/

#include <graphics.h>

#include <conio.h>

#include<math.h>

#include "MyTimer.h" //一个保证精确延时的类。下载于easyx官网

#pragma comment(lib,"Winmm.lib") //给游戏添加音乐要用到它

#define G 9.8 //重力加速度

#define XSIZE 512 //屏幕大小

#define YSIZE 384

#define X 64 //主角起始位置

#define Y 192

#define W 32 //主角的宽和高

#define H 32

#define STEP 4 //主角走一步相距的像素个数

#define HIGH (2\*role.w+1) //主角跳跃的最大高度

#define CMD\_LEFT 1 //方向键的宏定义

#define CMD\_RIGHT 2

#define CMD\_UP 4

#define CMD\_DOWN 8

#define CMD\_SHOOT 16

#define CMD\_ESC 32

int life; //全局变量，主角共有多少条生命

int score; //全局变量，主角获得的分数

struct ROLE

{

int id;

int x;//横坐标

int y;//纵坐标

int w;//图片宽度

int h;//图片高度

int xleft;//水平运动的左界限

int xright;//水平运动的右界限

int turn;//精灵的运动方向

int jump;//精灵是否跳跃

int iframe;//加载第几副精灵图，这样就能让精灵看上去动起来了

};

class MAP //储存地图的结构体

{

public:

int id;

int x;

int y;

};

class BULLET //子弹的结构体

{

public:

int x;

int y;

int turn;

int iframe;

int id;

};

class COINT //硬币的结构体

{

public:

int x;

int y;

double iframe;

};

class ENEMY //敌人的结构体

{

public:

int id;

int x;

int y;

int turn;

int iframe;

};

class game //整个游戏只设置了这一个类

{

private:

ROLE role;

MAP map[350];

BULLET bullet[20];

COINT coint[50];

ENEMY enemy[20];

IMAGE img\_mapsky,img\_p,img\_map,img\_ani,img\_mapbk,img\_home;

int xmapsky; //背景天空的起始横坐标

int xmap; //地图的起始坐标

double v0; //精灵跳跃的初速度

double h; //精灵跳跃的高度

double t; //精灵跳跃的时间

int ibullet; //第几颗子弹

int xbullet; //子弹的x坐标

int ybullet; //子弹的y坐标

int get\_bullet; //是否获得武器，0表示没有获得，1表示已获得

POINT icoint; //储存硬币的坐标

POINT bomb[20]; //储存哪些地方爆炸了的坐标

POINT temp; //临时坐标。储存哪些地方爆炸了的坐标

double score\_frame; //下面3个double型的变量用于控制各自图片的帧，以实现动画的效果。如画面中的流水

double bomb\_frame;

double mapbk\_frame;

int win; //玩家是否过关

int pause; //玩家是否按Esc（暂停键）

public:

game();

~game();

void start(); //处理游戏开始的界面，和按暂停键后的界面

void init(); //初始化各项变量

void move(); //控制主角移动

void show(); //显示画面

int isdie(); //判断主角是否已死

int GetCommand(); // 获取控制命令。参阅easyx

void left(); //主角向左运动

void right(); //主角向右运动

void up(); //主角跳跃

void init\_shoot(); //初始化发射子弹

void fall(); //主角自由落体或者向上跳跃

int is\_l\_touch(int id);//主角的左边是否碰到墙或敌人，以及敌人是否碰到陆地的左边界

int is\_r\_touch(int id);//主角的右边是否碰到墙或敌人，以及敌人是否碰到陆地的右边界

int is\_t\_touch(); //主角的头是否碰到墙

int is\_b\_touch(int id);//主角是否踩到敌人。

int is\_touch(); //主角是否吃到金币

int is\_land(ENEMY e); //敌人是否站在陆地上

void getbullet(); //获取子弹

void shoot(); //发射子弹

int eat(BULLET b); //子弹是否打到敌人或者墙壁

void end(); //处理游戏结束

};

game::game()

{

initgraph(XSIZE,YSIZE);

setbkcolor(WHITE);

cleardevice();

}

game::~game()

{

closegraph();

}

void game::start()

{

if(pause==1)//如果按了暂停键

{

BeginBatchDraw();

int points[8]={XSIZE/2-45,YSIZE/3,XSIZE/2+45,YSIZE/3,XSIZE/2+45,YSIZE/3+90,XSIZE/2-45,YSIZE/3+90};

setfillstyle(GREEN);

fillpoly(4, points);

setbkmode(TRANSPARENT);

setfont(20,0,"黑体");

RECT r2={XSIZE/2-45,YSIZE/3,XSIZE/2+45,YSIZE/3+30};rectangle(XSIZE/2-45,YSIZE/3,XSIZE/2+45,YSIZE/3+30);

drawtext("回到游戏", &r2, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

RECT r3={XSIZE/2-45,YSIZE/3+30,XSIZE/2+45,YSIZE/3+60};rectangle(XSIZE/2-45,YSIZE/3+30,XSIZE/2+45,YSIZE/3+60);

drawtext("重新开始", &r3, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

RECT r4={XSIZE/2-45,YSIZE/3+60,XSIZE/2+45,YSIZE/3+90};rectangle(XSIZE/2-45,YSIZE/3+60,XSIZE/2+45,YSIZE/3+90);

drawtext(" 主 菜 单 ", &r4, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

FlushBatchDraw();

