

QT 作业报告

这个需要你自己衡量队
崔鹤龄 郭翔宇 廖致远（队长）
（按姓氏首字母排列）

程序功能介绍：

概述：

本小组作业实现了一个基于地图探索的类 rougelike 式游戏。名为《生息演算 失尘秘境》，游戏的部分灵感和多数素材取材于游戏明日方舟的集成战略和生息演算模式。背景设定是信科的一位大一学生在某天早晨被一位神秘人（ocean cat）邀请，参与体验了一场模拟真实环境下的开放地图冒险游戏。玩家需要从驻扎地（起点），经过地图的不同支路到达终点。并在这一过程中遭遇各种突发事件，经历战斗，获得宝藏，购买道具，最终以尽可能高的评分到达终点。

右图为设计的地图。

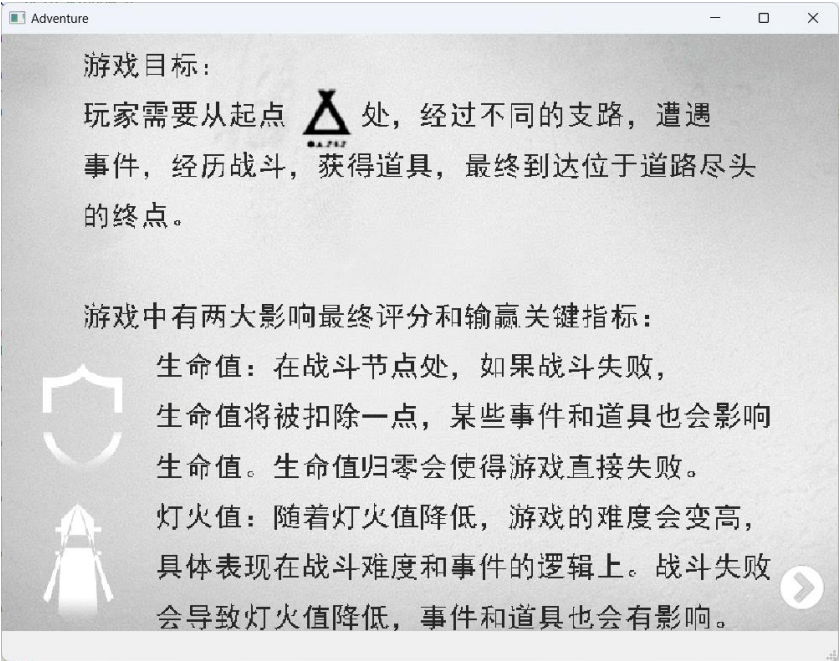


玩法介绍：

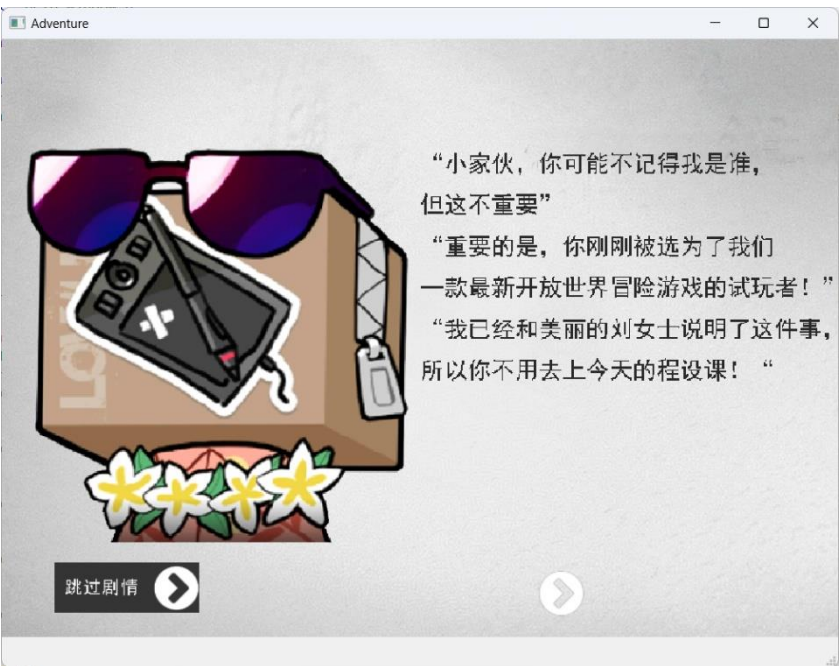
程序开始后，主界面是三个选项，分别对应着开始游戏，游戏帮助和退出按钮。首先介绍帮助按钮：



