//1.背景图像，人物图像，怪兽图像等铺设

#include <graphics.h>

#include <conio.h>

#pragma comment(lib,"MSIMG32.LIB") //实现透明贴图

//定义全局变量

#define Width 1024 //画面宽度

#define High 525 //画面高度

#define RoleWidth 42 //人物宽度

#define RoleHihg 75 //人物高度

#define MonsterWidth 128 //怪兽宽度

#define MonsterWidth 128 //怪兽高度

IMAGE img\_bk; //背景画面

IMAGE img\_role1, img\_role2, img\_role3, img\_role4, img\_role5, img\_role6, img\_role7, img\_role8, img\_role9, img\_role10; //人物图片

IMAGE\* p[10];//储存图片地址

void startup() //数据初始化

{

initgraph(Width, High);

loadimage(&img\_bk, \_T("C:\\Users\\Administrator\\Desktop\\Mario\\Debug\\Picture material\\背景图.png")); //背景图片导入

loadimage(&img\_role1,\_T("C:\\Users\\Administrator\\Desktop\\Mario\\Debug\\Picture material\\人物1.png"));//人物图像导入

loadimage(&img\_role2, \_T("C:\\Users\\Administrator\\Desktop\\Mario\\Debug\\Picture material\\人物2.png"));//人物图像导入

loadimage(&img\_role3, \_T("C:\\Users\\Administrator\\Desktop\\Mario\\Debug\\Picture material\\人物3.png"));//人物图像导入

loadimage(&img\_role4, \_T("C:\\Users\\Administrator\\Desktop\\Mario\\Debug\\Picture material\\人物4.png"));//人物图像导入

loadimage(&img\_role5, \_T("C:\\Users\\Administrator\\Desktop\\Mario\\Debug\\Picture material\\人物5.png"));//人物图像导入

loadimage(&img\_role6, \_T("C:\\Users\\Administrator\\Desktop\\Mario\\Debug\\Picture material\\人物6.png"));//人物图像导入

loadimage(&img\_role7, \_T("C:\\Users\\Administrator\\Desktop\\Mario\\Debug\\Picture material\\人物7.png"));//人物图像导入

loadimage(&img\_role8, \_T("C:\\Users\\Administrator\\Desktop\\Mario\\Debug\\Picture material\\人物8.png"));//人物图像导入

loadimage(&img\_role9, \_T("C:\\Users\\Administrator\\Desktop\\Mario\\Debug\\Picture material\\人物9.png"));//人物图像导入

loadimage(&img\_role10, \_T("C:\\Users\\Administrator\\Desktop\\Mario\\Debug\\Picture material\\人物10.png"));//人物图像导入

//loadimage(&img\_role2, \_T("C:\\Users\\Administrator\\Desktop\\Mario\\Debug\\Picture material\\掩码.png"));//人物掩码导入

BeginBatchDraw();

}

// 半透明贴图函数

// 参数：

// dstimg: 目标 IMAGE 对象指针。NULL 表示默认窗体

// x, y: 目标贴图位置

// srcimg: 源 IMAGE 对象指针。NULL 表示默认窗体

void transparentimage(IMAGE\* dstimg, int x, int y, IMAGE\* srcimg)

{

HDC dstDC = GetImageHDC(dstimg);

HDC srcDC = GetImageHDC(srcimg);

int w = srcimg->getwidth();

int h = srcimg->getheight();

// 结构体的第三个成员表示额外的透明度，0 表示全透明，255 表示不透明。

BLENDFUNCTION bf = { AC\_SRC\_OVER, 0, 255, AC\_SRC\_ALPHA };

// 使用 Windows GDI 函数实现半透明位图

AlphaBlend(dstDC, x,y , w, h, srcDC,0,0, w, h, bf);

}

void RoleWalk(void) //人物行走

{

//transparentimage(NULL, 0, 415, &img\_role1);

//putimage(0, 415, RoleWidth, RoleHihg, &img\_role1, i \* RoleWidth, 0, NOTSRCERASE);

//putimage(0, 415, RoleWidth, RoleHihg, &img\_role2, i \* RoleWidth, 0, SRCINVERT);

}

void show() //显示画面

{

putimage(0, 0,&img\_bk); //背景图显示

//transparentimage(NULL, 0, 415, &img\_role1);

//putimage(0, 415, RoleWidth, RoleHihg, &img\_role1, 0, 0,SRCAND); //人物图像显示

//putimage(0, 415, RoleWidth, RoleHihg, &img\_role2, 0, 0, SRCPAINT); // 人物掩码图

FlushBatchDraw();

}

void updateWithoutInput() //与输入无关的更新

{

}

void updateWithInput() //与输入有关的更新

{

/\*char input;

int i = 0;

if (\_kbhit())

{

input = \_getch();

if (input == 'd')

i++;

RoleWalk(i);

}\*/

RoleWalk();

}

int main(void)

{

startup();

while (1)

{

show();

updateWithoutInput();

updateWithInput();

Sleep(10);

}

return 0;

}