**迭代评估报告（技术原型迭代）**

　　　　　　　　　　　　　　　　　　评估日期：2020年11月17日

|  |  |  |  |
| --- | --- | --- | --- |
| 组号 | 23 | 项目名称 | Endless Memories |
| 迭代名称 | 迭代二 | 实际起止日期 | 2020年10月12日至2020年11月20日 |
| 任务达成情况：（完成的任务、实现的功能、进度、质量等）  完成了：  - 本次迭代的迭代计划  - 本次迭代的迭代评估报告  - 游戏内一个谜题的内容  - 物品所有功能的基本实现  - 游戏进度控制的实现  - 游戏整体路由逻辑实现  - 追赶者的关键算法的实现  - 场景内关键物品与提示物品的建模  - 概念建模  - 软件需求规约  - 架构设计  - 技术源代码  - ppt（阐述软件的价值、特性和优点，汇报《迭代评估报告》的概要情况）  原本的计划是对物品进行分类，但在深入讨论后我们发现关键物品与可消耗物品的分类是没有意义的，因为游戏中会用到的可消耗物品的种类可能总数不会超过5个，因此我们取消了这一分类，以及原本的选择物品弹出按钮再点击新的功能键的设计也非常不符合vr设备的使用习惯（使用vr设备按按钮是非常不便的），因此我们修改成了选择物品后按下物理按钮来实现不同的功能。  同时我们还将检视物品直接整合到左侧的面板（即不需要再按按钮来检视物品），否则在选择物品后还需要按按钮来检视物品对用户来说是很不便的。  HUD方面我们完成了鼠标靠近物品出现提示框。  追赶者（鬼）的基本巡逻AI在早期阶段完成。它会听取声音靠近声源，或看到逃生者时追踪。没有任何迹象时，鬼会定期巡逻几个巡逻点。因为进度很快，后面还追加了被鬼听到声音和被鬼直接看到时的独特BGM。逃生者被鬼触碰时，目前设定为直接游戏失败，跳转至GameOver界面。  PVP对战模式中的目前已完成了基本的功能实现，包括：“死神”能够传送鬼至指定位置，并且能在指定位置制造噪音吸引鬼；同一房间内客户端间数据同步且对游戏物体及脚本的控制权非常清晰，解决了同时操作同一物品的冲突现象；UI方面则在双方视角上方制作了游玩信息面板（倒计时、积分等），而“死神”的技能按钮目前针对键鼠进行了初步的设计，随后将根据vr设备的操作方式进行相应改进。  可以说本次迭代我们算是修改了比较多预期想要实现的目标，但是整体在效果上是一致的。 | | | |
| 评审/测试的结果：（执行了哪些评审和测试？评审和测试的结果如何？）   1. 技术评审测试，其中包括游戏内的各项功能正常使用，以及背包与游戏内物品的交互，是否正常。   结果：   1. 游戏能够正常运行。 2. 背包与游戏内物品交互正常。 3. 背包界面中物理按键与鼠标操作并不同步。 4. 玩家基本功能评审和测试 5. 基本操作正常。 6. 鬼的追踪系统的评审和测试 7. 鬼的追踪系统有一定缺陷，由于头部与身体固定，鬼的视野追踪能力极差，很容易通过拐弯造成一瞬间的跟丢。由于影响到BGM的播放，这里需要后面完善。 8. 鬼的触碰导致游戏失败逻辑正常。 9. 鬼的听辨声源能力正常。 10. 鬼的视野能力正常。   4. 进阶功能（PVP对战模式）  （1）同一房间中的客户端间数据正常同步。  （2）同一房间中的客户端拥有明确的脚本控制权限。  （3）“死神”角色玩家可对鬼进行位置传送，且可对指定地点制造噪音吸引鬼。 | | | |
| 问题、变更和返工：（遇到的问题、发生的变更、是否需要返工等）   1. 背包界面中物理按键与鼠标操作并不同步。   发生变更：修改了原本的计划后又对bug进行了修复，目前已经正常运行。   1. 进阶功能（PVP对战模式）：不同客户端间数据不同步，客户端间的脚本控制权限错乱。   发生变更：通过Photon自带的函数进行了数据同步，并在此基础上进行了改进以达到发送信号执行命令的效果，因此已解决以上问题。 | | | |
| 经验和教训：   1. 教训：需求设计不准确导致进行了大量的返工。   实际上我们因为在原始的设计上有许多缺陷，在项目开发期间就曾多次进行重新设计与返工，由此我们也做了许多无用功，因此本次迭代我们最大的问题应该是设计时没有能够从用户、玩家的角度进行设计，等到真正实现时才发现设计非常的不人性化。  其次，由于我们的初期工作顺序完全颠倒----在本该事先讨论、编写架构/需求/UML之前，就已经写好了极大部分的代码，因此不仅发生了上述的需求变更，由于已有的代码量很大导致一直不存在完善且详细的UML视图，导致沟通效率低下。我们应该先前讨论好架构，接口，需求，再开始进行编写。   1. 经验：对于整个项目，可以从不同的方面进行分析（进程视图等）   结合了课程的内容，我们在本次迭代中对项目的各个方面有了更系统化的分类与了解，由此能够在设计与实现项目时更加有效。  Unity的初次使用让我们体验了全新的开发模式，也带来了很多独特的问题。不好删除的未使用资源非常多（第三方追加的用例）需与当前使用的资源分开存放。脚本在哪个游戏物体的组件上放置需要约束来方便查找。.scene文件的git冲突解决十分困难，需要在分工时避免场景的多人编辑。这些问题在后面的开发环节上需继续留意。 | | | |
|  | | | |