移动硬盘1T；显卡为NVIDIA GeForce GTX 1060；声卡为Realtek High Definition Audio；网卡为Realtek 8821CE Wireless LAN 802.11ac PCI-E NIC。

**接口设备**：三台个人Windows10操作系统的电脑；两台安卓系统手机;一台苹果手机

**通信设备**：三台个人Windows10操作系统的电脑；两台安卓系统手机路由器；中国移动固定基站等。

**测试数据归约设备**：三台个人Windows10操作系统的电脑

**仪器设备**：三台个人Windows10操作系统的电脑；两台安卓系统手机。

**人员：**组内三个成员（必要时会和用户进行相关沟通和测试）

**测试计时设备**：两台安卓系统手机;一台苹果手机。

**测试事件记录仪**：两台安卓系统手机;一台苹果手机。

所有与硬件及固件项有关的测试，软件项目开发方均遵循严格的保密措施，对于其他可能的保密性与私密性问题，也将严格遵守相关法律法规，妥善处理。

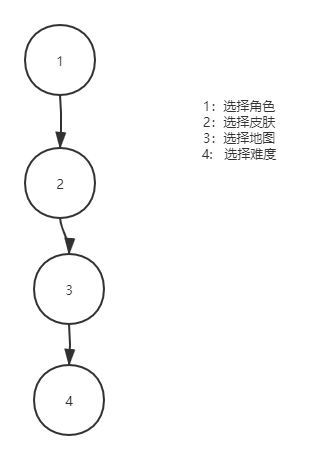
## 4测试

本章应描述计划测试的总范围并分条标识，并且描述本STR适用的每个测试。

### 4.1白盒测试

#### 4.1.1测试1（开始游戏前选择的白盒测试）

##### 4.1.1.1开始游戏前选择



首先根据V(G)=E-N+2计算出环形复杂度是1

然后再根据V(G)=P+1计算出环形复杂度是1，那么流图是正确的

路径1：1-2-3-4

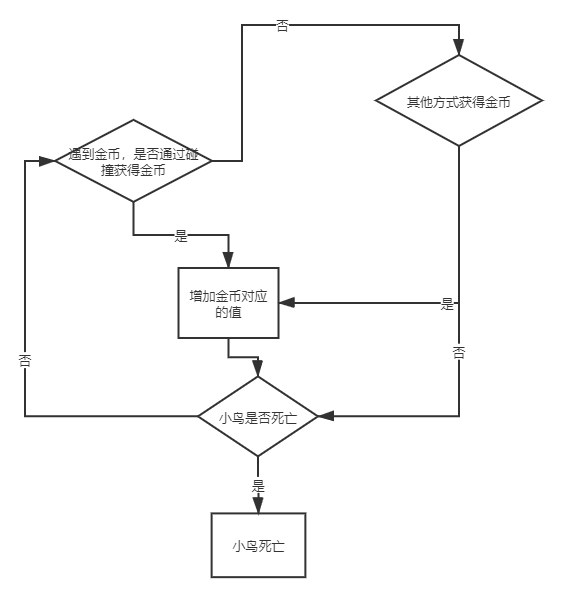
路径1：1-2-3-4

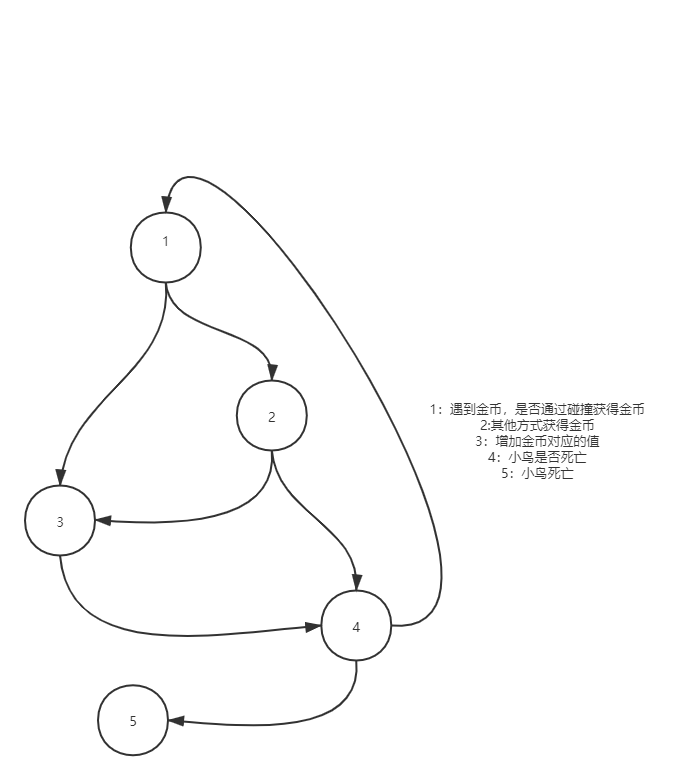
输入数据：玩家选择角色，选择皮肤，选择地图，选择难度

预期结果：玩家能够选择角色，选择皮肤，选择地图，选择难度在游戏中体现

#### 4.1.2测试2（游戏逻辑的白盒测试）

##### 4.1.2.1小鸟获得金币





首先根据V(G)=E-N+2计算出环形复杂度是4

然后再根据V(G)=P+1计算出环形复杂度是4，那么流图是正确的

路径1：1-3-4-5

路径2：1-2-4-5

路径3：1-2-3-4-5

路径4：1-2-4-1-3-4-5

路径1：1-3-4-5

输入数据：小鸟第一次碰撞到金币，然后死亡

预期结果：金币增加一次，小鸟死亡

路径2：1-2-4-5

输入数据：小鸟第一次没获得金币，然后死亡

