



HZK16汉字显示系统

3180103772 张溢弛

1.开发环境

- OS: Windows 10
- IDE: Visual Studio 2017
- Language: C++

2.程序结构与算法

- 本程序主要由以下几部分构成
 - 读取hzk16汉字库文件
 - 读入输入的中文字符
 - 计算偏移量offset，在汉字库中找到对应的字
 - $$\text{offset} = (94 * (\text{unsigned int})(\text{word}[0] - 0\text{xa0} - 1) + (\text{word}[1] - 0\text{xa0} - 1)) * 32;$$
 - 显示出若干种16*16的点阵汉字
 - 正常显示
 - 倒立显示
 - 360度旋转后显示
 - **斜体**显示汉字
 - 用二进制串显示汉字
 - 在文件中显示汉字
 - 放大显示汉字
 - **更改字体**显示
 - **彩色**显示
- 提示：**程序可以手动切换字体**
- 通过设置宏定义中的font的值可以切换自己想要的字体，默认值为1，宋体，楷体为2，黑体为3
- 每种显式方式实现的算法如下
 - 正常显示

```
printf("-----正常显示-----\n");
for (k = 0; k < 16; k++) {
    for (j = 0; j < 2; j++) {
```

```

        for (i = 0; i < 8; i++) {
            flag = buffer[k * 2 + j] & key[i];
            if (!flag) {
                printf("☆");
            }
            else {
                printf("★");
            }
        }
    }
    printf("\n");
}

```

◦ 倒立显示

```

printf("-----倒立字体-----\n");
    for (k = 15; k >= 0; k--) {
        for (j = 0; j < 2; j++) {
            for (i = 0; i < 8; i++) {
                flag = buffer[k * 2 + j] & key[i];
                if (!flag) {
                    printf("☆");
                }
                else {
                    printf("★");
                }
            }
        }
    }
    printf("\n");
}

```

◦ 360度旋转后显示

```

printf("-----反转字体-----\n");
    for (k = 15; k >= 0; k--) {
        for (j = 1; j >= 0; j--) {
            for (i = 7; i >= 0; i--) {
                flag = buffer[k * 2 + j] & key[i];
                if (!flag) {
                    printf("☆");
                }
                else {
                    printf("★");
                }
            }
        }
    }
    printf("\n");
}

```

◦ 斜体显示

```

printf("-----斜体显示-----\n");
    for (k = 0; k < 16; k++) {
        for (i = 0; i < 16 - k; i++) {
            printf("☆");
        }
    }

```

```

        for (j = 0; j < 2; j++) {
            for (i = 0; i < 8; i++) {
                flag = buffer[k * 2 + j] & key[i];
                if (!flag) {
                    printf("☆");
                }
                else {
                    printf("★");
                }
            }
        }
        for (i = 0; i < k; i++) {
            printf("☆");
        }
        printf("\n");
    }
}

```

◦ 二进制串显示

```

vector<string> word = change2binary(buffer); // 这里有一个编写的函数用于转换
printf("-----二进制字符表示-----\n");
for (i = 0; i < 16; i++) {
    cout << word[2 * i] << word[2 * i + 1] << endl;
}

```

◦ 文件中显示

```

printf("-----文件显示-----\n");
FILE *fp2 = fopen("result.txt", "w");
for (k = 0; k < 16; k++) {
    for (j = 0; j < 2; j++) {
        for (i = 0; i < 8; i++) {
            flag = buffer[k * 2 + j] & key[i];
            if (!flag) {
                fwrite(" ", sizeof(" "), 1, fp2);
            }
            else {
                fwrite("1", sizeof("0"), 1, fp2);
            }
        }
    }
    fwrite("\n", sizeof("\n"), 1, fp2);
}
printf("请在文件result中查看文件输出的结果! \n");
fclose(fp2);

```

◦ 放大显示

```

printf("-----放大显示-----\n");
for (k = 0; k < 16; k++) {
    for (j = 0; j < 2; j++) {
        for (i = 0; i < 8; i++) {
            flag = buffer[k * 2 + j] & key[i];
            if (!flag) {
                printf("☆☆");
            }
        }
    }
}

