**同步语文概要设计**

**趣味识字**

Revision: V6.1.0

Release Date: 2018-02-26

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Parts** | **Comments** |
| **V6.0.1** | **2018-02-26** | **卢浪平** | **趣味识字** |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **程序概要设计CheckList** | | |
| **序号** | **检查项** | **检查结果** |
| 1 | 文档是否为doc格式 | **是** |
| 2 | 文档中是否写明了日期、作者和文档的版本 | **是** |
| 3 | 文档中是否写明了对业务需求的理解和分析 | **是** |
| 4 | 文档中是否写明了对技术需求的理解和分析 | **是** |
| 5 | 文档中是否写明了对项目整体的框架设计 | **是** |
| 6 | 文档中是否写明了对外部资源（外部合作、内容资源等）的依赖 | **是** |
| 7 | 文档中是否写明了对第三方库等开源项目的依赖 | **无** |
| 8 | 文档中是否写明了对数据结构的设计 | **是** |
| 9 | 文档中是否写明了对接口的定义（后端接口、内部接口、外部接口等） | **是** |
| 10 | 文档中是否写明了风险项 | **是** |

同步语文V6.0.1概要设计

目录

[同步语文V6.0.1概要设计 3](#_Toc507418287)

[一、需求分析 4](#_Toc507418288)

[1.1.1产品需求： 4](#_Toc507418289)

[1.1.2技术需求 4](#_Toc507418290)

[1.2.1产品需求分析 4](#_Toc507418291)

[1.2.2技术需求分析 4](#_Toc507418292)

[1.2.3技术方案实现 4](#_Toc507418293)

[二、框架设计 7](#_Toc507418294)

[2.1整体框架图 7](#_Toc507418295)

[2.2模块图 8](#_Toc507418296)

[2.3界面结构图 9](#_Toc507418297)

[2.4核心功能流程图 10](#_Toc507418298)

[三、资源需求 10](#_Toc507418299)

[3.1跨部门资源 10](#_Toc507418300)

[3.2第三方资源 10](#_Toc507418301)

[3.3框架选型/开源项目选型 10](#_Toc507418302)

[四、接口设计及约定 10](#_Toc507418303)

[4.1数据格式定义 10](#_Toc507418304)

[4.2后台接口定义 11](#_Toc507418305)

[4.3对外接口定义 11](#_Toc507418306)

[4.4系统接口约定 11](#_Toc507418307)

[4.5依赖接口约定 11](#_Toc507418308)

[五、风险项 11](#_Toc507418309)

[六、时间安排 12](#_Toc507418310)

[附 12](#_Toc507418311)

# 一、需求分析

## 1.1.1产品需求：

1. 调整同步语文首页（书架）
2. 在书架新增加个人字词和同步字词库入口
3. 新增同步字词库
4. 新增个人字词库
5. 新增字词游戏个人等级详情说明和排行榜
6. 调整个人中心界面
7. 调整标记界面
8. 调整同步教辅UI，新增添加字词到个人字词库功能
9. 对问作业提供添加生字到个人字词库功能

## 1.1.2技术需求

1. 代码质量、代码管理、代码维护、编码规范符合要求
2. 应用性能符合标准
3. 清理技术债务，无A类、B类缺陷

## 1.2.1产品需求分析

详见：<【学习类 · 同步语文】趣味识字 V6.0.0 需求拆分-20180209-产品评优先级.xlsx>

## 1.2.2技术需求分析

1. 代码质量、代码管理、代码维护、编码规范符合要求，详见附录<代码质量指标>，无新增指标
2. 应用性能达标,详见附录<性能指标 – 同步语文>，无新增指标
3. 清理技术债务，修改bug，无A类、B类问题

