

汉字花样显示

3180102173 王子腾

实验描述

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。

原理介绍

- 区位码计算

GB2312 编码规定: 对任意一个图形字符都采用两个字节表示, 每个字节均采用七位编码表示, 其中高字节(第一个字节)称为区码, 低字节称为位码, 其编码的范围均为 0xA1~0xFE。GB2312 将编码分成 94 个区, 每个区有 94 位, 01~09 区为符号、数字区, 16~87 区为汉字区(0xB0~0xF7), 10~15 区、88~94 区是有待进一步标准化的空白区。GB2312 将收录的汉字分成两级: 第一级是常用汉字计 3755 个, 置于 16~55 区, 按汉语拼音字母 / 笔形顺序排列; 第二级汉字是次常用汉字计 3008 个, 置于 56~87 区, 按部首 / 笔画顺序排列。

因此区位码的计算方式为:

```
q = (hz[0] - 0xA0) & 255;  
w = (hz[1] - 0xA0) & 255;
```

- 字库偏移量计算

由于每一个汉字需要 $16 \times 16 = 256 \text{ bit} = 32 \text{ byte}$ 进行存储, 因此在国标码 GB2312-80 中偏移量的计算公式为:

$$\text{offset} = ((q-1) \times 94 + w-1) \times 32$$

- 16 * 16 点阵图绘制

- 1) 普通模式

```

case normal:
    for(int i = 0; i < 16; i++)
    {
        for(int j = 0; j < 8; j++)
        {
            if(hzcode[2*i] & (128 >> j))
                cout << '#';
            else cout << ' ';
        }
        for(int j = 0; j < 8; j++)
        {
            if(hzcode[2*i+1] & (128 >> j))
                cout << '#';
            else cout << ' ';
        }
        cout << endl;
    }
    return;

```

2) 倒立模式

```

case 1:
    for(int i = 15; i >= 0; i--)
    {
        for(int j = 0; j < 8; j++)
        {
            if(hzcode[2*i] & (0x80 >> j))
                cout << '#';
            else cout << ' ';
        }
        for(int j = 0; j < 8; j++)
        {
            if(hzcode[2*i+1] & (0x80 >> j))
                cout << '#';
            else cout << ' ';
        }
        cout << endl;
    }
    return;

```

3) 放大模式

```

case large:
    for(int i = 0; i < 16; i++)
    {
        for(int k = 0; k < 2; k++)
        {
            for(int j = 0; j < 8; j++)
            {
                if(hzcode[2*i] & (0x80 >> j))
                    cout << "###";
                else cout << "  ";
            }
            for(int j = 0; j < 8; j++)
            {
                if(hzcode[2*i+1] & (0x80 >> j))
                    cout << "##";
                else cout << "  ";
            }
            cout << endl;
        }
    }
    return;

```

4) 倾斜模式

```

case incline:
    for(int i = 0; i < 16; i++)
    {
        for(int j = 0; j < 16-i; j++)
            cout << ' ';
        for(int j = 0; j < 8; j++)
        {
            if(hzcode[2*i] & (0x80 >> j))
                cout << '#';
            else cout << ' ';
        }
        for(int j = 0; j < 8; j++)
        {
            if(hzcode[2*i+1] & (0x80 >> j))
                cout << '#';
            else cout << ' ';
        }
        cout << endl;
    }
    return;

```

5) 文件输出模式:

```

case file:
{
    ofstream File("HZOutput.txt");
    for(int i = 0; i < 16; i++)
    {
        for(int j = 0; j < 8; j++)
        {
            if(hzcode[2*i] & (128 >> j))
                File << '#';
            else File << ' ';
        }
        for(int j = 0; j < 8; j++)
        {
            if(hzcode[2*i+1] & (128 >> j))
                File << '#';
            else File << ' ';
        }
        File << endl;
    }
    return;
}

```

- 6) 颜色模式（修改输出字体颜色后执行 normal 模式）

```

case red:
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), FOREGROUND_RED);
    break;
case green:
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), FOREGROUND_GREEN);
    break;
case blue:
    SetConsoleTextAttribute(GetStdHandle(STD_OUTPUT_HANDLE), FOREGROUND_BLUE);
    break;

```

实验结果

```

enum{
    normal, reverse, large, incline, red, green, blue, file
};

```

样例驱动

```

for(int i = 0; i <= 7; i++)
{
    Display(Hzcode, i);
    cout << endl << endl;
}

```

输出界面：

[illegible]



 HZOutput - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
#
#  #  ##
## # ###
# # #
# #####
## # # #
# # #####
# # ## #
# ## # #
# ### # #
### # # #
# # # #
# # # #
# # # #
# # ## #
# # #
```