

INDIVIDUAL REPORT

FULL NAME: 李梦凡
STUDENT ID: 1612860

DIRECTIONS

To design and develop a game of “三国华容道” based on mobile application development technologies.



EXECUTIVE SUMMARY

REPORT NAME:

《曹操必须死》

Features summary

INSTRUCTIONS: Briefly summarize the main features and the unique/special things about your game.

1. 登录系统：可以存储游戏数据到数据库，不同手机登录同一账号即可共享数据。
2. 排行系统：所有玩家统一排名，分星星榜和积分榜。
3. 选关系统：目前共有 10 个地图，可以动态扩展。
4. 星星系统：每一关有三星，达到指定条件可以解锁星星。
5. 积分系统：根据每次通关记录的步数和时间计算积分，时间增加的越多，对积分的减少效果越弱，步数越少，对积分的增加效果越强。
6. 快速开始：在用户没有关卡选择的前提下，随机选取一关快速开始。
7. 游戏系统：用户可以通过滑动屏幕来实现棋子的移动，滑动起点所在的棋子既是移动的棋子，曹操棋子移动到指定的位置即判定为获胜，并根据所用时间和步数计算积分。



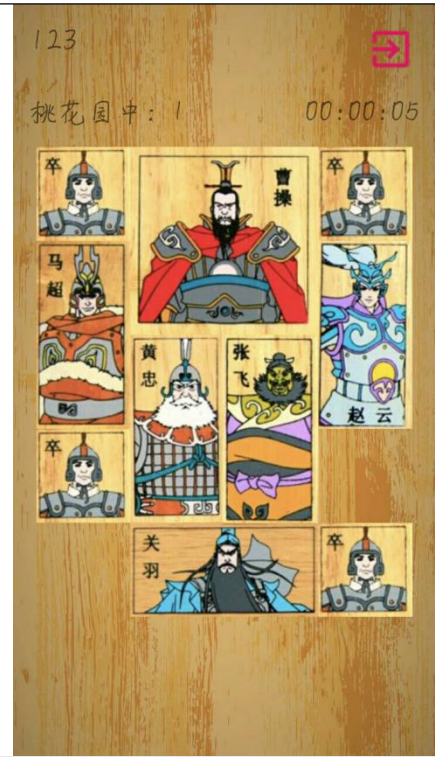
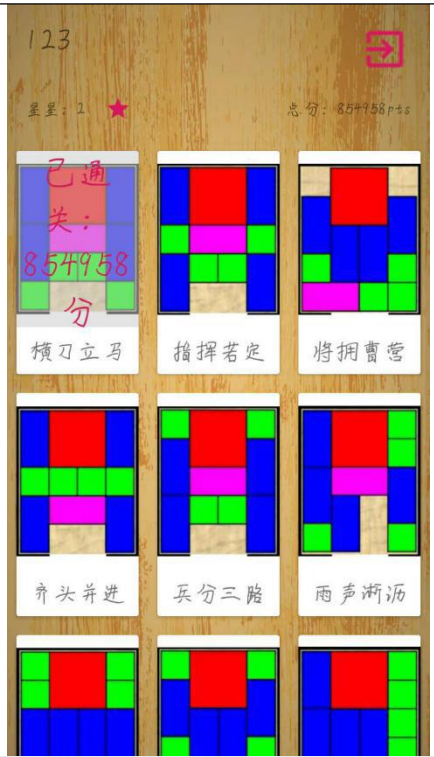

Solution summary


INSTRUCTIONS: Briefly summarize the tools and technologies used in your game.

