

# 课程资源管理器

## 一、为什么选择本项目

面对繁杂的课程资料，许多人的选择是将他们用各种文件夹分类。但每当取用不同的课程资料时，人们便要为此付出代价——用鼠标双击开一层层文件夹。往往不知不觉间，任务栏中已有六七个待关闭的文件夹。这无疑是一种值得被优化的体验。

## 二、项目特点

那么本项目相对于普通的文件管理器有何特点？

1. **致力于削减操作**，能单击触发则不双击（如展开文件树），能直接获得便不用搜索（如“最近打开”窗口）。
2. **看重实用性与布局紧凑性**，在一个窗口中实现尽可能多的功能，而不是频繁的切换窗口或者让屏幕被各种跳出的窗口淹没。
3. **特殊的文件系统**。开创了“转运机制”，从此下载课件的课件只需全部扔到“中转站”中，随后便可流水线式将各种文件分类至设定的存储文件夹中；同理，此功能也可用于整理杂乱的文件夹。
4. **文件筛选方式自由**，包括名称正则匹配，学科、类型、修改时间与重要程度（可自己设定）筛选。

## 三、具体模块与功能

### 3.1 后端模块（DataManager）

#### 3.1.0 文件结构与转运机制

所有的课程文件均保存在 `glob_dir` 目录下（下称根目录），用户可通过左上角的“设置”按钮修改此变量。同时用户可以随时可以点击主界面左上方的“转运文件按钮”将 `trans_dir` 目录（下称中转目录）中的文件转移到根目录中。

根目录下使用三层树形文件结构，一级子目录与科目同名，二级子目录与类型同名，二级子目录下存放所有的文件。

#### 3.1.1 CourseFile 类

`CourseFile` 类记录课程的科目、类型（标签）、文件名，以及优先级、使用频率、上次使用时间。重载了 `==`，认定两个 `CourseFile` 对象相同当且仅当其科目、类型和文件名都相同。

根据文件系统结构的约定，每个 `CourseFile` 与根目录下一个文件的路径一一对应，简化了操作。

#### 3.1.2 CourseFileManager 类

`CourseFileManager` 类包含一个 `vector<CourseFile>` 类型的成员变量 `courses`，其内部存放根目录下所有文件的信息。整个程序只有唯一一个该类的全局对象 `cfm`。构造时读取程序运行目录下 `course_info.json` 文件中的信息（如果存在），包括，并在析构时向 `course_info.json` 中输出更新后的文件信息。`cfm` 能根据根目录的变化自主增加或删除文件，也能在文件每次被打开时更新相应的频次、时间信息。

为实现文件信息的提取和修改功能，cfm 为前端提供了 filter\_file 和 transfrom\_file 两个方法。前者返回 courses 中所有满足给定条件 CourseFile 对象的复制，不进行 cfm 进行任何修改；后者进一步对 courses 中的满足条件的对象作用一个自定义函数，其筛选函数默认为恒 True，将对 cfm 中的文件信息进行修改，同时所有的对 course 的修改都只能通过此接口进行。

cfm 提供 add\_file 和 erase\_file 方法，分别对应文件的添加和删除功能。两者均接受一个 CourseFile 形参，前者需额外传入一个文件原始路径。如遇到文件被打开，路径不存在等问题，cfm 将不进行任何操作并返回 0。反之若操作成功则返回 1。

cfm 提供 open\_file 方法，允许用户以系统默认应用打开文件。其内部实现为 transfrom\_file。同样失败返回 0，成功返回 1。

### 3.1.3 MyWatcher类

该类只用一个全局对象 mv，用于监控根目录及中转目录的变化。提供 transfer\_files（转运）方法，将中转目录中的文件转移至根目录下（此过程中用户定义文件标签信息），同时检测并根目录并更新 cfm 中的相应信息。

## 3.2 前端模块

### 3.2.0 界面大观

不同组件的实现分布于不同的文件，最后在 MainWindow 中采用 QHBoxLayout 与 QVBoxLayout 进行组合，且可以灵活改变布局。效果如下。





### 3.2.1 文件树(上图区域①)

由 FileTree 类实现，继承自 QTreeWidget 类，能够展现存储库文件夹的文件结构。

**主要功能：**

1. 左键单击文件夹节点（前两层）可将其展开，双击文件节点（最后一层）可打开文件
2. 右键单击各节点会在鼠标处出现菜单选项。若被右键的为文件夹节点，选项为打开文件夹；若被右键的为文件节点，则菜单选项包括删除文件、打开文件。

