

座右铭：任何未执行的功能都不要盲目地下定义

## 一、游戏测试

### 1.游戏测试的定义

#### 1) 两个保证：

保证版本的正常（安装、解析、运行、卸载）运行、  
内部质量（功能质量、兼容性、弱网、性能）

#### 2) 两个建议：

易用性

可玩性

### 2.游戏测试的理解

1) 游戏是软件的一种形式，游戏测试的一部分内容和软件测试相同，但又有自己的特殊性。->

2) 游戏测试比软测更加专业，包括了软测和游戏专业测试（可玩性、美术、策划合理性等，是更多站在玩家角度考虑）->

3) 游戏测试要求测试人员具备极大耐心、恒心、细心，因为一些 bug 会让游戏出现致命影响）

### 3.游戏测试组成

组成分为传统的软件测试和游戏本身的测试（游戏世界测试），  
游戏本身的测试又有游戏测试的特性和游戏可玩性测试

## 4.传统的软件测试

- 1、测试的目的是发现软件中存在的缺陷
- 2、测试都是需要测试人员按照需求（产品行为）描述来实施
- 3、每一种测试都需要产品运行在真实（正式服、白名单账号）或是模拟（测试服）环境之下
- 4、每一种测试都要求以系统方法展示产品功能（测试点、测试用例），执行记录证明测试结果是否有效（用例执行记录），提交发现其中出错的原因（Bug 单）让程序人员进行改造。

## 5.游戏测试的特性

### 1) 游戏情节的测试:

主要指游戏世界中的任务系统的组成，即游戏世界的事件驱动，也可称游戏情感世界的测试

### 2) 游戏世界的平衡测试:

主要表现在经济平衡，能力平衡（包含技能、属性、种族、装备、经济系统等），保证游戏世界竞争公平

### 3) 游戏文化测试（不一定）：整个游戏的风格是中国文化还是日韩风格主导，人物行为设定符合形象本身

（游戏可玩性测试

游戏世界的搭建完整）

## 6.游戏测试和软件测试区别

1、Ui：软件 ui 是**实用性**最重要（一个**加分项**），用户希望能尽可能精简；游戏中贴图精美程度关系到玩家的受众，玩家每一步操作愉快感都是在毫秒之间传递（娱乐性最重要）

