

专项实战

深圳市门道信息咨询有限公司
Shenzhen MT Information Consulting Co., LTD
版权所有. 侵权必究

Chapter 1 WEB测试技术

Chapter 2 APP测试技术

Chapter 3 游戏测试技术

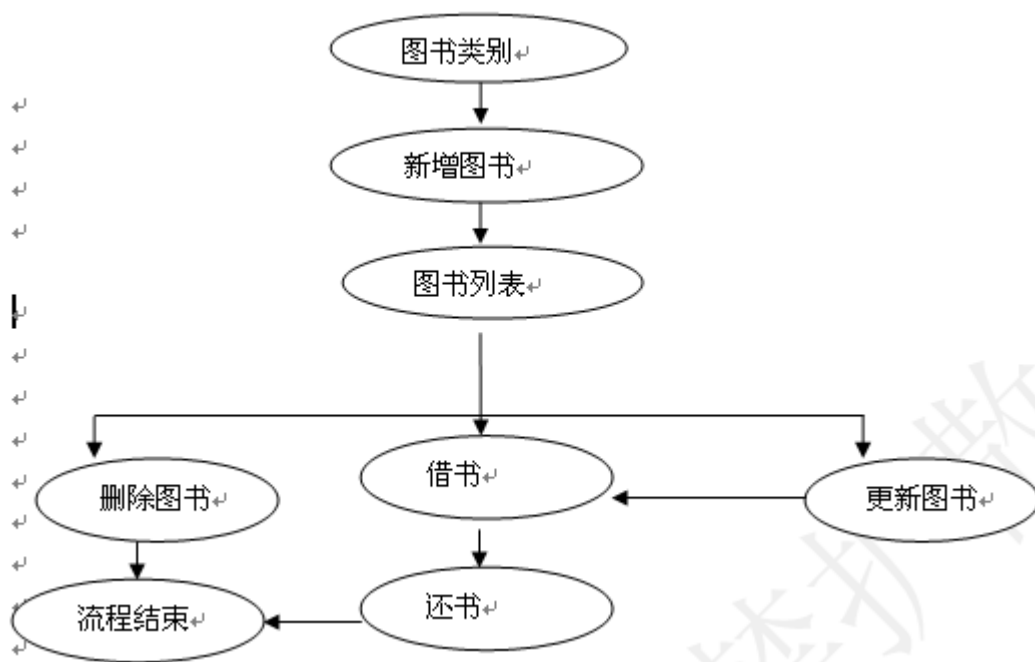
1.1 数据流测试

1.2 组合测试

1.3 单个测试

1.1.1 数据流测试（重点）

- 数据流测试主要采用场景法进行设计用例，主要用户版本验证测试（BVT），或者冒烟测试。
- 流程图：我们实现图书管理中的借还流程。



1.1.1 数据流测试（重点）

■ 测试点

- ① 正常的图书借用流程
- ② 正常的图书还书流程
- ③ 删除图书后，结束图书借还流程
- ④ 更新图书后，仍可进行借还流程

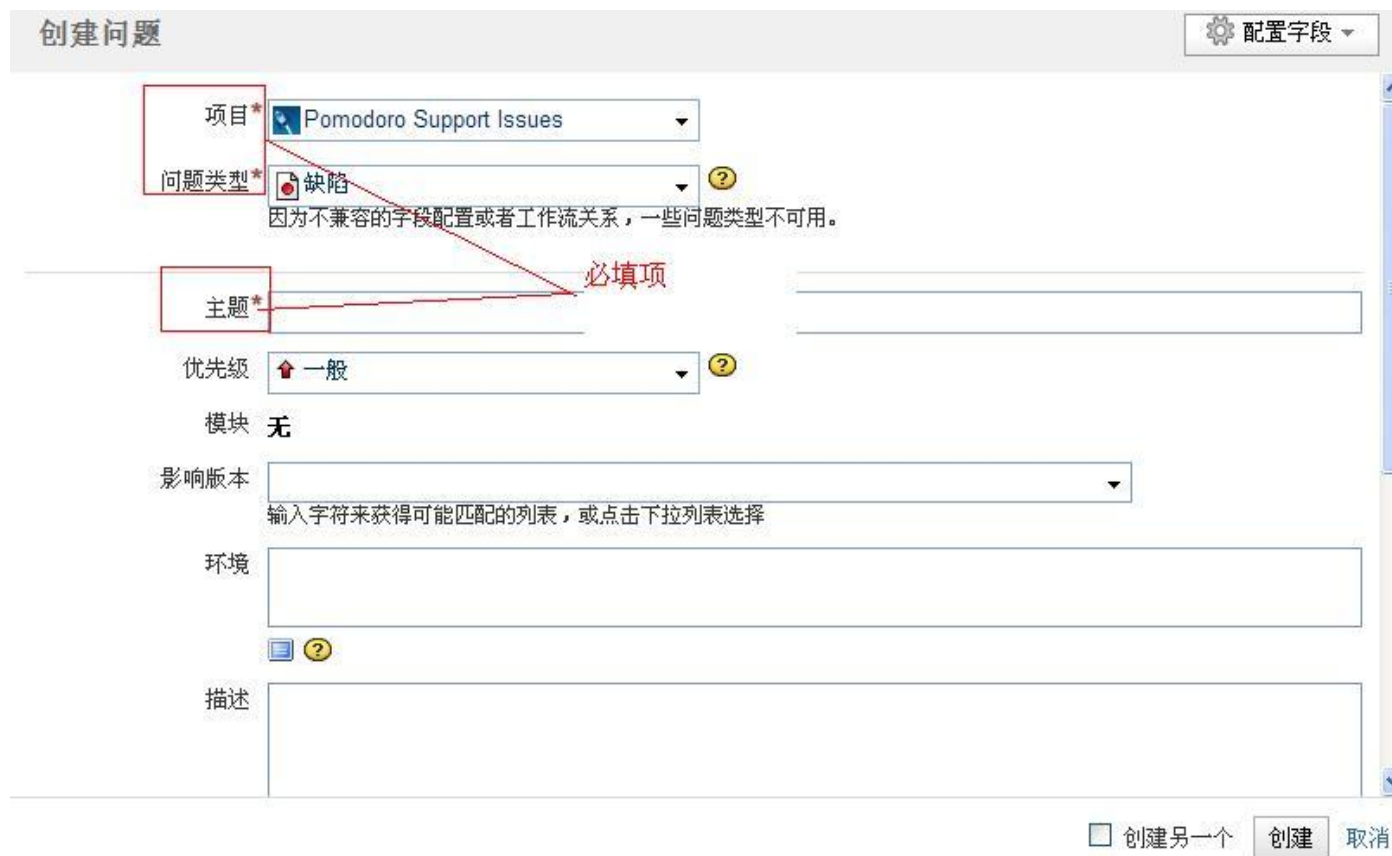
■ 测试用例：测试点对应的就是测试用例名称，但步骤就相对比较长。

- ① 在新增类别页面，新增名称为历史的类别
- ② 在图书新增页面，选择历史，并新增图书为甄嬛传
- ③ 在图书列表页面，选择甄嬛传，并点击借用

1.2.1 组合测试（重点）

1、必填键检查应该填写的项没有填写时系统是否都做了处理，
对必填项是否有提示信息，如在必填项前加*

2、实例：



创建问题

配置字段

项目* Pomodoro Support Issues

问题类型* 缺陷 ?
因为不兼容的字段配置或者工作流关系，一些问题类型不可用。

主题* 必填项

优先级 一般 ?