## 1.2.3技术方案实现

1. **个人字词库**
2. **总体方案**
   * + 1. 同步语文(课文、听写、字词学习等)、问作业添加字词到本地个人字词数据库，添加字词时，如果已存在则将字词重新修改为新添加
       2. APP本地缓存个人字词库信息，同时网络也存储个人字词库信息，使用机器码作为唯一标志，保存的信息包括上次更新时间、字词信息、字词状态等
       3. APP启动进入个人字词库首页、网络重新连接且需要更新、数据变化时都会触发更新本地数据到网络
       4. APP每次上传都会生成本次更新时间，并将本次更新时间上传到网络保存，上传成功后，本地也同时保存本次更新时间
       5. 每次触发上传动作时，将会根据本地保存的上次更新时间是否存在和网络保存的上次更新时间是否存来确定是否同步数据和采用的同步策略
       6. 累计修改数据>N条时才上传数据，避免频繁联网
3. **更新时机**
   1. 启动同步语文并进入个人字词库主界面
   2. 添加字词
   3. 消除字词
   4. 清空已消除字词
   5. 网络重新连接且需要更新时
4. **更新策略**

【上次更新时间】在网络和本地都有存储，其对应值在不同端的有无状态以及对应策略如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **本地** | **网络** | **状态** | **策略** | **备注** |
| 有 | 有 | 正常 | 不需要同步网络数据  将本地修改的数据上传到网络 | 这里不比较本地和网络更新时间是否一致，默认一致 |
| 有 | 无 | 错误  （网络数据被清除？） | 不需要同步网络数据  将本地修改的数据上传到网络 | 异常：网络数据会出现部分缺失 |
| 无 | 有 | 本地数据被清除 | 1. 本地无缓存数据：   获取网络数据保存到本地   1. 本地有缓存数据：   获取网络数据保存到本地，并将原来本地缓存的数据合并到当前数据，覆盖已存在数据? |  |
| 无 | 无 | 未使用 | 将本地修改的数据上传到网络  第一次本地无修改数据也要上传更新时间，避免不修改数据时多次联网 |  |
|  |  |  |  |  |

1. **更新流程**

****

1. **如何确定本地数据是否修改**

本地创建修改记录表，记录修改信息，只记录修改字词在数据库表中对应的ID；上传数据时只上传修改记录表中记录的字词

1. **方案风险以及可能出现的问题**
2. 方案以平板机器系统时间为准，用户可以调整系统时间，字词添加、消除、清除时间会出现逻辑上的混乱，不影响使用
3. 网络数据清除后，会导致旧数据丢失，且本地旧数据也不会上传，新修改数据会上传
4. 长时间不联网的情况下，突然联网可能会产生大量数据上传到网络，需要考虑大数据上传和数据压缩，修改数据>N条，分批上传
5. 本地数据被清除后，同步网络数据到本地，可能会一次拉取大量数据，耗时可能会偏长
6. 可能会出现上传不及时的情况(未联网或者上传失败)，下次进入会再次上传，不清除APP缓存数据的情况下不影响使用
7. **同步词库**
   1. **数据来源：**从已有同步教辅数据中解析每课生字
8. **个人字词库、同步字词库题目生成方案**
9. 题目生成规则

待定

1. **字词学习个人信息**
2. **本地/网络个人信息同步规则**：
   * + 1. 本地和网络都存储完整个人信息，本地存总经验、新增经验值，网络只存总经验
       2. 如果本地没有保存有完整个人信息（未使用或者本地数据被清除），则从网络获取完整个人信息和总经验值，保存在本地，然后上传本地缓存的新增经验值
       3. 如果本地保存有完整个人信息，直接上传本地缓存的新增经验
       4. 上传新增经验值时，如果成功，服务器返回总经验值，本地保存返回的总经验，本地缓存的新增经验值减去刚刚上传的部分新增经验值；如果失败，等待下次上传时机触发再次上传
       5. 经验值变化后，实时上传到服务器
3. **本地/网络个人信息同步时机：**
4. 启动同步语文，进入主界面？进入个人中心？
5. 经验值变化(增加)，实时上传
6. 网络重新连接且需要同步时
7. **游戏方案**
8. 启动游戏：同步语文App获取到游戏需要的字词信息后，从本地数据库获取一些状态信息（如果需要），然后从题目资料包中获取题目信息，将这些信息通过Intent传递到游戏App中。
9. 游戏结束返回结果给同步语文App