MOUSEMSG m;

while(true)

{

BeginBatchDraw();

m=GetMouseMsg();

switch(m.uMsg)

{

case WM\_LBUTTONDOWN:

EndBatchDraw();

if(m.x>XSIZE/2-45&&m.x<XSIZE/2+45&&m.y>YSIZE/3&&m.y<YSIZE/3+30)

return;

else if(m.x>XSIZE/2-45&&m.x<XSIZE/2+45&&m.y>YSIZE/3+30&&m.y<YSIZE/3+60)

{

mciSendString("close all", NULL, 0, NULL);

pause=0;

score=0;

return;

}

else if(m.x>XSIZE/2-45&&m.x<XSIZE/2+45&&m.y>YSIZE/3+60&&m.y<YSIZE/3+90)

{

mciSendString("close all", NULL, 0, NULL);

pause=0;

score=0;

life=0;

cleardevice();

break;

}

else

break;

case WM\_MOUSEMOVE:

RECT r;

int i;

for(i=0;i<3;i++)

{

if(m.x>XSIZE/2-45&&m.x<XSIZE/2+45&&m.y>YSIZE/3+i\*30&&m.y<YSIZE/3+30+i\*30)

{

r.left=XSIZE/2-45;

r.top=YSIZE/3+i\*30;

r.right=XSIZE/2+45;

r.bottom=YSIZE/3+30+i\*30;

int points[8]={r.left,r.top,r.right,r.top,r.right,r.bottom,r.left,r.bottom};

setfillstyle(RED);

fillpoly(4, points);

setbkmode(TRANSPARENT);

switch(i)

{

case 0:

drawtext("回到游戏", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

case 1:

drawtext("重新开始", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

case 2:

drawtext(" 主 菜 单 ", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

}

}

else

{

if(getpixel(XSIZE/2-45+1,YSIZE/3+i\*30+1)==RED)

{

r.left=XSIZE/2-45;

r.top=YSIZE/3+i\*30;

r.right=XSIZE/2+45;

r.bottom=YSIZE/3+30+i\*30;

int points[8]={r.left,r.top,r.right,r.top,r.right,r.bottom,r.left,r.bottom};

setfillstyle(GREEN);

fillpoly(4, points);

setbkmode(TRANSPARENT);

switch(i)

{

case 0:

drawtext("回到游戏", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

case 1:

drawtext("重新开始", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

case 2:

drawtext(" 主 菜 单 ", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

}

}

FlushBatchDraw();

}

}

}

if(pause==0)

break;

}

}

if(life==1||life==2)

return;

life=3;

score=0;

settextstyle(40,0,"黑体");

settextcolor(RGB(106,181,255));

RECT r1 = {0, 0, XSIZE, YSIZE/3};

drawtext("超级关东煮", &r1, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);//文字水平垂直居中并显示在一行

settextstyle(20,0,"楷体");

settextcolor(RGB(134, 255, 150));

setlinecolor(RGB(255, 250, 134));

RECT r2={XSIZE/2-45,YSIZE/3,XSIZE/2+45,YSIZE/3+30};

rectangle(XSIZE/2-45,YSIZE/3,XSIZE/2+45,YSIZE/3+30);

drawtext("开始游戏", &r2, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

RECT r3={XSIZE/2-45,YSIZE/3+30,XSIZE/2+45,YSIZE/3+60};

rectangle(XSIZE/2-45,YSIZE/3+30,XSIZE/2+45,YSIZE/3+60);

drawtext("游戏介绍", &r3, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

RECT r4={XSIZE/2-45,YSIZE/3+60,XSIZE/2+45,YSIZE/3+90};

rectangle(XSIZE/2-45,YSIZE/3+60,XSIZE/2+45,YSIZE/3+90);

drawtext("操作说明", &r4, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

RECT r5={XSIZE/2-45,YSIZE/3+90,XSIZE/2+45,YSIZE/3+120};

rectangle(XSIZE/2-45,YSIZE/3+90,XSIZE/2+45,YSIZE/3+120);

drawtext("退出游戏", &r5, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

int flag1=1,flag2=0,flag3=0;

MOUSEMSG m;

while(flag1==1)

{

BeginBatchDraw();//开始批量绘制

m=GetMouseMsg();//获得鼠标信息

switch(m.uMsg)//匹配鼠标信息

{

case WM\_LBUTTONDOWN://左键点击

EndBatchDraw();//停止批量绘制

if(m.x>XSIZE/2-45&&m.x<XSIZE/2+45&&m.y>YSIZE/3&&m.y<YSIZE/3+30&&flag1==1&&flag2==0&&flag3==0)//开始游戏

{

flag1=0;

break;

}

else if(m.x>XSIZE/2-45&&m.x<XSIZE/2+45&&m.y>YSIZE/3+30&&m.y<YSIZE/3+60&&flag1==1&&flag3==0)//游戏介绍

{

flag2=1;

cleardevice();

rectangle(50,50,213,220);

outtextxy(52,52,"游戏介绍：");

outtextxy(52,82,"超级关东煮");

outtextxy(52,102,"开发者：");

outtextxy(52,132,"关东组");

