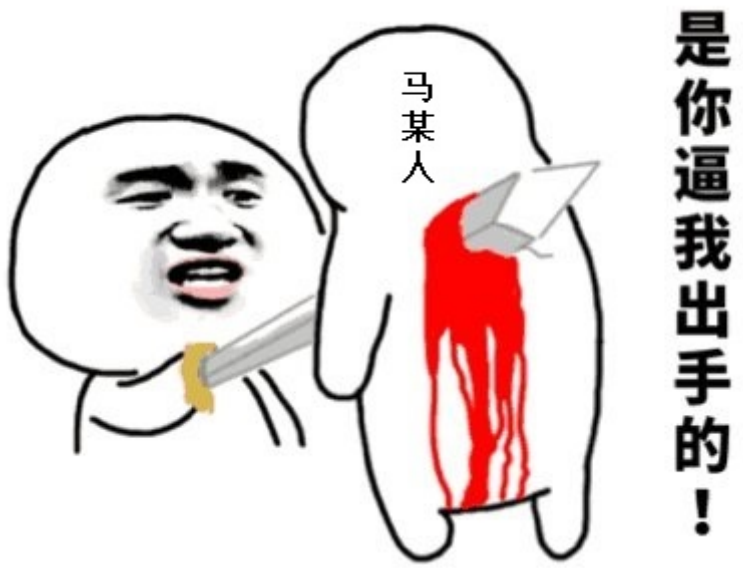
【奇牛学院 C/C++游戏辅助系列课程】

今晚课题：让“马某人”彻夜难眠的十五行代码！！

主讲老师: 九夏老师 开车时间：20:05 -- 22:00



道路千万条，安全第一条，行车不规范，亲人两行泪~！ *----电影：流浪地球*

交通规则：

1.**九夏老师**毕业于**秋名山老司机培训学校**，上课开车是常态。当然，翻车也属正常。

2.**既来之，则安之。**没有重要的事情就不要离开课堂。

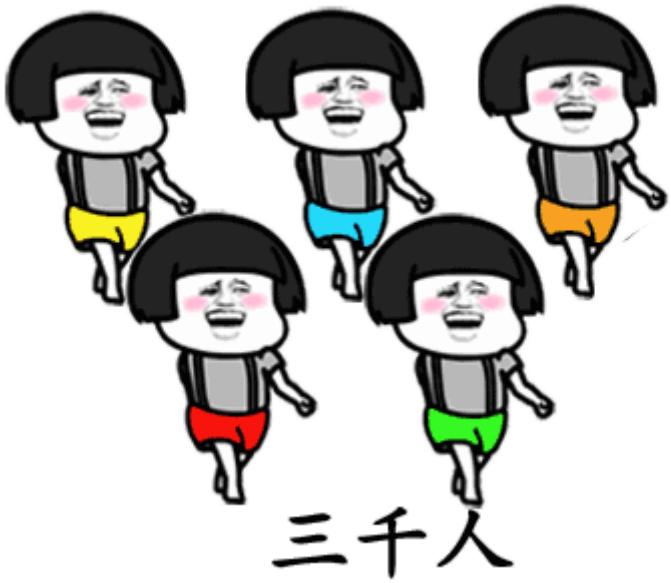
3.**为爱鼓掌。**啪啪啪~~！

不要和我说什么技术分析，老夫写代码就是：一把梭~！ *----网络表情包*

项目目标：

1. 编写代码，编译成一个exe。
2. 点击exe，实现一键“梭哈”腾讯游戏连连看。
3. 体验一次”老子天下第一“的优越感。（哈哈哈哈哈~）

来吧！ 展示~！



我们的眼睛就是自己的监狱，目光所及便是高墙。 *----哲学家：尼采*

涉及知识点：

1. C/C++ 基本语法。 基本数据类型 运算符 分支 循环 数组 指针 函数
2. Windows API接口函数。 C/C++ 必学！ mfc qt cocos2d unity3d UE
3. 第三方EasyX图形界面库。 新手准备的一个方便简单 图形图像库

