|  |
| --- |
| Code or die 小组 |
| 【C语言编辑器】 初步设计方案 |
| 界面设计、功能设计、数据结构设计 |

|  |
| --- |
| 朱长昊 张佳明 刘震宇 湛蓝蓝 蒋雨彤  2019-8-31 |

目录

[一、 界面设计方案 2](#_Toc18176930)

[(一) 界面整体布局设计方案 2](#_Toc18176931)

[1. 界面整体布局设计图 2](#_Toc18176932)

[2. 界面整体布局预览 2](#_Toc18176933)

[(二) 标题栏栏设计方案 2](#_Toc18176934)

[(三) 菜单栏设计方案 3](#_Toc18176935)

[1. 菜单栏整体设计方案 3](#_Toc18176936)

[2. 文件区域设计方案 3](#_Toc18176937)

[3. 编辑区域设计方案 3](#_Toc18176938)

[4. 编译运行区域设计方案 4](#_Toc18176939)

[5. 格式区域设计方案 4](#_Toc18176940)

[6. 帮助区域设计方案 5](#_Toc18176941)

[(四) 工具栏设计方案 6](#_Toc18176942)

[(五) 代码编辑区设计方案 7](#_Toc18176943)

[(六) 文件资源管理区设计方案 7](#_Toc18176944)

[(七) 状态栏设计方案 8](#_Toc18176945)

[(八) 控制台设计方案 8](#_Toc18176946)

[二、 功能设计方案 9](#_Toc18176947)

[(一) 功能整体预览 9](#_Toc18176948)

[(二) 文件管理方面 11](#_Toc18176949)

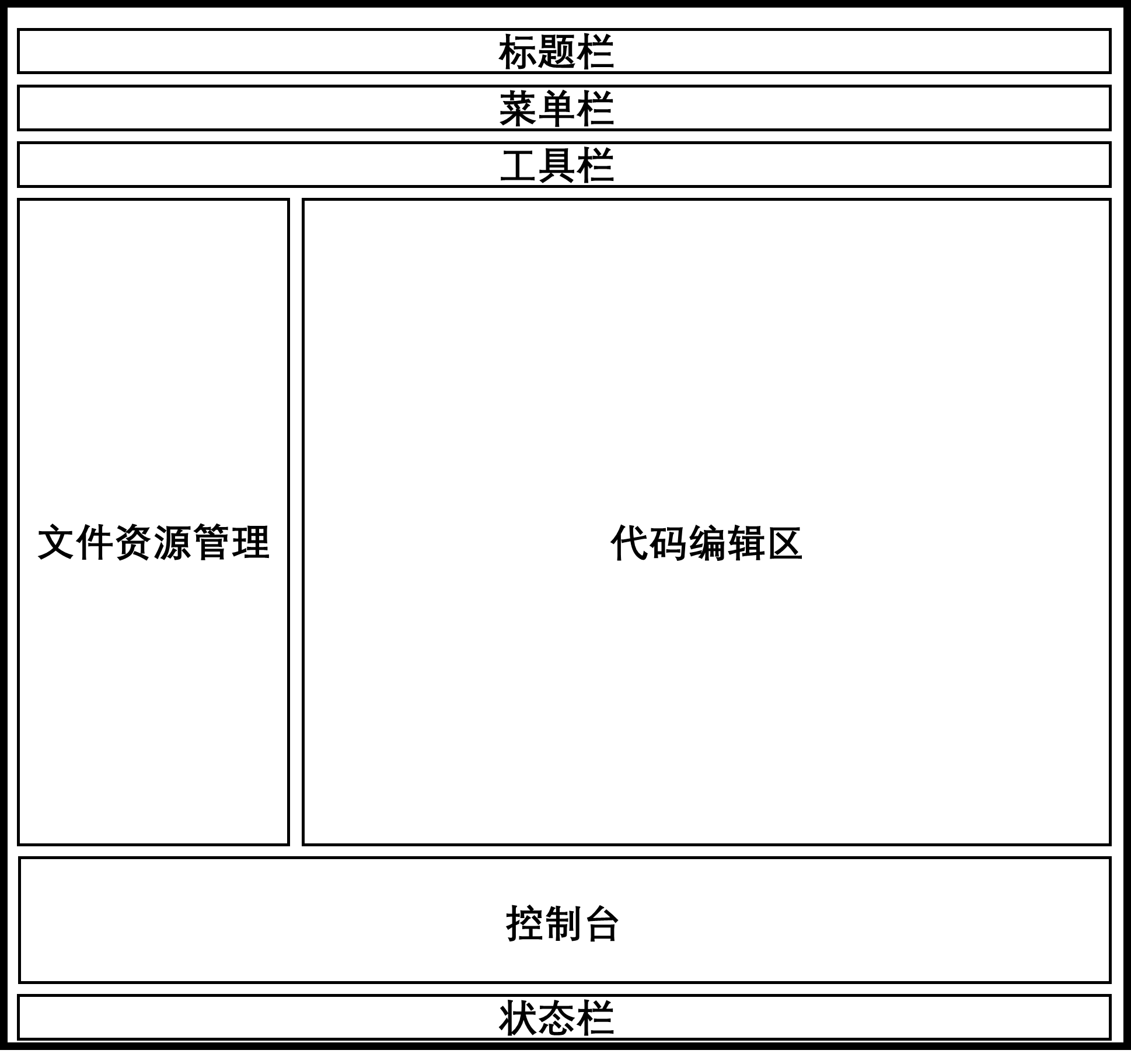
[(三) 文本编辑方面 11](#_Toc18176950)