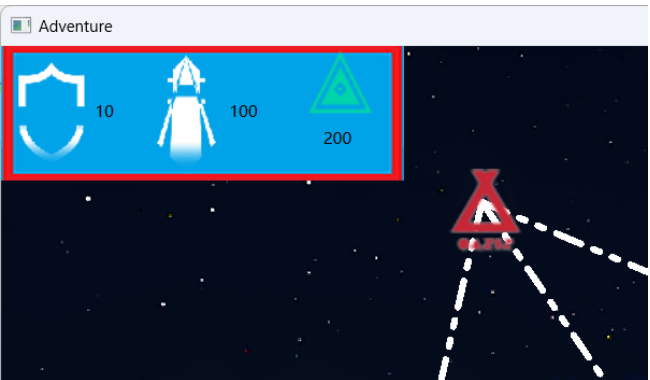
帮助界面：介绍了游戏的基本玩法，分为若干页。右下角包含了一个翻页按钮。



游戏界面介绍：点击开始探索按钮之后，首先会进入剧情提要剧情界面，剧情分为若干页，可以选择直接跳过剧情。



剧情界面结束之后，会跳转到正式的游戏地图。在游戏地图左上角可以看到玩家的生命值，灯火值和货币声明。这三个值会随着探索进度实时更新，并且会影响到玩家的游戏难度和体验。玩家会在各种突发事件中增加或者减少三种指标中的一个，但值得注意的是，玩家的生命值不会在事件节点中降为 0，最多降至 1。而灯火值始终限制在 0-100 范围内。当玩家

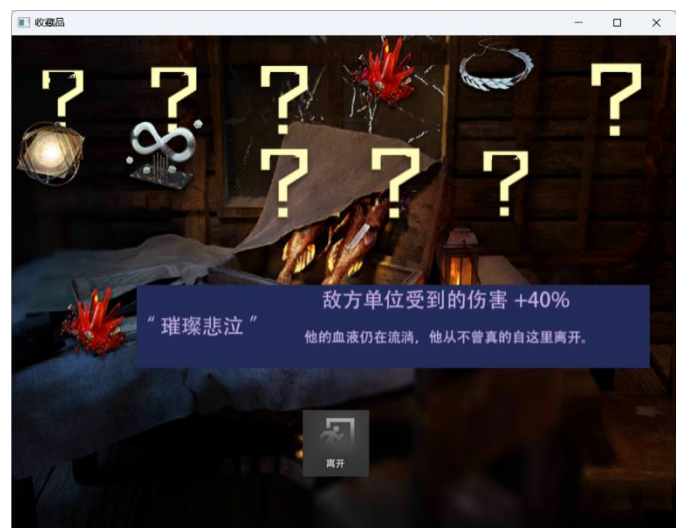


在一场战斗中失败，生命值会降低 5。灯火值会降低 10。当玩家在战斗中失败且生命值降为 0 时，游戏会失败。而随着玩家的灯火值降低，玩家的游戏会越发困难，具体表现在遇到更强的敌人和在突发事件中更容易遇到负面结果。而货币则用于在商店节点中购买道具。