2、数值：软件不需要主动创造，逻辑单一，产出固定；游戏需要不断创造，交

互频繁

3、活动：软件的频率、类型、产构 没有游戏频繁

4、进度（都赶进度）：软件的更新方式、交付对象是甲方（可以协商）  
没有那么赶

游戏的进度必须实现实时，不然会有大量老玩家脱坑（DNF 之前两次没有实时，大量老玩家脱坑）

5、性能：软件关注服务器性能；游戏同时关注服务器和客户端（CPU、内存、FPS)的性能

6、安全：对两者来说都是致命的

7、合服：目的是增加玩家活跃度，减少服务器成本

8、交互：软件：方式、类型；游戏包括软件+动作、表情、功能等交互

9、网络：软件可接受延迟（单一用户）；游戏不可接受（影响游戏的所有玩家）

## 7.测试工具

文件管理：SVN、Git

Bug 管理：TAPD、禅道、jira

流程图：Visio、wps 自带、亿图

思维导图：X-mind

接口工具：postman、Jmeter

抓包工具：Fiddler、Charles、F12、【ADB、客户端自带、第三方物件】

弱网工具：Fiddler、Perfdog、Network

性能工具：Jmeter、LR(LoadRunner)、Perfdog

自动化工具：Python、selenium、uniapp、pytest、allure

## 8.玩家的期望

对操作的期望（一致性）

对目标的期望（明确的指导）

对界面的期望（体贴、友好）

对**感觉**的期望（**沉浸**）

对**规则**的期望（**逻辑**）

**警示：一切未知的地方都存在风险**

**一切与需求不符的都是 bug,比如标点符号、多音字（细心）**

## 9.游戏的分类

端游 页游 手游 H5 小游戏

1.MOBA（多人在线战术竞技游戏），一般需要购买装备，两队玩家互相竞争

2.FPS（射击游戏），第一人称射击游戏

3.MMORPG（大型多人在线角色扮演游戏）

4.ACT（动作游戏），以动作为主要表现形式

5.SLG（策略类游戏）

6. 体育竞技类游戏，竞技运动即比赛性的体育活动设定为背景相关游戏：火箭联盟

7. 休闲游戏：消消乐、贪吃蛇

## 二、测试的基础知识

### 1.游戏测试职责

1.功能测试：保证游戏各功能的正常，与需求一致

2.可玩性

3.尽早介入（尽早熟悉项目进度），尽快完成（熟悉项目内容，完成产物产出，发现 BUG，推进 BUG 修复），持续不断（测试过程反复不断）

4.玩家立场评测

## 2.开发流程

1. 市场
2. 策划
3. 开发
4. 运营
5. 市场

## 3.游戏测试流程

1. 开始
2. 需求分析
3. 需求评审
4. 测试计划制定
5. 提取测试点
6. 编写测试用例
7. 用例评审
8. 测试前准备
9. 冒烟测试
10. 执行测试用例
11. BUG 提交
12. BUG 追踪
13. BUG 回归
14. 最终包全量测试
15. 编写测试总结
16. 汇总测试报告
17. 结束

## 4. 新版本测试执行策略

完整=》随机=》回归=》发布=》冒烟=》完整

分布式服务器：

游戏不同功能模块在不同的服务器

Server <-DB ->商场 好友 匹配(不同服务器)

功能测试流程：

需求分析：熟悉、分析功能需求（策划提供新需求给到测试组）

## 5.游戏生命周期

- 1.立项
- 2.首版 ->
- 3.技术封测（内部测试或研发测试） ->
- 4.删档内测（ACC 数据） ->
- 5.不删档内测（保留账号/密码） ->
- 6.公测 ->（不开放充值商业化功能，保留数据） ->
- 7.正式上线（申请版号，19年新政策） ->
- 8.产品下线

## 6.游戏项目规模

小型项目：

人员：策划 1 人 美工 1-2 人 开发 2-5 人 测试 1-2（一组）

周期：0-1 个月策划案、一个月开发、1-2 个月测试+开发 2-4 个月

版本周期：大版本 1-2 周一给、小版本 1-3 天一给

用例：800-1500 条

Bug：200-600 个

中型项目：

人员：策划 3-5 人 美工 2-4 人 开发 5-10 人 测试 2-4（一组）（10-20 人）

周期：0.5-1 个月策划案、一个月开发、3-5 个月测试+开发 4-6 个月

版本周期：大版本 2-3 周一给、小版本 2-4 天一给

用例：3000-5000 条

Bug：1000-1500 个

大型项目：

人员：策划 7-13 人 美工 5-8 人 开发 20-30 人 测试 5-8（一组）（30-40 人）

周期：1-3 个月策划案、1-2 个月开发、3-5 个月测试+开发 5-8 个月

版本周期：大版本 1-1.5 周一给、小版本 3-5 天一给

用例：5000-12000 条

Bug：2000-4000 个

## 二、需求分析

### 1.需求分析-评审

1. 分析：理解需求内容，对需求的测试，验证需求的正确性、完整性、无二义性
  2. 评审：沟通、质疑和确认需求内容，以及开发人员确认需求是否可以开发，测试人员确认开发的实现方式
  3. 参与人员：策划、开发、测试
  4. 产出：新版本需求文档
- 注意：需求评审一定开吗？
- 要求：新增需求多一定开

### 2.需求分析的范围

1. 需求的背景、目标和影响范围（交互模块等）界面、道具展示、界面交互等
2. 提示的方式，异常处理方式
3. 主流程，操作流程和步骤说明是否合理
4. 需求上下文的一致性，是否与其他需求发生冲突（**例如是否有死锁的情况**）
5. 需求的逻辑是否足够清晰，每个条款是否都包含描述问题和解决问题
6. 挖掘隐形需求和相互依赖的需求（**前置条件是否有清晰的说明**）

### 3.需求分析的目的

熟悉需求内容（主干、分支、异常），把不明确、疑问、隐形需求在本地记录下来

### 4.需求分析的方式

#### 1.全局观考虑：

- a 了解需求主要目的，了解功能的业务流程，硬件环境，软件环境，交互模块等
- b 分析并列出需求中的疑问或问题，包括但不限于时间限制、奖励方式、步骤顺序、用户操作场景
- c 列出需求中不可实现、不合逻辑、存在歧义或含义不明的语并与策划确认
- d 挖掘隐藏需求（如活动次数、角色关系、系统与系统之间的关系）