RECT R1={XSIZE-46,YSIZE-26,XSIZE-2,YSIZE-2};

rectangle(XSIZE-46,YSIZE-26,XSIZE-2,YSIZE-2);

drawtext("返回", &R1, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

}

else if(m.x>XSIZE/2-45&&m.x<XSIZE/2+45&&m.y>YSIZE/3+60&&m.y<YSIZE/3+90&&flag1==1&&flag2==0)//操作说明

{

flag3=1;

cleardevice();

rectangle(50,50,213,220);

outtextxy(52,52,"操作说明：");

outtextxy(52,72,"左移：A键");

outtextxy(52,92,"右移：D键");

outtextxy(52,112,"发射：J键");

outtextxy(52,132,"跳跃：W键/K键");

outtextxy(52,152,"暂停：Esc键");

RECT R2={XSIZE-46,YSIZE-26,XSIZE-2,YSIZE-2};

rectangle(XSIZE-46,YSIZE-26,XSIZE-2,YSIZE-2);

drawtext("返回", &R2, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

}

else if(m.x>XSIZE/2-45&&m.x<XSIZE/2+45&&m.y>YSIZE/3+90&&m.y<YSIZE/3+120&&flag1==1&&flag2==0&&flag3==0)//退出游戏

exit(0);

else if(m.x>XSIZE-46&&m.x<XSIZE-3&&m.y>YSIZE-26&&m.y<YSIZE-3&&(flag2==1||flag3==1))

{

cleardevice();

flag1=0,flag2=0,flag3=0;

start();

}

else

break;

case WM\_MOUSEMOVE:

RECT r;

if(flag2==1||flag3==1)

{

if(m.x>XSIZE-46&&m.x<XSIZE-3&&m.y>YSIZE-26&&m.y<YSIZE-3)

{

r.left=XSIZE-46;

r.top=YSIZE-26;

r.right=XSIZE-2;

r.bottom=YSIZE-2;

int points[8]={r.left,r.top,r.right,r.top,r.right,r.bottom,r.left,r.bottom};

setfillstyle(RED);

fillpoly(4, points);

setbkmode(TRANSPARENT);

drawtext("返回", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

}

else

{

if(getpixel(XSIZE-46+1,YSIZE-26+1)==RED)

{

r.left=XSIZE-46;

r.top=YSIZE-26;

r.right=XSIZE-2;

r.bottom=YSIZE-2;

int points[8]={r.left,r.top,r.right,r.top,r.right,r.bottom,r.left,r.bottom};

setfillstyle(BLACK);

fillpoly(4, points);

setbkmode(TRANSPARENT);

drawtext("返回", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

}

}

}

else

{

for(int i=0;i<4;i++)

{

if(m.x>XSIZE/2-45&&m.x<XSIZE/2+45&&m.y>YSIZE/3+i\*30&&m.y<YSIZE/3+30+i\*30)

{

r.left=XSIZE/2-45;

r.top=YSIZE/3+i\*30;

r.right=XSIZE/2+45;

r.bottom=YSIZE/3+30+i\*30;

int points[8]={r.left,r.top,r.right,r.top,r.right,r.bottom,r.left,r.bottom};

setfillstyle(RED);

fillpoly(4, points);

setbkmode(TRANSPARENT);

switch(i)

{

case 0:

drawtext("开始游戏", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

case 1:

drawtext("游戏介绍", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

case 2:

drawtext("操作说明", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

case 3:

drawtext("退出游戏", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

}

}

else

{

if(getpixel(XSIZE/2-45+1,YSIZE/3+i\*30+1)==RED)

{

r.left=XSIZE/2-45;

r.top=YSIZE/3+i\*30;

r.right=XSIZE/2+45;

r.bottom=YSIZE/3+30+i\*30;

int points[8]={r.left,r.top,r.right,r.top,r.right,r.bottom,r.left,r.bottom};

setfillstyle(BLACK);

fillpoly(4, points);

setbkmode(TRANSPARENT);

switch(i)

{

case 0:

drawtext("开始游戏", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

case 1:

drawtext("游戏介绍", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

case 2:

drawtext("操作说明", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

case 3:

drawtext("退出游戏", &r, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

break;

}

}

}

}

}

FlushBatchDraw();//刷新绘制图像

break;

default:

break;

}

}

}

void game::init()

{

if(pause==1)

return;

role.id=1;

role.x=X;

role.y=Y;

role.w=W;

role.h=H;

role.xleft=0;

role.xright=role.w\*6+STEP;

role.iframe=1;

role.turn=1;

role.jump=0;

xmapsky=0;

xmap=0;

v0=0;

h=0;

t=0;

ibullet=-1;

icoint.x=-1;

icoint.y=-1;

score\_frame=0;

bomb\_frame=1;

mapbk\_frame=1;

temp.x=-1;

temp.y=-1;

xbullet=41\*role.w-10;

ybullet=4\*role.h-25;

get\_bullet=0;

win=0;

pause=0;

score=0;

int i;

for(i=0;i<350;i++)

{

if (i < 50)

{

coint[i].x = -1;

coint[i].y = -1;

coint[i].iframe = 1;

}

if (i < 20)

