浙江大学

本科实验报告

课程名称：C程序设计专题

姓　　名：黄文翀

学　　院：数学科学学院

专　　业：数学与应用数学（强基计划）

学　　号：3200100006

指导老师：翁恺

2021年5月11日

单行文本编辑器 实验报告

总体设计

程序共分**文本编辑框、状态显示栏、功能区、红太阳区**四个板块。

文本编辑框实现了单行文本的编辑，能够显示光标，支持`Shift`、`CapsLock`、`Insert`、`Delete`、`Backspace`、`Home`、`End`等与单行文本编辑相关的功能键，但是不支持中文以及小键盘按键。

状态显示栏显示编辑模式（插入或覆盖），以及光标位置、文本总长度、单行允许的最大文本长度

功能区实现了几个按钮，并且制作了点击效果，使用鼠标左键点击，可以在一定范围内调节字号、颜色和字体，但只支持了三种基本的等宽字体

红太阳区是娱乐板块，打开程序时显示一个巨大的正方形按钮，点击后该按钮消失，随后显示翁恺老师的头像字符画

文本编辑框

文本框是一个900×50的矩形，每当有新输入或删除字符时，用白色矩形覆盖文本框区域实现局部清屏，然后更新字符串数组，再显示新的文本。输入回车后，将字符串数组在终端输出，然后清空字符串数组与文本框区域。以下是具体细节。

编辑模式的切换

用一个全局变量`insertMode`来表示用户的编辑模式，1代表插入，0代表覆盖。程序启动时将其初值设为1，也即默认用户的输入模式为插入

当程序判断到用户按下`insert`按键时，更新`insertMode`的值以及光标的形状。插入模式下，光标是一条黑色细直线；覆盖模式下，光标是与单个字符等宽的黑色方块。

字符的添加

设计一个`inputEvent(char c)`函数，实现用户输入字符`c`后，将其添加到字符数组的功能。另外需要一个全局变量`curPosition`用于记录用户的当前位置。

需要判断此时的输入模式：如果是插入模式，先将当前位置之后所有字符往后移一位，再将当前位置的字符改为`c`，然后令当前位置向后移动一位；如果是覆盖模式，直接将当前位置的字符改为`c`，然后令当前位置向后移动一位

大写锁的处理

调用函数`GetKeyState(VK_CAPITAL)`可以获取大写锁的开启状态，返回`1`时表示大写锁开启，`0`表示关闭。

当用户输入的是英文字母（对应虚拟键码65至90）时，判断大写锁的开启状态，如果开启则调用函数添加对应的大写字母，否则添加小写字母。

shift按键的处理

在`keyEvent`函数中用一个局部静态变量`shift`来记录shift按键的按下状态，初值设为0。当读取到shift键按下时，将其设为1；当读取到shift按键弹起时，将其重新设为0。

当用户输入时，判断`shift`是否为1，如果是则添加按键上对应的上方字符，否则添加按键上的原字符。由于虚拟键码和ASCII码在特殊字符的编码方面没有规律，得不到通用的转换公式，只能为每个按键单独处理。此外，按照通常习惯，按住shift按键时输入的英文字母应为大写，这里也进行特别处理。

Backspace与Delete的处理

当用户按下Backspace按键时，将`curPosition`的前一位置上的字符置为空，然后将它之后（包括它）的所有字符向前移动一位，最后使`curPosition--`。特别地，如果当前位置已经是字符串开头，则发出错误警报声音，并不对字符串做处理。

当用户按下Delete按键时，将`curPosition`的前上的字符置为空，然后将它之后（不包括它）的所有字符向前移动一位，最后`curPosition`仍然保持不动。特别地，如果当前位置已经是字符串结尾，则发出错误警报声音，并不对字符串做处理。

方向键的处理

按下左右方向键时，更新`curPosition`，并根据字符宽度计算出光标的新位置并调整。特别地，若已经到达字符串开头继续按左，或已经到底字符串结尾继续按右，则只发出错误警报声音，不做其它处理。

文本宽度的处理

这是一个非常麻烦的工作，因为同一个字号，不同的字体对应的字符所占宽度不同，且有些字体找不到通过字号计算字符宽度的通用公式。这里进行了大量的实验，最终确定了支持的三种字体和26种字号中，每种情况的字符宽度，并将其写入了函数`getTextWidth`中。

超出文本框的处理

此项处理不易实现，因此直接对字符串长度进行限制，不允许在到达指定最大长度后继续添加字符。

功能区

功能区实现了文本大小、颜色、字体调节的功能，当然，这里的调节都是针对整行文本的调节，暂未实现针对单个字符的调节。以下分别说明。

按钮的设计

按钮设计为一个22×22大小的矩形，并在里面显示有`+`、`-`或`...`的字样。

为每个按钮配置一个全局变量记录其按下状态，初始值置为0，当检测到鼠标位于按键的矩形区域内，并且鼠标左键按下时，将改变量的值置为1，当鼠标松开后将其值重新置为0

绘制按钮时，判断按钮对应的状态变量的值：如果值为0，表明按钮没被按下，使用白色笔刷、黑色文本绘制按钮；如果值为1，表明按下，则使用黑色笔刷、白色文本绘制按钮。这样实现了按住按钮的显示效果。

按钮的触发时机设在按下的瞬间，而不是松开的瞬间（当然这并没有什么用意）。触发后执行相应的功能

文本大小调节

在字号显示位置的左、右方分别置一个按钮，文本分别显示为`-`与`+`，按下`-`时令字号减一，按下`+`时令字号加一，允许的字号调节范围为15至40。

每次字号调节之后，对每行能容纳的字符个数进行重新计算，如果不能容纳原本的字符串，则直接将其截断，舍去不能容纳的部分。此外，文本框的高度、文本的宽度、光标的高度也要根据字号进行计算。

颜色调节

支持黑、红、绿、蓝、青、品红、黄七种颜色，在颜色名称作边置一个按钮，显示文本`...`，点击按钮后按顺序切换到下一种颜色，到黄之后再循环切换到黑。

颜色调节后，用新的颜色重新显示文本框中的文本。

字体调节

支持`Consolas`、`Courier`、`Courier New`三种等宽字体，在字体名称旁边置一个按钮，显示文本`...`，点击按钮后按顺序切换到下一种字体，最后再循环切换回第一个。

字体调节后，文本宽度受到影响，需要调用`getTextWidth`重新获取文本宽度，并重新显示文本框中的文本、更新光标位置。

红太阳区

这是一个娱乐板块。按钮的设计与功能区大同小异，只不过为了更显眼，按钮被设计成了200×200的大小，里面用直线绘制了一个巨大的加号。并且为了展示点击效果，将按钮的触发时机改为点击后松开的时刻。

当按钮被触发后，按钮原来所在的区域被白色矩形覆盖实现局部清屏，随后显示由字符组成的翁恺老师头像，并显示预先写好的文字，以表明对翁恺老师的感谢与尊重，以及虔诚的膜拜。