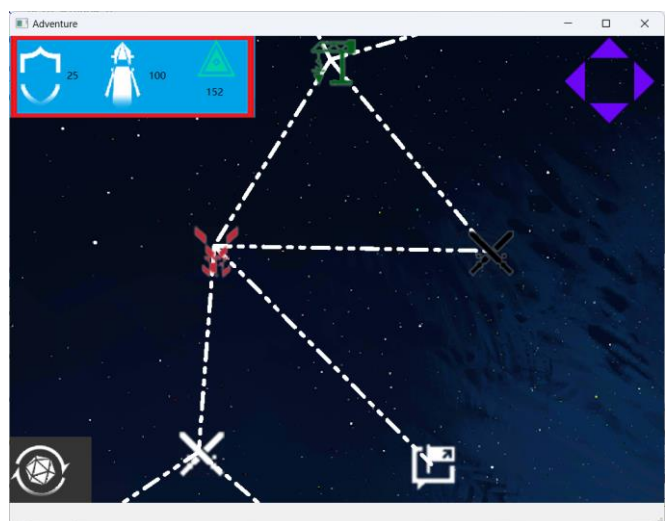
界面的右上角是控制地图视野的按钮，点击对应的按钮会将地图视野移动到不同的范围。



界面的左下角是道具列表，点击之后可以查看获得的各种道具，并且能查看各类道具信息



地图的各节点的事件都会有对应的进入按钮和退出按钮。进入意味着玩家在地图上从上一个节点走进这一节点地图上这一节点会变为红色，而走过的节点则变为绿色。值得一提的是，玩家只能从一个节点前进到特定的节点，即白色标注的节点，不能回头走之前未选择的支路，这些支路会用黑色标注。而更远处的节点则用问号覆盖，玩家无从得知其节点类型。