预期结果：金币不增加，小鸟死亡

路径3：1-2-3-4-5

输入数据：小鸟第一次没碰撞到金币，通过其他方式获得金币，然后死亡

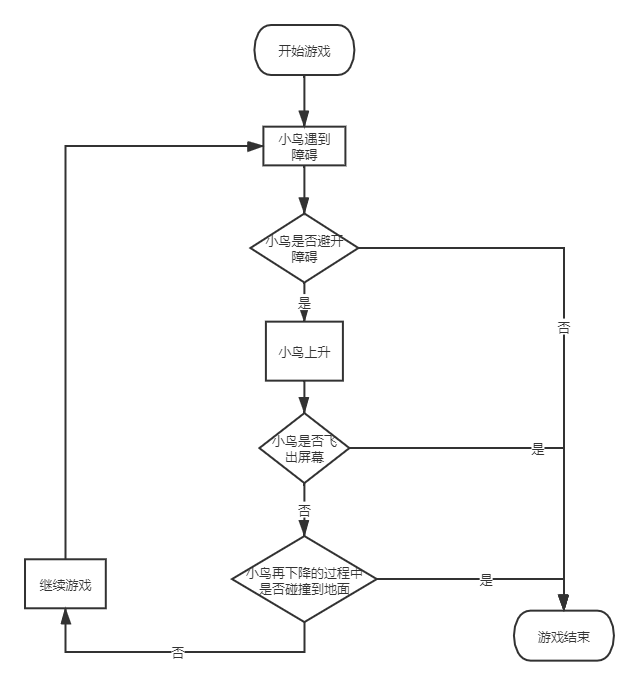
预期结果：金币增加一次，小鸟死亡

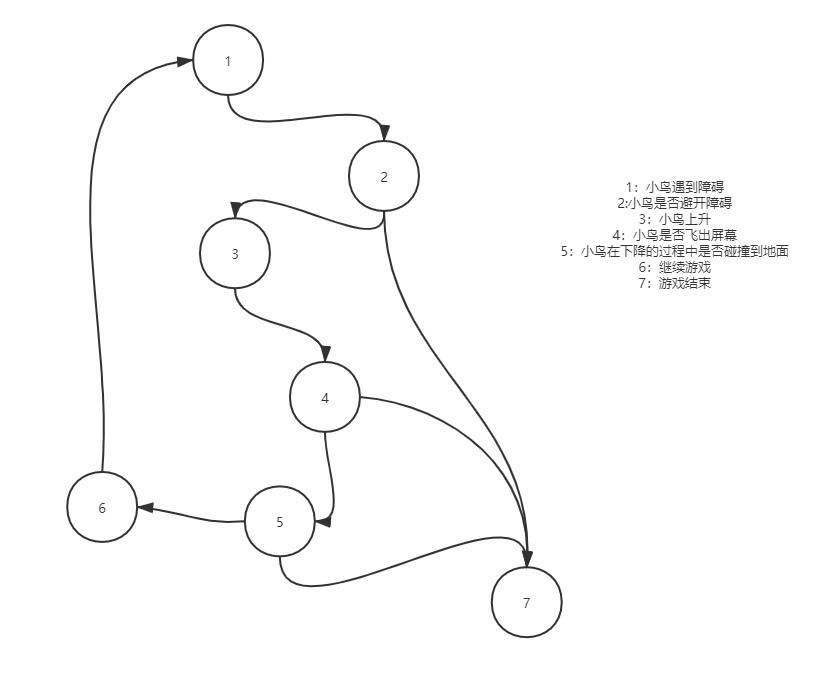
路径4：1-2-4-1-3-4-5

输入数据：小鸟第一次没有获得金币，第二次通过碰撞获得金币，小鸟死亡

预期结果：小鸟第一次没获得金币，第二次获得金币，小鸟死亡

##### 4.2.2.2小鸟避障





首先根据V(G)=E-N+2计算出环形复杂度是4

然后再根据V(G)=P+1计算出环形复杂度是4，那么流图是正确的

路径1：1-2-7

路径2：1-2-3-4-7

路径3：1-2-3-4-5-7

路径4：1-2-3-4-5-6-1-2-7

路径1：1-2-7

输入数据：小鸟没通过第一个障碍

预期结果：小鸟没通过第一个障碍死亡

路径2：1-2-3-4-7

输入数据： 小鸟通过第一个障碍而且飞出屏幕

预期结果：小鸟通过第一个障碍但是飞出屏幕死亡

路径3：1-2-3-4-5-7

输入数据：小鸟通过第一个障碍但是撞击到地面

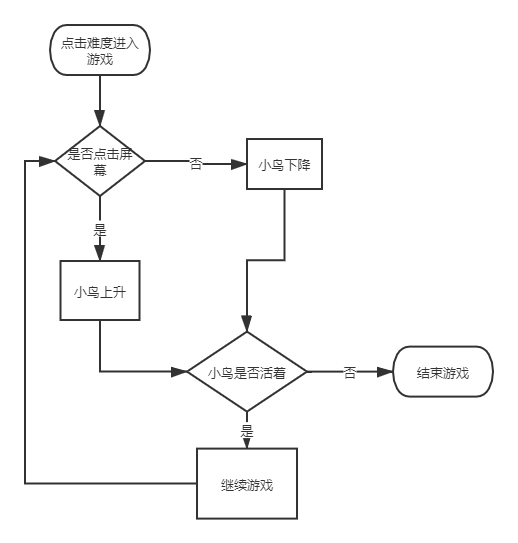
预期结果：小鸟通过第一个障碍但是撞击到地面死亡

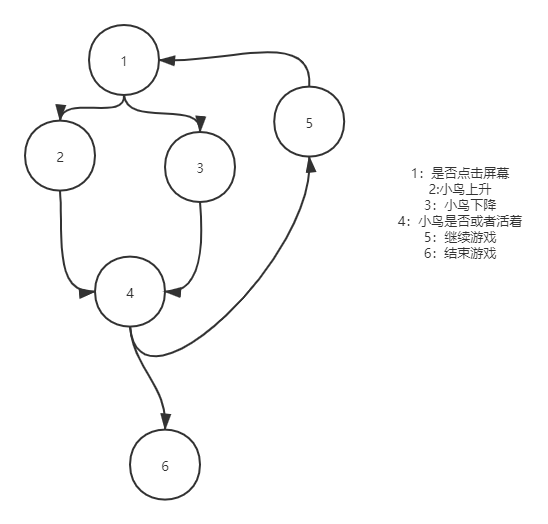
路径4：1-2-3-4-5-6-1-2-7

输入数据：小鸟通过第一个障碍而且在第二个障碍的时候碰撞

预期结果：小鸟通过第一个障碍但是在第二个障碍的时候碰撞死亡

##### 4.2.2.3小鸟跳跃





首先根据V(G)=E-N+2计算出环形复杂度是3

然后再根据V(G)=P+1计算出环形复杂度是3，那么流图是正确的

路径1：1-2-4-6

路径2：1-3-4-6

路径3：1-2-4-5-1-3-4-6

路径1：1-2-4-6

输入数据：点击屏幕

预期结果：小鸟点击屏幕后上升飞出屏幕或者碰到水管死亡结束游戏

路径2：1-3-4-6

输入数据： 不点击屏幕

预期结果：小鸟不点击屏幕后下降碰到水管或者碰撞地面死亡结束游戏

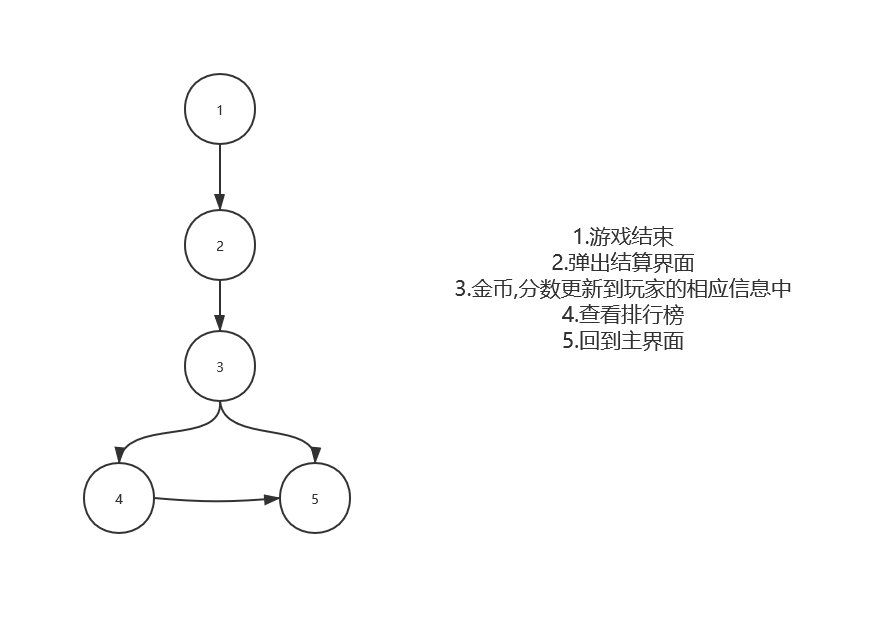
路径3：1-2-4-5-1-3-4-6

输入数据：点击一下屏幕

预期结果：小鸟点击屏幕上升后开始下降降碰到水管或者碰撞地面死亡结束游戏