[(四) 代码编辑方面 11](#_Toc18176951)

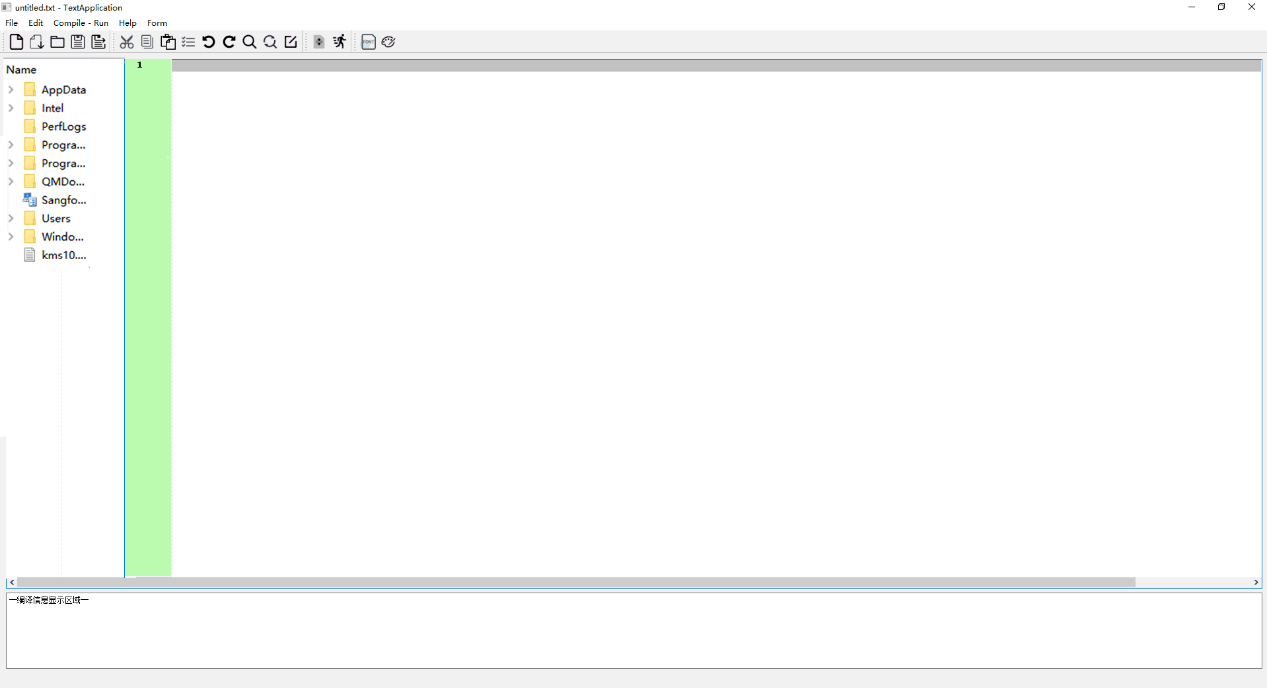
[(五) 代码运行方面 12](#_Toc18176952)

