**项目总结报告**

日期：2023/06/17

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 21 | 项目名称 | HyperCraft |
| 编程语言 | C++ | 开发平台和框架 | 跨平台开发, Vulkan |

|  |
| --- |
| **项目工作小结** |
| 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。  新增需求： 1. 引入光照计算  2. 能创建新世界，同时对已有的世界进行修改世界配置的修改以及删除已有世界  未实现的需求： 1. server 联机功能  2.采用哪种架构风格？哪些设计模式？  架构风格：并行架构  设计模式：策略模式，模板方法模式，外观模式  3.技术方案有哪些亮点？  本项目的技术方案亮点有：   1. 基于Vulkan设计实现高效的游戏渲染管线，使用了GPU-Driven Pipeline技术和Render Graph技术 2. 多线程的游戏数据处理系统和更新系统 3. 使用跨平台的图形API Vulkan以及其他跨平台的C++库，能够在Windows、Linux等平台编译并正常运行。   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？  本项目进行了单元测试、系统功能测试、性能测试以及兼容性、可靠性、压力、部署测试。其中   * 单元测试包括对Block类的属性正确性测试 * 系统功能测试包括对体素世界管理、游戏基本功能的测试 * 性能测试包括游戏的平均帧率测试 * 可靠性测试包括游戏的稳定运行时常测试 * 压力测试包括地形频繁加载测试以及超量渲染测试 * 部署测试与兼容性测试包括对游戏免安装性与跨平台性的测试 |
| **项目组成员对项目的贡献度（%）** |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 姓名 | 需求 | 设计 | 编码 | 测试 | 项目管理\* | 小计 | | 袁翊天 | 5 | 0 | 60 | 10 | 40 | 31 | | 黄予晗 | 30 | 30 | 13 | 10 | 20 | 23 | | 夏嘉阳 | 25 | 30 | 15 | 10 | 20 | 23 | | 任柏俊 | 40 | 20 | 15 | 8 | 20 | 23 |   注： 项目管理包括项目计划、报告、沟通与协调等。 |

|  |  |
| --- | --- |
| **软件度量** | |
| 软件代码行数（不包括注解行、空行和复用代码）： | 4997 |
| 复用第三方代码行数： | 299697 |
| 类的个数： | 49 |

|  |
| --- |
| **经验、教训和建议** |
| 经验和教训：   1. 项目前期开发过程中，没有善用Github协作功能。未来可以多利用Github并行开发。 2. 在Client-Server通信以及游戏机制的API接口还未设计完成的情况下，应该进行深入的讨论和交流，以明确接口的实现和使用。这样可以避免在后期开发过程中出现不必要的问题和延误。 3. 由于团队成员的C++掌握程度不同，代码沟通难度较大，导致出现了一些不必要的问题和延误。因此，在开发过程中，应该采用清晰的代码结构和注释，以提高沟通效率。 4. 由于游戏开发技术的学习成本较高，再加上繁重的学业压力，应当在保证完成项目开发的前提下适当宽松迭代的期限，减轻组员的工作压力。   建议：   1. 希望在课程早期提供一些git的教程。几乎所有项目都需要通过git来实现项目管理，在使用git的过程中也遇到了很多问题，耗费了大量时间摸索。而大学课程期间没有专门介绍git的课程。 2. 最好在互联网开发这门课上完之后再来讲这门课，在上完互联网开发之后才开始理解架构风格和设计模式。所以建议软院删除一些工科平台的无用课程，尽早把专业课放到大一和大二上。 |

项目组各成员签字：