```

```

        }
        else {
            printf("★★");
        }
    }
}
printf("\n");
for (j = 0; j < 2; j++) {
    for (i = 0; i < 8; i++) {
        flag = buffer[k * 2 + j] & key[i];
        if (!flag) {
            printf("☆☆");
        }
        else {
            printf("★★");
        }
    }
}
printf("\n");
}

```

◦ 彩色显示

```

system("color F2");
printf("-----彩色显示-----\n");
for (k = 0; k < 16; k++) {
    for (j = 0; j < 2; j++) {
        for (i = 0; i < 8; i++) {
            flag = buffer[k * 2 + j] & key[i];
            if (!flag) {
                printf("☆");
            }
            else {
                printf("★");
            }
        }
    }
}
printf("\n");
}

```

3. 程序运行样例

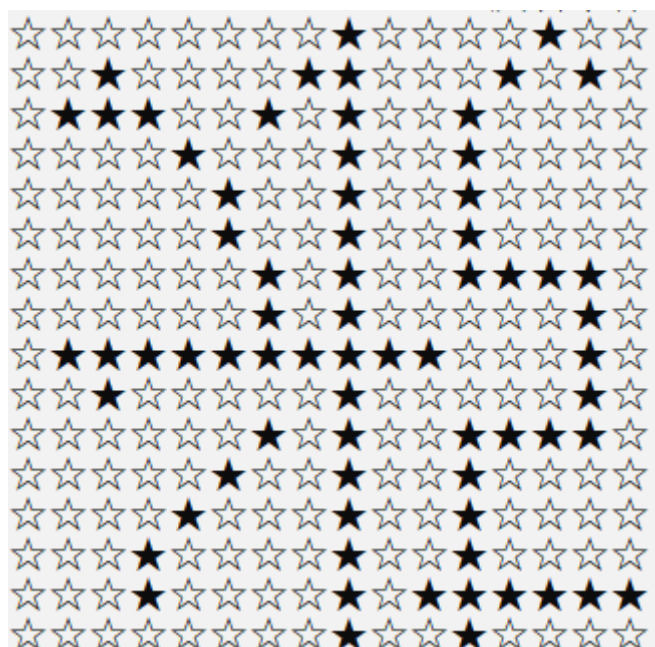
- 正常显示字体



- 倒立显示



- 旋转360度



- 斜体显示




- 用二进制串来显示

```

0000100100000000
1111110100001000
0000100100001000
0000100100010000
0000100100100000
0000100100100000
0111100101000000
0100000100000100
0100011111111110
0100000101000000
0111100101000000
0000100100100000
0000100100100000
0000100100010000
0000100101001110
0101000110000100
0010000100000000