#### 2.业务流程分析：

- a 画流程图，确认合理性、准确性等问题
- b 检查是否缺失分支流程
- c 检查是否存在异常分支、不确定范围

d 流程图列出每个阶段测试点

### 3.补充测试点:

- a 补充测试点，如 UI 测试点、进入方式等
- b 探索式测试方法补充测试点
- c 对需求中的疑问和测试范围进行再次确认

**举例：疑问查批注 、数字找边界 、时间问刷新 、交易“重”到账 、任务画流程**

## 三、测试计划制定

测试主管/组长会把新增的需求--分配到测试组员

## 五、测试用例的编写

### 1 概念

为了任意一个需求/功能，编写出的一组数据（输入、操作步骤、预期与需求相符），方便测试全面进行验证。

### 2 要素 7

用例编号（序号） 所属模块 用例描述 前提条件 操作步骤 预期结果 备注

### 3 用例种类 5

全功能用例 通用用例 最小检查点 冒烟用例

冒烟用例：不单独编写，功能用例的 5%提取出来

### 4 用例设计方法 5

**等价类划分法：**有效等价类，有意义的输入数据构成的集合|无效等价类，反之

简单理解（有逻辑判定功能）

1 输入值判定，如，玩家昵称(要求 3-9 位中文)，有效：中文、3-9 位，无效：其他类型文字、<3、

2 功能判定，如角色装备穿戴、攻击目标判定、释放范围判定、商品物品货币出售类型判定

**边界值分析法：**对等价类划分法的补充

简单理解（用数字要求的位置）

如时间、等级、限制、范围、组队人数、物品购买价格等

**错误推断法：**异常操作的判定

**判定表法：**多逻辑下执行不同操作的情况

条件桩 列出所有条件 条件项 左列条件的取值

动作桩 动作项

**场景设计法：**需覆盖用例的主成功场景和扩展场景，补充正反面的测试用例和考虑出现用例



的异常

## 六、用例评审：

时间：组内协商

主导人：测试部门（各功能模块负责人）

方式：组内评审-组内成员讲解->完成后优化好的测试用例

目的：用例完整性、覆盖度更全面

过程：测试组（所有用例编写完成后）协商时间，然后测试主导召开会议/工位共享，测试组员提出问题和建议，评审完成后修改、优化测试用例并输出到大群查看，修改完成后，进行用例上传归档。

## 七、测试前准备

1. 硬件：设备（测试功能用中、高端机），
2. 网络：测试组专用 WiFi（网络 IP 是否需要添加白名单根据项目组定义）
3. 账号：（项目组提供/自己注册）
4. 测试资源：SVN、沟通群下载、云盘下载