{

bullet[i].id = 0;

bullet[i].x = -1;

bullet[i].y = -1;

bullet[i].iframe = 1;

bullet[i].turn = -1;

enemy[i].id = 0;

enemy[i].x = -1;

enemy[i].y = -1;

enemy[i].turn = 1;

enemy[i].iframe = 1;

bomb[i].x = -1;

bomb[i].y = -1;

}

map[i].id=0;

map[i].x=-1;

map[i].y=-1;

}

loadimage(&img\_mapsky,"res\\mapsky.bmp",XSIZE,YSIZE\*4);//加载天空

loadimage(&img\_p,"res\\role.bmp");//加载人物

loadimage(&img\_map,"res\\map.bmp");//加载地图障碍

loadimage(&img\_ani,"res\\ani.bmp");//加载怪物

loadimage(&img\_mapbk,"res\\mapbk.bmp");//加载最后场景

loadimage(&img\_home,"res\\home.bmp",XSIZE,YSIZE\*5);//加载游戏结束画面

//加载音乐

mciSendString("open 背景音乐.mp3 alias mymusic1", NULL, 0, NULL);

mciSendString("open 子弹.mp3 alias mymusic2", NULL, 0, NULL);

mciSendString("open 金币.mp3 alias mymusic3", NULL, 0, NULL);

mciSendString("open 跳.mp3 alias mymusic4", NULL, 0, NULL);

mciSendString("open 子弹打到敌人.mp3 alias mymusic5", NULL, 0, NULL);

mciSendString("open 子弹撞墙.mp3 alias mymusic6", NULL, 0, NULL);

mciSendString("open 踩敌人.mp3 alias mymusic7", NULL, 0, NULL);

mciSendString("open 吃到武器.mp3 alias mymusic8", NULL, 0, NULL);

mciSendString("open 胜利.mp3 alias mymusic9", NULL, 0, NULL);

mciSendString("open 死亡1.mp3 alias mymusic10", NULL, 0, NULL);

mciSendString("open 死亡2.mp3 alias mymusic11", NULL, 0, NULL);

for(i=0;i<300;i++) //以下都是编辑地图

{

map[i].id=1;

map[i].x=i%100\*role.w;

if(i<100)

map[i].y=9\*role.h;

else if(i>=100&&i<200)

map[i].y=10\*role.h;

else

map[i].y=11\*role.h;

}

map[15].id=1,map[15].x=18\*role.w,map[15].y=8\*role.h;

map[115].id=1,map[115].x=19\*role.w,map[115].y=8\*role.h;

map[215].id=1,map[215].x=20\*role.w,map[215].y=8\*role.h;

map[16].id=1,map[16].x=21\*role.w,map[16].y=8\*role.h;

map[116].id=1,map[116].x=22\*role.w,map[116].y=8\*role.h;

map[216].id=1,map[216].x=23\*role.w,map[216].y=8\*role.h;

map[17].id=1,map[17].x=24\*role.w,map[17].y=8\*role.h;

map[117].id=1,map[117].x=25\*role.w,map[117].y=8\*role.h;

map[217].id=1,map[217].x=26\*role.w,map[217].y=8\*role.h;

map[300].id=2,map[300].x=10\*role.w,map[300].y=6\*role.h;

map[301].id=2,map[301].x=11\*role.w,map[301].y=6\*role.h;

map[302].id=2,map[302].x=12\*role.w,map[302].y=6\*role.h;

map[303].id=3,map[303].x=36\*role.w,map[303].y=7\*role.h;

map[304].id=3,map[304].x=44\*role.w,map[304].y=7\*role.h;

map[305].id=2,map[305].x=40\*role.w,map[305].y=4\*role.h;

map[306].id=2,map[306].x=41\*role.w,map[306].y=4\*role.h;

map[307].id=2,map[307].x=42\*role.w,map[307].y=4\*role.h;

map[308].id=2,map[308].x=13\*role.w,map[308].y=6\*role.h;

map[309].id=4,map[309].x=15\*role.w,map[309].y=10\*role.h;

map[310].id=5,map[310].x=19\*role.w,map[310].y=6\*role.h;

map[311].id=5,map[311].x=23\*role.w,map[311].y=6\*role.h;

map[312].id=5,map[312].x=32\*role.w,map[312].y=7\*role.h;

map[313].id=5,map[313].x=48\*role.w,map[313].y=7\*role.h;

map[314].id=5,map[314].x=52\*role.w,map[314].y=7\*role.h;

map[315].id=5,map[315].x=56\*role.w,map[315].y=7\*role.h;

map[316].id=3,map[316].x=80\*role.w,map[316].y=7\*role.h;

map[317].id=3,map[317].x=90\*role.w,map[317].y=7\*role.h;

map[318].id=2,map[318].x=62\*role.w,map[318].y=6\*role.h;

map[319].id=2,map[319].x=65\*role.w,map[319].y=3\*role.h;

map[320].id=2,map[320].x=66\*role.w,map[320].y=3\*role.h;

map[321].id=2,map[321].x=67\*role.w,map[321].y=3\*role.h;

map[322].id=2,map[322].x=68\*role.w,map[322].y=3\*role.h;

map[323].id=2,map[323].x=69\*role.w,map[323].y=3\*role.h;

map[349].id=6,map[349].x=97\*role.w,map[349].y=7\*role.h;

for(i=64;i<300;i+=100)

