

浙江大学实验报告

实验名称: 汉字花样显示

学生姓名: 翁罗轩 专业: 软件工程 学号: 3170103226

实验日期: 2019 年 4 月 20 日

一、 实验目的

利用十六点阵汉字库 **hzk16**, 编写图形模式下汉字显示程序。要求在普通显示之外, 增加多种显示方式, 如: 文件显示、放大、斜体、倒立、彩色、空心、中英文混搭显示、动态显示等。

- 实现 **GB** 码与 **ZB** 码的相互转换。
- 设计一种汉字输入法, 并编程实现。

二、 实验原理

程序主要使用 **hzk16** 与 **mhzk** 这两个汉字库来实现点阵输出, 并实现放大、斜体、倒立、中英文显示、汉字输入法等功能。

为实现汉字输入法功能, 需找到每个汉字与其拼音的对应文件。这里使用了网上下载的包含 **3500** 个常用汉字的文件, 并为读取方便修改了其格式。文件中每一行起始为一不带声调的拼音, 后为相应的各个汉字。由于控制台界面不支持中文输入法, 实现汉字输出功能时就需要使用这一自创输入法。用户输入一个不带声调的拼音后, 程序搜索 **py.txt** 文件, 找到对应的行, 首先输出待选汉字, 用户选择后, 该汉字即可以点阵形式输出。

若为中文输出, 则通过搜索相应的字库来得到该汉字的区码与位码, 分别保存在 **buf1[0]** 与 **buf1[1]** 中。由于每个区中有 **96** 个汉字, 因此该汉字的序号 $order = 94 * (buf1[0] - 0XA1) + (buf1[1] - 0XA1)$ (**hzk16** 字库) 或 $order = 94 * (buf1[0] - 0XA1) + (buf1[1] - 0XA1) + 256$ (**mhzk** 字库), 其中 **0XA1** 为区位的起始存储地址。又因为每个汉字占 **32** 个字节, 所以该汉字在字库文件的起始位置 $position = order * 32$ 。为方便后续输出, 将该汉字的点阵信息读入二维数组 **tmp[16][2]** 中。

若为英文输出, 则该字母在字库中的位置 $position = order * 32 = buf2[0] * 32$ 。同样将它的点阵信息读入二维数组 **tmp[16][2]** 中。

输出时, 遍历 **tmp**, 判断当前的点阵信息是否为空, 若非空, 则输出"●", 若否, 则输出"○", 这样, 汉字或字母将以 **16 * 16** 点阵输出。

实现放大功能时, 只需同步增加各种输出的内容即可 (以循环实现), 最终结果将以 $(16 * times) * (16 * times)$ 点阵输出, 其中 **times** 为放大倍数。

实现倒立功能时, 只需将之前的 **tmp** 数组反向输出, 并同步调整各项参数

（如循环时的 i 改为 $15 - i$ ， j 改为 $1 - j$ ， k 改为 $7 - k$ 等等，具体可见代码）即可。

实现斜体功能时，只需在每行前面逐步增加一定数量的空格（逐行递增）即可。为方便显示，这里将输出时的"●"改为"***"，"○"改为" "。

由于时间有限，本次实验未使用图形库，而是在控制台界面显示，因此无法实现比如彩色显示、动态显示等功能。但总的来说，实现的基本原理都是一样的，因此若有必要，将其改为图形库界面并不是一件困难的事。

此外，在实际应用中，花样显示的整合可能更加重要，比如在实现放大的同时实现倾斜。这也非常简单，可以通过修改极少量的代码来实现。

三、 使用方法

1. 经测试，该代码在 Visual Studio 2017 与 Dev-C++ 中均可正常编译通过。由于代码中使用了某些特殊字符，不同的编译器编码方式不同，可能在打开源代码时会产生乱码，因此建议直接运行 HZ.exe 程序。
2. 运行程序，根据提示输入数字选择相应的操作，1 为汉字输出，2 为英文字母输出，0 为退出。
3. 接下来继续根据提示来选择操作与编码方式，如 GB 码与 ZB 码的选择，不同花样显示的选择，拼音汉字的选择等等。

```
1. Chinese character    2. English character    0. Exit
1
Select the operation you want to take:
1. Normal display.
2. Enlargement display.
3. Upside-down display.
4. Incline display.
1
Select the code you want to use.
1. GB code      2. ZB code
1
Enter the pinyin of a Chinese character.
liu
1.溜 2.留 3.刘 4.流 5.柳 6.六
Enter your choice.
_
```

```
1. Chinese character    2. English character    0. Exit
2
Select the operation you want to take:
1. Normal display.
2. Enlargement display.
3. Upside-down display.
4. Incline display.
1
Enter an English character:
_
```

四、 测试结果

1. 汉字输入法

```
Enter the pinyin of a Chinese character.  
li  
1. 厘 2. 梨 3. 黎 4. 离 5. 理 6. 李 7. 里 8. 鲤 9. 礼 10. 莉  
11. 荔 12. 栗 13. 丽 14. 厉 15. 励 16. 历 17. 利 18. 例 19. 俐 20. 立  
21. 粒 22. 力 23. 璃  
Enter your choice.
```

2. 普通输出

```
Enter your choice.  
2  
○○○○○○○●○○○○○●○○○○  
○○●●●●○○○○○○○○●○○○  
○○○○○○●○○○○○○○○●○○○  
○●●●●●●●●●●○○○○○○○○  
○○○○○○●○○○○○○○○●○○○  
○○○○○○●●●●○○●○○○○○○  
○○●●○○○○○○○○○○○○○○○○  
○○○○○○●○○○○○○○○●○○○  
○○○○○○○○○○○○○○○○○○  
○●●●●●●●●●●●●●●●●○○  
○○○○○○○○●●●●○○●○○○○  
○○○○●●●○○○○○○○○○○○○  
○○○●●○○○○○○○○○○○○○○  
○●●○○○○○○○○○○○○○○○○  
○○○○○○○○○●○○○○○○○○○  
○○○○○○○○○●○○○○○○○○○
```

3. 放大输出

Enter your choice.

```
Input enlarging times(integer).
```

- #### 4. 倒立输出