#### 4.1.3测试3（游戏结算的白盒测试）

##### 4.1.3.1游戏结算



首先根据V(G)=E-N+2计算出环形复杂度是2

然后再根据V(G)=P+1计算出环形复杂度是2，那么流图是正确的

语句覆盖：

路径1：1-2-3-4

路径2：1-2-3-5

路径1：1-2-3-4

输入数据：游戏结束弹出结算界面后玩家点击排行榜

预期结果：排行榜界面正确显示

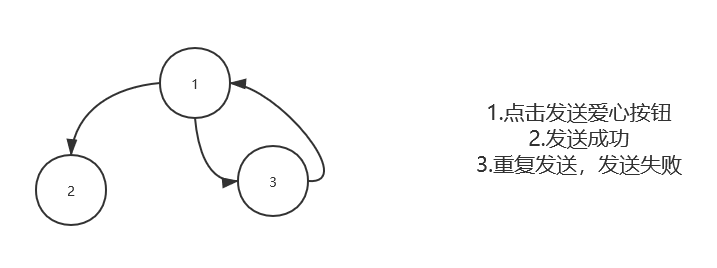
路径2：1-2-3-5

输入数据：游戏结束弹出结算界面后玩家点击主菜单按钮

预期结果：回到主菜单

#### 4.1.4测试4（好友功能的白盒测试）

##### 4.1.4.1发送爱心



首先根据V(G)=E-N+2计算出环形复杂度是2

然后再根据V(G)=P+1计算出环形复杂度是2，那么流图是正确的

语句覆盖：

路径1: 1-2

路径2: 1-3-1

路径1: 1-2

输入数据：点击发送爱心

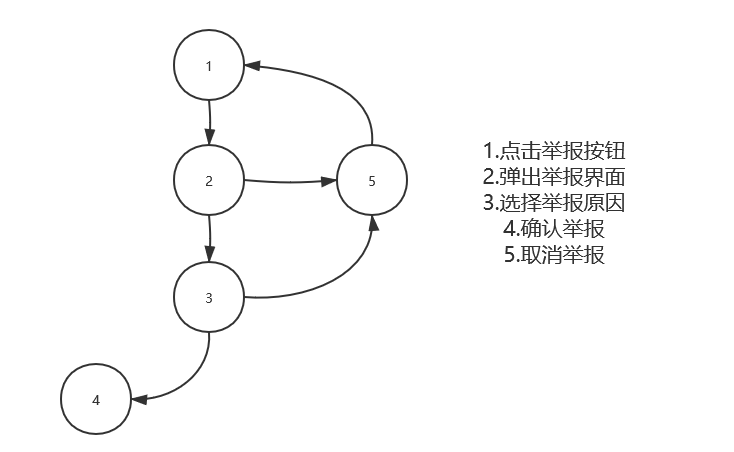
预期结果：成功发送

路径2: 1-3-1

输入数据：多次点击发送爱心

预期结果：第一次发送爱心成功发送，后面提示发送失败

##### 4.1.4.2举报功能



首先根据V(G)=E-N+2计算出环形复杂度是3

然后再根据V(G)=P+1计算出环形复杂度是3，那么流图是正确的

语句覆盖：

路径1: 1-2-3-4

路径2: 1-2-3-5-1

路径3: 1-2-5-1

路径1: 1-2-3-4

输入数据：点击举报按钮，选择举报原因后点击确认举报

预期结果：举报成功

路径2: 1-2-3-5-1

输入数据：点击举报按钮，选择举报原因后点击取消举报

预期结果：取消举报返回页面

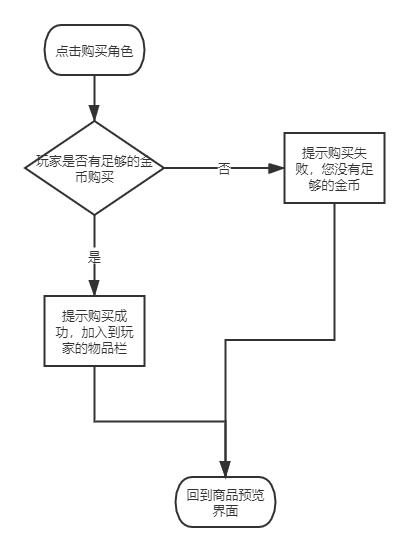
路径2: 1-2-5-1

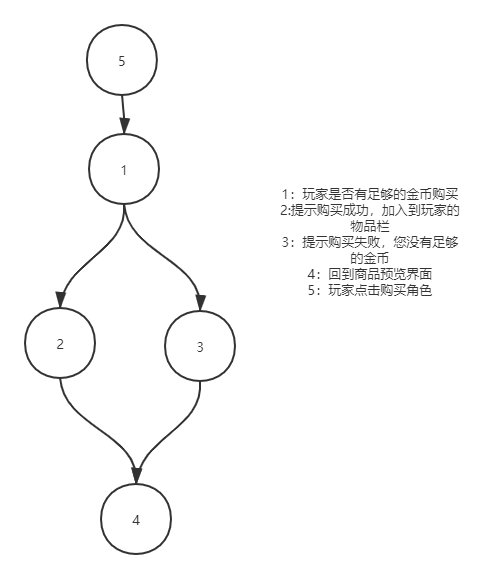
输入数据：点击举报按钮，未选择举报原因便点击取消举报

预期结果：取消举报返回页面

#### 4.1.5测试5（商店购买功能的白盒测试）

##### 4.1.5.1角色购买





首先根据V(G)=E-N+2计算出环形复杂度是2

然后再根据V(G)=P+1计算出环形复杂度是2，那么流图是正确的

语句覆盖：

路径1：5-1-2-4

路径2：5-1-3-4

路径1：5-1-2-4

输入数据：玩家点击购买角色

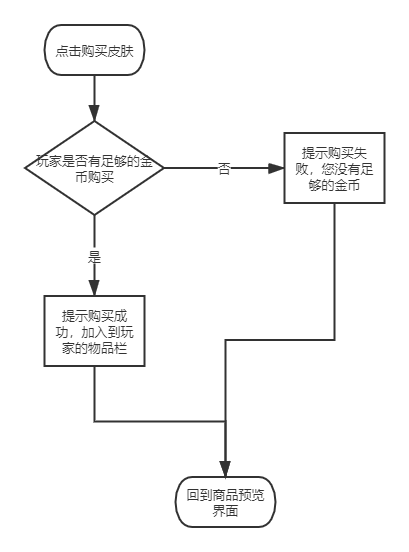