测试包获取，测试包版本号核对->安装

道具表、配置表资源的获取

## 八、冒烟测试

冒烟测试可不单独编写

执行结果：

Pass:**P**(大写、加粗、绿色字体)-**通过**

Fail:**F**(大写、加粗、红色字体)-**不通过**

**N/A**（大写、加粗、黑色字体）-**未实现**

BUG 表单：

一、要素

1.标题：【平台】安卓/iOS、【模块】简要描述 BUG 情境

eg:战队转让：战队，队长转让队长权限时，该玩家退出战队异常

2.测试环境概述：硬件设备、操作系统、网络环境、账号

3.复现步骤：

1.队长 A 向队员 B 转让队长权限时

2.队员 B 退出战队/在转让界面

3.队长点击【确认】按钮

4.查看界面表现

4.复现概率必现：100% 偶现：机率

5.实际结果：BUG 详细结果：游戏崩溃，功能无法使用，备注：战队中所有成员都无法使用

## 【战队功能】

### 6、严重程度：

A 禅道：致命、严重、一般、轻微

BTAPD:致命、严重、一般、提示、建议

CJIRA:S、A、B、C、Q

### 7.优先级：

A 禅道：紧急、高、中、低

BTAPD:紧急、高、中、低、无关紧要

CJIRA:紧急、高、中、低、无关紧要

### 8、指派人：BUG 解决者/（开发/策划）

### 9、附件：视频、图片、log 日志

### BUG:

#### 1.严重程度：

致命：闪退、卡死、崩溃、与 R 货币相关

严重：功能无响应、功能错误、数据（接口）

一般：ui、音效

轻微：语法错误、文字错误（建议相关）

#### 2.优先级：

紧急：完全卡进度、立即修复

高：卡部分进度、尽快修复

中：不卡进度、本版本修复

低：可不修复（本版本）

### Adb 命令



Android调试常用  
命令.docx

## Bug（执行）：

### 执行结果：

Pass:P(大写、加粗、绿色字体)-通过

Fail:F(大写、加粗、红色字体)-不通过

N/A（大写、加粗、黑色字体）-未实现

执行工具：禅道、TAPD

### Bug 状态流转（生命周期流转）：

测试提交新的 bug，开发/策划对新的 bug 进行查看/分析，开发/策划接受处理可选择：

A 拒绝 B 挂起 C 搁置 D 已修复，待验证，之后等待下一个状态流转，测试对 A 处理方式---  
查看拒绝原因，原因合理的话可以关闭 bug，如果不合理继续拿出更多证据去说服开发，如果双方都不能被说服的话，就找策划一起三方沟通，让策划来定夺关闭 bug 还是重新打开 bug。

对 B/C：查看评论挂起修复时间，及时去推动修复流程

对 D:以验证之后，重新打开 bug 或关闭 bug,把验证版本新附件进行更新

8.

## Bug 单 (log 日志)

分类: A 客户端日志 (前台)、B 服务器日志 (后台)、C 外部日志)

内容: 账号信息、运行时间、请求时间、处理方式、结果、报错 (error)

获取方式: A 安卓: 应用指定存储路径 (询问组员、开发)

ios: 工具获取 (iTunes/爱思助手)

itunes: 1 连接设备->2.BUG 复现->3.文件共享->选择对应游戏 4.找到文件-共享即可

爱思助手: 连接设备->工具箱->实时日志->复现 bug->停止导出

B 与开发沟通获取 (工具)

C 安卓: 工具 (adb)

adb(安卓调试桥)取外部日志:

1.设备连接: 设备开启开发者 (开启、USB 打开) PC<=>手机, 输入 adb devices

2.log 日志抓取: adb logcat (命令调动) -v time >D:\log 名称.txt(Cpan 受保护) (ctrl c 停止截取日志)

3.shell 命令: 查看应用包名:

adb shell 进入 shell 权限

pm list package

4.随机事件测试 monkey (压力)

adb shell monkey -p 应用包名 -v 次数 >D:\monkeylow.txt

5.安装应用 adb install 拖入应用 APK

文字描述类 bug:

所有活动的标签页、标签页中的时间显示、文字描述、礼包详情、商城物品描述等都需要对照文档一一查看, 并且查看具体文字是否准确

需求不一致 bug:

主要包括文字与文档不符、奖励与文档不符、奖励数量与文档不符、图片、按钮显示位置与文档不符等

界面显示类 bug:

主要包括文字显示格式、图片、图标重叠、界面显示与内容不一致等

功能操作类 bug:

功能阻碍、操作后与预期结果不符、卡顿

## bug 问题:

1.测试时发现一个已关闭的 Bug,后续版本测试时又复现了, 该如何处理?

首先排查 bug 的关闭原因, 若误关闭的话, 与开发进行良好的沟通后重新打开 bug; 如若正常关闭的话, 分析 bug 单中内容是否与之前一致, 一致的话就在原有的 bug 单增加新附件、内容、版本号后重新打开 bug, 不一致的话就提交新的 bug 单。

2.如果发现一个偶现的 Bug, 你会怎么处理?

第一时间保留 bug 相关信息, 尝试多次复现, 还是不能复现的话, 发组内成员多人复现, 还是不行的话, 第一时间记录信息提交 bug 单并配合开发定位复现, 验证时询问开发复现步骤, 并多次验证, 在后续版本中持续关注。

3.你提交的 Bug，开发拒绝你的 Bug 单，你会怎么做？

先查看，后沟通，测试主导，实在不行上报组长

4.如果你负责的版本内容，在版本发布后，收到外网玩家投诉，你会怎么做？

收到外网玩家问题后尝试进行复现，不能复现的话，让客服与玩家沟通获取更多详细信息，如果可以复现的话，第一时间提交 bug 单，群内进行评估；程度严重立即紧急修复，程度不严重后续版本修复，整个测试过程中要持续关注、配合；内网可以复现的话，就属于漏测，分析错误原因，整改优化用例，待定。