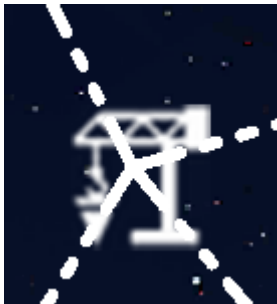
各节点介绍：

事件节点：玩家在这一节点会遇到或者正面或者负面的事件，表现为获得货币，生命值，道具或者失去生命值等。某些事件节点的有多种动作结果，概率受灯火值影响。



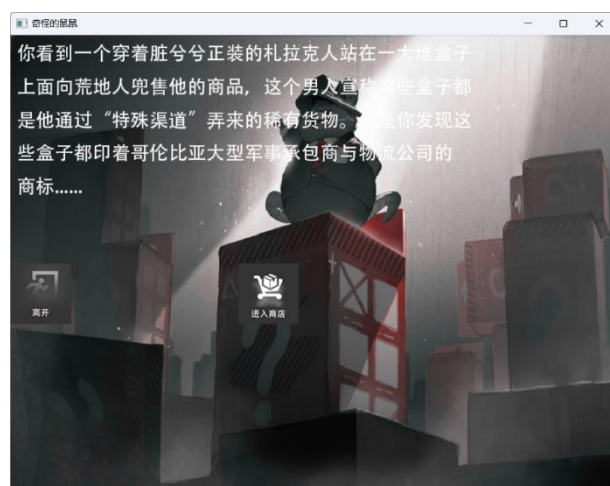
宝藏节点：

随机获得一件道具。

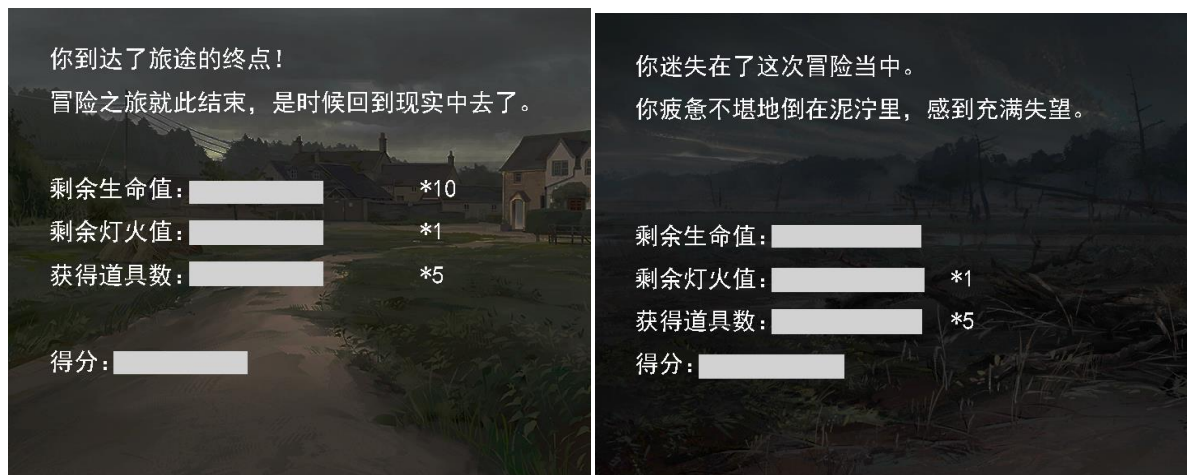
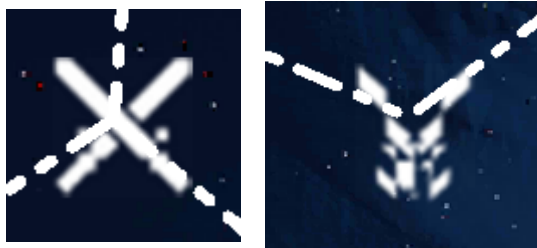


商店节点：

玩家可以购买道具物品。



战斗或敌袭（困难战斗节点），玩家会在这里遭遇一场战斗。



当玩家到达地图上的另一端点（终点）时，游戏会获得胜利，并跳转结算界面。玩家获得的道具，剩余的生命值，灯火值都会转换成最终结算的分数。

如果玩家在战斗中失败，失去了所有生命值，则游戏失败，跳转至失败界面。

各模块和类设计细节和分工：

1. `mainwindow` 类：是游戏的主窗口，`ui` 内部是一个 `stackedwidgets` 类，内部包含了一个主界面 `start_menu`，一个帮助界面 `help`，一个剧情界面 `game_story`，一个游戏地图界面 `map`（最主要的界面），一个胜利结算界面 `victory`，一个失败结算界面 `lose`。各个界面有对应的信号函数，主界面在收到各界面传来的信号（如返回按钮、继续按钮等的按动）时，通过 `setCurrentIndex` 函数切换界面。
该部分代码量约 80 行主要由廖致远负责完成。
2. `game_story_window` 类：是游戏的前情提要剧情窗口。界面通过一个成员 `index` 控制当前的剧情页面标号，并通过 `update` 函数调用 `paintevent` 函数重新绘制背景实现剧情的翻页。类中包含了一个继续按钮和跳过剧情按钮，通过控制 `index` 变化，在 `index` 到最大值时发出继续信号，使得主窗口翻页。
该部分代码量约 80 行，主要由廖致远负责完成。
3. `help_window` 类，游戏的帮助界面窗口，内部实现类似于 `game_story_window` 类，不同的是，每次进入的时候都会把 `index` 置初值，方便重复观看。
该部分代码量约 40 行，主要由廖致远负责完成。
4. `get_start` 类，游戏主界面，较为简单，除去背景绘制之外添加了三个按钮，分别发出信号让主窗口跳转到游戏，帮助界面以及退出游戏。
该部分代码量约 20 行，主要由廖致远负责完成。
5. `m_Button` 文件：游戏中涉及到的各种按钮和按钮点击之后跳出的窗口设计。游戏程序主要核心之一，代码量约 1200 行，由三人分工完成：

m_button 类：重载 Qpushbutton，通过 setCursor, setFixsize, setIcon, setMask 等各种函数并导入素材实现了形状各异的按钮。同时通过重载 press_event 函数和 release 函数实现按钮的按动效果。是整个程序按钮类设计的基础。代码量约 40 行，主要由廖致远负责完成。