### 3.2.2 文件筛选列表(上图区域②)

由 FilterWidget 类实现，主体为 QTableWidget 类，给用户提供个性化的筛选与文件信息呈现机制。

**主要功能：**

1. 通过表格上方的筛选栏可进行高级筛选：支持**文件名的正则匹配**；支持学科、类别、优先级、打开时间筛选；支持多种筛选标准**共同作用于结果**；在未筛选时默认显示所有文件的信息。
2. 左键点击表头可使该列文字完全显示/重新部分隐藏。
3. 右键某一单元会在右键处显示菜单，功能包括：修改文件信息，即重新填入该文件该项信息；**新增筛选窗口**(最多同时存在两个)；删除筛选窗口(最少保留一个)。

### 3.2.3 多功能模块(上图区域③)

包括最近打开、备忘便签、常用网址三个功能，由三个组件构成：Recent，NotesWidget 与 Weblist。以一个 QStackedLayout 组织，用主界面下方的后三个按钮切换；在主界面左上方的设置菜单中可选择隐藏整个多功能模块。三个组件的效果如下：



**最近打开：**在列表中按照时间先后顺序展示打开的文件。单击可打开文件。

**备忘便签：**在列表中展示之前存储的便签。点击“添加便签”按钮可添加便签。右键单击便签会显示菜单，菜单选项包括置顶该便签（不影响其他便签顺序）与删除便签。

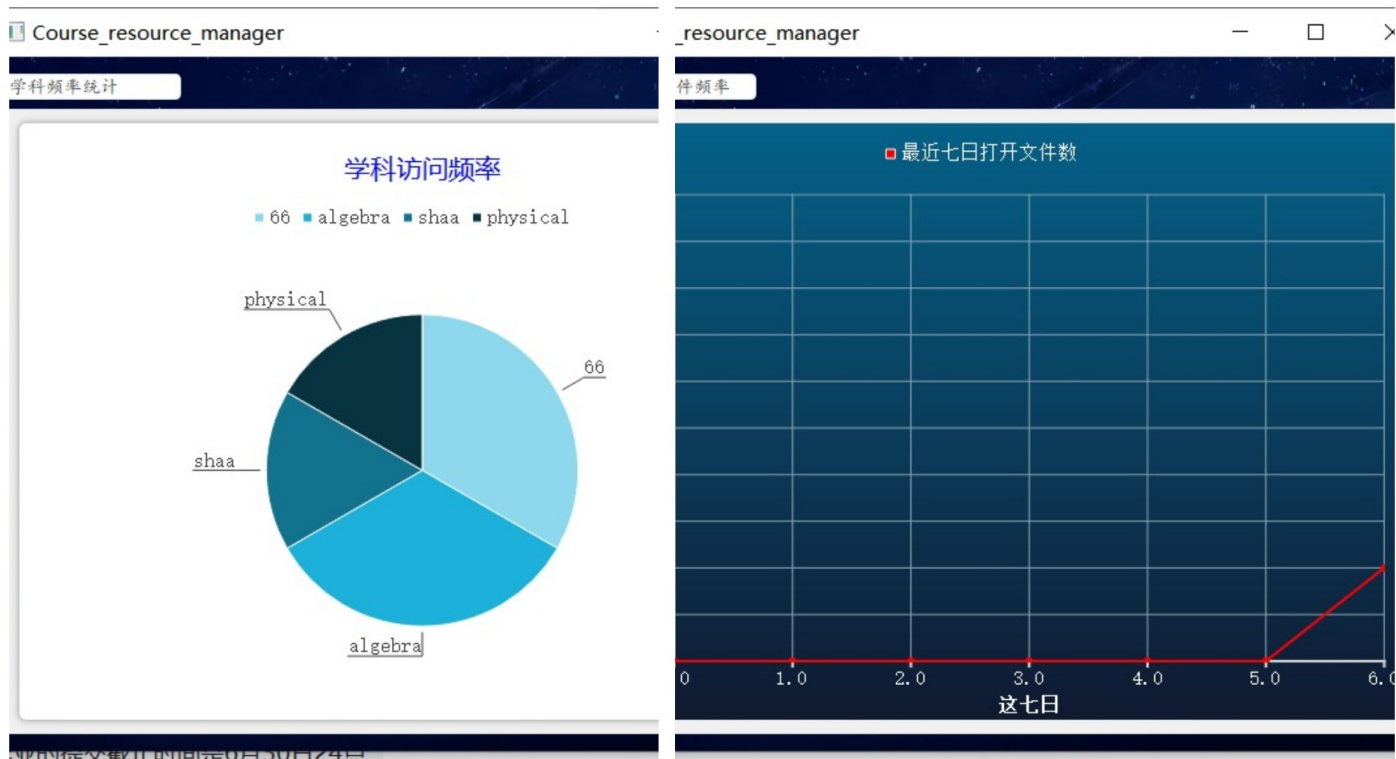
**常用网址：**点击“新建链接”按钮可设置链接的名称与URL；单击链接可跳转至该链接。

