**12组迭代计划（简化版）**

制定日期：2020/ 10/ 16

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 13 | 项目名称 | My PC Logo |
| 迭代名称 | 技术原型迭代 | 计划起止日期 | 2020/10/16-2020/11/16 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 1 | 选择架构风格，设计架构视图 | 10/16-10/18 | 徐珺涵，敖宇晨 | | 2 | 优化界面设计布局 | 10/16-10/20 | 窦嘉伟，付玉晗，莫兰 | | 2 | 设计关键算法，主要是对命令行及命令文本的解析与响应 | 10/16-10/20 | 敖宇晨，徐珺涵，窦嘉伟 | | 3 | 撰写和评审软件架构文档并搭建起软件架构 | 10/18-10/22 | 徐珺涵，敖宇晨 | | 4 | 选定或撰写编程规范 | 10/20-10/22 | 窦嘉伟，付玉晗，莫兰 | | 5 | 集体学习编程规范 | 10/23-10/24 | 全体成员 | | 6 | 开发界面，包括登录注册界面，单机界面以及联网界面，各自负责相应的界面开发 | 10/25-11/06 | 徐珺涵，窦嘉伟，付玉晗，莫兰 | | 7 | 开发后台，本迭代中只需要支持注册登录、联网匹配 | 10/31-11/06 | 徐珺涵 | | 7 | 实现上述关键算法，即LOGO语言解释的逻辑，包括语义分析 | 10/25-11/06 | 敖宇晨 | | 8 | 语言逻辑层与界面层的交接、前后台的交接 | 11/07-11/09 | 敖宇晨，徐珺涵，窦嘉伟 | | 9 | 测试技术原型，包括基本功能实现的正确性 | 11/10-11/13 | 敖宇晨，徐珺涵，窦嘉伟 | | 10 | 编写《迭代评估报告》评审记录、测试记录以及开发总结 | 11/13-11/16 | 全体成员 | | | | |
| **预期成果：**  1、文档：《迭代评估报告》、架构视图、《编程规范》，《软件架构》  2、模型：项目概念模型、用例分析模型  3、源代码：项目的源代码，放在本次迭代仓库的目录《My\_PCLogo》  4、安装包：本次迭代完成后发布的应用程序安装包 | | | |
| **主要的风险和应对方案：**   1. **技术风险 - 关键算法出现语义分析不明或者错误的bug：**   应对方案：尽量全面测试各种刁钻的命令语句，尽可能测试到所有可以想象到的corner-case，使其预定的技术要求。用户在使用过程中出现bug提供反馈选项（目前来看只能在github上提issue，显然这是对青少年不够友好的），开发者及时修复和更新。   1. **技术风险 - 在联网模式中后台崩溃：**   应对方案：当后台崩溃时前台进行错误提示，并引导用户尝试进行单机模式的使用，同时在项目前期不具备自动监测后台能力时，期望用户向开发人员反馈崩溃信息   1. **技术风险 - 界面美观及趣味性达不到要求：**   应对方案：可以参考网络上现有的界面设计，或者尝试引入他人的界面组件，多征询目标用户对界面的意见，并进行返工。更为进步的，我们期望开发出自定义界面设置，方便用户根据自身的喜好设置界面，如更换背景、字体、颜色等   1. **进度风险 – 在规定时间内无法完成本次迭代所期待的全部目标：**   应对方案：全体开发人员加班，增大开发强度，若仍存在较大风险，可依据实际情况，逐步降低预期目标到本次迭代通过所要求的最低水平  **5、技术风险 - 用户在下载安装包出现时问题：**  应对方案：给出详细的安装指引文档，包含多种可能存在的问题，期望用户尽可能自行解决安装上的问题。对于后续本项目若存在更大的规模（企业级），可以开发出用户社区用以解决用户的问题。我们也会参考其它成熟应用软件的安装包构建方案，尽可能减少此风险的发生概率 | | | |