模块 无

影响版本
输入字符来获得可能匹配的列表，或点击下拉列表选择

环境

描述

☐ 创建另一个 创建 取消

1.2.1 组合测试（重点）

■ 根据一个页面里面的所有控件、文本框等条件构成判定表、因果图，进行用例的设计，并且简化用例。

■ 先根据必选框进行集成设计用例

		1	2	3	4
选项	客户名称	1	1	0	0
	地区	1	0	1	0
动作	添加成功	1			
	添加失败		1	1	1

■ 化简：

		1	2	3
选项	客户名称	1	1	0
	地区	1	0	-
动作	添加成功	1		
	添加失败		1	1

■ 覆盖用例：

用例覆盖	1	2	3
客户名称	金威啤酒	青岛华生	空
地区	天津和平区	不选	香港沙田区

1.3 单个测试（重点）

■ 页面元素

- (1) 文本框:
- (2) 下拉:
- (3) 链接 **Link**:
- (4) 单选框:
- (5) 复选框:
- (6) 分页:
- (7) 表单
- (8) 向上、向下
- (9) 月历控件
- (10) **Ajax** 技术
- (11) 按钮
- (12) 搜索
- (13) 热键
- (14) **Back** 回退
- (15) 附件
- (16) 更新
- (17) 本地化
- (18) **Loading**
- (19) 价格
- (20) **Cookie**
- (21) **Session**
- (22) **Config** 配置文件
- (23) 文本区域

1.3.1 文本框测试-字符长度测试（重点）

- 1、字符串长度检查：输入超出需求所说明的字符串长度的内容，看系统是否检查字符串长度，会不会出错。
- 2、实例：

会员名：
41个字符

 5-25个字符

1.3.2 文本框测试-字符类型检查（重点）

- 1、字符类型检查：在应该输入指定类型的内容的地方输入其他类型的内容(如在应该输入整型的地方输入其他字符类型),看系统是否检查字符类型,会否报错。
- 2、实例：



1.3.3 文本框测试-标点符号测试（重点）

- 1、标点符号检查输入内容：包括各种标点符号，特别是空格，各种引号，回车键，看系统处理是否正确。
- 2、实例：

1 填写账户信息

2 验证账户信息

会员名：

6个字符

✖ 包含非法字符

1.3.4 文本框测试-中文字符测试（重点）

- 1、中文字符：处理在可以输入中文的系统输入中文,看会否出现乱码或出错。
- 2、实例：因为中文，日文，韩文等是unicode8的，英文是AscII。

1.3.4 文本框测试-测试用例（重点）

1、主要方式：等价类划分和边界值法

等价类划分分为有效等价类、无效等价类

首先，确定校验参数：

数字 0, 1, 2, 3……

字母 a, b, c, d……

特殊字符 ~ @ # \$ % & _ 空格

运算字符 + - * / () []

汉字（占两个字符）

根据以上参数，分别划分有效等价类和无效等价类

		有效等价类	无效等价类
输入条件	文本框		
输出条件	合法文本框		

2、边界值法

首先设定好文本框的边界范围，如该文本框的范围为【3, 14】

	覆盖用例(编写数据)
输入 2 个字符	Ab
输入 3 个字符	abc
输入 4 个字符	abcd
输入 13 个字符	
输入 14 个字符	
输入 15 个字符	

当然，写边界值的用例可以结合等价类划分的用例，两者同时覆盖。

1.3.5 重复测试（重点）

- 1、信息重复在一些需要命名，且名字应该唯一的信息输入重复的名字或ID，看系统有没有处理，会否报错，重名包括是否区分大小写，以及在输入内容的前后输入空格，系统是否作出正确处理。

2、实例：

注册

 请输入你的详细信息。

用户名*
用户名已存在。

密码*

确认密码*

全名*

邮件*

1.3.5 重复测试用例（重点）

- 1、输入相同的用户名
- 2、输入相同的ID

此处最主要是验证数据库里的关键字，关键字是不允许重复的。

1.3.6 下拉框测试-测试用例（重点）

- 1、测试默认值正常显示
- 2、测试下拉框中的显示值与后台模块设置值一致
- 3、测试下拉框的滚动条，滚动的时候如果到底，不会溢出页面
- 4、测试下拉框正常点开，正常选择
- 5、测试下拉框数据显示正常，不会被下拉框挡住
- 6、测试下拉框数据设置默认颜色为黑色
- 7、测试下拉框数据排序
- 8、测试下拉框数据按照需求如果只有4个，不需要滚动条

1.3.7 WEB链接测试（重点）

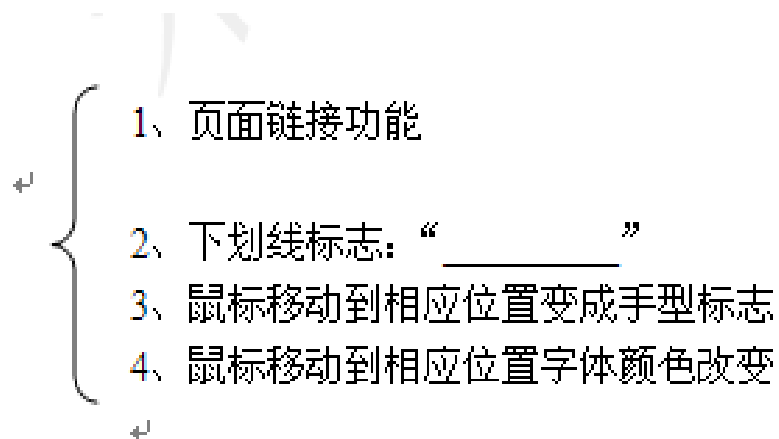
- 1、页面链接测试是检查每一个链接是否都有对应的页面，并且页面之间切换正确； 链接检查工具 Xenu。
- 2、实例：

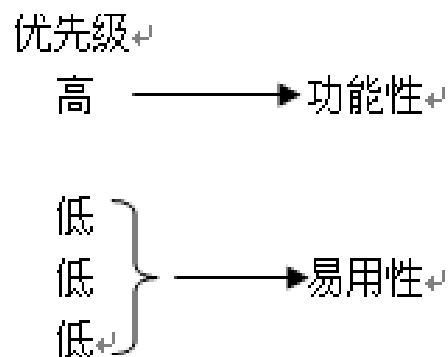


[新闻](#) [网页](#) [贴吧](#) [知道](#) [音乐](#) [图片](#) [视频](#) [地图](#)

