

Qt作业报告

Haiweng Xu

2024.6.28

目录

1	综述	1
2	程序功能介绍	2
2.1	菜单	2
2.2	主界面	2
2.3	个性化设置功能	3
2.4	地图	3
2.5	说明	3
3	项目各模块与类设计细节	4
3.1	主界面模块	4
3.2	地图模块PKUMap	4
3.3	事件类Event	5
3.4	设置模块Config	5
3.5	选择模块Selection	5
3.6	倒计时模块CountDownTimer	5
3.7	文件读写模块FileIO	5
3.8	爬虫模块	5
4	小组成员分工情况	5
5	项目总结与反思	5

1 综述

- 我们的项目的目标是实现一个能够满足北大学生尤其是新生个性化需求的日程管理应用，希望借此帮助用户提高效率，关键是尝试走出舒适区，尝试未尝试过的事情，活出多彩的人生。
- 与通常的日程管理软件不同，我们的项目的突出特点在其个性化的设计，希望项目能够帮助有缺乏规划意识的同学提高效率，同时能够降低走出舒适区所需要的各类成本

2 程序功能介绍

2.1 菜单

- 在菜单界面点击启动按钮可以进入主界面

2.2 主界面

导入课表功能

- 点击导入课表按钮，可以从文件资源管理器中打开从选课网或信息门户上下载的xls/xlsx格式课表，作为之后日程自动推荐的基础。
- 导入的课表会被自动保存到本地，用户重新打开应用时无需重新导入课表。

日程自动推荐功能

- 双击日历中未被点击且本地没有相应数据的一天时，程序将会根据用户的个性化设置生成一个当天的基本日程规划，并且展示在日程表中，基本日程包含就餐和自习场所的信息。
- 当双击的一天已被规划或本地已经有数据时，右侧的日程表将会自动回到那天已有的日程规划，用户可以点击重新安排按钮对当天进行的日程进行重新安排。

个性化事件推荐功能

- 点击个性化推荐按钮，程序将会自动生成3个推荐事件，并展示在弹出窗口中，用户可以点击其中的任意事件进行添加或重开按钮进行重新生成。

日程本地保存功能

- 选定日历中的某一天时，我们可以点击保存到本地按钮，将当天的日程记录到本地，关闭程序重新打开后再次双击该天可以将日程切换为当天的日程。

日程表交互

- 右键日程表任意一个位置会弹出包含删除，添加，取消，修改等四个按钮：点击删除会删除当前行；点击添加会弹出一个设置事件信息的窗口，设置完成后可以添加事件，如果添加的事件存在时间冲突会进行弹窗警告；点击修改则可以按照当前单元格的类别对当前单元格内容进行修改。
- 日程表下方有一个日程倒计时器，将会显示当前到日程表中的下一个日程的剩余事件，为了提醒用户及时通勤，将会在倒计时结束10分钟前响铃并弹窗提醒用户。

讲座信息获取功能

- 选定日历中的一天时，点击讲座信息将会利用爬虫从北京大学讲座网中获取当天的讲座信息，并显示出来，帮助用户获取讲座信息，拓宽自己的眼界。

手动倒计时功能

- 为了在内卷模式下提高用户的内卷效率，用户可以操作主界面左下方的倒计时器，手动设计一个倒计时，到时提醒用户。

2.3 个性化设置功能

- 点击个性化设置按钮将会弹出一个个性化设置界面，主要的各项功能如下：

模式设置

- 用户可以选择合适的推荐模式：
 1. 探索模式(默认模式)：在这个模式下，在日程自动推荐功能中，我们会采用智能的方式为用户规划自习和就餐的时间和地点，当我们勾选记录并使用日程记录改进推荐选项时，程序将会优先推荐用户前往那些前往次数较少的就餐和自习场所，当不勾选该选项时，程序将会随机进行推荐。
 2. 内卷模式：在这个模式下，在日程自动推荐功能中，程序会采用最短路径的原则为用户推荐自习和就餐地点，并且会将上午，下午，晚上没有课程的时间全部排满，提高用户的内卷效率。
 3. 摆烂模式：在这个模式下，用户可以获得酣畅淋漓的摆烂体验。

偏好设置

- 用户可以勾选或者取消日程自动推荐功能中是否自动推荐早餐，中餐，晚餐的选项。

杂项

1. 用户可以勾选明确的答复选项，关闭用户添加事件发生时间冲突时的弹窗提醒。
2. 用户可以修改宿舍位置，作为我们地图中路径的起始点和终止点。
3. 用户可以点击倒计时设置中的修改图片按钮，修改倒计时弹窗的图片。
4. 用户可以点击清空历史记录按钮，清空本地的课表，日程和访问记录等数据。

2.4 地图

- 点击主界面上方菜单栏中的地图项可以切换主界面至地图界面。
- 在地图界面中可以使用界面中的按钮上一步，下一步，显示全部，隐藏全部在地图上分步或一次性显示路径

2.5 说明

- 点击主界面上方菜单栏中的说明项会自动在浏览器中打开本项目的Github仓库，在这里用户可以查看README文档获取应用使用说明。

3 项目各模块与类设计细节

3.1 主界面模块

- 主界面模块主要包含日历，日程表(包含倒计时)，课表导入三部分组成，由于彼此相互关联程度高，因此没有对它们做进一步的细分，我们可以分别介绍其相关设计细节。

日历

- 当我们点击日历上任意一天时，会触发...