第一站：基本概念补充

1. **使用windows函数需要加上头文件：windows.h**
2. **使用EasyX函数需要加上头文件：easyx.h**

```
#include <windows.h>
#include <easyx.h>

int main(){


    return 0;
}
```

3. **窗口句柄**：HWND　Handle of Window

windows系统用于**标记某一个窗口的ID号。**有了这个ID号，就可以对这个窗口 为所欲为。

```
HWND gameHwnd;  //定义一个用于保存游戏窗口ID的变量  gameHwnd
```

4. **绘图句柄**：HDC　Handle to Device Context

windows系统用于**标记某一个绘图设备的ID号。**有了这个ID号，就可以对这个绘图设备 为所欲为。

```
HDC gameHdc;  //定义一个用于保存游戏绘图设备ID的变量 gameHdc
HDC imageHdc;  //定义一个用于保存图片绘图设备ID的变量 imageHdc
```

第二站：一些新的函数，盘它：

```
//Windows 函数
HWND FindWindow(LPCSTR lpClassName, LPCSTR lpWindowName);
HDC GetDC(HWND hWnd);
BOOL BitBlt(HDC hdcDst, int x, int y, int cx, int cy, HDC hdcSrc, int x1, int
y1, DWORD rop);

//EasyX 函数
HDC GetImageHDC(IMAGE* pImg = NULL);
void SetWorkingImage(IMAGE* pImg = NULL);
void setorigin(int x, int y);
void getimage(IMAGE *pDstImg, int srcX, int srcY, int srcWidth, int srcHeight);
COLORREF getpixel(int x, int y);
```

1.查找窗口ID

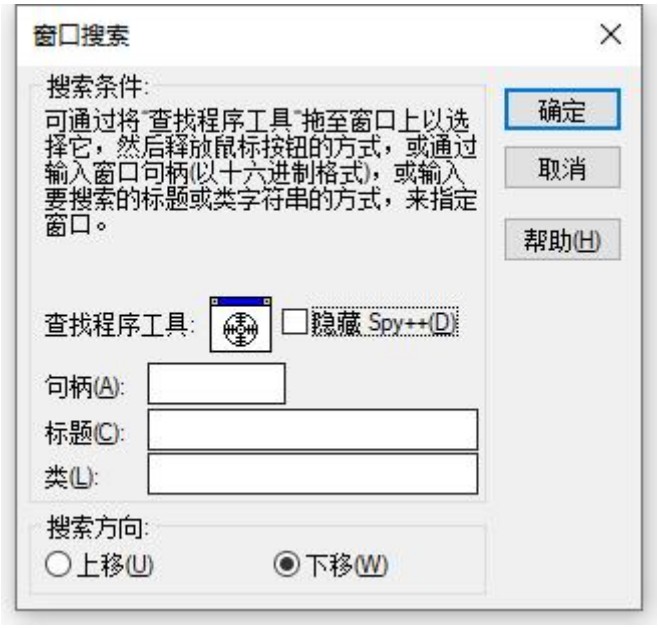
```
//通过FindWindow函数可以查找到windows系统中所有窗口的ID号。并返回。
HWND FindWindowA(LPCSTR lpClassName, LPCSTR lpWindowName);

//参数1：lpClassName     需要查找窗口的类名称 ，没有就写NULL
//参数2：lpWindowName    需要查找窗口的标题，没有就写NULL
```

示例：

```
hwnd = FindWindowA(NULL,"QQ游戏 - 连连看角色版");  //获取QQ游戏连连看的窗口ID
```

【补充】：窗口的类名称和标题，可以使用 `工具` ---> `spy++` ---> `搜索` ---> `查找窗口` 来获取。这货长这样：



2.获取绘图设备ID

```
//windows 通过窗口ID 获得该窗口的绘图设备ID
HDC GetDC(HWND hWnd);

//easyx 获取一张图片的绘图设备ID
HDC GetImageHDC(IMAGE* pImg = NULL);
```