游戏结束后，调用同步语文服务提供的接口将结果传回给同步语文

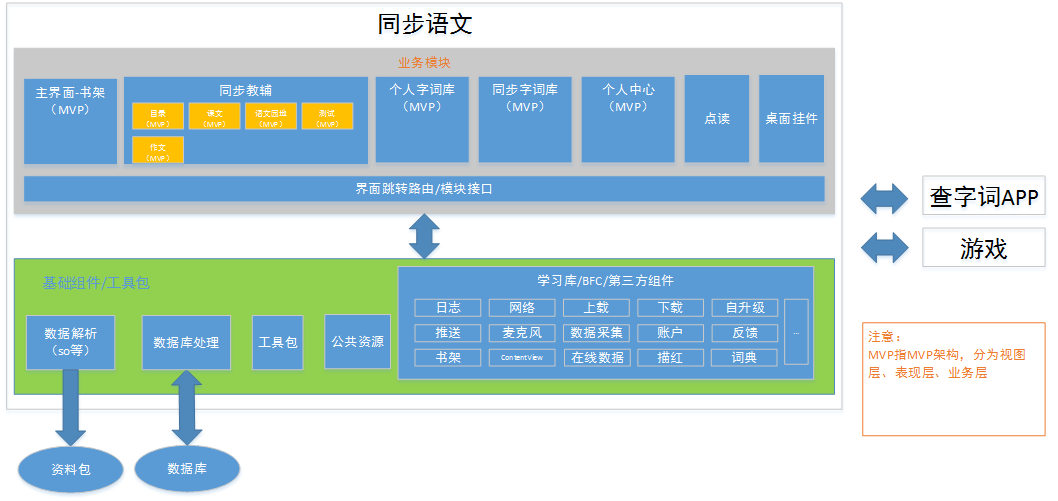
1. **字**

# 二、框架设计

## 2.1整体框架图

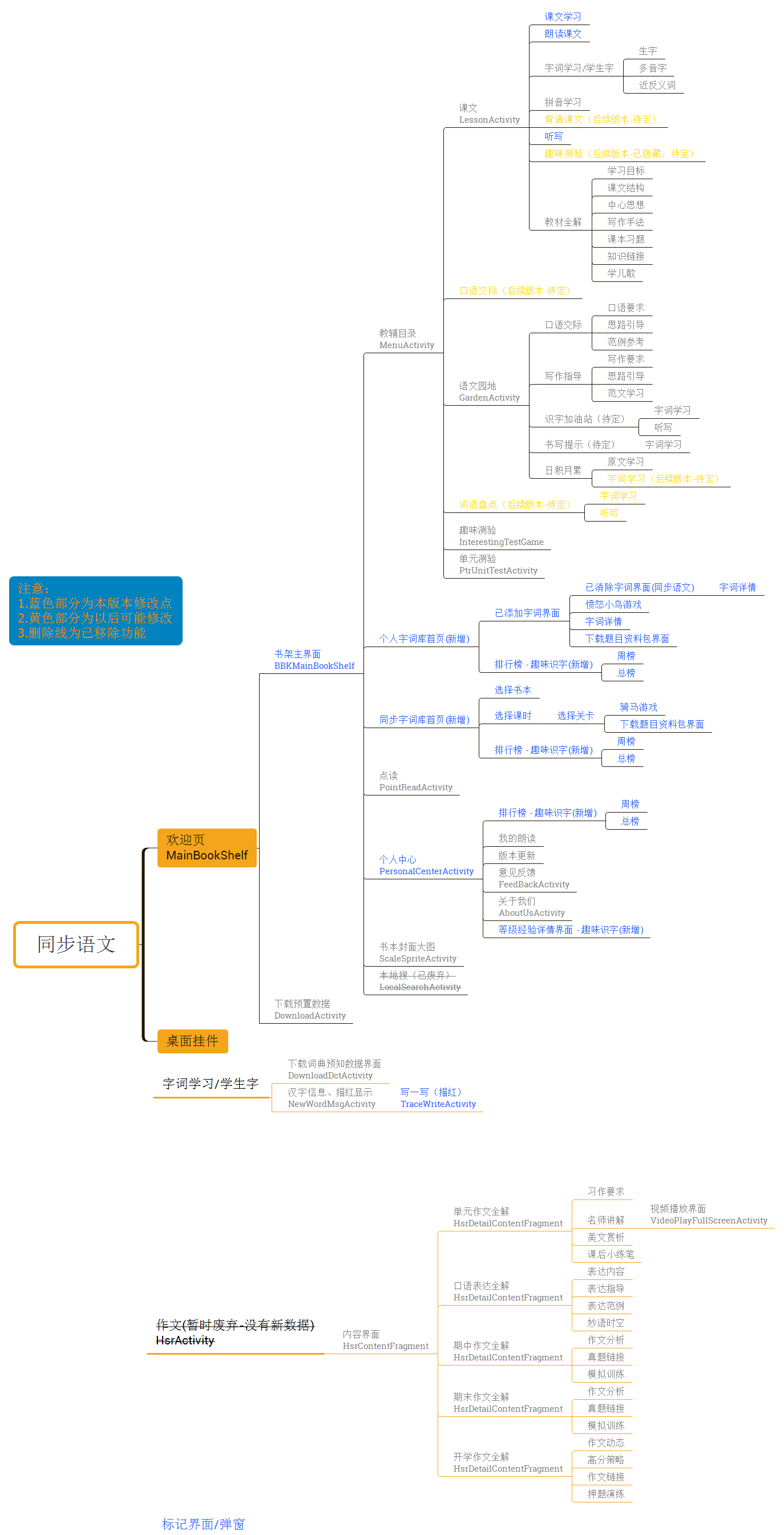
无调整

## 2.2模块图



新增个人字词库、同步字词库模块

## 2.3界面结构图



## 2.4核心功能流程图

1. 个人字词库

详见：1.2.3技术方案实现中的流程图

# 三、资源需求

## 3.1跨部门资源

* 1. 需要内容计划组协助制定和打包数据
  2. 需要后台小组完成网络数据存储并对客户端提供相应接口
  3. 需要游戏小组完成游戏开发
  4. 需要视觉出效果图和切图

## 3.2第三方资源

无

## 3.3框架选型/开源项目选型

无

# 四、接口设计及约定

## 4.1数据格式定义

1. 题目数据结构

待定，在开发前详细设计中完成

1. Android本地数据库
   1. 个人字词库

待定，在开发前详细设计中完成

* 1. 同步字词库

待定，在开发前详细设计中完成

* 1. 个人信息

待定，在开发前详细设计中完成

* 1. 正在学的书本信息

待定，在开发前详细设计中完成

## 4.2后台接口定义

4.2.1 **个人字词库**

1. 根据机器码获取上次更新时间，返回更新时间
2. 根据机器码获取所有字词信息，返回个人字词库所有字词信息