[三、 UML设计 12](#_Toc18176953)

1. **界面设计方案**
2. **界面整体布局设计方案**
3. 界面整体布局设计图



1. 界面整体布局预览



1. **标题栏栏设计方案**



显示当前文件名（未保存时为untitled.txt）

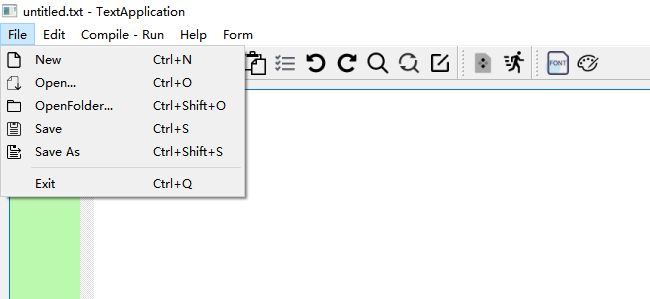
1. **菜单栏设计方案**
2. 菜单栏整体设计方案



1. 文件区域设计方案

文件区域实现功能：

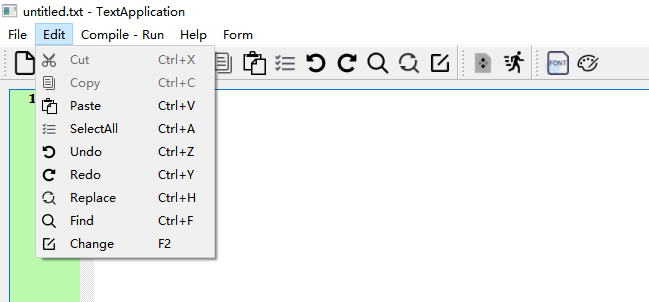
1. 新建文件 Ctrl + N
2. 打开文件 Ctrl + O
3. 打开文件夹 Ctrl + Shift + O
4. 保存 Ctrl + S
5. 另存为 Ctrl+ Shift + S
6. 退出程序 Ctrl + Q



1. 编辑区域设计方案

编辑区域实现功能：

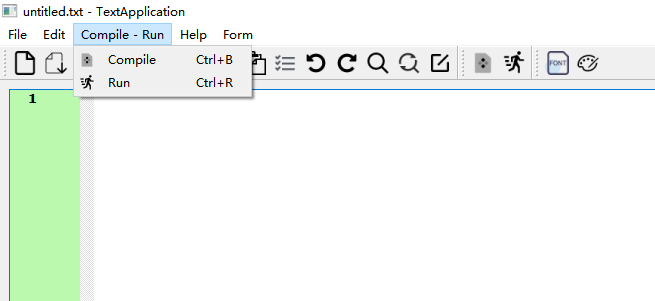
1. 复制 Ctrl + C
2. 粘贴 Ctrl + V
3. 剪切 Ctrl + X
4. 全选 Ctrl + A
5. 撤销 Ctrl + Z
6. 重做 Ctrl + Y
7. 查找 Ctrl + F
8. 替换 Ctrl + H
9. 变量名重命名 F2