1. 前端采用安卓实现。
2. 后端采用 springboot 实现。
3. 数据库采用 mysql 实现。
4. 前后端交互使用 AsyncHttpClient+ResponseResult。
5. 代码的迭代和版本管理使用 github。

SCREENSHOTS AND SCENARIOS

| Scene 1 | Scene 2 | Scene 3 |
|--|---|---|
| | | |
| <p>Description: 登录注册页面，用户名不允许重复</p> | <p>Description: 游戏大厅界面，可以快速开始，选择关卡，个人成绩，系统排行</p> | <p>Description: 选择关卡界面，左右滑动即可切换关卡，点击进入</p> |

| Scene 4 | Scene 5 | Scene 6 |
|---|---|---|
|  |  |  |
| <p>Description: 游戏界面，顶栏显示地图名称，步数和计时</p> | <p>Description: 个人成就界面，所有地图，已通关的地图有积分显示，顶栏可以查看星星总数和积分总数</p> | <p>Description: 系统排行界面，分为星星排行榜和积分排行榜，分别显示前五名</p> |

| Scene 7 | Scene 8 | Scene 9 |
|---|---------|---------|
|  | | |

| | | |
|---|---------------------|---------------------|
| Description: 游戏通关界面，显示分数和星星数，分数为步数和时间的函数，通关获得一颗星，100 步内通关获得一颗星，5 分钟内通关获得一颗星 | Description: | Description: |
|---|---------------------|---------------------|

TECHNOLOGIES SPECIFICATION

Frontend Design and Implementation

INSTRUCTIONS: Provide additional detail about the **frontend** design and implementation in your game which should include:

- UI (User Interface)
- UE (User Experience)
- IxD (Interaction Design)

设计

1. 参照原木色背景和棋子，给用户以真实的华容道体验。
2. 以红色为强调色，与木色形成对比，更加吸引用户的注意。
3. 排行榜前三名使用彩色强调色和大字号字体，使得用户有竞争更高分数的动力，以及荣誉感
4. 保存用户的登录信息，第一次登录成功后之后可以自动填充用户名和密码，优化用户体验。
5. 采用滑动屏幕来实现选择关卡，更加贴合用户的习惯。
6. 游戏通关的积分显示框设置了提示按钮，用户可以查看积分以及星星的来源。

实现

1. 所有的顶栏相同，因此可以提出来一个 **topbar** 布局的引用，每个布局通过 **include** 引用同一实例。
2. 游戏界面由三个独立的布局组合而成，分别为 **topbar**，**playing_scorebar**，**playing_map**。
3. 选择关卡布局由水平滑动，占据屏幕宽度且限制滚动范围为一页的循环布局来实现。
4. 成就布局由一个三列的循环布局来实现，排行榜布局由一个一列的循环布局来实现。
5. 游戏布局由一个约束布局来实现，每次移动棋子遍历并改变所有棋子的约束。

Backend Design and Implementation

INSTRUCTIONS: Provide additional detail about the **backend** design and implementation in your game which should include:

- Game strategy
- Service architecture
- Logical algorithm
- Data storage

Controller 层

后端传递给前端的数据格式由 `ResponseResult` 封装，分为三部分：成功/失败，数据和信息。

后端使用 `SpringBoot` 框架，采用 `MVC` 结构，提供给前端的控制器为：

1. `UserController`：注册登录，获取用户信息。
2. `LevelMapController`：获取地图信息。
3. `UserLevelMapController`：写入，获取用户闯关信息。

Service 层

`Service` 层与 `Controller` 层对应，分别提供 `UserService`，`LevelMapService` 和 `UserLevelMapService`。

`Service` 层采用接口加实现的形式，可以便捷的改变实现而不影响对 `Controller` 层的接口。

Model 层

与数据库的链接采用 `MyBatis` 做模型到视图的映射处理。

数据层为三个表：

1. `User`：存储用户信息
2. `LevelMap`：存储地图信息
3. `UserLevelMap` 存储闯关信息。

数据层额外提供三个视图：

1. `UserInfo`
2. `LevelMapInfo`
3. `UserLevelMapInfo`

视图是多个表的联合查询，提供了模型的全部数据，例如，`UserInfo` 视图拥有 `User` 表和 `UserLevelMapInfo` 表联合查询得到的星星数目信息和积分总数信息。

OPERATIONS SPECIFICATION

INSTRUCTIONS: Describe any important software, hardware, or other information technology that you have used to operate your game. That might include a register solution for managing your users, an achievement system to attract attention, a notification and feedback service for optimizing your game, a payment service for profit, and so on.

