

浙江大学



Object-Oriented Programming Project 中期报告

潘子曰	3180105354
艾琪	3180105950
张雯琪	3180103770
吴沁琤	3180103776

2019~2020 春夏学期 2020 年 04 月 11 日

1. 项目介绍

我们常常需要做一些基础的文字编辑工作。然而pc电脑上自带的txt文档编辑器功能繁杂，界面粗糙，在文字排版上非常难以使用。为此，我们希望开发一款基础的文字编辑器，其只需要完成基本的文字编辑工作，并且Tab Policy改成4个空格而非简单粗暴的10个字距。

总而言之，我们希望开发一款极轻量级，只包含文字编辑的核心功能，并且符合程序员习惯的文字编辑器。



普通的NotePad

2. 项目使用的IDE介绍

1.1 IDE介绍

我们主要使用的IDE是Qt Creator，它是一款跨平台的集成开发环境，特别针对Qt开发者，是QtSDK组成的一部分。它包括一个可视化调试工具和集成的GUI版面和外形设计师，功能包括C++代码编辑器、文件项目操作、调试编译，以及相比其他IDE而言Qt Creator特有的Qt相关构建工具等。在Windows，默认安装它可以使用MinGW或MSVC。本项目采用MinGW编译运行。



Qt 框架

1.2 Compiler介绍

在Windows安装Qt Creator默认使用MinGW。MinGW精简的C/C++开源编译器。它是一些头文件和端口库的集合，无需第三方动态链接库就能使用GCC产生Windows程序。MinGW的特点是轻便，无需链接其他第三方库，与Qt库可以完美兼容使用。



MinGW编译器

1.3 库介绍

本项目主要使用Qt 5.14.1，这一系列的C++库被广泛地应用于开发各类跨平台GUI程序，在各类注重运行效率的情景中使用十分常见。它最大的特点是支持平台的原生界面，同时又可以使用诸如QML这类标记语言实现复杂界面，功能强大。诸如奔驰的汽车中控界面、LG的高端智能电视系统就是通过Qt实现的。



Qt开发的应用程序

Qt的运行需要C++编译器支持。市面上大多数C++编译器都支持对Qt的配置。而与C++少许不同的是Qt对许多C++标准类进行了再封装和设计，针对平台界面开发专门设计了API接口，大大简化了开发者的开发难度。

3. 项目需要完成的功能目标

本项目需要完成的功能指标可以用下图来概述：



具体的需求分析如下：

- **文件交互：**
 - 新建文件：会检查当前文档有无改动。如果有改动，会提示是否存储当前文档，并新建文件。
 - 打开文件：同样检查当前文档有无改动情况，并且从文件管理器打开.txt后缀的文本文件。
 - 存储：存储当前文档。
 - 另存为：将此文档另存为一份文件，并且将文件指针指向新存储的文件。
 - 退出：会先检查是否有未存储的内容，并提示是否要存储。
- **文字编辑：**
 - 复制、粘贴、剪切、撤销、重做：基础的文字编辑功能。
- **附加改进：**
 - 快捷键支持：全功能支持快捷键，可以完全无鼠标化操作。