预期结果：玩家成功购买角色

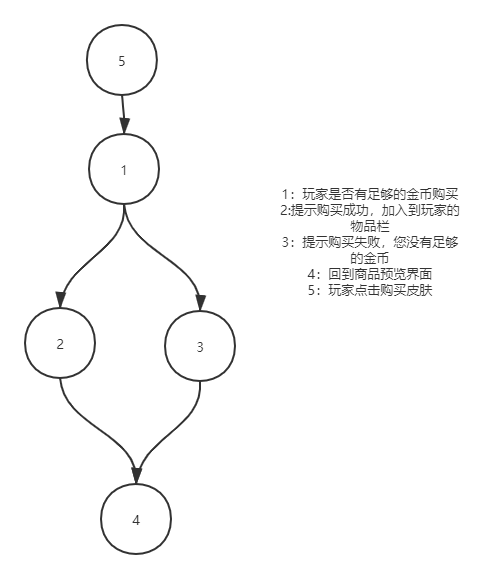
路径2：5-1-3-4

输入数据：玩家点击购买角色

预期结果：玩家失败购买角色

##### 4.1.5.2皮肤购买





首先根据V(G)=E-N+2计算出环形复杂度是2

然后再根据V(G)=P+1计算出环形复杂度是2，那么流图是正确的

语句覆盖：

路径1：5-1-2-4

路径2：5-1-3-4

路径1：5-1-2-4

输入数据：玩家点击购买皮肤

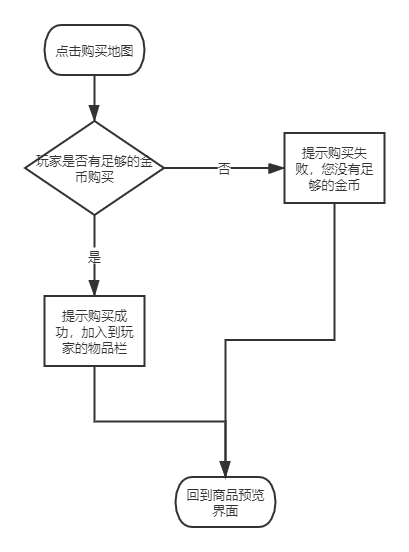
预期结果：玩家成功购买皮肤

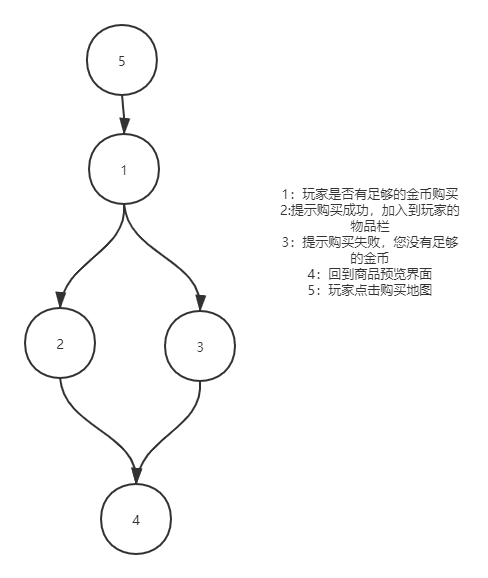
路径2：5-1-3-4

输入数据：玩家点击购买皮肤

预期结果：玩家失败购买皮肤

##### 4.1.5.3地图购买





首先根据V(G)=E-N+2计算出环形复杂度是2

然后再根据V(G)=P+1计算出环形复杂度是2，那么流图是正确的

语句覆盖：

路径1：5-1-2-4

路径2：5-1-3-4

路径1：5-1-2-4

输入数据：玩家点击购买地图

预期结果：玩家成功购买地图

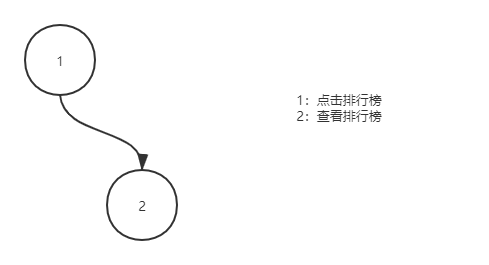
路径2：5-1-3-4

输入数据：玩家点击购买地图

预期结果：玩家失败购买地图

#### 4.1.6测试6（排行榜功能的单元测试）

##### 4.1.6.1排行榜功能



首先根据V(G)=E-N+2计算出环形复杂度是1

然后再根据V(G)=P+1计算出环形复杂度是1，那么流图是正确的

语句覆盖：

路径1：1-2

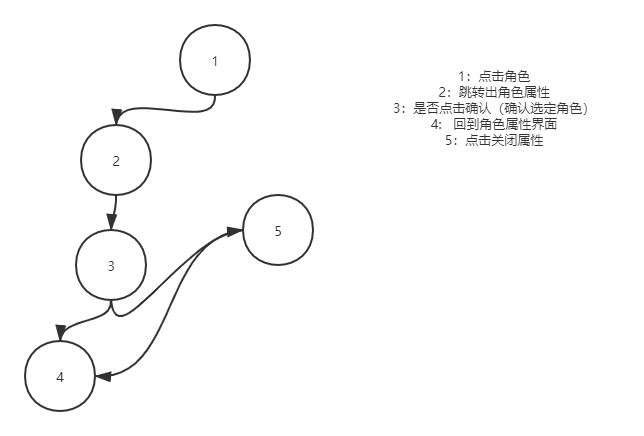
路径1：1-2

输入数据：点击排行榜

预期结果：玩家能够正确的查看到排行榜

#### 4.1.7测试7（角色功能的单元测试）

##### 4.1.7.1角色功能



首先根据V(G)=E-N+2计算出环形复杂度是2

然后再根据V(G)=P+1计算出环形复杂度是2，那么流图是正确的

语句覆盖：

路径1：1-2-3-4-5

路径2：1-2-3-5

路径1：1-2-3-4-5

输入数据：在属性界面点击确认角色

预期结果：玩家成功选定角色

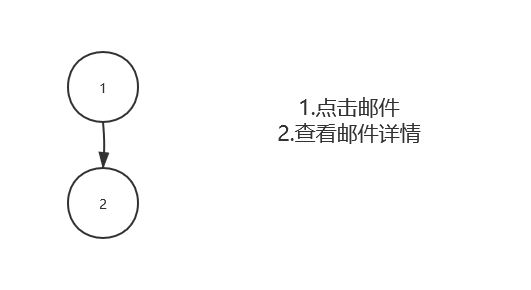
路径2：1-2-3-5

输入数据：点击角色属性