1.3.7 链接测试-测试用例（重点）

■ 链接测试用例设计思路

- 
- 1、页面链接功能
 - 2、下划线标志：“_____”
 - 3、鼠标移动到相应位置变成手型标志
 - 4、鼠标移动到相应位置字体颜色改变



1.3.8 单选框测试-测试用例（重点）

■ 单选框测试用例设计思路

☒ 审核文档 ☐ 更新HTML ☐ 移动文档 ☐ 删除文档

↵

↵

需验证

- 1、可点击↵
- 2、点击后圆圈内有实心小黑点，表示选中↵
- 3、在同一选项中，点击另一个前一个被取消，未选中↵
- 4、选中之后的关联结果↵

↵

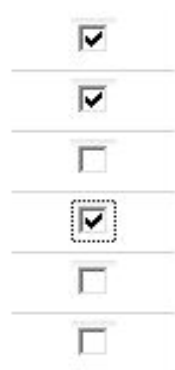
↵

异常

- 1、不能点击↵
- 2、勾选后位置错误（乱跳）↵
- 3、...↵
- ↵
- ↵

↵

■ 复选框测试用例设计思路



需测试

- 1、可选，可多选
- 2、点击后，正方形框内显示“√”，表示选中
- 3、对同一个复选框先选中，再点击一次，正方形框内“√”消失，取消选中
- 4、测试选中后的关联结果

异常

- 1、不可选
- 2、可选，但不可多选
- 3、勾选后位置错误（乱跳）
- 4、对同一个复选框先选中，再点击一次，正方形框内“√”未消失，功能失效

■ 分页测试用例设计思路

共 3 页/73条记录 首页 1 2 3 下页 末页

需测试

- 1、可点击分页，并跳转点击的分页
- 2、测第一页数据条数
- 3、当页面处于第一页时，“首页”按钮成灰色，不可点击
- 4、同理，当页面位于最后一页时，“末页”按钮也不可点击
- 5、当页面处于非首页或非末页时，点击数字“2”、“上页”、“下页”、“末页”测试可点击
- 6、测末页数据条数
- 7、当某页面数据位 0 条时，分页格式显示：“共 0 页/0 条记录”信息
- 8、两页以上才有“翻页”
- 9、验证倒数第二页（边界值测试）

异常

- 1、不可点击
- 2、页面显示数据不正确
- 3、同一条数据在不同页面显示
- 4、...

■ 表单测试用例设计思路

数据正确性↵

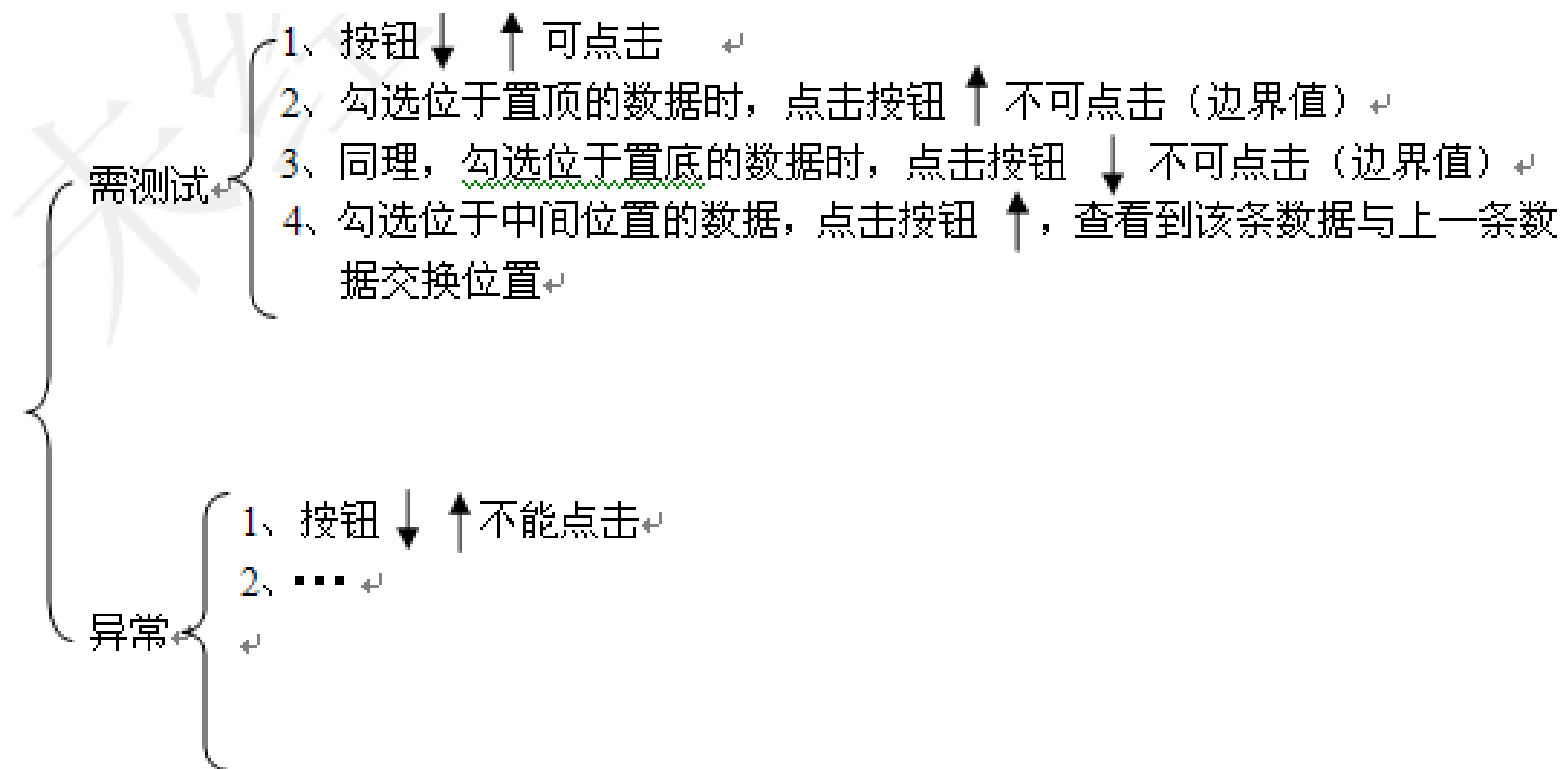
全选/反选: ↵

需测试↵

- 1、按钮可点击↵
- 2、点击“全选”按钮后，当前页面所有数据前的复选框显示“√”状态，被选中↵
- 3、点击“全选”按钮选中，再点击“反选”按钮，当前页面所有数据前的复选框中的“√”消失，取消选中↵
- 4、只勾选其中某一、两条数据后，点击“反选”按钮，之前被勾选的数据前面的复选框中的“√”消失，选中其他数据↵

- 1、按钮不可点击↵
- 2、点击“全选”按钮后，未全部选中↵
- 3、... ↵

■ 向上向下测试用例设计思路



■ 日历控件测试用例设计思路

- 1、选一个日期，文本框正确显示
- 2、点击“《《”，则后推一年
- 3、点击“》》”，则前进一年
- 4、过去的日期（根据需求，确定为可选或不可选）
- 5、对应的文本框可否输入
- 6、根据平年/闰年测2月份的28/29号
- 7、选择“今天”日期
- 8、星期六/天（也是根据需求，确定为可选或不可选）
- 9、月历中，非本月的日期显示为灰色
- 10、验证点击年份出现的下拉框
- 11、验证点击月份出现的下拉框
- 12、开始时间要小于停止时间，否则报错