- 当前行提示：当前行会被标注为淡绿色。
- Tab缩进优化：改普通的10个column为4个空格，优化排版功能。

4. 基本实现思路

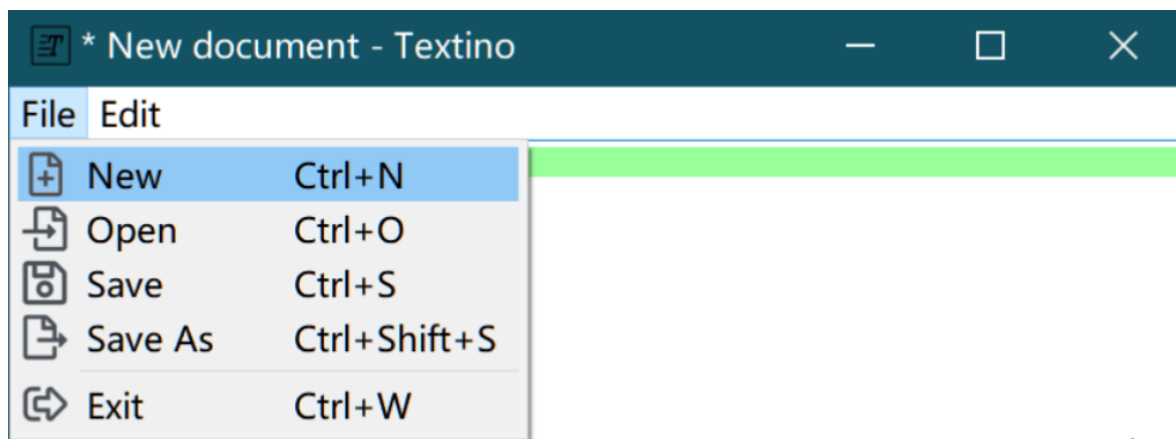
4.1 开发环境的配置和使用

从 [QT官方网站](#) 上下载得到 [QT](#) 库和 [QT Creator](#) 的安装程序，使用默认配置安装即可。

4.2 各功能的实现

我们的文字编辑器实现了用户交互界面和功能逻辑部分。使用功能强大的库Qt实现用户交互界面，便于用户操作，精确地获取并理解用户的需求，调用相关的后端功能。以下是各个功能的实现介绍：

4.2.1 文件交互



文件交互界面

- **新建文件**：检查当前文档有无改动。如果有改动，会提示是否存储当前文档，并新建文件。

```
1 void Textino::ActionNew(){
2     if(changed)           // 判断是否有改动
3         SavePrevious(); // 保存之前的文档
4
5     if(!changed){         // 新建文档
6         main_editor.clear(); // 清空当前文档
7         changed = false;    // 更新当前状态
8         file_path = nullptr;
9         setWindowTitle("New document - Textino");
10    }
11 }
```

- **打开文件**：检查当前文档有无改动情况，如果有改动，会提示是否存储当前文档，并且从文件管理器打开.txt后缀的文本文件。

```

1 void Textino::ActionOpen(){
2     if(changed)
3         SavePrevious(); // 存储当前文档
4     if(!changed){       // 打开新文档
5         QString open_path = ShowDialog(QFileDialog::AcceptOpen, "Open",
6             ":img/imgs/icon.png");
7         OpenFile(open_path);
8     }
9 }

```

- **存储和另存为**：存储当前文档。另存为则是将此文档另存为一份文件，并且将文件指针指向新存储的文件。

```

1 void Textino::ActionSave(){
2     QString save_path = SaveFile(file_path, "Save");
3     if(save_path!="")
4         file_path = save_path;
5 }
6 void Textino::ActionSaveAs(){
7     QString save_path = SaveFile(nullptr, "Save As");
8     if(save_path!="")
9         file_path = save_path;
10 }

```

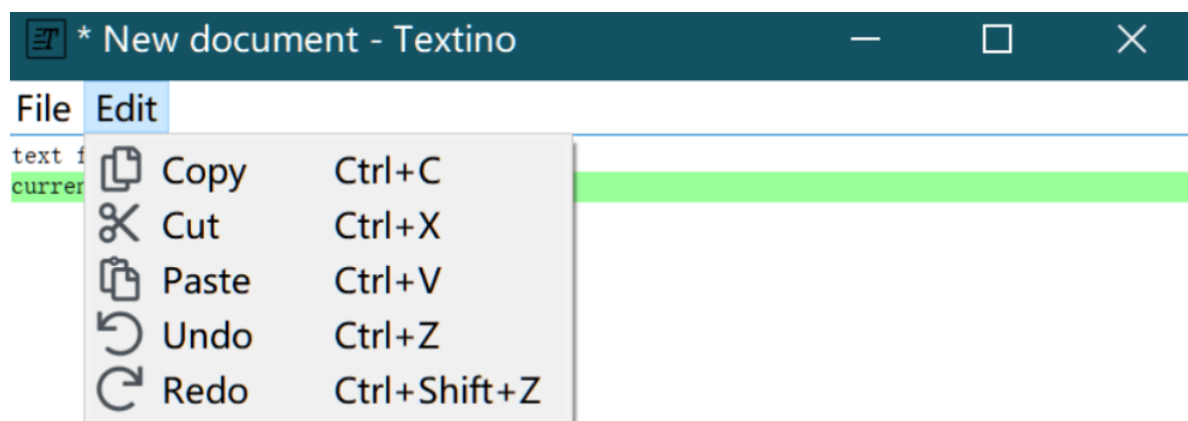
- **退出**：先检查是否有未存储的内容，如果有会提示是否存储当前文档，并且执行退出操作。

```

1 void Textino::ActionExit(){
2     if(changed)
3         SavePrevious();
4     if(!changed)
5         close();
6 }

