《高等代数》阅读器

程序功能介绍

本程序主要实现了丘砖阅读器的功能。最基础的功能是可以展现所有页面。由于本书页数过多，所以需要特殊设计才能在翻阅时可以把每一页展现出来。另外可以随意缩放，更方便多软件协同。本阅读器针对《高等代数》的原因是，由《高等代数》中有许多例题需要依托于之前的例题和定理，在翻阅的时候十分不方便，所以本程序旨在简化向前“迭代”这一过程。通过快捷键可以搜索对应的例题，定理，并且跳转到书的对应位置。在搜索例题的时候可以弹出多道例题，多个定理，方便不同例题间的比较。同时我们还实现了书签，高光标记例题等功能，通过不同的快捷键可以实现，更加方便复习。

具体介绍：

Ctrl+F：搜索指定定理/命题/例题/引理/定义/推论，此时会弹出一个框，有参考格式，按照参考格式输入即可跳转到对应页；

Ctrl+H：弹出一个框，在此输入指定例题，给例题添加高光标记；

Ctrl+A：给当前页添加书签并添加备注名，在弹出的框中输入备注名；

Ctrl+S：弹出一个框，点击下三角查找已添加的书签备注，并可以翻到指定页码；

Ctrl+N：可以弹出显示相同页码的另一个窗口

程序模块与类内设计

本程序可分为三个部分：mainwindow.h 作为头文件储存所有的类的声明以及全局变量的声明，mainwindow.cpp 实现所有的成员函数，main.cpp 负责预处理和让第一个窗口弹出。

预处理：众所周知《高等代数》的电子版是图片形式并不支持查找文字，为了获得《高等代数》全书各种定理/命题/例题/引理/定义/推论的位置，我们先用了 python 和第三方库 easyocr 来文字识别，将信息筛选、整理成“章.节.小节名字”（名字为类型+序号，如例 3、推论 2）的形式，储存在 quetopage.txt 中，再在 main 函数中读入，存在 QMap<QString,int> mp 中。

本程序一共设计了四个自定义类：

myScene 继承自 QGraphicsScene，QGraphicsScene 通常和 QGraphicsView 一起使用，类似电视节目中的舞台和电视机，可以通过多个 View 来观看 Scene 中的舞台。这里定义 myScene 是为了在 QGraphicsScene 的基础上实现“懒加载”。

```
class myScene: public QGraphicsScene
{
    Q_OBJECT
public:
    myScene();
    void myUpdate(int);
    void updatepage(int);
    void del(int);
    QGraphicsPixmapItem *item[720];
    int cnt[720];
};
```

《高等代数》一共有七百多页，大于 1 个 G，内存不足以同时储存七百页。myScene 只有在有 QGraphicsView 在看这一页时才会加载这一页，当所有 QGraphicsView 都不看这一页时便会将这一页丢掉。当某个 View 的视野范围换页时，会调用 myScene::del() 和 myScene::myUpdate(int)，将原来看的页以及前后两页的“观看数“减一，若观看数为零则本页丢弃，并将新翻到的页以及前后两页的”观看数“加一，若原来观看数是零则加载本页。

```
void myScene::del(int page)
{
    for(int i=qMax(-1,-page+1);i<=1;i++)
    {
//        qDebug()<<"del:"<<i+page<<endl;
        cnt[i+page]--;
        if(cnt[i+page]!=0) continue;
        item[page+i]->setPixmap(QPixmap());
        removeItem(item[page+i]);
    }
}
void myScene::myUpdate(int page)
{
    QDir cur = QDir::currentPath();
    cur.cdUp();
    for(int i=qMax(-1,-page+1);i<=1;i++)
    {
//        qDebug()<<"add:"<<i+page<<endl;
        cnt[i+page]++;
        if(cnt[i+page]!=1) continue;
        QPixmap pixmap(QString(cur.path()+"/NewTry/gddsImage/%1.png").arg(page+i-1));
        pixmap = pixmap.scaled(Size.width(),Size.height(),Qt::KeepAspectRatioByExpanding,
        item[page+i]->setPixmap(pixmap);

        addItem(item[page+i]);
        item[page+i]->setPos(0,Size.height()*(page+i-1));
    }
}
```

imageView 则是继承自 QGraphicsView，是实现大多数功能的类，也是阅读器的主体部分。

```
class imageViewer : public QGraphicsView
{
    Q_OBJECT

public:
    imageViewer(myScene *scene,TAG *Tag,QGraphicsView *parent = nullptr);
    ~imageViewer();
    void resizeEvent(QResizeEvent *event) override;
    void closeEvent(QCloseEvent *event) override;
    myScene *scene=nullptr;
    void turnto(int i);
    void turnto1(int y);
    int currentPage;
    bool scrollblock = false;
    QSpinBox *spinbox;
    TAG *tag;
    imageViewer *sons[200];
    int cntSon = 0;
protected:
    void scrollContentsBy(int dx,int dy) override;
public slots:
    void onFind();
    void highlight();
    void addtag();
    void searchtag();
    void copy();
};
```

resizeEvent：View 自带函数的重写，当窗口放大缩小时，书页也随之放大缩小
turnto & turnto1：翻页或翻到某个坐标
scrollContensBy：View 自带函数的重写，在滑动时实时更新 currentPage，并在翻页时配合 myScene 懒加载。
onFind：Ctrl+F 触发，弹出对话框并跳转到指定定理/命题/例题/引理/定义/推论，位置信息已储存在预处理 QMap<QString,int> mp 中。
highlight：Ctrl+H 触发，给指定例题加高光
addtag：Ctrl+A 触发，给当前页加书签并添加备注
searchtag：Ctrl+S 触发，查找备注并翻到指定书签页
copy：Ctrl+N 触发，复制当前 View

TAG 继承自 QDialog，主要是为了实现一个支持下拉选择已添加的备注的对话框，并且随着用户操作实时更新选项，以配合上面提到的 searchtag 函数。

```
class TAG:public QDialog
{
public:
    TAG();
    QComboBox *comboBox;
    QStringList beizhu;
    QMap<QString,int> mp;
    int cnt=0;
    void addtag(QString,int);
    int searchtag();
    QString getSelectedItem() const { return comboBox->currentText(); }
    QString showComboBoxDialog();
};
```

FluoreRectItem 继承自 QGraphicsRectItem 是为了配合高亮功能的一块荧光黄的半透明矩形，在类内提前设定好 QPen 和 QBrush 以方便使用，只需设定好大小和位置，再添加到 Scene 中就可以显示一块荧光黄的半透明矩形，也就是高光。

```
class FluorescentRectItem : public QGraphicsRectItem {
public:
    FluorescentRectItem(const QRectF &rect,QGraphicsItem *parent = nullptr) :
        QGraphicsRectItem(rect,parent) {
        QPen pen;
        pen.setColor(Qt::yellow);
        pen.setWidth(3); // 边框宽度
        pen.setCosmetic(true); // 确保边框在缩放时保持一致
        setPen(pen);

        QColor c = Qt::yellow;
        c.setAlpha(120);
        QBrush brush(c);
        brush.setStyle(Qt::SolidPattern);
        setBrush(brush);
        setZValue(1);
    }
};
```

小组分工

董一凡：设计 imageView 类及对应功能，将所有类进行统合

梁僖叡：进行文字处理，筛选信息，设计快捷键，其他类及对应功能
首雨欣：设计 myScene 类及对应功能，整理预处理信息

项目总结与反思

项目成果大体完成了之前的规划，基本实现了一个阅读器应具备的功能。但仍有几点不足：美工方面相对简陋，预处理占据时间过多，在多台电脑上协作不协调等。