### 3.2.4 统计图表

采用 QCharts 绘制，包括不同学科文件打开比例的饼状图与这七天文件打开次数的折线图。

点击主界面下方的“统计图表”按钮即可调出图表窗口。

点击图表窗口左上方的 QComboBox 可切换图表。



### 3.2.5 杂项

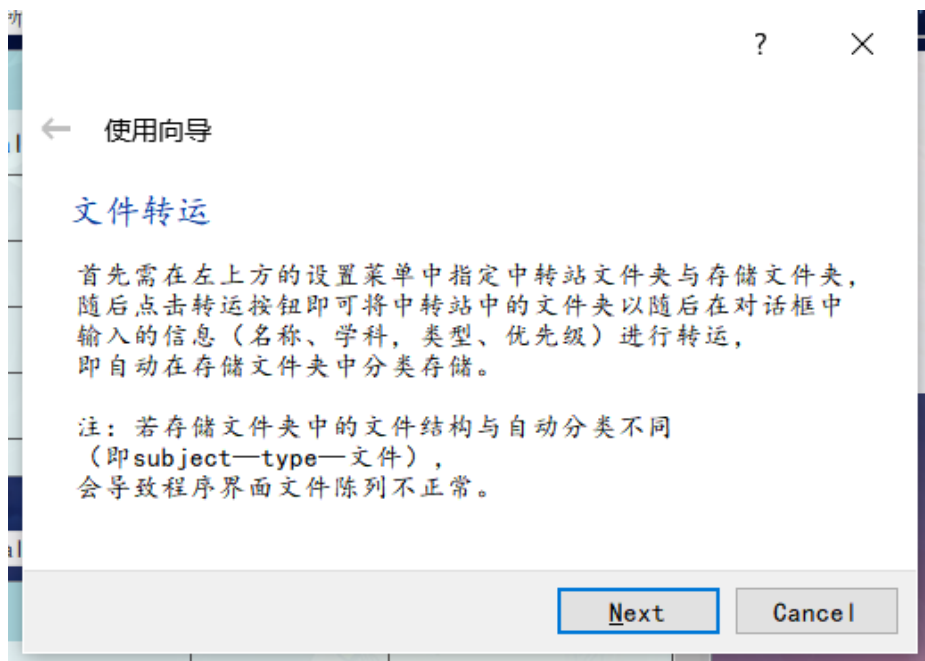
**设置文件信息:**由 `AddFileDialog` 类实现。当点击转运文件按钮后会弹出此窗口。既可手动输入学科等信息，也可下拉选择已有的学科或者类型。还可通过滑动条设置重要程度。

The "设置文件信息" (Set File Information) dialog box contains the following fields and controls:

- name:** A text input field containing "Qt.docx".
- subject:** A dropdown menu with "自定义" (Custom) selected.
- type:** A dropdown menu with "自定义" (Custom) selected.
- priority:** A horizontal slider bar.
- Buttons:** "submit" and "cancel" buttons at the bottom.

**帮助窗口:** 提供帮助信息的窗口，以 `IntroWidget` 实现，主体为 `QWizard`。可在主界面左上方调出。





### 3.3 美化

1. **整体**：设置字体为KaiTi，添加图标，修改文件名称，调整透明度为0.95（视觉感官更好），设置背景图。
2. **QPushButton**：圆角，设置灰色层次边框，调整背景为透明，根据背景分别调整文字颜色，设置按下之后状态变换。**为主要按钮专门写下 MyPushbutton 类制作动画**，并且为了给予更好的体验为按钮设置延时。
3. **FileTree**：修改颜色。
4. **QListWidget**：增加间距，为光标选中与点击设定提醒，修改颜色。
5. **QComboBox**：圆角，增加质感，为部分下拉框隐藏下拉箭头。
6. **QLabel**：修改颜色。
7. **QLineEdit**：圆角。
8. **QTableWidget**：设置颜色，设置选中提醒状态。
9. **QSlider**：修改滑块样式，设置光标选中滑块提示，设置滑动条颜色，设置滑动条已选部分提示。

## 四、分工

后端与部分前端：蒋轩林

主要前端：刘宇政

所有的美化：李锴

## 五、总结与思考

作为目前我们在大学阶段遇到的最大的项目，此次作业给我们带来了很多方面的挑战。首先是设计理念问题，我们最终决定做个实用的东西而不是司空见惯的游戏，我们也计划自己能在之后的生活中使用这个程序。在开发过程中，我们要自主学习Qt这个工具，其中的过程包括搜索博客、查阅文档等，这锻炼了我们的检索能力。如何让合作发挥更好的效果也是需要考虑的问题，为此我们最终选择了前端、后端、美化分离的分工模式，并且采用git进行版本管理，使得开发到后期仍较有效率。当然，若是助教没耐心看到这里就当我说（

