**迭代评估报告（简化版）**

　　　　　　　　　　　　　　　　　　评估日期：2020.11.17

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 15 | 项目名称 | MY PC LOGO |
| 迭代名称 | 技术原型迭代 | 实际起止日期 | 2020.10.11-2020.11.17 |
| 任务达成情况：（完成的任务、实现的功能、进度、质量等）  已完成任务：   1. 文档及设计   《迭代计划》、《vision文档》、《软件需求规约》、《软件架构文档》、  《编程规范》、Use case、UML分析与设计模型   1. 后端框架完成，可以初步实现用户的登录，存取文件等操作 2. 前端完成编程界面的整体布局以及绘图组件   实现的功能：   1. 用户可通过登录界面注册或登录 2. 在命令行或命令文件中输入命令控制小海龟的移动（目前只有部分命令）   进度：  完成项目的架构搭建、完成技术原型，符合老师要求进度。  后端基本完成，前端绘图组件和登录页面完成。  质量：  基本达标 | | | |
| 评审/测试的结果：  技术原型的整体架构符合项目需求，后端采用go语言开发，可以支持高并发；前端采用react框架，界面符合界面原型的设计。文档内容较为完整、充实。  组内评审：  暂无  老师与助教评审：  暂无  测试：  暂无: | | | |
| 问题、变更和返工：（遇到的问题、发生的变更、是否需要返工等）  遇到的问题：   1. 后端使用springboot的框架，对双人房间的高并发支持可能不足。 2. 在服务器端解析代码会导致服务器压力过大。 3. 对软件设计模式不够熟悉，不能灵活运用设计模式优化软件的架构设计。   变更：   1. 采用go语言开发后端，提高服务器端并发性能。 2. 参考codesandbox等在线编程网站，修改架构设计，将代码的解析移至前端，降低后端压力。 3. 将前端由单纯的View更改为MVC架构，便于储存代码、解析代码和绘图。   是否需要返工：  暂时不需要 | | | |
| 经验和教训：   1. 学习了用例实现，到软件架构、物理架构，再到设计模式等多种优化软件设计的方法，理解了优秀的设计对于编码的帮助。 2. 广泛查阅资料，在设计时尝试突破常规，在老师和助教的帮助下，将不寻常的设计合理化，对软件的设计有了更加深刻的理解。 3. 小组成员在讨论软件设计时经常会有不同的想法，在出现争议时，我们通过查阅资料、咨询有经验的同学以及向老师请教等方式，最后达成一致的意见，在这个过程中不仅锻炼了与合作开发者沟通的能力，也学到了更多不同的设计思想。 | | | |
|  | | | |