■ JS测试用例设计思路

- 1、鼠标在相应位置可点击并且可修改（例如：√变为×）
- 2、修改后在页面中可直接查看结果，但需到前台或数据库中验证修改结果。

1.3.15 按钮测试（重点）

- 1、检查按钮的功能：是否正确，如update, cancel, delete, save等功能是否正确。
- 2、实例：



登录名： [手机动态密码登录](#)

手机号/会员名/邮箱

登录密码： [忘记登录密码?](#)

☐ 安全控件登录

登 录

[支付宝账户登录](#) [免费注册](#)

■ 按钮测试用例设计思路

- 1、可点击，并有相应的功能
- 2、鼠标移到该位置，鼠标箭头变成相应设定的形状标志（例如：手型）
- 3、按钮显示的文字正确

1.3.16 删除测试（重点）

- 1、检查删除功能在一些可以一次删除多个信息的地方，不选择任何信息，按” delete”，看系统如何处理，会否出错；然后选择一个和多个信息，进行删除，看是否正确处理。
- 2、实例：



1.3.16 删除测试（重点）

■ 删除测试用例设计思路

- 1、多选删除功能
- 2、单选删除
- 3、不选删除
- 4、全选删除
- 5、翻页多选删除
- 6、关联删除，例如图书已经借，故此图书类别不可删除
- 7、确定
- 8、取消
- 9、直接删除

1.3.17 搜索测试（重点）

- 1、Search检查在有search功能的地方输入系统存在和不存在的內容，看search结果是否正确。如果可以输入多个search条件，可以同时添加合理和不合理的条件，看系统处理是否正确。
- 2、输入信息位置注意在光标停留的地方输入信息时，光标和所输入的信息会否跳到别的地方。
- 3、实例：

搜索

另存为

项目: 全部 ▼

问题类型: 全部 ▼

状态: 全部 ▼

经办人: 全部 ▼

dd



+ 添加过滤器

没有搜索到符合条件的问题

■ 搜索测试用例设计思路

- 1、可点击
- 2、鼠标移到该位置，鼠标箭头变成手型标志
- 3、搜索可按文本框、下拉、复选框等组合条件进行搜索
- 4、可以自动显示匹配出的数据
- 5、搜索的数据根据库存进行排序
- 6、搜索的结果正确

1.3.18 键盘测试-快捷键测试（重点）

- 1、快捷键检查是否支持常用快捷键，如Ctrl+C Ctrl+V Backspace等，对一些不允许输入信息的字段，如选人，选日期对快捷方式是否也做了限制。

2、实例：



The screenshot shows a web form with the following fields and a context menu:

- 主题***: A text input field. Below it, a red error message reads: **你必须填写问题的主题。**
- 优先级**: A dropdown menu showing **一般** (General) with a yellow question mark icon to its right.
- 模块**: A text input field containing the text **无** (None).
- 影响版本**: A text input field with a placeholder text: **输入字符来获得可能匹配的列表，或点击下拉列表选择** (Enter characters to get a list of possible matches, or click the dropdown list to select).

A context menu is open over the '主题' field, containing the following options:

- 撤销 (U) (Undo)
- 剪切 (T) (Cut)
- 复制 (C) (Copy)
- 粘贴 (P) (Paste)
- 删除 (D) (Delete)
- 全选 (A) (Select All)

- 1、测试“Ctrl+C、Ctrl+V”是否支持，例如：是否可以直接Ctrl+C复制出密码框的“***”号密码
- 2、测试系统设定的热键

1.3.19 WEB浏览器中的Back测试（重点）

- 1、浏览器上的back或者键盘上的back：重复提交表单一一条已经成功提交的纪录，back后再提交，看看系统是否做了处理。
- 2、检查多次使用back键的情况在有back的地方，back，回到原来页面，再back，重复多次，看会否出错。
- 3、实例：



1.3.19 键盘测试-回车/Back键测试（重点）

1、回车键检查在输入结束后直接按回车键,看系统处理如何,会否报错。

2、实例：



创建问题 配置字段

项目* Pomodoro Support Issues

问题类型* 缺陷 ?
因为不兼容的字段配置或者 workflow 关系，一些问题类型不可用。

主题*
你必须填写问题的主题。

优先级 一般 ?

模块 无

影响版本
输入字符来获得可能匹配的列表，或点击下拉列表选择

环境

描述

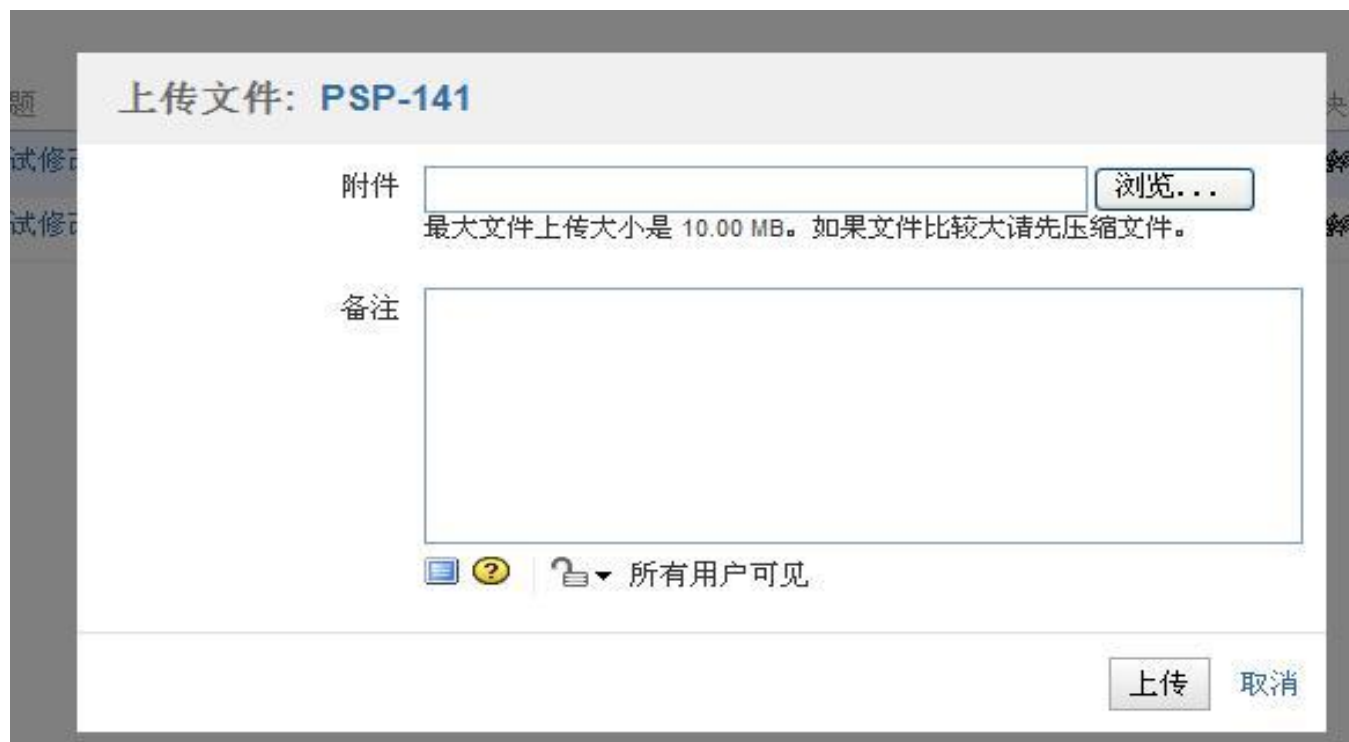
☐ 创建另一个 创建 取消

1.3.19 键盘测试-回车/Back键测试（重点）

- 1、测试回车键后系统可正常相应。
- 2、测试能否返回到登录页面，例如：网银的付款页面不能够回退，因为付款的操作已经进行了，回退也没用。
- 3、回退back后，数据是否正确。