示例:

```
gameHdc = GetDC(gameHwnd);          //获取游戏窗口的 图像
imageHdc = GetImageHDC(&image);     //获取image图片的 图像
```

3.在2个设备之间拷贝图像 dst目标 Src源

```
BOOL BitBlt(HDC hdcDst, int x, int y, int cx, int cy, HDC hdcSrc, int x1, int
y1, DWORD rop);

//参数1:      hdcDst    目标设备
//参数2、3:    x y       目标设备中 起始坐标
//参数4、5:    cx cy     目标设备中 绘图宽度 高度
//参数6:      hdcSrc    源设备
//参数7、8:    x1 y1     源设备中 起始坐标
//参数9:      rop       绘图方式  一般写拷贝 SRCCOPY
```

示例:

```
//将 游戏窗口的图像 拷贝到 image图片中
BitBlt(imageHdc, 0, 0, 800, 600, gameHdc, 0, 0, SRCCOPY);
```

4.设置当前工作区

```
void SetWorkingImage(IMAGE* pImg = NULL);
```

工作区：

easyx的默认工作区是窗口。如果要将工作区暂时调整为某一个image图片，使用此函数。

示例：

```
SetWorkingImage(&image);  //操作对象变为 image

//小黑屋  为所欲为~~!

SetWorkingImage(NULL);  //操作对象变为 默认工作区
```

图 1

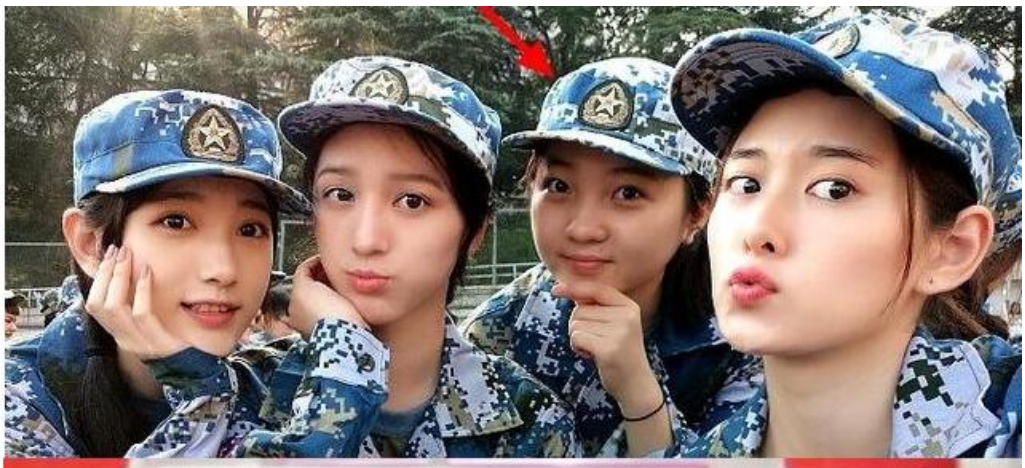


图 2

5.重置坐标原点 （当前工作区）

```
//将当前工作区的坐标原点 重置为 x y的位置
void setorigin(int x, int y);
```

6.获取部分图像 (当前工作区)

```
//从当前工作对象中获取部分图像： 起点srcX srcY  宽度高度：srcWidth srcHeight
void getimage(IMAGE *pDstImg, int srcX, int srcY, int srcWidth, int srcHeight);
```

7.获取像素点的颜色值 （当前工作区）

```
//从当前工作对象获取 x y 处的像素颜色值。
COLORREF getpixel(int x, int y);
```

功能1：判断一张图片是否是空图片

```
bool IsBlank(IMAGE* pimg)
{
    SetWorkingImage(pimg);

    for(int i=0;i<10;i++)
    {
        for(int j=0;j<10;j++)
        {
            if(getpixel(j,i)!=BACKCOLOR)
                return false;
        }
    }
    return true;
}
```