{

map[i].id=0;map[i].x=-1;map[i].y=-1;

map[i+1].id=0;map[i+1].x=-1;map[i+1].y=-1;

map[i+2].id=0;map[i+2].x=-1;map[i+2].y=-1;

map[i+7].id=0;map[i].x=-1;map[i].y=-1;

map[i+8].id=0;map[i+1].x=-1;map[i+1].y=-1;

map[i+9].id=0;map[i+1].x=-1;map[i+1].y=-1;

map[i+11].id=0;map[i].x=-1;map[i].y=-1;

map[i+12].id=0;map[i+1].x=-1;map[i+1].y=-1;

map[i+13].id=0;map[i+1].x=-1;map[i+1].y=-1;

}

map[64].id=4,map[64].x=64\*role.w,map[64].y=10\*role.h;

map[71].id=4,map[71].x=71\*role.w,map[71].y=10\*role.h;

map[75].id=4,map[75].x=75\*role.w,map[75].y=10\*role.h;

enemy[0].id=1;enemy[0].x=6\*role.w;enemy[0].y=8\*role.h;enemy[0].turn=1;enemy[0].iframe=1;

enemy[1].id=1;enemy[1].x=8\*role.w;enemy[1].y=8\*role.h;enemy[1].turn=1;enemy[1].iframe=1;

enemy[2].id=1;enemy[2].x=27\*role.w;enemy[2].y=8\*role.h;enemy[2].turn=1;enemy[2].iframe=1;

enemy[3].id=1;enemy[3].x=29\*role.w;enemy[3].y=8\*role.h;enemy[3].turn=1;enemy[3].iframe=1;

enemy[4].id=1;enemy[4].x=31\*role.w;enemy[4].y=8\*role.h;enemy[4].turn=1;enemy[4].iframe=1;

enemy[5].id=1;enemy[5].x=33\*role.w;enemy[5].y=8\*role.h;enemy[5].turn=1;enemy[5].iframe=1;

enemy[6].id=1;enemy[6].x=35\*role.w;enemy[6].y=8\*role.h;enemy[6].turn=1;enemy[6].iframe=1;

enemy[7].id=1;enemy[7].x=40\*role.w;enemy[7].y=8\*role.h;enemy[7].turn=1;enemy[7].iframe=1;

enemy[8].id=1;enemy[8].x=82\*role.w;enemy[8].y=8\*role.h;enemy[8].turn=1;enemy[8].iframe=1;

enemy[9].id=1;enemy[9].x=65\*role.w;enemy[9].y=2\*role.h;enemy[9].turn=1;enemy[9].iframe=1;

enemy[10].id=1;enemy[10].x=69\*role.w;enemy[10].y=2\*role.h;enemy[10].turn=1;enemy[10].iframe=1;

enemy[11].id=1;enemy[11].x=85\*role.w;enemy[11].y=8\*role.h;enemy[11].turn=1;enemy[11].iframe=1;

for(i=0;i<4;i++)

{

coint[i].x=(10+i)\*role.w;

coint[i].y=5\*role.h;

coint[i+4].x=(67+i)\*role.w;

coint[i+4].y=8\*role.w;

coint[i+8].x=74\*role.w;

coint[i+8].y=(4+i)\*role.w;

}

for(i=12;i<18;i++)

{

coint[i].x=(83-12+i)\*role.w;

coint[i].y=6\*role.h;

coint[i+6].x=(83-12+i)\*role.w;

coint[i+6].y=7\*role.w;

}

}

void game::move()

{

MyTimer tt;

int c;

int k=0; //控制发射子弹的频率和敌人的移动速度

int n=0; //控制发射子弹的频率

while(true)

{

tt.Sleep(25);

t=sqrt(2\*HIGH/G)/14;

k++;

if(k==1000)

k=0;

if(kbhit()&&win==0)

{

c=GetCommand();

if(c&CMD\_LEFT)

left();

if(c&CMD\_RIGHT)

right();

if((c&CMD\_UP)&&role.jump==0)

up();

if(c&CMD\_ESC)

{

pause=1;

break;

}

if(c&CMD\_SHOOT&&get\_bullet==1)

{

if(n==0)

{

init\_shoot();

n=1;

}

n++;

if(k%10==0&&n>10)

{

init\_shoot();

}

}

else

n=0;

}

if(-xmap+role.x==97\*role.w)

{

mciSendString("stop mymusic1", NULL, 0, NULL);

mciSendString("play mymusic9", NULL, 0, NULL);

}

if(-xmap+role.x>95\*role.w)

{

win=1;

role.x+=STEP;

if(role.x-STEP>XSIZE)

break;

}

if(is\_b\_touch(1)==0)

role.jump=1;

if(role.jump==1)

fall();

if(isdie()==1)

{

mciSendString("stop mymusic1", NULL, 0, NULL);

mciSendString("play mymusic11", NULL, 0, NULL);

life--;

return;

}

if(k%2==0) //敌人的运动

{

for(int i=0;i<20;i++)

{

if(enemy[i].id==1)

{

if(is\_land(enemy[i])==1)