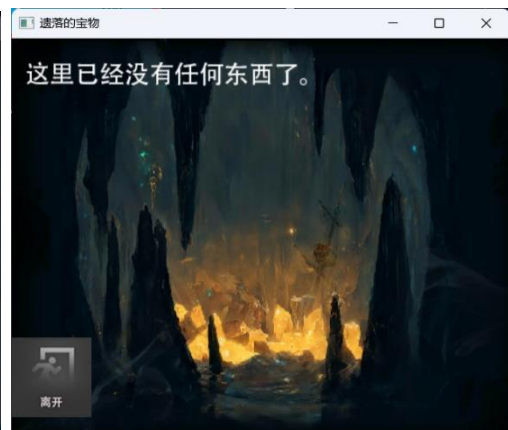
m_board 类：m_button 类不可按动的变种，主要用于实现图片块的导入。代码量约 40 行，主要由廖致远负责完成。

collection_symbol 类：道具列表按钮。内部成员存储了道具数量和道具列表。道具列表由二进制串表示，第 i 位是否为 1 表示当前是否由该道具，从而决定了相应道具按钮可否按动，是否加载道具图片。内部的窗口包含所有的道具对应的 m_button 按钮和两个 m_board 说明板，用于说明道具的属性。每当按动上方的道具按钮，下面就会通过接收信号调用相应的 λ 表达式实时改变说明版的图标。由于该部分有大量重复工作：每个按钮都要单独设置，因此代码量较高，约 300 行，主要由廖致远负责完成

my_point 类：由 m_button 类派生而来，表示地图节点，内部由 explored 成员代表是否探索过，can_be_chosen 成员代表是否能够选择前进，connected_point[6] 表示与之相连的所有节点，forward_point[6] 代表它下一步能到达的节点，forward_num 表示下一步能到达的节点数目 connected_num 与之相连的节点数目，located_image_path，当前节点状态下的贴图路径，explored_image_path 已探索状态下的贴图路径，forbid_image_path 无法选择状态下的贴图路径。其中相连节点列表，下一步节点列表和可否选择成员都会在 game_map 类中随游戏进行实时更新。主要通过 change_can_chosen close_can_chosen add_forward_point connected_to 四个函数更新。而 change_located 和 restore 则负责更改贴图状态以及可否选择的状态。代码量约 50 行，主要由廖致远负责完成。

Start_point 类：探索开始节点。通过信号槽机制将点击信号和自身的 clicked_action 函数相链接，达到点击之后跳出窗口的效果。代码量约 50 行，主要由廖致远负责完成

Treasure_point 类：第一个正式探索中出现的节点。该节点的成员中包含了一个道具序号栏。在构造函数中出了退出按钮之外加入了另一个按钮：step_forward 按钮，通过这个按钮实现前进到这一节点，改变周围的节点可否选择的状态，同时让玩家获得这个节点对应的道具（在 game_map 类中道具列表对应位置发生异或）。同时，点击这个按钮将会改变 click_action 中子的窗口状态，表现在出现新的按钮和贴图上，这些操作均通过信号槽机制实现。此外，在通过这个节点时，该节点的 explored 成员属性将会改变，导致再次点击时窗口不再包含 step_forward 按钮，玩家不能重复探索。代码量约 130 行，主要由廖致远负责完成。



war_point 类和 aboutFight 文件：存储了有关战斗的窗口设计，逻辑设计和按钮设计。代码量约 800 行，主要由郭翔宇负责完成。（旁注：story_point 类也是一个特殊的战斗节点，由于前期原本打算设计成剧情线，后更改了计划）