```

4.2.2 文字编辑功能



文字编辑界面

- **复制、粘贴、剪切、撤销、重做**：可以直接调用Qt库中 `QPlainText` 中的槽函数。

```

1 void Textino::ActionCopy(){main_editor.copy();}
2 void Textino::ActionCut(){main_editor.cut();}
3 void Textino::ActionPaste(){main_editor.paste();}
4 void Textino::ActionUndo(){main_editor.undo();}
5 void Textino::ActionRedo(){main_editor.redo();}

```

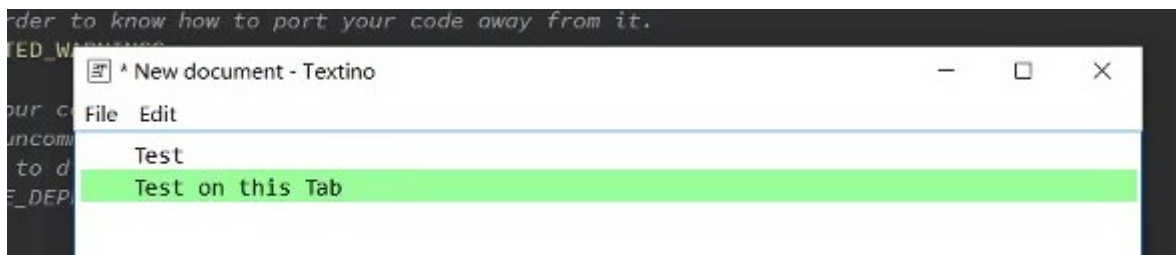
- **文字变动提示：**调用槽函数 `ActionChanged()` 来实现。当有新的改动没有保存时，设置标题前加 "*"。

```

1 void Textino::ActionChanged(){
2     if(!changed)
3         setWindowTitle("* "+windowTitle());
4     changed = true;
5     status_label.setText("length: " +
        QString::number(main_editor.toPlainText().length()) + "    lines: " +
        QString::number(main_editor.document()->lineCount()));
6 }

```

4.2.3 附加改进



附加改进

- **快捷键支持：**全功能支持快捷键，可以完全无鼠标化操作。
 - 快捷键主要依赖Qt库的槽函数设计。通过链接动作和槽函数可以加入快捷键支持。
 - 以 **新建文件** 功能作为示例展现以下槽函数的运行方式：

```

1 CreateAction(action, menu_file, "New", Qt::CTRL+Qt::Key_N,
    ":img/imgs/new.png"); // 创建动作
2 connect(action, SIGNAL(triggered()), this, SLOT(ActionNew())); // 链接动作和
    槽函数
3 menu_file->addAction(action); // 嵌入菜单栏
4
5 class Textino{ // 主类
6     // ..
7 private slots: // 主类中的槽函数限定符
8     void ActionNew(); // 槽函数
9     // ...
10 }

```

- **当前行提示：**光标所在当前行会被标注为淡绿色，方便文字编辑者阅读：

```

1  void Textino::ActionHighlight(){
2      QList<QTextEdit::ExtraSelection> extraSelections;
3      QTextEdit::ExtraSelection selection;
4      QColor lineColor = QColor(Qt::green).lighter(160);
5
6      selection.format.setBackground(lineColor);
7      selection.format.setProperty(QTextFormat::FullWidthSelection, true);
8      selection.cursor = main_editor.textCursor();
9      selection.cursor.clearSelection();
10     extraSelections.append(selection);
11     main_editor.setExtraSelections(extraSelections);
12 }