1. 编译运行区域设计方案

编译运行区域实现功能：

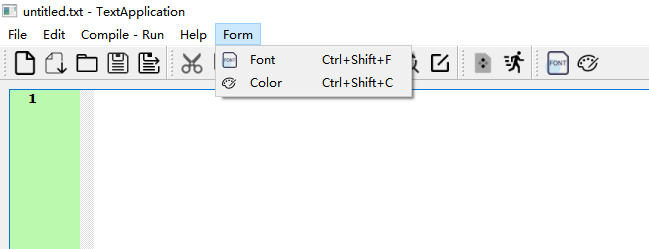
1. 编译 Ctrl + B
2. 运行 Ctrl + R



1. 格式区域设计方案

编译运行区域实现功能：

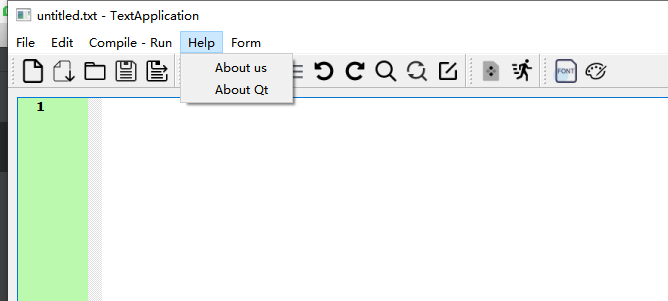
1. 字体大小、样式
2. 字体颜色



1. 帮助区域设计方案

帮助区域实现功能：

1. 关于我们
2. 关于Qt

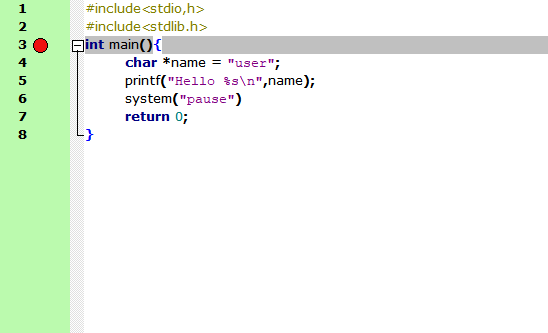


1. **工具栏设计方案**



工具栏实现功能

1. 新建文件 Ctrl + N
2. 打开文件 Ctrl + O
3. 打开文件夹 Ctrl + Shift + O
4. 保存 Ctrl + S
5. 另存为 Ctrl+ Shift + S
6. 复制 Ctrl + C
7. 粘贴 Ctrl + V
8. 剪切 Ctrl + X
9. 全选 Ctrl + A
10. 撤销 Ctrl + Z
11. 重做 Ctrl + Y
12. 查找 Ctrl + F
13. 替换 Ctrl + H
14. 变量名重命名 F2
15. 编译 Ctrl + B
16. 运行 Ctrl + R
17. 字体大小、样式
18. 字体颜色
19. **代码编辑区设计方案**

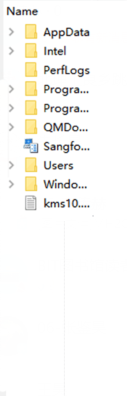


代码编辑区实现功能：

1. 调整字体大小 Ctrl + mouseWheelEvent
2. 关键字高亮
3. 关键字补全
4. 括号匹配高亮
5. 括号、引号自动补全
6. 自动缩进
7. 代码自动整理
8. 代码整体整理 Ctrl + K + D
9. 代码块折叠
10. 多行代码注释 Ctrl + /
11. 行号显示
12. 当前行阴影
13. 添加断点
14. **文件资源管理区设计方案**

文件资源管理区实现“打开文件夹”功能的可视化

提供 可折叠展开的树形管理机制 和 “打开文件”功能的引用

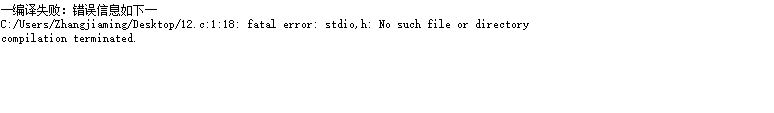


1. **状态栏设计方案**



状态栏提供功能：

1. 提供鼠标悬停所在功能的提示
2. 提供功能实现后的反馈提示
3. 提供编写时当前代码的行列显示
4. **控制台设计方案**



状态栏提供功能：

1. 提供当前文件编译结果的显示
2. 提供该程序运行的输入和输出
3. **功能设计方案**
4. **功能整体预览**

|  |  |
| --- | --- |
| 文本编辑方面 | 复制ctrl+c |
| 粘贴ctrl+v |
| 查找ctrl+f |
| 替换ctrl+h |
| 撤销ctrl+z |
| 重做ctrl+y |
| 在状态栏显示当前行数 |
| 自动保存（可选择是否开启此项功能） |
| 调整编辑区字体大小ctrl + wheel |
|  |
|  |
| 代码编辑方面 | 关键字补全、识别高亮 |
| 括号自动匹配高亮 |
| 自动补全 1.() 2. {} 3."" |
| 联想提示关键字 |
| 变量重命名F2 |
| 换行后自动标齐位置 |
| 行格式排版 |
| 整体格式排版 |
| 多行注释 快捷键 |
| 代码块折叠 |
| 代码运行方面 | run gcc / g++ |
| 提取编译错误信息 并显示 |
| 控制台显示错误信息 |
| 控制台生成结果 |
| 多文件编译 |
| debug gbk |
| 文件管理方面 | 新建文件 |
| 打开文件 |
| 打开文件夹 |
| 保存文件 |
| 另存为文件 |
| 树形文件管理(点击可扩展式) |
| 多文件编辑(菜单栏下册&&左侧的属性管理) |