{

if(enemy[i].turn==1)

enemy[i].x+=STEP;

else

enemy[i].x-=STEP;

}

if(is\_land(enemy[i])==0||is\_l\_touch(3)==1||is\_r\_touch(3)==1)

{

if(enemy[i].turn==1)

enemy[i].x-=STEP;

else

enemy[i].x+=STEP;

enemy[i].turn\*=-1;

}

enemy[i].iframe\*=-1;

}

}

}

int boom=0;

if(is\_b\_touch(2)==1) //如果主角“踩到”敌人

boom=1;

getbullet(); //获取子弹

if(get\_bullet==1)

shoot();

BeginBatchDraw();

show();

FlushBatchDraw();

if((is\_l\_touch(2)==1||is\_r\_touch(2)==1))

{

mciSendString("stop mymusic1", NULL, 0, NULL);

mciSendString("play mymusic10", NULL, 0, NULL);

life--;

pause=0;

putimage(role.x,role.y,role.w,role.h,&img\_p,2\*role.w,role.h,SRCAND);

putimage(role.x,role.y,role.w,role.h,&img\_p,2\*role.w,0,SRCPAINT);

return;

}

}

}

void game::show()

{

if(xmapsky==-XSIZE)

xmapsky=0;

putimage(xmapsky,0,&img\_mapsky); //显示背景

putimage(XSIZE+xmapsky,0,&img\_mapsky);

if(is\_touch()==1)

score\_frame=1;

if(score\_frame!=0) //碰到硬币，显示得分

{

switch((int)score\_frame)

{

case 1:

putimage(xmap+icoint.x,icoint.y,role.w,role.h,&img\_ani,0,11\*role.h,SRCAND);

putimage(xmap+icoint.x,icoint.y,role.w,role.h,&img\_ani,0,10\*role.h,SRCPAINT);

break;

case 2:

putimage(xmap+icoint.x,icoint.y,role.w,role.h,&img\_ani,role.w,11\*role.h,SRCAND);

putimage(xmap+icoint.x,icoint.y,role.w,role.h,&img\_ani,role.w,10\*role.h,SRCPAINT);

break;

case 3:

putimage(xmap+icoint.x,icoint.y,role.w,role.h,&img\_ani,2\*role.w,11\*role.h,SRCAND);

putimage(xmap+icoint.x,icoint.y,role.w,role.h,&img\_ani,2\*role.w,10\*role.h,SRCPAINT);

break;

case 4:

putimage(xmap+icoint.x,icoint.y,role.w,role.h,&img\_ani,3\*role.w,11\*role.h,SRCAND);

putimage(xmap+icoint.x,icoint.y,role.w,role.h,&img\_ani,3\*role.w,10\*role.h,SRCPAINT);

break;

default:

break;

}

score\_frame+=0.2;

if(score\_frame==5)

score\_frame=0;

}

int i;

for(i=0;i<350;i++) //显示地图，天空上的地图和硬币

{

if(map[i].id==1)

{

putimage(xmap+map[i].x,map[i].y,role.w,role.h,&img\_map,0,0);

}

else if(map[i].id==2)

{

putimage(xmap+map[i].x,map[i].y,role.w,role.h,&img\_map,0,role.h);

}

else if(map[i].id==3)

{

putimage(xmap+map[i].x,map[i].y,2\*role.w,2\*role.h,&img\_map,0,9\*role.h);

}

else

{

if(map[i].id==4)

{

switch((int)mapbk\_frame)

{

case 1:

putimage(xmap+map[i].x,map[i].y,3\*role.w,2\*role.h,&img\_mapbk,0,10\*role.h,SRCAND);

putimage(xmap+map[i].x,map[i].y,3\*role.w,2\*role.h,&img\_mapbk,0,8\*role.h,SRCPAINT);

break;

case 2:

putimage(xmap+map[i].x,map[i].y,3\*role.w,2\*role.h,&img\_mapbk,3\*role.w,10\*role.h,SRCAND);

putimage(xmap+map[i].x,map[i].y,3\*role.w,2\*role.h,&img\_mapbk,3\*role.w,8\*role.h,SRCPAINT);

break;

default:

break;

}

}

else if(map[i].id==5)

{

switch((int)mapbk\_frame)

{

case 1:

putimage(xmap+map[i].x,map[i].y,3\*role.w,2\*role.h,&img\_mapbk,0,2\*role.h,SRCAND);

putimage(xmap+map[i].x,map[i].y,3\*role.w,2\*role.h,&img\_mapbk,0,0,SRCPAINT);

break;

case 2:

putimage(xmap+map[i].x,map[i].y,3\*role.w,2\*role.h,&img\_mapbk,3\*role.w,2\*role.h,SRCAND);

putimage(xmap+map[i].x,map[i].y,3\*role.w,2\*role.h,&img\_mapbk,3\*role.w,0,SRCPAINT);

break;

default:

break;

}

}

else if(map[i].id==6)

{

switch((int)mapbk\_frame)