```

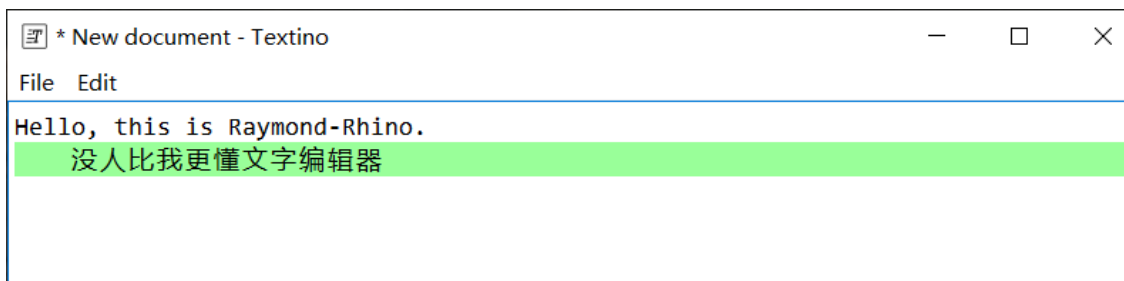
- **Tab缩进优化**：改普通的10 column的Tab键为4个空格，优化排版，避免可能出现的bug
 - 主要通过更改 `QPlainTextEdit` 中的内置事件实现

```

1  void PlainTextEdit::keyPressEvent(QKeyEvent* event)
2  {
3      if (event->key() == Qt::Key_Tab)
4          this->insertPlainText("    ");
5      else
6          QPlainTextEdit::keyPressEvent(event);
7  }

```

- **优化字体**：对于有中英文输入需求的人来说最合适的字体是每个中文字符对应两个英文字符。
 - 采用了 `YaHei Consolas Hybrid` 字符集来实现

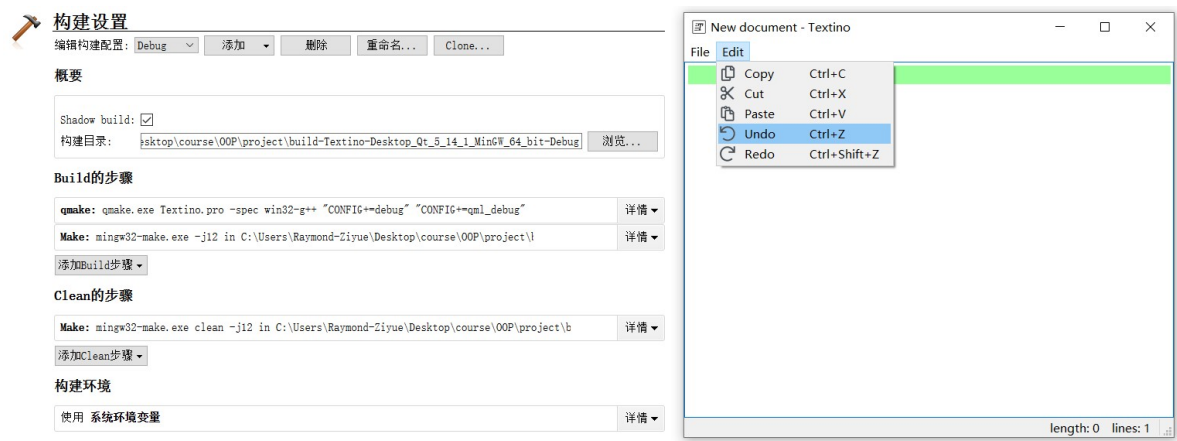


4.3 测试运行

我们的测试过程包括测试程序是否能够**正常编译运行**，**打开、关闭、保存、新建和另存为**是否能够正常运行，**撤销、重做、粘贴、复制、剪切**的功能以及**修改Tab键为4个空格、当前行提示**的附加功能。

