**项目总结报告**

日期：2021/1/8

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 23 | 项目名称 | Endless memories |
| 编程语言 | C# | 开发平台和框架 | Unity |

|  |  |
| --- | --- |
| **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。  项目开始时的需求是代入感强、趣味性强，剧情画面吸引用户，保证游戏统一性，界面易用性等。  基本上实现了所有的需求，并且没有新增的需求。  2.采用哪种架构风格？哪些设计模式？  采用三层架构。但Unity只让我们需要编辑业务层。  设计模式在游戏内唯一的物体上采用单例模式，在存在魔法数字（MagicNumber）的Unity函数调用地方（如场景跳转）使用中介者模式。  3.技术方案有哪些亮点？  Unity帮助我们实现高复用性，可以把常用的游戏物体与组件的组合保存为模板进行复用，我们不仅在此项目，在其它项目内也可  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？  无单元测试——项目大部分依赖的代码在Unity内部（黑盒状态），因此没有白合测试  系统测试——基于游戏体验进行测试：游戏内容、易用性、吸引力  性能测试——测是否影响游戏体验 | |
| **项目组成员对项目的贡献度（%）** | |
| 33% JunHanaizumi：实体逻辑、场景逻辑、模型收集  30% Naomi：物品交互、VR调试、HUD界面实现  27% EdwardRaymond：PVP对战  10% 叶鹏鹏：部分UI | |
| **软件度量** | |
| 软件代码行数（不包括注解行、空行和复用代码）： | 200+300+500+50=1050 |
| 复用他人代码行数： | 无法估量（20000+） |
| 类的个数： | 22 |

|  |
| --- |
| **经验、教训和建议** |
| 技术风险的应对：  我们在项目的开发途中原本以为已经在第二次迭代的时候已经解决了技术风险，但没想到我们实际上只是解决了部分的技术风险，即Unity方面的技术风险，却忽略了VR方面的技术风险。我们真正实现VR的时间点过于靠后，导致这一技术风险被发现的时候也已经是项目后期了，原本应该用于丰富软件内容的时间用于解决这一风险就又造成了项目的进度风险。  万幸的是我们还是在最后解决了这两个风险，不过这还是给了我们一个教训就是我们不应该小看或忽略一些未知因素对项目造成的影响，早一点开始是最好的解决方式。 |