1. **文件管理方面**
2. 新建文件

在编辑区新建空白文件编辑区（检测当前文件是否保存）；

1. 打开文件

用户通过弹窗选择打开的文件，并将文件加载在文件编辑区（检测当前文件是否保存）；

1. 打开文件夹

用户通过弹窗选择想要打开的文件夹，并将文件夹展现在文件资源管理区；

1. 保存文件

用户通过弹窗选择想要保存的文件夹，并将当前文件保存；

1. 另存为

用户通过弹窗选择想要保存的文件夹，并将当前文件保存；

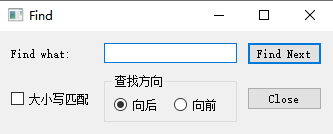
1. 树形文件资源管理

将用户选择的文件夹通过树形展示，并可双击打开其中文件；

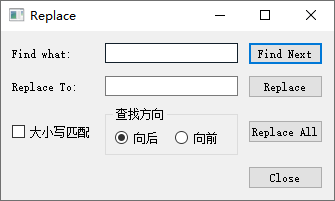
1. 多文件编辑

将用户所有打开的文件通过选项卡进行展示；

1. **文本编辑方面**
2. 复制：复制用户在编辑区内选中的文本；
3. 粘贴：粘贴内容；
4. 剪切：剪切用户再编辑区内选中的文本；
5. 全选：全部选中编辑区内所有文本；
6. 查找：查找编辑区所含用户输入关键字的区域；



1. 替换：查找编辑区所含用户输入关键字的区域，并将其用所替换的文字进行替换；



1. 撤销：撤销用户当前执行的操作；
2. 重做：重新执行用户撤销的操作；
3. 自动保存：系统自动保存当前文本于临时存储区，可设置是否执行此项功能及间隔时长；
4. 调整字体：通过“ctrl + wheel”实现编辑区的字体大小调整；
5. **代码编辑方面**
6. 关键字联想补全提示：根据用户当前输入提供关键字提示列表，按Tab进行补全；
7. 关键字识别高亮：检测编辑区文本是否为关键字，如果是则高亮显示
8. 括号自动匹配高亮：查找当前光标所在括号的所匹配的括号，并通过高亮显示
9. 自动补全：当用户输入“ **(** ”，” **[** ”，” **{** ”，” **”** “，” **’** “时进行自动补全；
10. 变量重命名：检测当前光标所在的变量，提供输入框将所有该变量进行重命名；
11. 代码自动缩进：实现代码的自动缩进；
12. 行格式自动排版：当用户输入 **;** ，对当前行的代码进行排版；
13. 整体格式排版：将本文件的代码格式进行整理排版；
14. 多行注释：将用户选择文本区域通过添加 **//** 进行注释；
15. 代码块折叠：在编辑区侧边栏提供代码块折叠；
16. **代码运行方面**
17. 编译

调用gcc编译器在当前文件路径下生成可执行.exe文件，将编译结果显示在控制台。如果文件未保存则优先提示用户保存文件，弹出保存页面。

1. 运行：编译并运行.exe文件；
2. 调试：可打断点进行单步逐语句调试等调试功能；
3. **UML设计**

编辑器采用QsciScintilla类，词法分析器采用QsciLexerCPP类，可以实现关键字识别高亮、括号自动匹配高亮、自动排版、代码块折叠等功能。

编译功能使用gcc。当用户点击编译按钮，系统会自动调用cmd里的gcc编译命令，通过gcc编译实现程序的编译，运行即直接运行编译好的.exe文件。

窗口方面，mainWIndow继承自QMainWindow类，findDialog继承自QDialog类，replaceDialog类继承自findDialog类。

按键监听为自定义类，继承自QObject。

UML如下：