{

case 1:

putimage(xmap+map[i].x,map[i].y,3\*role.w,2\*role.h,&img\_mapbk,0,6\*role.h,SRCAND);

putimage(xmap+map[i].x,map[i].y,3\*role.w,2\*role.h,&img\_mapbk,0,4\*role.h,SRCPAINT);

break;

case 2:

putimage(xmap+map[i].x,map[i].y,3\*role.w,2\*role.h,&img\_mapbk,3\*role.w,6\*role.h,SRCAND);

putimage(xmap+map[i].x,map[i].y,3\*role.w,2\*role.h,&img\_mapbk,3\*role.w,4\*role.h,SRCPAINT);

break;

default:

break;

}

}

mapbk\_frame+=0.003;

if(mapbk\_frame>2.9)

{

mapbk\_frame=1;

}

}

if(i<50)

{

if(coint[i].x!=-1||coint[i].y!=-1)

{

switch((int)coint[i].iframe)

{

case 1:

putimage(xmap+coint[i].x,coint[i].y,role.w,role.h,&img\_ani,0,9\*role.h,SRCAND);

putimage(xmap+coint[i].x,coint[i].y,role.w,role.h,&img\_ani,0,8\*role.h,SRCPAINT);

break;

case 2:

putimage(xmap+coint[i].x,coint[i].y,role.w,role.h,&img\_ani,role.w,9\*role.h,SRCAND);

putimage(xmap+coint[i].x,coint[i].y,role.w,role.h,&img\_ani,role.w,8\*role.h,SRCPAINT);

break;

case 3:

putimage(xmap+coint[i].x,coint[i].y,role.w,role.h,&img\_ani,2\*role.w,9\*role.h,SRCAND);

putimage(xmap+coint[i].x,coint[i].y,role.w,role.h,&img\_ani,2\*role.w,8\*role.h,SRCPAINT);

break;

case 4:

putimage(xmap+coint[i].x,coint[i].y,role.w,role.h,&img\_ani,3\*role.w,9\*role.h,SRCAND);

putimage(xmap+coint[i].x,coint[i].y,role.w,role.h,&img\_ani,3\*role.w,8\*role.h,SRCPAINT);

break;

default:

break;

}

coint[i].iframe+=0.125;

if(coint[i].iframe==5)

coint[i].iframe=1;

}

}

}

if(get\_bullet==0)

{

switch((int)mapbk\_frame)

{

case 1:

putimage(xmap+xbullet,ybullet,52,25,&img\_ani,0,12\*role.h+25,SRCAND);

putimage(xmap+xbullet,ybullet,52,25,&img\_ani,0,12\*role.h,SRCPAINT);

break;

case 2:

putimage(xmap+xbullet,ybullet,52,25,&img\_ani,52,12\*role.h+25,SRCAND);

putimage(xmap+xbullet,ybullet,52,25,&img\_ani,52,12\*role.h,SRCPAINT);

break;

default:

break;

}

}

for(i=0;i<20;i++) //显示子弹

{

if(get\_bullet==1)

{

if(bullet[i].id==1)

{

if(bullet[i].iframe==1)

{

putimage(bullet[i].x,bullet[i].y,role.w,role.h,&img\_ani,0,3\*role.h,SRCAND);

putimage(bullet[i].x,bullet[i].y,role.w,role.h,&img\_ani,0,2\*role.h,SRCPAINT);

}

else

{

putimage(bullet[i].x,bullet[i].y,role.w,role.h,&img\_ani,role.w,3\*role.h,SRCAND);

putimage(bullet[i].x,bullet[i].y,role.w,role.h,&img\_ani,role.w,2\*role.h,SRCPAINT);

}

}

}

if(enemy[i].id==1)

{

if(enemy[i].iframe==1) //显示敌人

{

putimage(xmap+enemy[i].x,enemy[i].y,role.w,role.h,&img\_ani,0,role.h,SRCAND);

putimage(xmap+enemy[i].x,enemy[i].y,role.w,role.h,&img\_ani,0,0,SRCPAINT);

}

else

{

putimage(xmap+enemy[i].x,enemy[i].y,role.w,role.h,&img\_ani,role.w,role.h,SRCAND);

putimage(xmap+enemy[i].x,enemy[i].y,role.w,role.h,&img\_ani,role.w,0,SRCPAINT);

}

}

if(bomb[i].x!=-1||bomb[i].y!=-1)

{

switch((int)bomb\_frame)

{

case 1:

putimage(xmap+bomb[i].x-role.w/2,bomb[i].y-role.h/2,2\*role.w,2\*role.h,&img\_ani,0,6\*role.h,SRCAND);

putimage(xmap+bomb[i].x-role.w/2,bomb[i].y-role.h/2,2\*role.w,2\*role.h,&img\_ani,0,4\*role.h,SRCPAINT);

break;

case 2:

putimage(xmap+bomb[i].x-role.w/2,bomb[i].y-role.h/2,2\*role.w,2\*role.h,&img\_ani,2\*role.w,6\*role.h,SRCAND);

putimage(xmap+bomb[i].x-role.w/2,bomb[i].y-role.h/2,2\*role.w,2\*role.h,&img\_ani,2\*role.w,4\*role.h,SRCPAINT);

break;

case 3:

putimage(xmap+bomb[i].x-role.w/2,bomb[i].y-role.h/2,2\*role.w,2\*role.h,&img\_ani,4\*role.w,6\*role.h,SRCAND);

putimage(xmap+bomb[i].x-role.w/2,bomb[i].y-role.h/2,2\*role.w,2\*role.h,&img\_ani,4\*role.w,4\*role.h,SRCPAINT);

break;

case 4:

putimage(xmap+bomb[i].x-role.w/2,bomb[i].y-role.h/2,2\*role.w,2\*role.h,&img\_ani,6\*role.w,6\*role.h,SRCAND);

putimage(xmap+bomb[i].x-role.w/2,bomb[i].y-role.h/2,2\*role.w,2\*role.h,&img\_ani,6\*role.w,4\*role.h,SRCPAINT);

break;

default:

break;

