**迭代计划**

　　　　　　　　　　　　　制定日期：2023年3月18日

| 组号 | 21 | 项目名称 | HyperCraft开放世界体素游戏 |
| --- | --- | --- | --- |
| 迭代名称 | 迭代#1 | 计划起止日期 | 3/18-5/2 |
| 任务、进度安排和人员分配：   |  |  |  |  | | --- | --- | --- | --- | | No | 任务 | 起止日期 | 人员 | | 0 | 进一步完善接口，丰富游戏功能和交互模式。 | 3/18-4/15 | 袁翊天 | | 1 | 熟悉现有接口的使用方法，继续学习相应的开发框架。 | 3/18-4/4 | 黄予晗、任柏俊、夏嘉阳 | | 2 | 确定游戏玩法，设计游戏模式和交互逻辑，并用使用现有接口初步实现。 | 3/31-4/15 | 黄予晗 | | 3 | 初步实现服务器与客户端间的简单交互，客户端可以发送请求，服务器端实现请求的相应。 | 3/31-4/15 | 任柏俊 | | 4 | 利用噪声算法设计地形生成的接口，初步测试和评估生成效果，必要时进行算法的修改调整。 | 3/31-4/15 | 夏嘉阳 | | 5 | 针对已经实现的功能进行项目的评估和调整，对各个模块之间的耦合进行优化调整。 | 4/15-5/2 | 袁翊天，黄予晗，任柏俊，夏嘉阳 | | 5 | 编写《迭代评估报告》 | 4/15-5/2 | 袁翊天，黄予晗，任柏俊，夏嘉阳 | | | | |
| 预期成果：   1. 本迭代的迭代计划 2. Vision文档 3. 软件架构文档 4. UML模型（含用例模型、分析模型和设计模型） 5. 编程规范 6. 技术原型的代码 7. 本迭代的迭代评估报告 | | | |
| 主要的风险和应对方案：  整体上和迭代#0时的风险类似，本次迭代中新增的一些风险如下：   1. 模块耦合风险  * 整体功能调试的时候可能会遇到来自不同模块的bug，由于不同模块的代码开发相对独立，可能会造成调试难度的明显增大。 应对方案：各成员规范代码风格，合理编写注释和文档；调试时可以先记录下问题和minimal reproducible example，便于bug的复现；开发过程中统一时间进行整体返工（如果遇到比较性质比较恶劣的bug，可以提高修复的优先级）。  1. 接口管理风险  * 本迭代中，很多接口会得以实现，项目可调用的API会变得十分繁杂，为工程的管理带来了一些不便。如果错误地使用接口，很可能会造成影响深远的bug。 应对方案：规范注释和文档；利用文档生成工具Doxygen和IDE集中管理项目API。 | | | |