4.3.1 编译运行

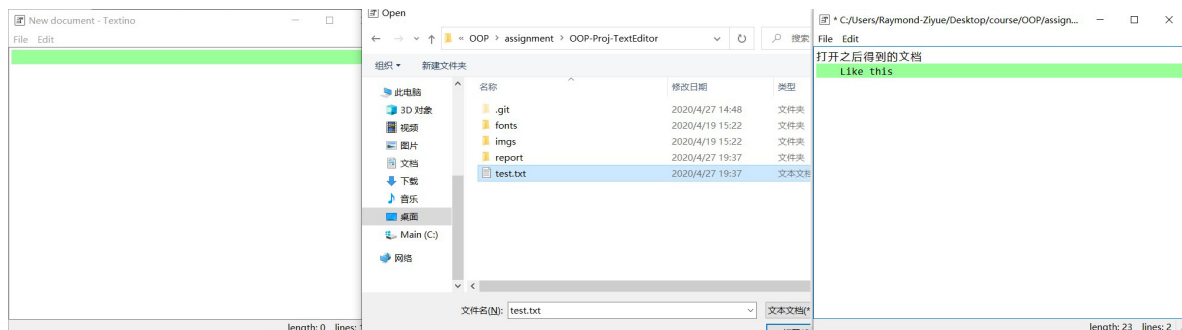
首先我们配置好项目，使用MinGW编译器，并点击运行后，程序运行无误，弹出编译得到的运行程序，窗口、菜单、图标、快捷键提示和程序名显示都没有问题。