aboutFight 实现的类如下：

stateIcon 类：显示战斗属性的小图标（物攻、法攻、物抗、法抗等），继承自 QLabel，abstractFighter 类中有其对象。

HPLine 类：顾名思义显示血条，继承自 QGraphicsView，abstractFighter 类中有其对象。

abstractFighter 类：是 fighter 类和怪物类（目前有 juniorDemon 和 eliteDemon）的抽象基类，它包含了对战时的属性（物攻、法攻、物抗、法抗等）以及显示这些属性所需的图标（对应的 stateIcon 和 HPLine）；在调用 abstractFighter 的 showAll 函数时，所有图标等 会一并刷新显示。

skillUnit 类：每一个 skillUnit 是一个技能单元，分为 13 种，有增加物攻法强，增加行动点，减少对面属性等；重写了 mousePressEvent 函数，在鼠标点击时 skillUnit 会移动到相应位置。

skillGroup 类：是一个技能组，内部包含一个 skillUnit 数组 group，技能组的等级 level 是其能包含的技能单元数；generate 函数：接受一个等级（整数），随机生成四个相应等级的技能（skillUnit）供玩家选择，并将玩家选择的那个技能添加到 group 中；skillEffect 函数：将此技能组作用于双方产生效果，伤害计算增益减益等效果在此实现。

fightWidget 类：QStackedWidget 的子类，战斗实现的窗口；内部通过 left,right 两个数组储存对战双方；oneFight 函数（此函数实现了具体战斗过程）：进入战斗后，首先根据灯火值等信息生成怪物（怪物数据在一定程度上随机，分为普通怪和精英怪），然后计算道具效果，最后进入主角的回合函数 oneTurn，主角回合结束后会发出信号，此时进入其他对象的回合函数；回合函数：首先生成对象的技能组（主角通过调用 skillGroup 的 generate 函数生成技能组，怪物的技能组在设计时确定），接着调用技能组的 skillEffect 函数对双方造成影响，最后回合结束。

Accident_point 类和 an_action an_incident action_button 类：存储了有关事件节点的窗口设计，逻辑设计和按钮设计：该节点的成员中包含了一个道具序号栏。在构造函数中除了退出按钮之外还有另一种按钮：action_button 按钮，代表人物应对事件的不同动作。每个 action_button 与对应的 an_action 对象绑定。后者存储以下信息：按钮图像的编号，所执行的动作（加血，扣血，加光，扣光，获得货币等），以及选择该动作前后的说明文字。所有的 an_action 都存储在 game_map 类下的一个向量（vector）actions 内。每个 accident_point 类含有 1~3 个 action_button，用 step_forward_buttons[] 存储，此外还有对应的 Q_label 形式的介绍事件描述，动作描述和动作结果的标签 about_point 和 desc[]。Descs 和 step_forward_button 中的元素一一对应。点击一个 action_button 后，原本描述事件的 about_point 会变为对所选动作的描述，而描述所选选项动作及结果的 desc 会转而描述所选动作的结果。（一个动作有可能引发多个结果，视概率而定）。Action_button 按钮与描述其他动作的 desc 都将被隐藏，使玩家只能选择一个动作。此外，走到一个事件节点时，将随机选取一个事件发生。事件的相关信息用 an_incident 类存储，存储信息为事件的描述文字及各选项在 actions 内的存储序号。所有的事件存在 game_map 下的向量 incidents 内，点击 accident_point 时，若发生的事件未确定，则会从 incidents 中随机抽取一个并保存其编号 typ，使得退出后下次进入时节点事件不变。与 treasure_button 类似，玩家不能