3. 根据机器码批量更新字词信息并修改更新时间(可能存在只修改更新时间，不更新字词信息的情况)

4.2.2 **字词学习个人信息**

1. 根据机器码获取字词学习个人信息，包括总经验值等，总经验值初始状态为-1
2. 根据机器码上传字词学习个人信息，包括新增经验值等，返回总经验值

4.2.3 题目资料包

1. 根据题目资料包下载地址下载资料包数据

## 4.3对外接口定义

4.3.1 愤怒的小鸟游戏接口

1. 启动游戏接口
2. 游戏结束结果返回接口

4.3.2 骑马游戏接口

1. 启动游戏接口
2. 游戏结束结果返回接口

## 4.4系统接口约定

系统预置游戏

## 4.5依赖接口约定

不依赖其他应用接口，需要依赖bfc库、学习库

# 五、风险项

|  |  |
| --- | --- |
| **风险项** | **应对方案** |
| 题目资料包，需要制定题目规则、数据结构，并打包相应数据，提供Demo，这些都还没有定 | 需要尽快落实并确定，保持紧密沟通 |
| 需求多，工作量大，测试时间短 | 按模块和界面分部开发，完成一个尽早提测一个 |
| 因为测试提前介入，测试版本提前提供，要保证每个提测版本可测试 | 测试、视觉需要和软件紧密配合，及时沟通，视觉必须在软件开发前一周完成出图且出图要完整 |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 六、时间安排