注册系统：

1. 后台会存储用户的登录信息和闯关信息。
2. 前端会采用 `SharedPreferences` 存储用户的登录信息方便用户下次便捷地登录。

闯关系统：

1. 前端闯关成功后会将时间和步数信息传递给后端，后端进行分数的计算并查询后会返回分数和是否为最高分数。

排行系统：

1. 后端通过 `LevelMapInfo` 和 `UserLevelMapInfo` 的联合查询，可以获取前 n 名的用户信息和成绩， n 为前端传给后端的参数，所以可以支持扩展排行系统为前 10 名或前 50 名而不需要改动其他的代码。

前后端交互：

1. 前端采用 `AsyncHttpClient` 向后端发送 `Http` 请求，并将 `Get` 和 `Post` 请求都封装到了 `HttpUtil` 里，同时前端拥有和后端相同的 `ResponseResult` 类来解析后端的返回数据。
2. 前端通过继承 `AsyncHttpHandler` 实现了一个 `BaseAsyncHttpHandler` 来进行对后端返回值的处理，主要实现的是对返回值成功的处理，每次需要创建一个 `BaseAsyncHttpHandler` 的子类作为匿名类传递给 `HttpUtil` 的 `Get` 或 `Post` 方法。
3. 后端与数据库的交互采用 `MyBatis` 实现 `orm` 的映射。

EXTENSIONS

Game Distribution

INSTRUCTIONS: The game is supposed to be an installable and usable mobile app including necessary interaction with its backend service. Provide the URLs about your game:

- The source code URL: <https://github.com/lilixi/hua.git>
- The released app URL: <https://aifabu.com/IfMj> (爱发布, 应用服务平台)
- The game video (<1 minute) URL:
https://v.youku.com/v_show/id_XNDIyNTk1NTUxNg==.html?spm=a2h3j.8428770.3416059.1

Conclusion and Expectations

INSTRUCTIONS: Summarize your project and give future expectations about yourself and this course.

项目的开发周期为大概一个月左右, 先是提前完成了 10 个单元的所有实验, 之后开始了前端的设计和开发, 前端解决了关键的问题之后才开始后端的设计和开发, 因为比较熟悉 Java 所以后端采用 SpringBoot 框架, 在前后端交互时遇到了困难, 最终采用了 AsyncHttpClient 进行前后端的交互。

之后希望可以在游戏中加入匹配系统的段位系统。

匹配系统: 玩家可以联机进行游戏, 以时间长短和步数多少进行分数的计算, 分数高者为胜, 可以增加游戏的竞争性和可玩性。

段位系统: 玩家在匹配系统中进行游戏, 胜利即可累计积分, 提升段位, 失败则积分会下降, 段位也会下降, 提高了游戏的竞争性。

APPENDIX

Note:

1. This report should be a Microsoft Word or PDF document **no more than 12 pages**.
2. Each student should submit an individual report to **18720429@qq.com** named **“studentID-Name.doc/.pdf”** before **June-15-2019**.
3. If your game has been deployed in a certain app store, it will be appreciated and may get higher final score.

Fiction Story:

“诸葛亮智算华容，关云长义释曹操”，是世人皆知的一个故事，《三国演义》中的重点篇章。

小说写赤壁战前，诸葛亮算定曹操必败走华容，且夜观天象，曹操不当身亡，考虑到曹操与关羽有恩，于是派关云长把守华容道，留个人情与关羽做。小说又写曹操果然由乌林向华容道败退，并在途中三次大笑诸葛亮、周瑜智谋不足，未在险要处暗设伏兵。

然而，一笑笑出赵子龙，多亏徐晃、张郃二人双敌赵云，才使曹操得以逃脱；

二笑笑出张翼德，又是张辽、徐晃二将抵挡张飞，使曹操再次脱险；

三笑非同小可，笑出了关云长，且又在有一夫当关之险的华容狭路上，加之曹军几经打击，此时已无力再战，无奈，曹操只得亲自哀求关羽放行，关羽念旧日恩情，义释曹操，使曹操得以回到江陵。

后关羽为孙权所杀，人头交由曹操，曹操为其塑木制身体。