重复探索。该部分含代码约 300 行，实现主要由崔鹤龄负责完成，事件和选项的设计由廖致远负责。

Branch_point 类和 goods 类：存储了有关商店节点的窗口设计，逻辑设计和按钮设计。在动 step_forward 按钮之后，会发出信号让新的 goods 按钮出现，进入商店界面。Goods 按钮是一个特殊的 m_Button，按动之后会出现商品信息界面，说明商品的具体效果和价格以及购买按键，该按钮的 buy_信号会让原窗口货币量减去相应的价格并改变列表。该节点在地图上被按动时，地图会向其传递当前的货币，生命值，道具列表等变量，存储到其相应的成员当中，相应地生成商品列表，在购买途中会改变这些成员的值，并在关闭节点的时候恢复 game_map 相应的指标，从而实现购买的效果。



该部分代码量约 380 行，由廖致远负责完成。

1. game_map 类：游戏的主要页面，成员包括灯火值，生命值，货币值，道具列表，当前节点和 QLine 列表 path。该部分是一个范围很大窗口。大部分节点的位置都在窗口显示范围之外，主要通过四个视野移动按钮和各个节点相链接，通过按钮位置移动来实现地图视野变动。而路径是通过改变 QLine 列表中各个线成员的 xy 值并通过 update 重新绘制背景来实现。地图窗口左上角是相关的指标列表，并通过 QLabel 的 setnum 函数实时更新。内部声明了各个节点，并按照地图上的位置设计移动位置。最后通过 connect 函数和 add_forward 函数实现节点之间的互联和逻辑转换。内部设计了大量信号槽来实现节点窗口和 map 成员的相互交互。该部分是游戏最主要的界面整体约 600 行，地图设计和节点互联由廖致远负责，accident 事件导入由崔鹤龄负责，其余信号槽由各负责的人实现。
2. victory_window 和 lose_window：游戏的失败和胜利结算页面，在 map 类中收到胜利或者失败信号之后跳转，并读取 map 类中各项指标来计算分数。两部分共 60 行代码，由廖致远负责实现。

补充：原本实现了一个属性升级模块（崔鹤龄设计），但是由于时间不够来不及完整实现以及和游戏其他部分衔接，这部分只得放弃，这也是项目时间分配不均衡和前期设计不到位问题的体现。

项目总结与反思：本次项目在最初设想的时候体量很大。并且还设想了很多其他功能，比如小地图，存档，沙盒模式等等，但是开始时间晚（5 月底），前期完成内容太少（截止到 6 月底也只完成了地图基本设计，节点事件一概未动），导致最后只能潦草完工，质量没有想象中好。此外，很多基本的功能也没有得到充分测试的完善，一个最基本的例子就是中途退出

的断点保护，在中间窗口中如果不慎点击关闭，可能会出现道具丢失等隐患。人机界面设计的也不是很友好，游玩过程中的帮助，中途退出和继续游戏都没有完善，导致整体质量略显粗糙。甚至为了节省时间将原本的剧情线节点删去。同时由于缺乏充分的测试，游戏中可能会出现一些隐患和 bug，这些都是这一项目中留下的遗憾。

反思总结教训如下：

1. 有关“到底应该做什么”的事情没有讲清楚。这个作业一开始的设计想法是队长匆忙决定，没有综合各小组成员的意见和讨论，在具体怎么实现上也缺乏明确的、可操作性的提纲。导致大部分时间都是组长在根据自己的主观想法自作主张推进这个项目，其他组员无从下手的情况。
 2. 小组之间缺乏讨论。虽然小组组建了微信群，很多问题也无法实时传递到，队长也没有把自己的设计想法很高效地传递到各个组员方面。比如各个版本的代码由哪些改进，代码的设计思路这块组员之间没有相互了解，出现三个组员之间互相不理解对方代码的情况。
 3. 没有做到明确的分工。由于是第一次接触到多人协作项目，分工没有得到很好的解决。除却各个任务分配不均衡外甚至有出现各个组员重复做某一板块的情况。
 4. 对项目地体量估计不到位。组员普遍反映很多好的想法没有时间实现，通常是想象中觉得比较简单，但遇到实际情况无从下手，或者只能草草低效了事。
- 3.