1.3.20 上传下载测试（重点）

- 1、上传下载文件检查上传下载文件的功能是否实现，上传文件是否能打开。对上传文件的格式有何规定，系统是否有解释信息，并检查系统是否能够做到。
- 2、实例：






上传文件: PSP-141

附件 浏览...

最大文件上传大小是 10.00 MB。如果文件比较大请先压缩文件。

备注

   所有用户可见

上传 取消

1.3.20 上传下载测试（重点）

- 1、测试点击浏览按钮弹出windows路径选择界面框
- 2、测试选择后出现在文本框的path格式
- 3、测试图片单个、多个上传
- 4、存储位置的校验，在FTP或者在数据库
- 5、能够上传附件的大小边界值，例如：最大能传10M的附件

1.3.21 更新测试（重点）

- 1、检查添加和修改：是否一致检查添加和修改信息的要求是否一致，例如添加要求必填的项，修改也应该必填；添加规定为整型的项，修改也必须为整型。
- 2、检查修改重名：修改时把不能重名的项改为已存在的内容，看会否处理报错。同时也要注意：会不会报和自己重名的错误。

1.3.21 更新测试用例（重点）

■ 按钮测试用例设计思路

- 1、系统是否即时刷新页面，保持数据在线变化
- 2、刷新过后的数据正确性

■ 表单测试用例设计思路

- 1、语言文字（注意：德语的字符是英语的4倍，所以相应的文本框大小要变为4倍）
- 2、时间格式（根据不同国家更换时区以及跟换时间格式，如格林时间）
- 3、货币符号的改变
- 4、Mapping（页面各个模块层的位置）
- 5、习惯性功能（个性化需求），如：有些页面的主页不能有广告

■ Loading测试用例设计思路

- 1、程序加载几秒需要loading页面？
- 2、Loading页面的提示文字正确性。

1.3.24 Cookie/Session测试用例（重点）

■ Cookie测试用例设计思路

- 1、先登录一次，验证cookie是否保存
- 2、清除cookie，再登陆页面数据为空

■ Session：将页面停留一段时间，不做任何操作，看会不会页面过期

■ Config 配置文件测试用例设计思路

- 1、用来配置系统的初始化，所以要测试系统启动的页面默认值
- 2、修改其他值，需要检验系统正确相应
- 3、如果修改为无效的值，系统报错会如何提示

■富文本控件和Office控件测试用例设计思路

- 1、copy excel、web、或者link格式的内容在文本区域中，
会变为什么格式
- 2、测试文本区域的FCK（副文本控件）的编辑功能

1.3.27 相关性测试（重点）

- 1、相关性检查：删除/增加一项会不会对其他项产生影响，如果产生影响，这些影响是否都正确。
- 2、实例1：人员-公司名称-职位绑定，当选择人后，公司名称和职位自动赋值。一旦把职位删除，人员无法绑定赋值。
- 3、实例2：页面按钮是可变动的，一旦删除了中间的按钮，对左右按钮的位置，功能是否有影响。

2.1 认识移动终端

2.2 APP测试要领

2.3 APP测试规程

2.1.1 移动终端种类



2.1.2 终端OS

■IOS

■Android

■Windows Phone

■Symbian

■Windows Mobile

■Blackberry



2.1.3 IOS介绍

iOS是由苹果公司开发的手持设备操作系统。苹果公司最早于2007年1月9日的Macworld大会上公布这个系统，最初是设计给iPhone使用的，后来陆续套用到iPod touch、iPad以及Apple TV等苹果产品上。iOS与苹果的Mac OS X操作系统一样，它也是以Darwin为基础的，因此同样属于类Unix的商业操作系统。原本这个系统名为iPhone OS，直到2010年6月7日WWDC大会上宣布改名为iOS。截止至2011年11月，根据Canalys的数据显示，iOS已经占据了全球智能手机系统市场份额的30%，在美国的市场占有率为43%。