功能2：判断2张图片是否相同

```
bool IsSimilar(IMAGE* pimg1, IMAGE* pimg2)
{
    //10 10
    SetWorkingImage(pimg1);
    COLORREF color11 = getpixel(3, 3);
    COLORREF color12 = getpixel(3, 6);
    COLORREF color13 = getpixel(6, 3);
    COLORREF color14 = getpixel(6, 6);
    COLORREF color15 = getpixel(5, 5);
    SetWorkingImage(pimg2);
    COLORREF color21 = getpixel(3, 3);
    COLORREF color22 = getpixel(3, 6);
    COLORREF color23 = getpixel(6, 3);
    COLORREF color24 = getpixel(6, 6);
    COLORREF color25 = getpixel(5, 5);

    if (color11 == color21 && color12 == color22 && color13 == color23 &&
color14 == color24 && color15 == color25)
        return true;
    else
        return false;
}
```

功能3：将不同的图片 对应不同的数字 保存在二维数组中，并打印出来。

```
void ImageToMap()
{
    bool isfirst = true;
    int num = 1;
    int flag = 0;
    for (int i = 0; i < ROW; i++) {
        for (int j = 0; j < COL; j++)  //遍历img数组
        {
            if (IsBlank(&img[i][j]))   //如果是空图片 map[i][j]赋值为0
                map[i][j] = 0;
            else   //如果不是空图片
            {
                //第一张打个样
                if (isfirst)  //第一张非空图片
                {
                    map[i][j] = 1;  //第一张为1
                    isfirst = false;  //isfirst = false
                    continue;
                }
                for (int m = 0; m < ROW; m++)
                {
                    for (int n = 0; n < COL; n++)  //遍历所有图片
                    {
                        if (!IsBlank(&img[m][n]))  //不是空的图片
                        {
                            if (!(m == i && n == j))   //不能是自己
                            {
                                if (IsSimilar(&img[m][n], &img[i][j])) //2张图片相
同
                                {
                                    map[i][j] = map[m][n];  //新的数组元素=相同的那
一张
                                    flag = 1;  //赋值成功
                                    break;  //下一个
                                }
                            }
                        }
                        //只能遍历到g_img[i][j]的前一张
                        if ((j == 0 && m == i - 1 && n == 18) ||
                            (j != 0 && m == i && n == j - 1))
                        {
                            map[i][j] = ++num;     //map[i][j]新ID
                            flag = 1;  //赋值成功
                            break;  //下一个
                        }
                    }
                    if (flag)
                    {
                        flag = 0;
                        break;
                    }
                }
            }
```

```
        }
    }
}
```