编译运行

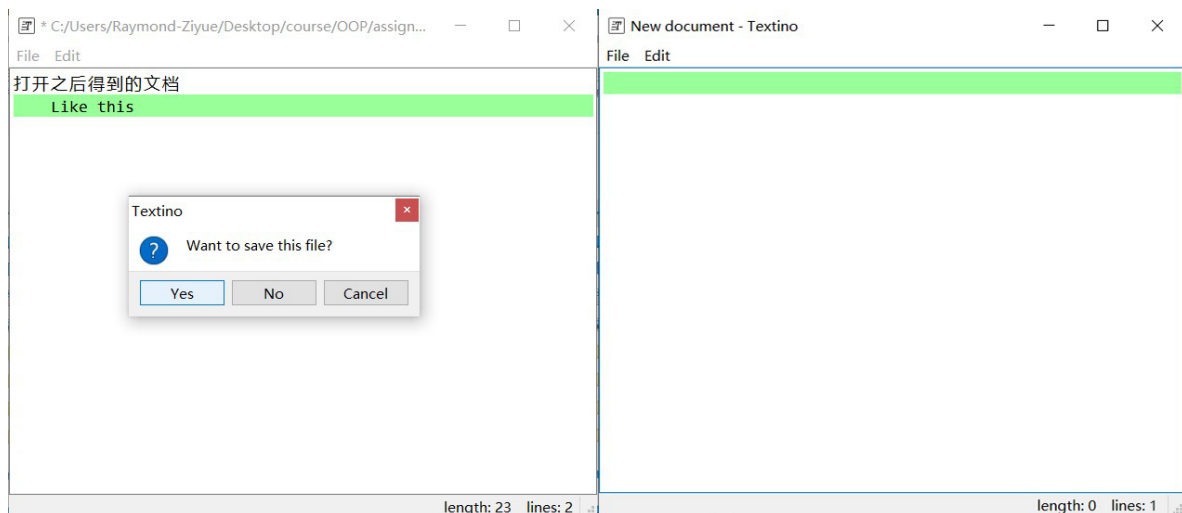
4.3.2 打开、关闭、保存、新建和另存为

我们首先测试打开一个文件。打开之后应用程序的标题显示了当前打开文档的绝对路径，并且显示了其中的内容。打开功能运行无误。



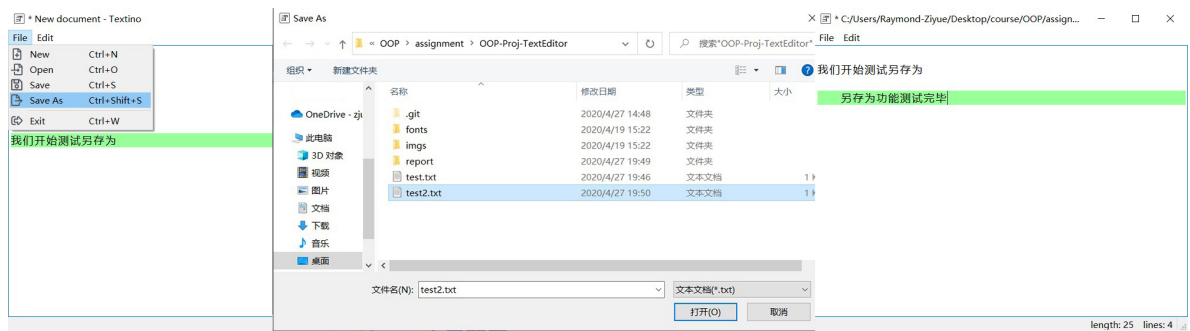
打开

接着我们测试新建文件和保存文件。由于上此打开的文档有改动，因此程序首先询问是否保存当前文件。选择是后，程序打开了新文档：



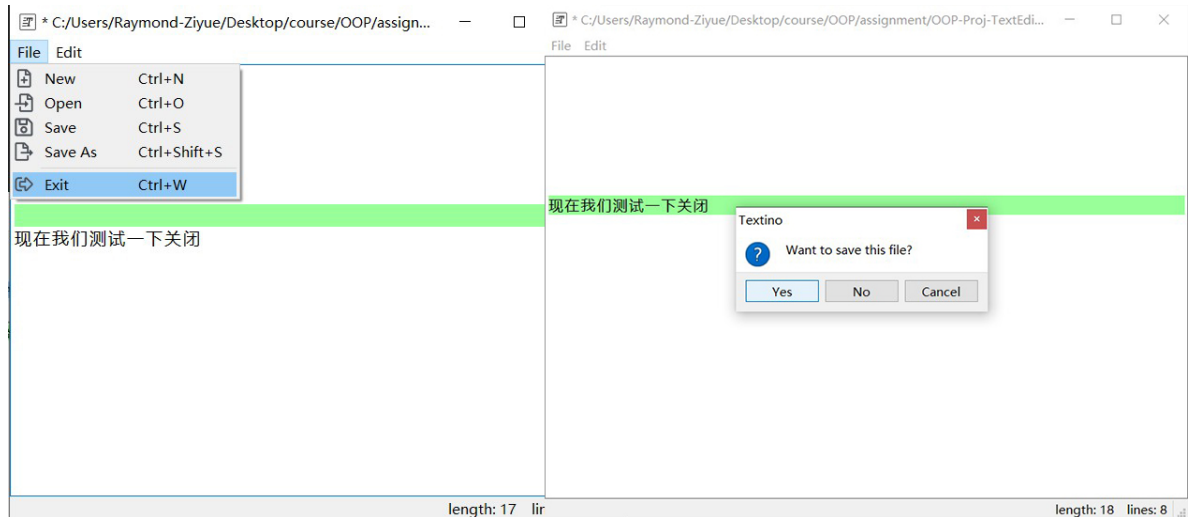
新建 & 保存

我们测试另存为功能。点击另存为，保存到本地的新建文件，则程序会将当前存储的文件路径显示出来，说明另存为功能测试成功。



另存为

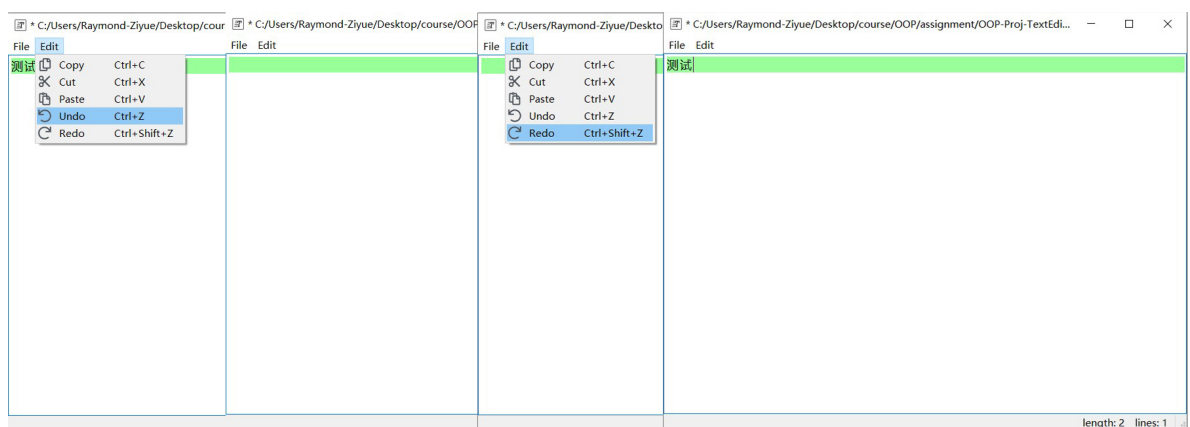
再来测试关闭文件的功能。由于有未保存的内容，因此首先会提示保存文档，然后再对关闭事件做操作。