- 出色的触控体验
- 强大的APP Store
- 安全性及扩展性强

2.1.3 IOS介绍

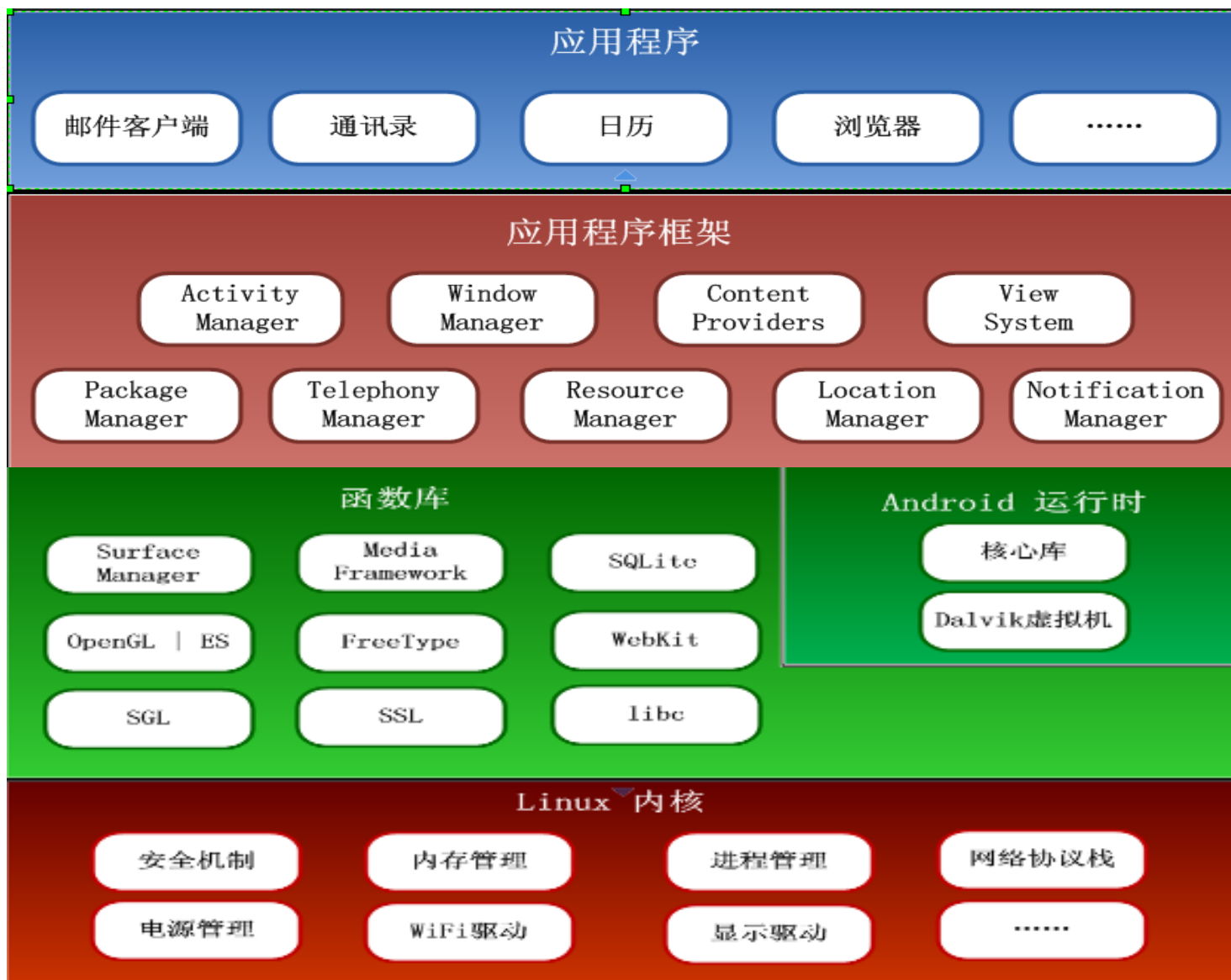


2.1.4 Android介绍

Android是一种以Linux为基础的开放源代码操作系统，主要使用于便携设备。尚未有统一中文名称，中国大陆地区较多人使用“安卓”或“安致”。Android操作系统最初由Andy Rubin开发，最初主要支持手机。2005年由Google收购注资，并组建开放手机联盟开发改良，逐渐扩展到平板电脑及其他领域上。Android的主要竞争对手是苹果公司的iOS以及RIM的Blackberry OS。2011年第一季度，Android在全球的市场份额首次超过塞班系统，跃居全球第一。2012年7月数据，Android占据全球智能手机操作系统市场59%的份额，中国市场占有率为76.7%。

- 全新开源系统
- 自由度高
- 安全性低

2.1.4 Android介绍



2.2.1 测试类别

- 平台测试（终端OS供应商）
- 整机测试（终端厂商）
- APP测试（APP应用提供商）

2.2.2 测试要领

■ UI界面

- ① 控件显示（位置、大小、特效）
- ② 文字、图片、图标显示（位置、大小）
- ③ 触控操作（触摸、按键）
- ④ 控件焦点
- ⑤ 提示框、对话框的显示及操作
- ⑥ 整体排版及控件顺序
- ⑦ 界面回退机制
- ⑧ 特殊语种

■ 功能流程（基本功能与常规软件测试相同）

- ① 应用本身之间的流程
- ② 应用与应用之间的流程
- ③ 应用与系统之间的流程
- ④ 消息处理机制（中断优先级）
- ⑤ 安装、卸载、更新

2.2.2 测试要领

■ 易用性

- ① 提示性文字、图片、图标易于理解
- ② 操作流程常规化，符合通用标准，简单明了
- ③ 各种操作控制符合人体工学设计
- ④ 整体视觉效果

■ 兼容性

- ① 与系统本身兼容
- ② 与系统中其他软件兼容

■ 安全性

- ① 用户信息安全
- ② 数据存储可靠

2.2.2 测试要领

■ 性能

- ① 软件启动
- ② 触控响应
- ③ 数据上传下载
- ④ 其他功能性能

■ 资源测试

- ① 系统内存
- ② 设备磁盘空间
- ③ 控件队列
- ④ 文件句柄
- ⑤ 网络（网络类型、运营商）

2.2.2 测试要领

■ 设备相关

- ① SIM卡（单卡/双卡）
- ② 有线耳机
- ③ 蓝牙设备（耳机及其他蓝牙设备）
- ④ 数据线及电源设备
- ⑤ 其他配件
- ⑥ 数据传输及第三方同步软件控制

■ 极限压力

- ① 字符串过长
- ② 文件过大或过多
- ③ 控件中信息量过大
- ④ 边界性的操作（音量过大/过小，文件太大/太小）
- ⑤ 反复操作及界面切换
- ⑥ 长时间运行
- ⑦ 网络信号强弱

2.2.2 测试要领

■ 中断

- ① 拔插中断（耳机、USB）
- ② 系统中断（闹钟、事务提醒）
- ③ 其他应用中断
- ④ 网络占用、内存占用中断

■ 收费相关：

- ① 计费点
- ② 会员制

2.2.3 测试思维

- 终端资源有限（CPU、内存、磁盘）
- 环境特殊（移动空间、网络环境、应用五花八门）
- 用户为核心（文化背景、操作习惯）
- 什么操作都有可能（变态思维）
- 用户体验至关重要

2.2.4 测试方法

- 手工测试
- 半自动化测试（借助测试辅助工具）
- 自动化测试

2.3.1 操作规程

- 系统刷机
- 软件安装卸载
- 系统监控
- 运行日志

2.3.2 测试设计与执行

- 测试环境
- 测试步骤
- 监控日志更胜一筹

2.3.3 APP常见Bug

- 获取通讯录问题
- 反复操作问题
- 网络切换问题
- 中断问题

- 3.1 游戏团队组成
- 3.2 游戏测试是什么
- 3.3 游戏软件测试组成
- 3.4 游戏软件测试特性（重点）
- 3.5 如何测试游戏（重点）
- 3.6 手机游戏测试

3.1.1 游戏团队组成

- 游戏策划/产品设计
- UI设计：这个与常规软件不一样，UI设计消耗人力较大
- 游戏开发
- 游戏测试：这个与职业玩家不一样

3.1.2 游戏测试日常工作

日常工作：测试部的员工每天都在玩游戏么？不，我们在对游戏进行系统的测试，并且不断的自我更新，提高工作效率，寻找更好的工作方法。

■ 第一个挑战——游戏内测

从最初的简单大厅两个人物一张地图到现在上千人一起参与内测，华丽的大厅，各种不同的人物造型，似曾相识却又新奇的地图，上上下下心血的付出，夜以继日看不到未来的疯狂度日。

■ 开展工作

- 1、测试部不是玩游戏而是测试游戏，我们针对每个版本给出相应的测试计划，每个人都有自己的任务，在版本最初的几天完成，然后组织所有人一起进行模拟类测试，压力测试。
- 2、测试并不像想象中，其实是一项比较枯燥的工作，大部分tester是新人，他们从不理解到理解，不适应到适应的过程中付出了很多，同时也收获了很多。部门的气氛也由原来彼此陌生，逐渐逐渐转为热情待人，相互尊重相互帮助。这也是测试部逐渐走向成熟的重要表现。