5.如果你发现你的同事测试的内容有 Bug，你会怎么做？

6.如果同事请假，你按照同事写的用例执行发现 Bug，你会怎么做？

## 测试总结：

项目名称《》

版本号：

测试类型：

一、测试环境概述：硬件、系统、时间、人员

二、BUG 概述：XXX 版本新增 XXX 个，致命 XXX 个，严重，一般，轻微

三、测试模板：XXX\XXX

四、测试结果（插入用例执行结束）

发布标准

## 专项测试：

测试主动或被动一些操作，验证程序的一些操作

常见中断：

媒体（音乐）、提醒（iOS）、锁屏解锁、照相按钮、短信、来电、通话挂断、手机语音、数据线插拔、充电、WiFi 连接提醒、蓝牙提醒、低电提醒、闹钟或闹钟提醒

其他：

暴力测试：拔电池、杀进程、关机、重启、长时间使用、存储卡插拔、软件搬家（存储转移）、前后台切换

面试问题:专项测试是在什么时间节点介入？

性能节点：功能测试完成后（跑完整包），在开始性能测试，

弱网节点：某一个功能完成（未完成 100%）时测弱网

性能测试定义：通过性能测试工具（perfdog）测试游戏运行在不同配置设备（高、中、低）上各游戏场景运行时，各项指标（cpu、内存、FPS、流量、温度）数据

目的：优化系统，运行流畅，功耗符合用户预期

## 性能测试流程：

（专职）：

- 1.接收性能需求（明确测试目标如：测试设备要求、性能指标/标准、测试的场景）
- 2.设计性能场景用例
- 3.搭建测试环境（工具安装、设备联调、指标数据设定）
- 4.执行性能测试并收集测试结果
- 5.汇总测试数据提交对应人员
- 6.分析指标数据，给出调优建议

（辅助：协助组长）：

- 1.接收性能测试任务（组长），分析性能需求及用例
- 2.搭建测试环境
- 3.执行性能测试收集数据结果
- 4.汇报组长

需求由开发提供，包括设备要求、测试场景、测试性能指标、标准数值

需求内容：硬件设备（高端机型：苹果 iPhone 15 Pro 系列，华为 Mate 60 Pro；中端机型：红米 K60，OPPO Reno9；低端机型：红米 Note 12），测试场景()，性能指标关注：1.CPU 2.内存消耗 3.FPS,4.流量，5.温度（整机温度≤37 摄氏度）

注意事项：

每次开测前需保证测试环境一致，数据一致

- 1.手机电量充满，取掉手机外壳（影响散热），手机起始温度<35,锁定手机屏幕亮度 50%
- 2.perfdog 勾上加号处所有可选数据，保证数据完整性
- 3.开启进程前，确保手机进程无关 app 全部杀掉进程
- 4.每个关卡都需要独立的标签，区分开

场景：

- 1.待机（挂机），目的：无异常数据表现，平稳性
- 2.功能切换场景（全量冒烟），目的关注内存释放是否积流，导致泄露情况
- 3.对局高并发场景（PVE、PVP），异常负载（团战），目的关注游戏局内高并发时，所有指标数据数值情况

性能要素：测试场景、操作步骤，时长

性能需求分析：

当需求为游戏大厅和直播类时，更多关注挂机后内存、温度、帧数

当需求为活动类型时，关注活动页签切换和重复打开时内存增长问题，针对帧数、内存、温度、GPU 占比等

当需求为局外页签类时，关注重复打开和所有页签切换后的内存变化、温度变化

当需求为 loading 类，把一条用例重复三次，取平均数和竞品或者迭代之前的版本做对比

当需求为道具类时，关注装备该道具时第一视角和第 N 视角使用时的性能数据，和比较稳定的同类型道具做对比，多对特效等方面做详细测试

性能测试报告组长编写，我有看过。

包括版本号、测试时间、测试服、风险等级、测试类型、测试平台、测试结论

测试阶段有多少个服务器？

性能 bug:安卓，击杀瞬间掉帧；10 人比单人 CPU 上升 10%；温度。

# 弱网测试

## 定义：

通过弱网工具，设置不同（弱网环境）损伤，查看弱网环境下各功能运行时表现

## 目的：

确保游戏在弱网或无网环境下的正常运行，避免卡死、闪退、崩溃现象，  
确保弱网提示机制的正常触发，局外的常规提示网络差、加载中，图标信号提示触发，局内的断线重连机制触发提示，恢复后数据的同步性、正确性、唯一性