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **计划内容** | **时间** | **人员安排** |
| 1 | 本地、后台、游戏确定数据结构和接口  数据库搭建  卡牌书架  个人字词库业务接口  游戏业务接口 | 2月23日 -3月2日 | 张春雷、卢浪平、阳荣克 |
| 2 | 同步语文主界面框架  同步教辅主界面  个人中心业务接口  同步字词库业务接口  游戏业务接口 | 3月5日 - 3月9日 | 卢浪平、阳荣克 |
| 3 | 个人字词库 + 同步教辅详细界面  挂件界面 | 3月12日 - 3月16日 | 卢浪平、阳荣克、张少杰 |
| 4 | 同步字词库  后台接口联调 | 3月19日 - 3月23日 | 卢浪平、阳荣克 |
| 5 | 个人中心以及其他  游戏接口联调 | 2月23日 - 3月30日 | 卢浪平、阳荣克 |
| 6 | 学习详情大数据统计 | 2月23日 - | 卢浪平、阳荣克 |
| 7 | AI语音指令 | 2月23日 - | 张春雷 |
| 8 | 题目数据包制作 | 2月23日 – 3.9日 | 魏誉荧 |
| 9 | 后台 | 2月23日 - | 吴智贤 |
| 10 | 游戏 | 2月23日 - | 徐栋梁 |
| 11 | 测试 | 3月12日 – 4月6日 | 张玉兰 |
|  |  |  |  |

# 附

* + - 1. 代码质量指标

1. 符合现有代码规范
2. link检测无严重问题
   * + 1. 性能指标 – 同步语文

|  |  |  |
| --- | --- | --- |
| 性能项 | 指标（开发人员填写） | 基准指标：基于Android 7.0 2G RAM（非特殊说明1G RAM机型要求相同） |
| **冷启动速度** | 参考基准指标2s | 1. 学习类应用：   无欢迎页面：**<=1.5s**  欢迎页面的延时时间为1.5s   1. 工具类应用：   无欢迎页面：**<=1.5s**  欢迎页面的延时时间为1.5s  说明：计算规则为桌面点击图标变灰至内容首页入场动画结束 |
| **过度绘制** | 参考基准指标 | 1.各个界面不能出现全屏（不包含状态栏）**红色及深红色**（>三层绘制）  2.在界面无浮动框（引导，弹框等）条件下，不能出现深红色绘制（四层）超过界面面积**50%**  3.与上个已发布版本应用，过度绘制在无交互修改的前提下只能优化不能增加  测试条件：开发者选项中开启过度绘制检测 |
| **内存泄漏** | 手动操作无内存泄漏异常 | 1. **500ms 100000次** monkey执行无应用内存泄漏（非系统内存泄漏）  2.基本功能（可参考P0级用例）使用无内存泄漏提示（非系统内存泄漏）  测试条件：使用debug版本应用（集成LeakCanary）进行测试 |
| **主线程卡顿** | 无超过1s的卡顿异常 | 1.**500ms 100000次** monkey执行无卡顿异常日志  2.基本功能使用（可参考P0级用例）无卡顿异常日志  测试条件：使用debug版本应用（集成BlockCanary）进行测试  *说明: 卡顿时间控制由BlockCanary工具去控制，将卡顿异常按照时间大小拆分为error, warning, info级别数据输出供开发及测试使用；分级阈值由性能优化组控制，定期提高* |
| **Monkey** | 参考基准指标 | 1. 应用在**12小时、时间间隔500ms（约等于500ms100000次）**无应用本身的必现monkey异常问题 |
| **Else(自定义)** | 功耗  定制化指标 | 1. 正常网络状态书本下载成功率100%； |