**项目总结报告**

日期：2021年1月6日星期三

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 15 | 项目名称 | MY\_PC\_LOGO |
| 编程语言 | Go+Javascript | 开发平台和框架 | React+Iris |

|  |  |
| --- | --- |
| **软件需求特性** | |
| **1.是否实现了项目立项时的所有需求？列出实现的新增需求和未实现的需求。**  未实现需求：双人双海龟模式  新增需求：支持定义变量语法  **2.采用哪种架构风格？哪些设计模式？**  BS架构。  组合模式、解释器模式、外观模式  **3.技术方案有哪些亮点？**  使用了SSE技术实现了双人模式下的实时通讯。  在前端手动维护了一棵文件树，便于用户进行项目管理。  在前端实现了LOGO语言的解释器，能进行语法提示和报错。  设计了任务机制，引导新手熟悉软件操作。任务完成后解锁新皮肤，给予用户激励，极大的激发了儿童对编程的兴趣。  编辑器功能完善，快捷键丰富。  构建了debug工具栏，帮助用户调试错误。  **4.是否做了单元测试？是否做了系统功能测试？是否做了性能测试？是否做了兼容性等其他非功能测试？**  前端未进行单元测试，后端进行了单元测试。  完成了所有系统功能测试。  进行了完整的性能测试。  此外，进行了不同浏览器和不同屏幕分辨率下的兼容性测试。 | |
| **项目组成员对项目的贡献度（%）** | |
| 吴侃真（25%）、邹立凯（25%）、王执（25%）、毛彦凯（25%） | |
| **软件度量** | |
| 软件代码行数（不包括注解行、空行和复用代码）： | 5000+ |
| 复用他人代码行数： | 通过npm管理他人的软件包，不便统计 |
| 类的个数： | 20+ |

|  |
| --- |
| **经验、教训和建议** |
| **经验**   1. 技术原型迭代期间，最开始计划采用CS架构，用qml来编写客户端。但当小组成员初步学习qml后，发现很难绘制出美观合适的界面，便抛弃了这一技术框架，选择react来构建前端。由于小组成员曾合作开发过react+spring框架的项目，因此减少了项目技术风险。 2. 在开发过程中，由于进度原因，我们减去了优先级较低的双人双海龟功能，将精力更多地投入到完善已开发功能上，使得项目能够按计划顺利完成。 3. 在开发前及开发过程中，我们使用绘制界面原型的方式确定界面改进的方向。当界面原型经过小组成员评估后，再进行编码实现，提高了开发效率。 4. 根据时间紧张程度，灵活地制定和调整项目迭代计划。在空闲的时间抓紧进度，在其他课程大作业繁重时稍缓进度。在保证了项目进度的同时，避免小组成员承受较大的开发负担。 5. 每个新功能开发完成后，先push到仓库的个人分支。代码经过评议后再合并至项目主分支，保证了项目的代码质量，实现了良好的版本管理。   **教训**   1. 项目在搭建框架时，引入了较多的开源模块。但在后续开发的过程中，这些开源模块引发了难以解决的冲突，不得已只能删去部分模块。在引入模块时没有经过慎重的考虑和评估，浪费了后期的不少精力。 2. 在保持软件兼容性的同时绘制出漂亮的软件界面，其难度大于我们的预期，为了优先保证不同屏幕下的兼容性，软件界面最终没能达到理想中的美观程度。 |