关闭

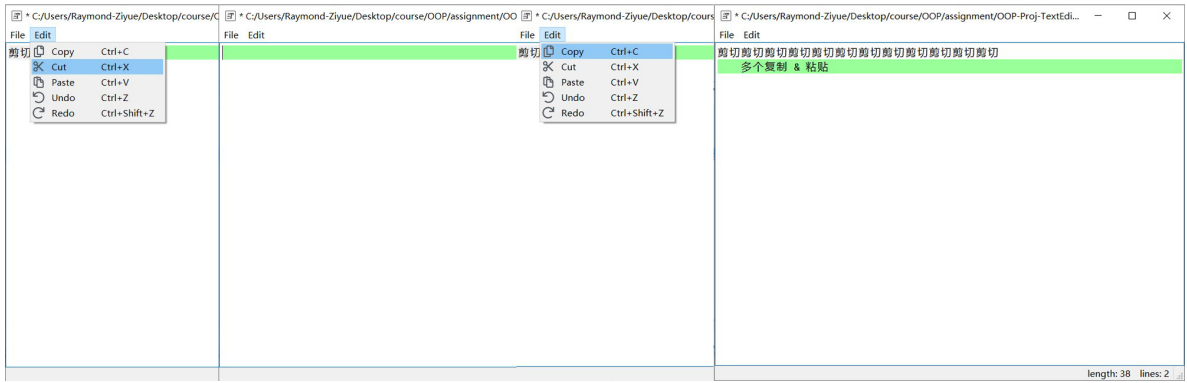
4.3.3 撤销、重做、粘贴、复制、剪切

首先测试撤销和重做，输入测试并撤销，然后重做：



撤销 & 重做

然后测试剪切、粘贴和复制。



剪切、粘贴和复制

4.3.4其他附加功能

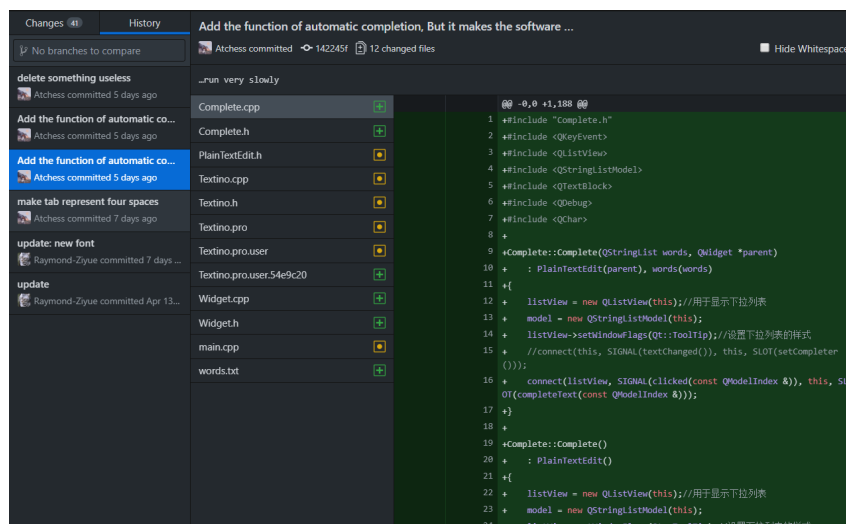
在前述的多张图片中可见当前行提示、文档改动提示以及Tab键改为4个空格的配置等等附加功能。