```

- 在txt文本文件中显示

 result - 记事本

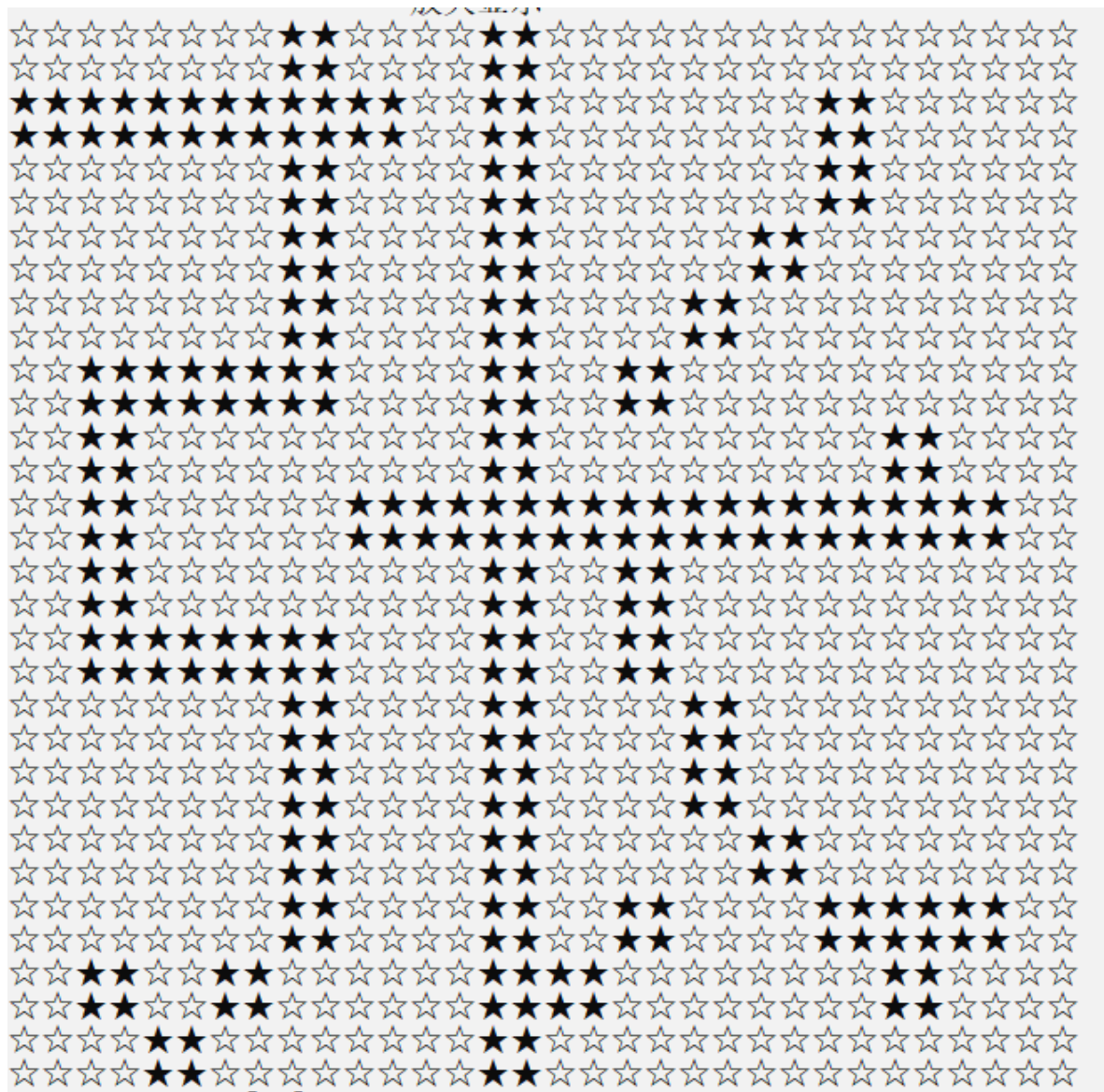
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```

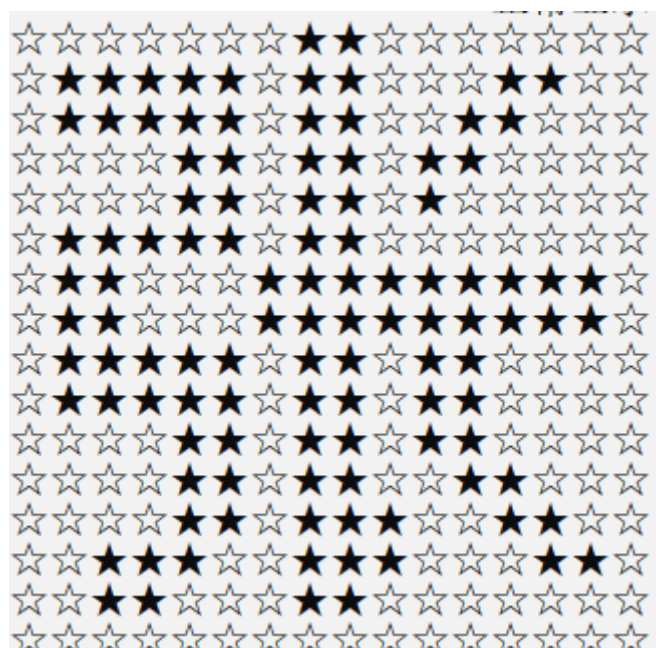
      1  1
1 1 1 1 1 1  1      1
      1  1      1
      1  1      1
      1  1  1
1 1 1 1  1  1
1      1      1
1      1 1 1 1 1 1 1 1 1
1      1  1
1 1 1 1  1  1
      1  1  1
      1  1  1
      1  1      1
      1  1  1  1 1 1
1  1      1 1      1
      1      1

```

- 放大显示



- 更改字体，例如黑体，可以看到和宋体有一定的区别



- 彩色显示

彩色显示