延时：(客户端->服务器)一个端点到另一个端点所需时间

延迟：一个端点到另一个端点的往返时间（不包含客户端、服务器的处理时间）

丢包率：接收包体的数据差占所发送数据包的比率

## 造成丢包原因：

- 1 网络状况（手机网络，WiFi）
- 2 客户端状况（性能、内存）
- 3 服务器状况（服务器人数多）

## 弱网测试流程：

- 1.接收组长分配的弱网测试任务（熟悉需求及用例）
- 2.搭建弱网测试环境（工具的安装、弱网参数的设定）
- 3.执行弱网用例，记录结果上报组长

弱网节点：某一个功能完成（未完成 100%）时测弱网

## 弱网场景

现象 1：和平地图下载或资源包下载的场景，地图数据加载中的场景

原因：数据下载中，下载失败后，未进行数据回滚，中止后重新下载，出现数据重复

解决方案：通过事务所处理数据下载逻辑，下载失败后，应用本地数据进行数据回滚

现象 2：（前端和后台请求交互的场景）进入副本场景消耗入场券

原因：数据上传过程中，由于失败重传机制，会出现连续两次写操作，并且未作唯一识别处理

解决方案：根据数据特性，对造成脏数据字段，通过关键字段，例如创建时间，**key-value** 值等生成 **hash** 键，标记记录唯一性，即数据写入时，检查 **hash** 键是否存在，如果已经存在，当前重复数据丢失

现象 3：在弱网环境下，用户输入用户名和密码点击登录，应用连接超时返回用户名和密码错误提示

原因：在弱网环境下的连接超时，按照强网业务逻辑处理，导致返回超时异常

解决方案：弱网连接超时后，检查应用本地数据库是否有用户登录信息，若存在，获取应用本地用户信息进行登录

现象 4：在弱网环境下，用户第一次输入搜索关键字没有得到响应后，再次输入全新关键字并发送请求，等待搜索结果返回后，当前结果页被之前的关键字搜索结果刷新覆盖

原因：之间的请求返回较慢，显示最终的结果后，之前请求返回的数据应不做处理

解决方案：对异步请求未完成任务进行 **cancel**

## 兼容性测试

### 定义

测试游戏运行在不同硬件（CPU、内存、）、操作系统上正常的安装、解析、运行、卸载，然后保证在不同的屏幕分辨率（1920、1080、960）和异形屏（刘海、曲面、打孔）的显示 **ui** 的清晰、无拉伸、无遮挡、超框。

### 方法

1. 人工测试即全手工测试兼容：top100 、top50

2. 借助第三方兼容性测试工具; wetest

备注：

人工测试量大，而且覆盖不全；第三方虽然工作量小，但是在主功能和主流程测试的时候没有侧重点，很难发现一些隐藏的问题；要说这两种方法哪一种更好，两者结合才是最好的兼容性测试方法。

## 测试流程

1. 接收适配测试需求（核心重点描述、新增场景位置描述、道具描述、注意事项描述、设备要求）

2. 获取测试资源文档（接收适配用例，Top 列表整理（机型、品牌、内存、CPU 主频、CPU 内核、GPU 渲染），测试包获取）
3. 执行适配测试并提交适配问题
4. 记录测试结果提交组长汇总

什么阶段执行兼容测试？

1. 新游戏已封包且功能完善可进行适配测试
2. 新增功能玩法、涉及界面较多、

一、兼容性主要关注哪些？

1. 硬件:(品牌、型号、配置(CPU 主频、内核、内存、GPU)
  2. 系统版本(Android 11.4 ios 15.4)
- ☆:1 和 2:重点关注游戏安装运行情况(如:无法安装，安装后无法正常运行)
3. 屏幕分辨率(1080P、2K、25K)
  4. 4.异形屏(刘海、曲面、折叠)
- 3 和 4:重点关注游戏运行过程中界面显示情况(超框显示，显示遮挡，显示不全，功能按点击无响应，渲染异常)

二、实际是如何进行测试？

1. 所有功能进行全部冒烟(根据兼容性测试用例)
2. 核心场景:完整的进行一次功能流程测试(如:匹配、快速完成一场对局、抽奖(抽一次))

三、兼容性测试经常遇见哪些问题？

1. 低端机型/低的操作系统:无法安装运行，运行时卡顿/闪退，
2. 不同分辨率/不同异形屏，界面显示遮挡，拉伸、按钮超框显示，点击无响应
3. 界面渲染异常(角色/建筑物模型显示异常)，界面花屏、卡屏。
4. 其他硬件周边问题如:音量按钮点击无效、音效异常、自带输入法输入异常等