一款好的游戏是“打磨”出来的。

3.2 游戏测试是什么

- 游戏软件测试作为软件测试的一部分，它具备了软件测试所有的一切共同的特性。
 - 1、测试就是发现问题并进行改进，从而提升软件产品的质量。
 - 2、测试都是需要测试人员按照产品行为描述来实施（产品行为描述可以是书面的规格说明书，需求文档，产品文件，或是用户手册，源代码，或是工作的可执行程序）。
 - 3、每一种测试都需要产品运行于真实的或是模拟环境之下。
 - 4、每一种测试都要求以系统方法展示产品功能，发现其中出错的原因，从而让程序人员进行改进。

3.3 游戏软件测试组成

游戏软件测试主要分为两部分组成：

- 传统的软件测试
- 游戏本身的测试

游戏特别是网络游戏，它相当于网上的虚拟世界，是人类社会的另一种方式的体现，所以也包含了人类社会的一部分特性。

同时它又是游戏所以还涉及到娱乐性，可玩性等独有特性，所以测试的面相当的广。这是游戏世界测试。

游戏从进入市场到开始运营，必须经过两次测试：内部测试和公开测试

很多玩家对于自己的感兴趣游戏，通常都是在游戏开始内部测试时便争取测试号，拿到拥有内测权限的帐号后就加入到游戏当中。

在测试时期便进行游戏，会给玩家带来两大好处：

- 事先了解游戏，测试时累积的经验在游戏开始运营后更好的进行游戏。
- 提先体验游戏，以便审核游戏是否符合自己的要求。

3.4 网络游戏测试

测试网络游戏的时候，需要考虑如下问题：

- 1、功能是否实现
- 2、人们在进行操作时会如何做
- 3、可能有多少种做法
- 4、这些做法应该有什么样的响应
- 5、哪些做法是被禁止的
- 6、在进行了被禁止的操作后应该有什么的响应

3.4 游戏软件测试特性

■ 游戏情节的测试

指游戏世界中的任务系统的组成，或称为游戏世界的事件驱动，并可称为游戏情感世界的测试。

■ 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，属性等等），保证游戏世界竞争公平。

■ 游戏文化的测试

比如整个游戏世界的风格，是中国文化主导，还是日韩风格等等，大到游戏整体，小到NPC(游戏世界人物)对话，比如一个书生，他的对话就必需斯文，不可以用江湖语言。

3.4 游戏中的游戏情节测试

- 前期的策划阶段只是对游戏情节大方向上的描述，并没有针对某一个具体的情节进行设计。
- 进入设计阶段时，某个游戏情节逻辑已经完整形成。
- 策划可以给出情节的详细设计说明书，称为任务说明书，通过任务说明书我们可以设计出任务测试案例。
 - 1、比如某一个门派的任务由那些组成，我们可以针对它们设计出完整的任务测试案例，从而保证测试可能最大化的覆盖到所有的任务逻辑。
 - 2、如果是简单任务，还可以提出自动化需求，采用机器人自动完成。

3.4 游戏软件测试特性

■ 平衡性

- ① 经济能力
- ② 技能
- ③ 属性
- ④ 角色
- ⑤ 装备
- ⑥ 升级
- ⑦ 关卡
- ⑧ 。 。 。

3.4 魔兽世界 - 平衡性



3.4 游戏可玩性测试人员

■ 内部的测试人员

他们都是精选的**职业玩家分析人员**，对游戏有很深的认识，在内部测试时，对前面的四点进行分析。

■ 游戏媒体专业人员

利用它们对游戏作分析与介绍，既可以达到宣传的效果，又可以达到测试的目的，通常这种方式是比较好的。

■ 外面玩家

利用外部一定数量的玩家，对外围系统进行测试，他们是普通的玩家，但却是我们最主要的目标，主要的来源是大中院校的学生等等，主要测试游戏的可玩性与易用性，发现一些外围的Bug。

3.5 如何测试游戏

在一款游戏中，我们需要查些什么？

■ 从内容角度

AI Behavior: 游戏中人物的动作或反应方面

Animation: 动画效果方面

Camera: 游戏中镜头视角的问题

Crash: 游戏死机或出错

Collision: 游戏中的碰撞

Controller: 与鼠标及键盘等相关的bug

Engine: 引擎方面的问题

Graphic: 贴图及图像方面的问题

Level Design: 关卡中的设计问题

Menu: 菜单

Music: 背景音乐及其他音乐方面

Sound Effect: 音效

Special Effect: 游戏中特效方面

Suggestion: 关于游戏的一些提议性问题

Text: 文字方面的bug

Weapon: 武器方面

User Interface: 用户界面方面

3.5 如何测试游戏

■ AI Behavior: 游戏中人物的动作或反应方面

包括游戏中的主要角色和其他角色（例如NPC）例如：一个性格懦弱的角色在受到攻击时，它的反映应该是逃跑，而不是反抗。

注意：

- 1、当你能够和一个角色产生互动时，请从不同角度进行接触。
- 2、检查一个AI的路径是否正确是需要大量时间的，耐心和想法很重要。
- 3、如果一个角色会依据你的行为作出不同的回应，那么请把所有此类角色都查一下。

3.5 如何测试游戏

■ Animation: 动画效果方面

Animation Bug指的是角色如何来做它的动作的。比如：跑步的动作，那些过分夸张或者不符合实际的动作都是此类bug。因此，在测试的时候务必把角色所有的动作都测试，更要留意各种组合动作。NPC类角色同样不能忽略。

■ Sound: (Music & Effect)

Sound 分为音乐和音效2种，声音在许多游戏中占有重要地位。测试声音时请时刻带上耳机。

3.5 如何测试游戏

■ Camera: 游戏中镜头视角的问题



轮胎悬空

3.5 如何测试游戏



在视频里，我们可以发现，这其实是个视觉bug，法师在施放冰霜之环的时候，如果取消掉，地上的冰霜法阵不会消失，还会停留很长一段时间，于是这个法师就铺天盖地的刷出了许许多多的霜环，当然，它们都是没有真实效果的，不然就真不得了了。

3.5 如何测试游戏

■ Graphic: 贴图及图像方面的问题

- ① 检查所有地图中的物体，每一个角度都不放过。
- ② 用Zoom In/Out (视角缩放) 复查。
- ③ 如果存在多种Camera设计，如第1人称，第3人称，或夜视，热视。都需要全部检查。

3.5 如何测试游戏

■ Graphic: 贴图及图像方面的问题 (Continued)



贴图丢失

贴图错误
膝盖、肚脐
长在背后



贴图错误

3.5 如何测试游戏

■ Collision: 游戏中的碰撞

如果说Graphic是游戏中玩家能看见的东西，那么Collision就是游戏中玩家通过操作人物能够“感觉”到的东西。

注意：

- 1、仔细检查所有的物体。
- 2、把重心放在容易出错的地方：角落、多种地形汇合处和地图边界。
- 3、思维活跃，不能只按照游戏主线进行测试。
- 4、耐心

3.5 如何测试游戏

■ Collision: 游戏中的碰撞 (Continued)