预期结果：玩家成功点击查看角色属性

#### 4.1.8测试8（收件箱功能的单元测试）

##### 4.1.8.1点击邮件



首先根据V(G)=E-N+2计算出环形复杂度是1

然后再根据V(G)=P+1计算出环形复杂度是1，那么流图是正确的

语句覆盖：

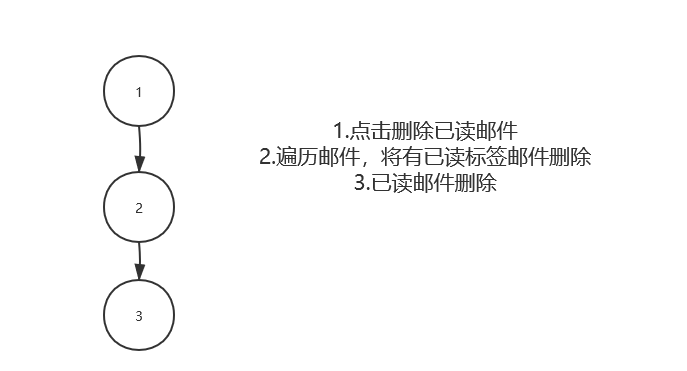
路径1: 1-2

路径1: 1-2

输入数据：点击邮件

预期结果：进入邮件详情页

##### 4.1.8.2点击删除已读邮件



首先根据V(G)=E-N+2计算出环形复杂度是1

然后再根据V(G)=P+1计算出环形复杂度是1，那么流图是正确的

语句覆盖：

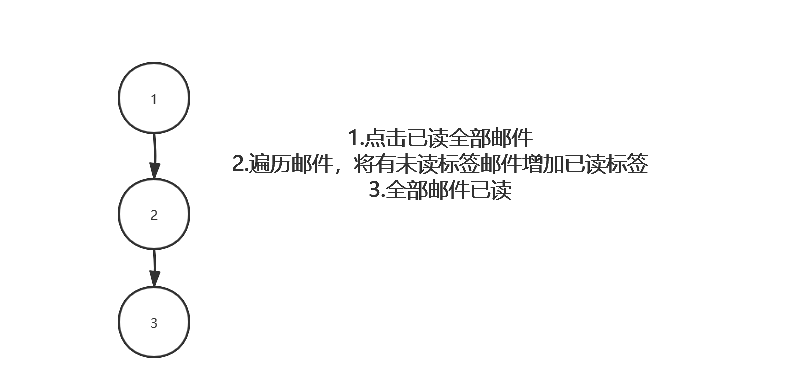
路径1: 1-2-3

路径1: 1-2-3

输入数据：点击删除已读邮件

预期结果：全部已读邮件删除

##### 4.1.8.3点击已读全部邮件



首先根据V(G)=E-N+2计算出环形复杂度是1

然后再根据V(G)=P+1计算出环形复杂度是1，那么流图是正确的

语句覆盖：

路径1: 1-2-3

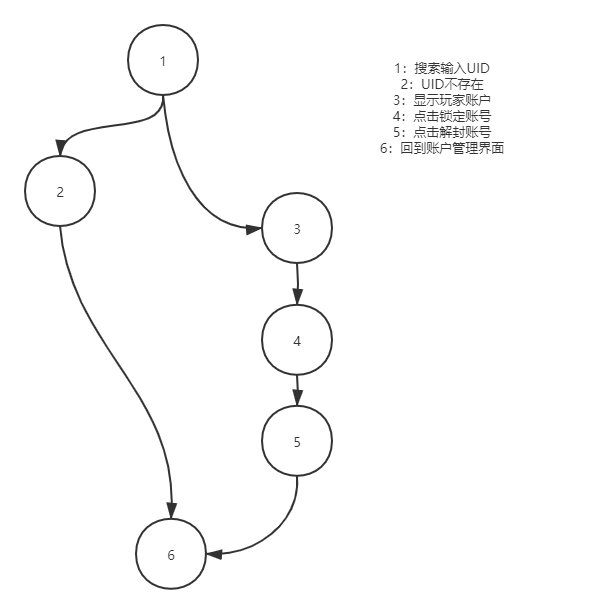
路径1: 1-2-3

输入数据：点击已读全部邮件

预期结果：全部邮件已读

#### 4.1.9测试9（管理员功能的单元测试）

##### 4.1.9.1账号管理



首先根据V(G)=E-N+2计算出环形复杂度是2

然后再根据V(G)=P+1计算出环形复杂度是2，那么流图是正确的

语句覆盖：

路径1：1-3-4-5-6

路径2：1-2-6

路径1：1-3-4-5-6

输入数据：输入存在的UID对其进行锁定解封操作

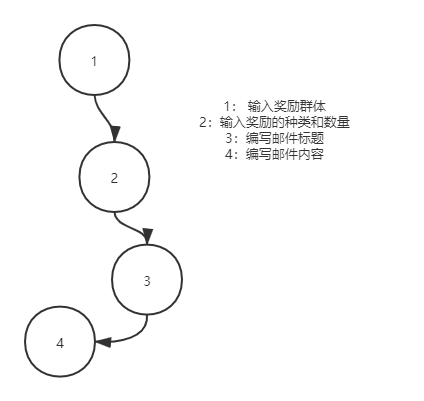
预期结果：能够搜索到存在的UID并对其进行账号的锁定解封操作

路径2：1-2-6

输入数据：输入不存在的UID

预期结果：当UID不存在的时候显示不存在

##### 4.1.9.2奖励发送



首先根据V(G)=E-N+2计算出环形复杂度是1

然后再根据V(G)=P+1计算出环形复杂度是1，那么流图是正确的

语句覆盖：

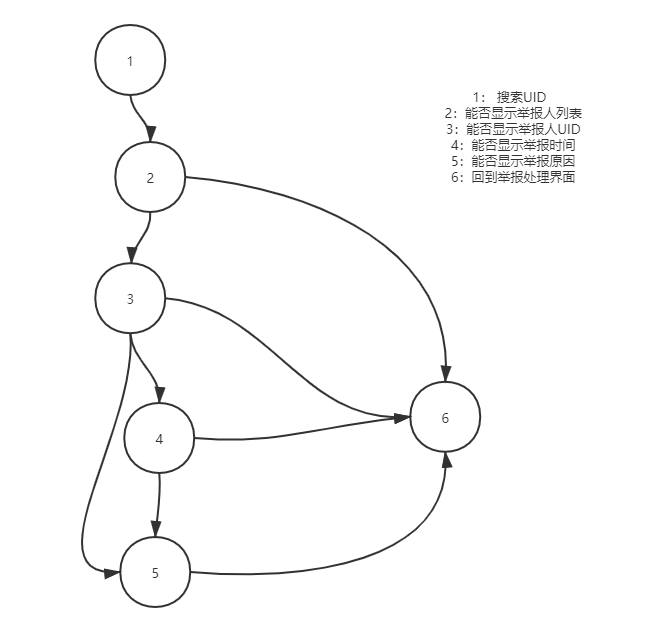
路径1：1-2-3-4

路径1：1-2-3-4

输入数据：输入奖励群体，奖励种类和数量，邮件标题和正文

预期结果：能够成功根据正确奖励群体，输入的奖励种类和数量，编写的邮件标题和正文发送邮件

