

程序设计实习大作业报告

项目名称: The Truth

报告小组: 未名四号

小组编号: 132

小组成员: 钱一略 2000017710 qyl@stu.pku.edu.cn

说明: 由于本组为单人小组, 大作业成果受项目复杂度及时间限制的影响较大。经过开发前的重新评估, 原有的程序设计无法在规定时间内完成。因此本项目实际成品与立项时的内容不同。

程序功能介绍

- 本程序模拟游戏《真探》(英文名: The Truth) 进行开发, 开发完成了其主要游戏界面的基本功能, 但仍有部分较为复杂的功能暂未实现。
- 本游戏的目的是让用户对游戏给出的案件线索进行分析和思考, 并从中发现新的案件线索。综合所有线索后完成案件的“侦破”。
- 本游戏在开始时提供给用户的仅有一个线索, 即“案件描述”, 描述案件的基本情况, 例如时间, 地点, 现场情况, 尸体情况等。用户从该线索中找到关键信息进行检索, 从而使游戏提供新的线索。下面是两个例子:
 - 线索中有死者姓名, 则死者姓名很可能为一个新的线索;
 - 线索中有死者现场发现的重要物件, 例如凶器、脚印、证件等, 则这些物件很可能为一个新的线索。
- 在所有线索中, 有可能出现人名、地名、企业机构名、重要物品、重要事件等。
- 用户在找齐所有线索后, 可以对案件进行分析破案。

项目各模块与类设计细节

- 主窗口 `detective`: 由标题(`QLabel`), 线索区域(`ClueArea`), 调查区域(`InvestArea`)组成。
- 程序中的后端类: `Clue` 类, 用于存储和处理后端线索信息。
- 程序中的第三方类(库): 使用了 `nlohmann::json` 来读取和存储线索信息。

线索区域

- 线索搜索框: `QLineEdit` 的实例。
- 线索列表 `ClueList`: 继承自 `QScrollArea`, 作为线索按钮的承载容器。
 - 线索项 `ClueItem`, 继承自 `QPushButton`, 每个实例代表一个可交互的线索按钮。
 - 线索未找到时按钮不可见、不可用。
 - 线索找到后, 当线索未读时, 按钮显示黄色, 线索已读时显示蓝色, 线索选中时显示更深的蓝色。按钮上的文本为线索名称。
 - 接收自身的 `clicked` 信号处理上述事件。
 - 接收线索搜索框按下回车的信号, 检查用户输入的线索是否能被找到, 若找到, 则将该线索对应的 `ClueItem` 设为可见, 并发送 `clueFound` 信号。

调查区域

- 基本信息区域 `BasicInvestArea`：继承自 `QFrame`，用于显示线索的基本调查结果。
 - 标签 `label`：用于显示标题。
 - 内容 `BIAcontext`：`QTextBrowser` 类，用于显示线索内容。
 - 接收 `ClueItem` 的 `clicked` 信号，调整内容。
- 询问笔录区域 `AskInvestArea`：继承自 `QStackedWidget`，对选中的不同线索显示不同的内容。对于没有询问笔录的线索，显示一个空的 `widget`，对于有询问笔录的线索，显示一个询问笔录类的实例。
 - 询问笔录类 `PoliceAsk`：继承自 `QTabWidget`，对于该线索中做过询问的每个人，用一个 `QTextBrowser` 实例显示笔录内容。
 - 当某个人作为一条线索还未被找到时，询问笔录中关于此人的页面将不可见，直到此人作为一条线索被用户找到。
 - 接受线索列表的 `clueFound` 信号，枚举所有的页，检查刚发现的线索是否是人名，且在此线索中做过询问。若有此种情况，将此页设为可见。
- 额外信息区域 `InfoArea`：用于显示线索的额外信息，如图片、声音等。暂未支持非文本信息。

其他类

- 线索类 `Clue`：初始化时，程序读取案件对应的 `json` 文件，并将其构造成多个线索类的实例。类提供访问线索类型、名称、内容、询问内容等的接口，使得在前端编码时不需要对 `json` 文件或 `json` 变量做处理。

设计细节

- 对不同显示分辨率及缩放的兼容性：设计了对显示器及系统缩放比例的检测，保证在显示像素不过低（不小于1080p）的情况下，在不同的显示器上窗口均占据83%的宽度。
- 整个窗口的设计通过代码完成，不依赖于UI文件，兼容性更好。多个线索按钮的位置自动生成而非手动指定，因此对不同数量的线索有较好的泛化能力。

小组分工情况

钱一略完成全部工作。

项目总结与反思

项目局限性

- 相比于原游戏《真探》，项目仍有多个功能没有实现：
 1. 未实现线索内的搜索：原游戏中，除了搜索线索，也可以设置是否搜索已有线索的文本内容；
 2. 未实现进度保存：由于游戏进度在本程序中有较大部分通过窗口的状态（是否可见等）体现，游戏进度的保存功能较为困难。
 3. 未实现线索交互：原游戏中有若干种类型的可交互线索，例如通过输入日期获取的监控录像、通过将线索A（人名）拖动到线索B（物品）的交互框进行指纹匹配等。这些交互内容开发难度大、特殊性强，需要对每种可交互的线索开发其交互逻辑。受限于开发时间，线索交互暂未完成。
 4. 未实现提示系统：提示系统需要根据目前用户的探案进度，给出提示，例如某个线索中仍有未发现的信息，某个可交互线索已经可交互等。提示算法的开发较为复杂。
- 在开发和测试过程中发现的问题：
 1. 控件样式目前以明文形式写在代码文件中，若以 `.qss` 文件的形式保存并读取，代码会更为美观；

2. 程序对高分屏的适配导致文字显示存在锯齿，暂未有好的解决办法。原游戏仅支持1080p的分辨率。

未来开发方向

1. 进度保存：记录用户操作日志文件，用恢复日志文件的方式进行线索保存；
2. 尝试进行可交互线索的开发；
3. 游戏欢迎界面的开发；
4. 游戏提示系统的开发；
5. 提供更丰富的案件；