3.5 如何测试游戏

■ Text: 文字方面的bug

- 1、Text包括游戏中出现的所有文字。
- 2、除了错别字和错句，一些需要屏蔽的文字也是测试内容。

■ Controller: 与鼠标及键盘等相关的bug

- 1、实际操作遇到与设计有出入的结果。例：按 ‘w’ 应为方向键前，实际操作中角色后退。

3.5 如何测试游戏

■ Level Design: 关卡中的设计问题

- 1、所有设计的关卡不经过实战测试，是无法确定其难度，平衡性，可玩性是否达到标准的。
- 2、寻找此类Bug时，需要客观的判断，不能因为自己能顺利够完成游戏就认为难度过低。同样，不能因为自己完不成而觉得需要修改。必要的时候询问其他测试员的意见可能会帮你解决难题。

3.5 如何测试游戏

■ Menu: 菜单

- 1、所有游戏中会出现的菜单都需要仔细的检查其功能。
- 2、菜单是为玩家精心设计的工具，一切以玩家使用的角度出发，避免一些多余的菜单。

■ User Interface: 用户界面方面

- 1、和菜单一样都是为玩家服务。
- 2、界面的友好度关系到玩家对游戏的评价。
- 3、UI的特点：只有玩家本人才能看到。例：主要角色的生命、防御等数值。

3.5 如何测试游戏

■ Weapon: 武器方面

- 1、一些游戏，特别是射击类游戏重点。
- 2、武器的平衡性需要大量时间测试，包括武器威力、弹数、种类等等。

■ Special Effect: 游戏中特效方面

- 1、同样以视觉效果出现，和Graphic不同的是：Graphic是关卡中已有的场景，而特效必须是玩家通过一些操作、互动产生的视觉效果。例：手雷爆炸、打开开关出现光影效果。
- 2、好的影片需要特效，好的游戏也一样。

3.5 如何测试游戏

■ Crash：游戏死机或出错

- 1、任何Crash对于一个游戏来说都是致命的。
- 2、任何地方都可能出现Crash。
- 3、每一个bug都有它的重现方法，那些所谓的random bug也是如此，尽力寻找重现方法，有时候你需要尝试上百遍。

■ Engine：引擎方面的问题

- 1、Engine处于游戏设计的最底层，许多普遍存在的bug可能都是Engine出错。例：所有地图的贴图都会闪烁。
- 2、大部分对于Engine的测试都是压力测试。我们会在同一时间内触发尽可能多的事件来观察游戏是否会Crash。

3.5 如何测试游戏

■ Suggestion: 关于游戏的一些提议性问题

- 1、设计文档中可能存在一些不足，又或许你有更好的想法。
当测试过程中遇到一些不足之处，你可以报此类bug。
- 2、通常一款游戏开发过程中，越到后期，这些Suggestion被采纳的可能性越低。当测试进入中后期，避免报此类的bug，多把时间放到其他测试中去。

3.5 如何测试游戏



3.5 评测角度(详细描述)

- 美术资源测试
- NPC对话测试
- NPC特定功能测试
- 道具使用中的数据测试
- 角色升级的人物属性变化
- 聊天系统测试
- 各种任务的测试
- 战斗测试

3.5 评测角度(详细描述)

- 物品捡取丢弃测试
- 登录的测试
- 技能使用测试
- 怪物死亡爆率测试
- 地图跳转点测试
- 文化（书生-书生语言）

3.5 性能测试/压力测试

- 各地图能承载的人数
- 各地图怪物的刷新率
- 地图内不同怪物的搭配测试
- 爆出能力测试
- 服务器承载量同时能力

3.6 手机游戏测试内容（一）

■ 功能性

功能性测试重点检测软件的安装与卸载、功能表现等手机使用条件对游戏的影响

- 1、游戏运行中接听电话、短信后，能否返回到中断的游戏画面继续游戏
- 2、游戏设置中是否可以关闭声音、振动功能
- 3、游戏菜单中是否有详细的操作帮助说明
- 4、棋牌益智类游戏能否积分上传

3.6 手机游戏测试内容（二）

游戏娱乐性内容的评价分为画面、游戏性、操作性三大部分

■ 游戏画面评价

- 1、游戏背景——游戏背景层次是否丰富鲜明，制作精细，发色数高，与前景用色对比明显
- 2、游戏前景——游戏场景中前景数量是否较多，造型丰富各有特点，细节刻画丰富，颜色丰富，与游戏内容相符
- 3、人物和物品造型——角色的肢体细节及物品造型是否丰富，比例正常，色彩艳丽
- 4、人物动作或物体运动状态——人物动作攻击、移动动作是否姿势丰富，流畅连贯制作精细，无跳越感
- 5、游戏特效——游戏中出现特效数量多少，效果是否细腻

3.6 手机游戏测试内容（三）

■ 游戏性评价

- 1、关卡设计——评价故事设定是否完整，游戏中任务、谜题安排的合理性，场景设计问题，关卡设计等
- 2、难度设计——评价游戏难度设计是否合理
- 3、游戏模式——评价游戏模式是否多样

■ 游戏操作性测试

- 1、游戏流畅性——评价游戏运行是否流畅，按键延迟的有无，操作有无惯性等
- 2、按键位置——评价游戏主要功能键位置是否合理顺手或键位可自行设定

3.6 手机游戏测试内容（四）

■ 手机游戏兼容性测试

- 1、手机操作系统—重点检测不同的手机操作系统对游戏的影响，该款游戏运行的可能性。
- 2、手机的内存—检测手机的内存对游戏的运行响应时间，画面的流畅程度，游戏能否顺利完成等的影响。
- 3、手机屏幕尺寸大小—检测游戏画面能否正确显示，是否会出现诸如手机游戏的画面不足屏幕宽度，甚至画面边缘超出屏幕宽度等严重影响游戏享受的问题。
- 4、手机屏幕的刷新频率—重点检测游戏图像能否正常显示。

■ 游戏的绿色标准

- 1、静态指标——包括暴力度、色情度、恐怖度、社会道德度、文化内涵度5项内容。
- 2、动态指标——包括PK（砍人）行为、非法程序（外挂）、聊天系统的文明度、游戏内部社会体系的秩序、游戏形象宣传、游戏时间限制、社会责任感7项内容。
- 3、游戏等级——按游戏定级标准来认证手机游戏的等级。

联系我们

电话：0755-83221336/13928429246（微信同步）

邮箱：service@mtesting.net

官网：www.mtesting.net

学习社区：www.mtesting.cn

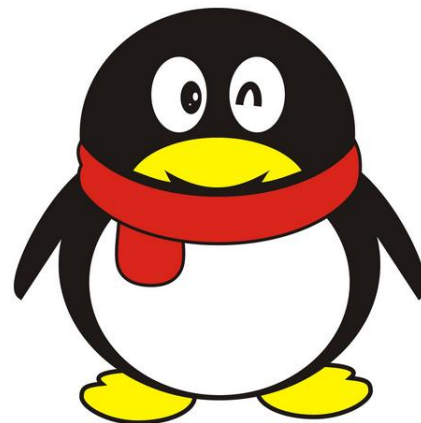
地址：深圳市福田区彩田南路深圳青年大厦五楼



手机扫描访问官网



手机扫描关注公众号



QQ群号:15233368

感谢您对我们的关注



Thanks&Best wishes for you!

多动脑 勤动手 定成功