第一节课完整代码：

```
#include <stdio.h>
#include <windows.h>
#include <easyx.h>
#pragma comment(lib,"Msimg32.lib")

#define OX 15       //游戏区的坐标
#define OY 182

#define ROW 11      //行和列
#define COL 19

#define WIDTH 800    //游戏窗口的大小
#define HEIGHT 600

#define SIZEX 31    //一个小图片的宽和高
#define SIZEY 35

#define BACKCOLOR   (RGB(48,76,112))

//判断一张图片是不是空图片 返回1  不是返回0
bool IsBlank(IMAGE* pimg)
{
    SetWorkingImage(pimg);
    for (int i = 0; i < 10; i++)  //最多100次循环   900+
    {
        for (int j = 0; j < 10; j++)
        {
            if (getpixel(i, j) != BACKCOLOR)
                return false;
        }
    }
    return true;
}

//判断2张图片是不是相同
bool IsSimilar(IMAGE* pimg1, IMAGE* pimg2)
{
    //刷一个小聪明
    SetWorkingImage(pimg1);
    COLORREF c11 = getpixel(3, 3);
    COLORREF c12 = getpixel(3, 6);
    COLORREF c13 = getpixel(6, 3);
    COLORREF c14 = getpixel(6, 6);
    COLORREF c15 = getpixel(5, 5);
    SetWorkingImage(pimg2);
    COLORREF c21 = getpixel(3, 3);
    COLORREF c22 = getpixel(3, 6);
    COLORREF c23 = getpixel(6, 3);
    COLORREF c24 = getpixel(6, 6);
```

```
    COLORREF c25 = getpixel(5, 5);
    if (c11 == c21 && c12 == c22 && c13 == c23 &&
        c14 == c24 && c15 == c25)
        return true;
    else
        return false;
}


int main()
{
    HWND gameHwnd; //定义一个用来保存游戏窗口的ID
    HDC  gameHdc;  //标记游戏的图像
    HDC  imageHdc; //标记一张图片的图像
    IMAGE image(800, 600); //空图片 宽度800 高度是600
    IMAGE img[ROW][COL];   //图片的数组  装所有的小图片

    //initgraph(800, 600);
    gameHwnd = FindWindowA(NULL,"QQ游戏 - 连连看角色版");

    gameHdc = GetDC(gameHwnd); //得到游戏的绘图设备
    imageHdc = GetImageHDC(&image);  //得到image的绘图设备

    BitBlt(imageHdc, 0, 0, 800, 600, gameHdc, 0, 0, SRCCOPY);
    //image里面就已经保存了游戏的图像!

    SetWorkingImage(&image); //接下来  搞事情 image

    setorigin(OX, OY);

    //从image 获取某一部分的图像
    for (int i = 0; i < ROW; i++)
    {
        for (int j = 0; j < COL; j++)
        {
            getimage(&img[i][j], j * SIZEX+10, i * SIZEY+10, 10, 10);
        }
    }
    SetWorkingImage(NULL);

    //将img[11][19]图片 转换为 整形的二维数组
    int map[ROW][COL] = {0};
    bool isfirst = true; //第一张
    int num = 1;
    bool flag = false;  //有没有搞定一个
    for (int i = 0; i < ROW; i++)
    {
        for (int j = 0; j < COL; j++) //遍历所有的图片  img[11][19]
        {
            //找到空白图片
            if (IsBlank(&img[i][j]))
            {
                map[i][j] = 0;
            }
            else  //如果不是空白图片
            {
                //是不是第一张?
                if (isfirst)  //第一张
```

```
                {
                    map[i][j] = num;  //1
                    isfirst = false;  //再也不会有第一张
                    continue;
                }

                //不是第一张 应该跟前面的所有图片进行一次比较!
                //比较对了有相同的
                //没有一张相同  新的数字
                for (int m = 0; m < ROW; m++) //遍历所有的小图片去比较
                {
                    for (int n = 0; n < COL; n++)
                    {
                        //非空图片
                        if (!IsBlank(&img[m][n]))
                        {
                            //不用跟自己比
                            if (!(m == i && n == j))
                            {
                                //找到一张相同
                                if (IsSimilar(&img[m][n], &img[i][j]))
                                {
                                    //相同的编号给他 完事了
                                    map[i][j] = map[m][n];
                                    flag = true;
                                    break;
                                }
                            }
                        }

                        //遍历到前一张
                        if (j == 0 && m == i - 1 && n == 18 ||
                            j != 0 && m == i && n == j - 1)
                        {
                            //新的编号
                            map[i][j] = ++num;
                            flag = true;
                            break;
                        }
                    }
                    if (flag)
                    {
                        flag = 0;
                        break;
                    }
                }
            }
        }
    }

    for(int i = 0; i < ROW; i++)
    {
        for(int j = 0; j < COL; j++)
        {
            if(map[i][j] == 0)
                printf("   ");
            else if(map[i][j] < 10)
                printf("0%d ", map[i][j]);
```

```
            else
                printf("%d ", map[i][j]);
        }
        printf("\n");
    }
    getchar();
    return 0;
}
```