##### 4.1.9.3举报处理



首先根据V(G)=E-N+2计算出环形复杂度是5

然后再根据V(G)=P+1计算出环形复杂度是5，那么流图是正确的

语句覆盖：

路径1：1-2-3-4-5-6

路径2：1-2-6

路径3：1-2-3-6

路径4：1-2-3-4-6

路径5：1-2-3-5-6

路径1：1-2-3-4-5-6

输入数据：输入存在的UID搜索

预期结果：能够成功根据正确UID搜索到举报信息（举报人UID，举报时间，举报原因）

路径2：1-2-6

输入数据：输入不存在的UID

预期结果：当UID不存在的时候显示不存在

路径3：1-2-3-6

输入数据：输入存在的UID搜索

预期结果：能够成功根据正确UID搜索到举报信息（举报人UID）

路径4：1-2-3-4-6

输入数据：输入存在的UID搜索

预期结果：能够成功根据正确UID搜索到举报信息（举报人UID，举报时间）

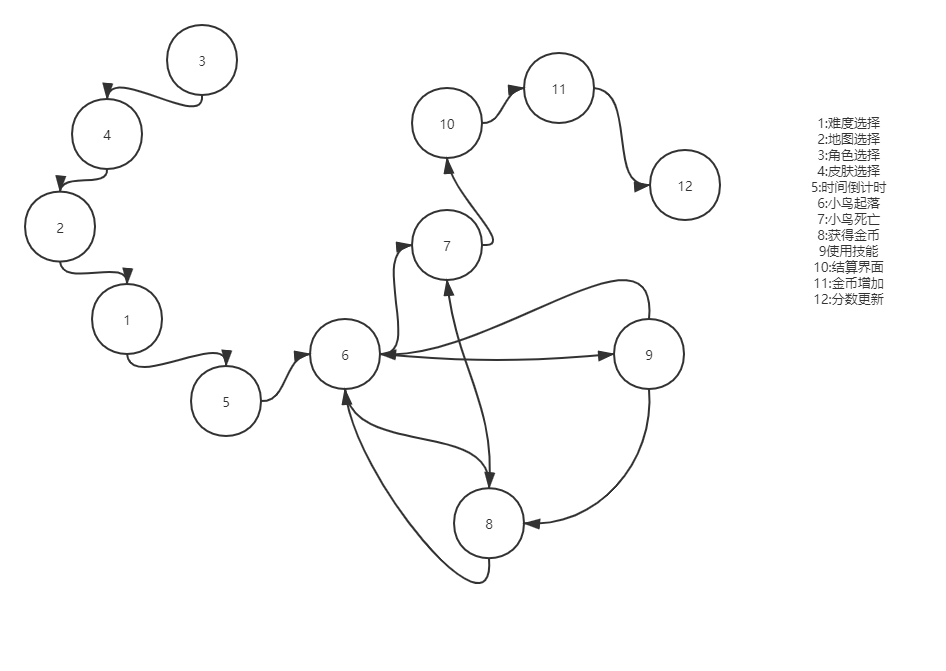
路径5：1-2-3-5-6

输入数据：输入存在的UID搜索

预期结果：能够成功根据正确UID搜索到举报信息（举报人UID，举报原因）

### 4.2集成测试

#### 4.2.1游戏模块集成测试



首先根据V(G)=E-N+2计算出环形复杂度是5

然后再根据V(G)=P+1计算出环形复杂度是5，那么流图是正确的

路径1：3-4-2-1-5-6-7-10-11-12

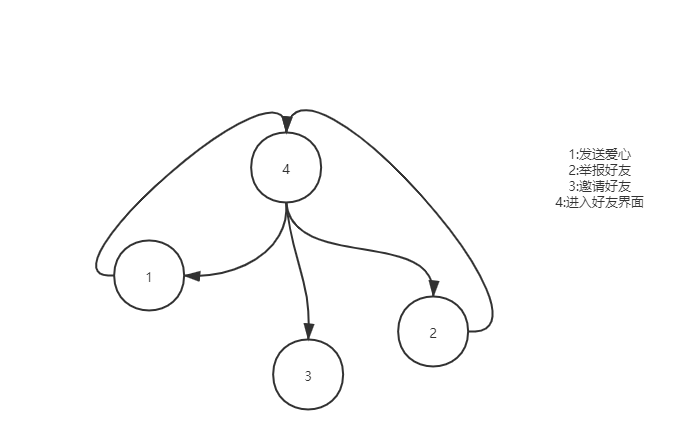
路径2：3-4-2-1-5-6-8-6-7-10-11-12

路径3：3-4-2-1-5-6-9-6-7-10-11-12

路径4：3-4-2-1-5-6-9-8-6-7-10-11-12

路径5：3-4-2-1-5-6-9-8-7-10-11-12

#### 4.2.2好友模块集成测试



首先根据V(G)=E-N+2计算出环形复杂度是3

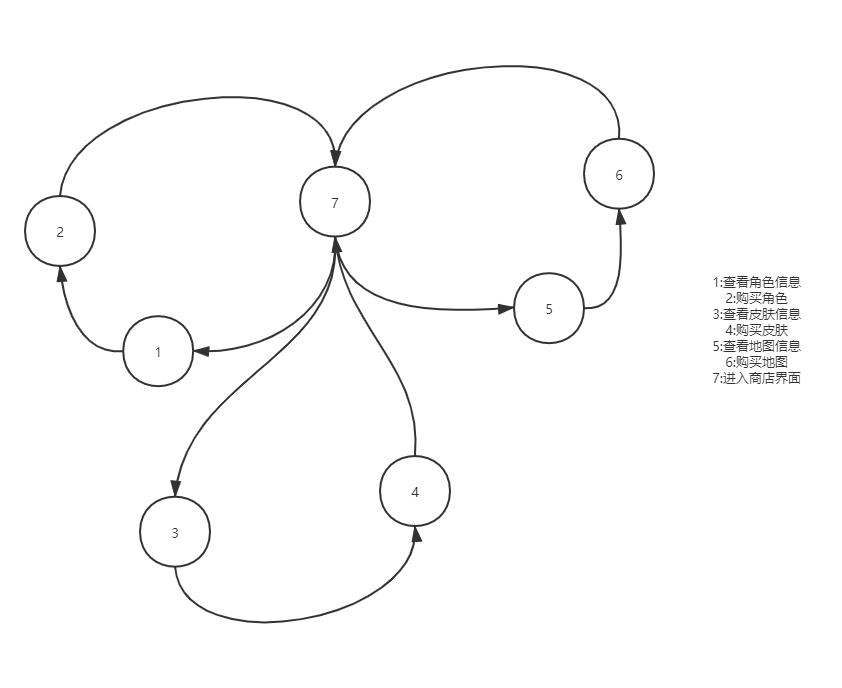
然后再根据V(G)=P+1计算出环形复杂度是3，那么流图是正确的

路径1：4-3

路径2：4-1-4

路径3：4-2-4

#### 4.2.3商店模块集成测试



首先根据V(G)=E-N+2计算出环形复杂度是4

然后再根据V(G)=P+1计算出环形复杂度是4，那么流图是正确的

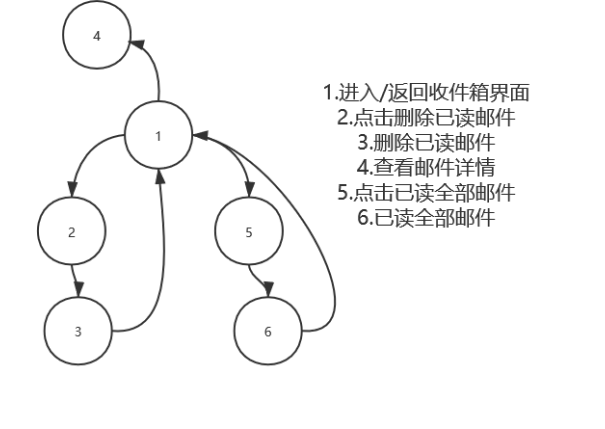