日程表

- 为了日程表高阶交互操作的构建方便，我们构建了一些基础接口例如AddRow，AddCol，AddEvent，DeleteEvent，在之后更为复杂的交互部分实现更加复杂的抽象。
- 我们右键日程表中任意处时，会触发。。。接口，并弹出。。。选项

课表导入

3.2 地图模块PKUMap

- 为了地图模块更高阶的交互的构建方便，我们首先定义了ShowPath，HidePath作为底层接口，它们接收一个整型参数idx表示我们的边的序号，分别用来显示/隐藏第idx条边。
- 地图模块中包含一个QGraphicsScene*指针，指向我们以燕园地图为像素图的一个QGraphicsScene对象，我们之后显示路径，添加滚轮功能等操作都是添加在这个QGraphicsScene上的。
- 设计了一个WheelEventFilter封闭类，过滤掉除鼠标滚轮外的信号，用于实现鼠标滚轮对图像的缩放，提升用户体验。
- 设计了一个ArrowLine封闭类，包含一个paint接口，用于绘制其在ui中可以表现为一个红色的箭头。
- 地图模块设置了四个按钮，每个按钮对应着一个槽函数，分别为Prev，Next，ShowAll，HideAll，用于逐步或者一次性显示/隐藏路径。其中路径会被保存在一个ArrowLine*类型的数组edges内。
- 在地图模块中我们设计了Update接口用来维护地图界面右下角显示的信息，告诉用户当天的行程的总边数，以及目前展示的线段数。

3.3 事件类Event

3.4 设置模块Config

3.5 选择模块Selection

3.6 倒计时模块CountDownTimer

3.7 文件读写模块FileIO

3.8 爬虫模块

4 小组成员分工情况

- 徐海翁：事件类Event，选择模块Selection，地图模块PKUMap，设置模块Config，文件读写模块FileIO等模块的设计和实现，主界面中日程自动推荐机制，手动个性化推荐功能的设计和实现，整合Python爬虫模块到Qt项目中。
- 黄源森：主界面中日历和日程表高级交互的设计和实现，事件添加的通用模板设计，课表导入功能的设计和实现，倒计时CountDownTimer模块的设计和实现，利用Python和Selenium库模拟用户实际操作实现网络爬虫功能
- 谢宇翔：日程倒计时机制的实现，菜单和主界面的UI美化。

5 项目总结与反思

- 模块设计上：我们对部分模块例如选择模块Selection，地图模块PKUMap，设置模块Config，文件读写模块FileIO进行了较好的封装以及模块化处理，有效地减少了代码地相互依赖性，并按照模块进行了较好的分工合作，使得代码结构相对清晰，且维护起来相对容易。当然由于时间限制以及部分代码的依赖性，在主界面中我们没有将日历和日程表这两个功能复杂且几种的组件进行模块化，导致mainwindow.cpp文件相对冗长，维护性受到些许影响，而自动推荐事件的部分功能由于过程性较强，出现了部分重复性较强的代码。
- 项目创意上：我们的项目聚焦日程安排的个性化，力图让各类用户在不同场景下(例如走出舒适区，内卷，摆烂)都能使用这个程序，因此花了较多精力在个性化的设计上，从用户的角度思考合适的推荐形式，给用户提供了多种选择。但与此同时由于个性化以及用户群体的广泛导致了我们的项目模块类别较多且独立，在部分细节的设计有些粗糙,UI的美化上都有很大改进空间。
- 技术使用上：由于我们项目模块类别较多且独立，在项目的完成中我们接触了非常多的模块与库，处理了各式各样的问题，锻炼了组员的学习能力和代码能力，我们主要使用的相对复杂的技术包括以下：
 1. 利用QAxObject操作excel等表格数据，并处理编码问题转化为Event类对象
 2. 使用QGraphicScene来展示燕园地图，从而支持路径在像素点上的精确显示

3. 使用Python中的Selenium外部库实现爬虫功能，模拟人类使用浏览器时进行的行为在北大讲座网中选择日期获取讲座信息。
 4. 在Qt中利用QProcess类实现运行Python代码，并且通过其标准输出获取运行结果
 5. 我们同样尝试了使用Qt内置的Network模块完成爬虫任务，但是由于我们使用的Qt版本相对较低，支持的OpenSSL版本较低，在配置OpenSSL以实现访问https：类型网页时遇到困难，因此我们采取了前面基于Python库Selenium的方法。
 6. 文件读写模块FileIO中读写了csv和json两种格式的文件，分别为程序预设数据如地点与用户数据两类。
- 沟通协作上：我们的项目使用Github进行代码托管，在项目早期小组成员共同讨论项目设计思路，探索各类设计的可行性，在代码的具体实现阶段，组长安排每一阶段工作任务，组员在小组群内及时沟通交流，分工到个人，沟通高效。但在项目早期由于相关经验缺乏，没有统一Qt的版本，之后花费了部分时间进行调整。