

魔兽可视化 项目报告

一. 程序功能介绍 概述:

本项目旨在帮助选程设的后人更好地 debug 魔兽大作业。通过实现文本对比功能和对魔兽世界的图形可视化，提供标准样例的世界变化过程，让用户可以看到程序运行后整个世界的实际变化过程，用于比对错误。

功能描述:

1. 文本对比器 （用于对比用户输出和标准输出）
2. 世界实体可视化：
 - 实现基地、城市的图形可视化
 - 基地、城市实现悬浮弹窗预览，以及点击

查看详情功能（出现弹窗），以查看该位置武士和对应属性（如城市 flag 状态 基地生命值等）

- 武士实现悬浮弹窗预览，以及点击查看详情功能（出现弹窗），以查看该武士武器和属性（hp, strength, base 等）
- 武器实现悬浮弹窗查看属性

3. 世界运行可视化

- 实现时间操控功能，允许查看不同时间世界状态（前进、回溯），以进行更好的 debug

4. 文字注意事项

- 传授正确的 output 办法
- 总结前人经验，把对题意的常见错误理解汇总，以帮助更好地阐明题干意思

二. 项目各模块与类设计细节

a) 按钮类：

i. Mybutton 类

1. 一个 QWidget 指针
2. 一个 layout 来控制弹窗布局
3. 一个 `vector<QLabel*>` 来储存弹窗中的信

息

4. 在构造函数中初始化了一个 `ToolTip` 的 `QWidget` 的窗口分配给该指针
 5. 并将 `layout` 初始化指向一个 `HBoxLayout` (垂直布局), 并分配给弹窗
 6. **重载了基类的 `enterEvent` 和 `leaveEvent` 函数**, 通过判断光标是否进入\离开按钮全局坐标范围来显示\隐藏悬浮弹窗, 再 `leaveEvent` 函数中, 在鼠标离开时通过 **`QTimer` 中的 `singleShot` 函数**设置了一个短暂延迟以判断光标是否移开到了弹窗的范围中, 确保在光标移开按钮但未移入弹窗的时候再隐藏弹窗。同时, 通过重载基类的**事件过滤器**并安装再悬浮弹窗上, 实现光标移出悬浮弹窗自动关闭弹窗的功能
- ii. `Mycitybutton` 和 `Mybasebutton` 类由 `Mybutton` 类派生而来, 并进一步实现点击弹出弹窗功能

b) 窗口类:

- i. 世界可视化界面窗口类 (`test` 类):

1. 采用了根据样例城市数动态初始化界面的方法，能够自适应不同城市数目的世界均匀排版
2. 使用两个时间操控按钮，通过槽函数链接到界面重载函数来显示不同时间世界状态，实现时间回溯前进功能

c) 文本对比器:

利用文件读取操作来实现，将两份文本分别读入两个文件中，清空 QTextEdit 后再输出，不一样的地方分别用红色、蓝色标注

d) 信息存储类

- i. GameState 类
包含一个样例中一个时间点所有的世界实体状态
- ii. Caseinfo 类
包含一个样例中所有的全局变量和这个类中所有时间的 GameState 状态
- iii. Allcases 类
包含整个程序输出和该组测试输入所有样例所有时间的状态

三. 小组成员分工

先鼎巍：世界可视化页面实现、信息储存类、Mybutton 类实现

芮思铭： 程序整体逻辑框架实现、主界面、注意事项功能、输入功能的前后端实现；文本对比界面的前端实现；可视化世界城市和武士弹窗的前后端实现。

杨丰岭： 文本对比功能实现、项目图片素材收集

四. 项目总结与反思

选题新颖，主题明确，项目整体有比较强的实用性。类设计和前后端关联上思路比较清晰，分工明确、队员互帮互助，较好地利用了团队开发的优势。但由于时间原因，也存在一些尚未实现的功能。