路径1：7-1-2-7

路径2：7-5-6-7

路径3：7-3-4-7

路径4：7-3-4-7-5-6-7

#### 4.2.4收件箱模块集成测试



首先根据V(G)=E-N+2计算出环形复杂度是3

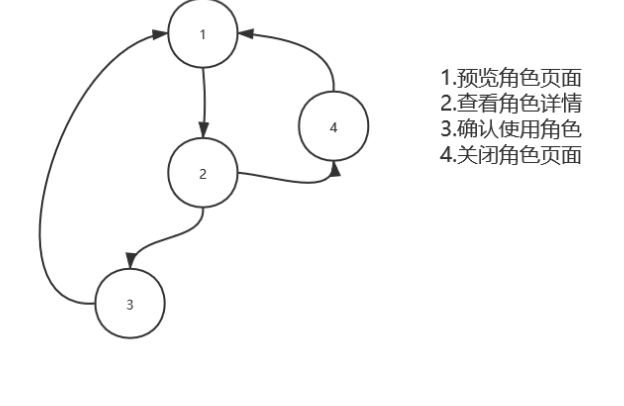
然后再根据V(G)=P+1计算出环形复杂度是3，那么流图是正确的

路径1：1-2-3-1

路径2：1-5-6-1

路径3：1-4

#### 4.2.5角色模块集成测试



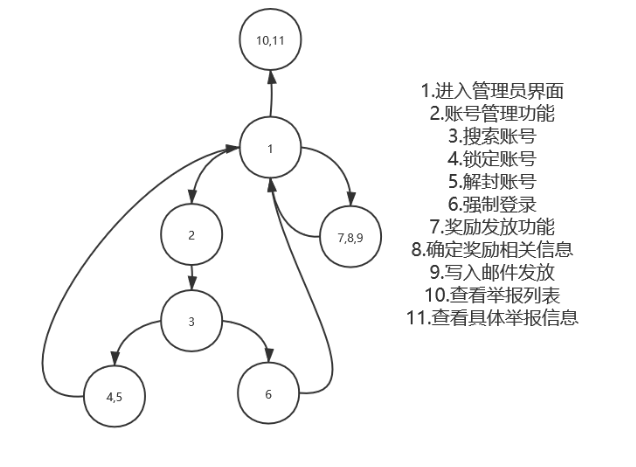
首先根据V(G)=E-N+2计算出环形复杂度是2

然后再根据V(G)=P+1计算出环形复杂度是2，那么流图是正确的

路径1:1-2-3-1

路径2:1-2-4-1

#### 4.2.6管理员模块集成测试



首先根据V(G)=E-N+2计算出环形复杂度是4

然后再根据V(G)=P+1计算出环形复杂度是4，那么流图是正确的

**语句覆盖：**

路径1：1-2-3-4-5-1

路径2：1-2-3-6-1

路径3：1-7-8-9-1

路径4：1-10-11

### 4.3系统测试

用户是否能流畅地进行游戏全功能：

1. 正常的授权，登录
2. 正常的进入游戏界面
3. 各个界面之间可以正常地自由跳转
4. 流畅地进行完游戏过程，游戏内容正常显示，游戏机制逻辑正常体现。
5. 用户之间可以进行举报，送爱心增加亲密度。

### 4.4验收测试

需要和用户一起确认，系统的功能和文档的描述之间是否有差异，适合符合用户需求。

### 4.5测试用例

## 5软件功能概述

### 5.1游戏

根据不同难度进入游戏，进入游戏后，玩家操控会自由下落的角色通过障碍物水管，通过点击屏幕的方式让小鸟上跳，点击技能按钮让小鸟使用技能，画面中障碍物水管会随机产生，同时也会随机产生金币，每跨过一个水管就会使分数加1，操控小鸟触碰到金币会增加金币。游戏结束后，分数和金币会出现在结算界面显示，同时加入到玩家的数据库中，更新信息。玩家可改变角色，角色皮肤以及地图和难度进入游戏游玩。

### 5.2商店与角色

游戏为玩家提供多个角色的使用，除了初始角色之外，其他的角色需要通过商城购买来解锁，角色界面用于玩家浏览角色的属性，技能，介绍等相关信息，玩家在该界面选择出战的角色；商城界面提供了角色的购买，皮肤的购买和地图的购买，皮肤包括角色皮肤和水管的皮肤，商城购买的货币是游戏中游玩或者观看广告来获得的货币。