5. 人员职责

角色	姓名	职责
组长	潘子曰	总体策划，负责协调活动 负责召集组内成员学习讨论 IDE的选择、下载、安装、调试 库的下载、安装、使用 Project总体架构设计 基础代码实现 中期报告撰写和调整
组员	张雯琪	库的下载、安装、使用 中期报告四部分撰写
组员	艾琪	库的下载、安装、使用 Tab Policy部分代码实现 Project总体测试
组员	吴沁珮	库的下载、安装、使用 中期报告二、五、附录部分撰写

附录

- 为了小组成员代码沟通的方便，我们将代码托管在GitHub上，GitHub地址为：
<https://github.com/2020-ZJU-OOP/OOP-Proj-TextEditor>



GitHub commit记录

- 我们也建立了QQ群，方便线上线下沟通，以下是一些关键消息记录截图：

潘子曰(3508736995) 2020/4/13 19:13:57

我现在有一个功能来分一下锅，有人愿意吗？

潘子曰(3508736995) 2020/4/13 19:15:14

具体就是：文字编辑器一般按一个Tab是会缩进10个字符位。但是这种方式看上去非常不美观，我们希望把它变成按一个Tab缩进4个字符位，并且每个字符位都是空格

潘子曰(3508736995) 2020/4/13 19:16:47

我还需要另一个人帮我写一下报告

吴沁玥(982586719) 2020/4/13 19:22:55

或许我可以写报告吗 🤖，还不是会很弄QT

潘子曰(3508736995) 2020/4/13 19:23:07

sure

潘子曰(3508736995) 2020/4/13 19:24:29

可能你不太会用git操作，我会把每一次修改的代码上传到群文件

潘子曰(3508736995) 2020/4/13 19:25:23

• 目前报告写作用的是markdown，你也可以按你的习惯换

潘子曰(3508736995) 2020/4/19 15:03:12

• 具体一下：

张雯琪写第4部分

吴沁玥在原基础上改进第2、5、附录部分

第1和第3部分我来写

艾琪(205331937) 2020/4/14 10:28:35

接着我们要做些什么功能

潘子曰(3508736995) 2020/4/14 10:29:39

• auto completion

• 有兴趣没？

艾琪(205331937) 2020/4/14 10:30:29

补全？

潘子曰(3508736995) 2020/4/14 10:30:33

对

艾琪(205331937) 2020/4/14 10:30:58

针对某种语言吗？

潘子曰(3508736995) 2020/4/14 10:31:14

英文单词的补全

艾琪(205331937) 2020/4/14 10:31:22

• 哦

• 可以试试

潘子曰(3508736995) 2020/4/14 10:34:14

ok，你去试试

潘子曰(3508736995) 2020/4/14 10:34:31

qt有这个小查补全 可以看看

艾琪(205331937) 2020/4/16 06:12:03

• 刚改了一个小bug

OOP-Proj-TextEditor04.16.01...

21.28 MB

来自群文件

下载 转发

艾琪(205331937) 2020/4/16 06:13:34

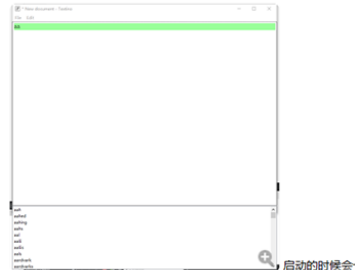
main函数里面文件的地址，不知道为什么vscode读不出相对地址，所以我写的绝对地址

潘子曰(3508736995) 2020/4/16 8:35:35

• 所有qt文件的地址要先加入Res.qrc中

• 然后用“...”来调用

艾琪(205331937) 2020/4/15 22:10:28



成PPT

艾琪(205331937) 2020/4/15 22:13:21

感觉从词库里面查找太费时间了

潘子曰(3508736995) 21:21:12

ok，我写了一个简单的文本编辑器，把中期报告里面那些redo，undo什么都实现了，还没有加一些特殊的功能

潘子曰(3508736995) 21:21:27

你们的github账号给我一下，我拉你们进协作者

学术中心-张雯琪(86107966) 21:28:52

？

学术中心-张雯琪(86107966) 21:29:19

您这效率太感人了 🤖