### 5.3好友以及收件箱

游戏有好友系统，好友为用户的微信通讯录中的好友，玩过该游戏的好友都会显示在好友列表中，同时可以邀请尚未玩过的好友游玩该游戏，邀请信息会以小程序分享的形式发送到同好友的对话框中，玩家可以向好友赠送爱心，每赠送一次爱心都会增加一次与好友的亲密度；收件箱用于接受官方发放的奖励或者活动相关以及更新相关的通知，除此以外，好友赠送的爱心会以邮件的形式在收件箱中出现来告知玩家。如对某些好友的行为有所不满，可以选定合适的原因向系统举报好友。

### 5.4排行榜

玩家的每场游戏结束会有分数统计，以此来衡量本场游戏玩家的水平。而分数即排行榜的排行标准，玩家之间的排行会以分数的高低先后显示，排行榜分为全球排行榜和好友排行榜。

### 5.5管理员

管理员具有3大功能：

1. 账号管理：管理员可以通过uid来搜索玩家，同时可以对玩家账号进行封锁和解封，如有需要，可以强制登录玩家的账号。
2. 奖励发放：管理员可以选定奖励群体和奖励物品如金币，钻石等来以邮件的形式给玩家发放奖励。
3. 举报信息：管理员会接收到系统得到的玩家的举报信息，从而根据举报信息来进行账号管理。

## 6总体分析

### 6.1功能分析

### 6.2存在缺陷

### 6.3后续补救

### 6.4评价与结论

## 7注解

**1、软件开发背景**

本产品是一款主要面向20-30岁有许多碎片时间的游戏爱好者的微信小程序，玩家可以在游戏中搜集拥有不同属性和技能的角色以及相应的皮肤，同时还可以体会到不同难度的游戏内容和地图。

**2、软件测试相关补充**

在软件实现中包括编码和测试两个阶段。按照传统的软件工程方法学，编码是在对软件进行了总体设计和详细设计之后进行的，它只不过是把软件设计的结果翻译成用某种程序设计语言书写的程序，因此，程序的质量基本上取决于设计的质量。程序内部的良好文档资料，有规律的数据说明格式，简单清晰的语句构造和输人输出格式等，都对提高程序的可读性有很大作用，也在相当大的程度上改进了程序的可维护性。

目前软件测试仍然是保证软件可靠性的主要手段。测试阶段的根本任务是发现并改正软件中的错误，软件测试是软件开发过程中最艰巨最繁重的任务，大型软件的测试应该分阶段地进行，通常至少分为单元测试、集成测试和验收测试3个基本阶段。设计测试方案是测试阶段的关键技术问题，基本目标是选用最少量的高效测试数据，做到尽可能完善的测试，从而尽可能多地发现软件中的问题。

白盒测式和黑盒测试是软件测试的两类基本方法,这两类方法各有所长，相互补充。通常，在测试过程的早期阶段主要使用白盒方法，而在测试过程的后期阶段主要使用黑盒方法。为了设计出有效的测试方案。软件工程师应该深人理解并坚持运用关于软件测试的基本准则。设计白盒测试方案的技术主要有，逻辑覆盖和控制结构测试；设计黑盒测试方案的技术主要有，等价划分、边界值分析和错误推测。在测试过程中发现的软件错误必须及时改正，这就是调试的任务。为了改正错误，首先必须确定错误的准确位置，这是调试过程中最困难的工作，需要审慎周密的思考和推理。为了改正错误往往需要修正原来的设计，必须通盘考虑统筹兼顾，而不能“头疼医头、脚疼医脚”,应该尽量避免在调试过程中引进新错误。

测试和调试是软件测试阶段中的两个关系非常密切的过程，它们往往交替进行。程序中潜藏的错误的数目，直接决定了软件的可靠性。通过测试可以估算出程序中剩余的错误数。根据测试和调试过程中已经发现和改正的错误数，可以估算软件的平均无故障时间；反之，根据要求达到的软件平均无故障时间，可以估算出应该改正的错误数，从而能够判断测试阶段何时可以结束。

**3、相关术语解释**

**单元测试**：对软件组成进行的测试。其目的是检验软件基本组成单位的正确性。测试对象是软件设计的最小单元：模块，又称为模块测试。

**集成测试**：也成联合测试，联调、组装测试，将程序模块采用适当的集成策略组装起来。

**系统测试**：将软件系统看成一个系统测试。包括对功能、性能以及软件所运行的硬软件环境进行测试。时间大部分在系统测试执行阶段，包括了回归测试和冒烟测试。

**验收测试**：部署软件之前的最后一个测试操作，它是技术测试室的最后一个阶段，也叫做交付测试，验收测试的目的是保证软件的准备就绪，按照项目合同、任务书、双方约定的验收依据文档，向软件的购买者展示该软件的原始的需求。

**α测试**：主要是由一个用户在开发环境进行的测试，也可以是公司内部的用户在模拟实际操作环境下进行的测试。主要的目的是：评价软件产品的FLURPS（即功能、局域化、可使用性、可靠性、性能和支持）。

**β测试**：由软件的最终的用户们在一个或者多个客户场所进行的测试。

**第三方测试**：介于开发方和用户之间的组织测试。

**静态测试**：静态测试值的是不运行程序本身，仅通过分析和检查源程序的语法、结构、过程、接口来检查程序的正确性。对需求规格说明书、软件设计说明书、流程图分析、符号执行来进行找错。

**动态测试**：指的就是运行被测的程序。检查运行结果与预期结果的差异，并分析运行效率、正确性和健壮性的等性能，这种方法主要是由三部分进行组成的：测试用例、执行程序、分析程序运行输出的结果。

**手工测试**：是由人一个一个的输入测试用例，然后观察结果、和机器测试相对应，属于比较原始，大事需要一个一个步骤进行测试。

自动化测试：在预设条件下运行系统或应用程序，评估运行结果、预先条件应该包括正常的条件和异常条件。简单的说自动化测试是把人为驱动的测试行为转化为机器执行的一种过程。

**黑盒测试（**Black-box-Testing）：黑盒测试也称为功能测试，测试中把被测的软件当成一个黑盒子，不关心盒子的内部结构是什么，指关心软件的输入数据和输出数据。

**白盒测试**（White-box-Testing）：白盒测试又称结构测试，透明盒测试、逻辑驱动测试或基于代码的测试。白盒值的是打开的盒子，去研究里面的源代码和程序结果。

灰盒测试（White-box-Testing）：灰盒测试是介于白盒测试与黑盒测试之间的一种测试，主要用于集成测试阶段。不仅观念朱输入输出的正确性。同时也关注程序内部的情况。

## 附录

附录可用来提供那些为便于文档维护而单独出版的信息(例如图表、分类数据)。为便于处理，附录可单独装订成册。附录应按字母顺序(A, B等)编排。