}

bomb\_frame+=0.25;

if(bomb\_frame==5)

{

bomb[i].x=-1;

bomb[i].y=-1;

bomb\_frame=1;

}

}

}

int n=score;

char s1[20]="当前得分：";

char s2[10];

itoa(n,s2,10);

RECT r1={10,10,110,40};

RECT r2={110,10,150,40};

setfont(20, 0,"宋体");

drawtext(s1, &r1, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

drawtext(s2, &r2, DT\_CENTER | DT\_VCENTER | DT\_SINGLELINE);

if(role.iframe==1) //显示主角

{

if(role.turn==1)

{

putimage(role.x,role.y,role.w,role.h,&img\_p,0,role.h,SRCAND);

putimage(role.x,role.y,role.w,role.h,&img\_p,0,0,SRCPAINT);

}

else

{

putimage(role.x,role.y,role.w,role.h,&img\_p,4\*role.w,role.h,SRCAND);

putimage(role.x,role.y,role.w,role.h,&img\_p,4\*role.w,0,SRCPAINT);

}

}

else

{

if(role.turn==1)

{

putimage(role.x,role.y,role.w,role.h,&img\_p,role.w,role.h,SRCAND);

putimage(role.x,role.y,role.w,role.h,&img\_p,role.w,0,SRCPAINT);

}

else

{

putimage(role.x,role.y,role.w,role.h,&img\_p,3\*role.w,role.h,SRCAND);

putimage(role.x,role.y,role.w,role.h,&img\_p,3\*role.w,0,SRCPAINT);

}

}

}

int game::isdie()

{

if(role.y>=YSIZE)

return 1;

else

return 0;

}

int game::GetCommand()

{

int c = 0;

if (GetAsyncKeyState('A') & 0x8000)

c |= CMD\_LEFT;

if (GetAsyncKeyState('D') & 0x8000)

c |= CMD\_RIGHT;

if ((GetAsyncKeyState('W') & 0x8000)||(GetAsyncKeyState('K') & 0x8000))

c |= CMD\_UP;

if (GetAsyncKeyState('S') & 0x8000)

c |= CMD\_DOWN;

if (GetAsyncKeyState('J') & 0x8000)

c |= CMD\_SHOOT;

if (GetAsyncKeyState(VK\_ESCAPE) & 0x8000)

c |= CMD\_ESC;

return c;

}

void game::left()

{

role.iframe\*=-1;

role.turn=-1;

role.x-=STEP;

if(is\_l\_touch(1)==1)

role.x+=STEP;

if(role.x<role.xleft)

role.x+=STEP;

}

void game::right()

{

role.iframe\*=-1;

role.turn=1;

role.x+=STEP;

if(is\_r\_touch(1)==1)

role.x-=STEP;

if(role.x>role.xright&&(-xmap+role.x<90\*role.w))

{

role.x-=STEP;

xmapsky-=1;

xmap-=STEP;

}

}

void game::up()

{

mciSendString("play mymusic4 from 0", NULL, 0, NULL);

role.iframe\*=-1;

v0=-sqrt(2\*G\*HIGH);

role.jump=1;

}

void game::init\_shoot()

{

mciSendString("play mymusic2 from 0", NULL, 0, NULL);

ibullet++;

if(ibullet==20)

ibullet=0;

bullet[ibullet].id=1;

bullet[ibullet].y=role.y+8;

bullet[ibullet].turn=role.turn;

if(bullet[ibullet].turn==1)

bullet[ibullet].x=role.x+10;

else

bullet[ibullet].x=role.x-26;

}

int game::is\_l\_touch(int id)

{

int x,y;

int i;

if(id==1) //id==1表示主角是否碰到id为1的地图，及游戏中黄色的地图

{

x=-xmap+role.x;

y=role.y;

for(i=0;i<350;i++)

{

if(map[i].id!=0&&map[i].id<4)

{

POINT m[2];

m[0].x=map[i].x;

m[0].y=map[i].y;

m[1].x=map[i].x+role.w;

m[1].y=map[i].y;

if(map[i].id==3)

{

if(((y-m[1].y)/role.h==0||(y-m[1].y-role.h)/role.h==0)&&x>m[1].x&&x<m[1].x+role.w)

return 1;

}

else

{

if((y-m[1].y)/role.h==0&&x>m[0].x&&x<m[1].x)

return 1;

}

}

}

return 0;

}

else if(id==2) //id==2表示主角是否碰到敌人的左边

{

x=-xmap+role.x;

y=role.y;

for(i=0;i<20;i++)

{

if(enemy[i].id!=0)

{

POINT m[2];

m[0].x=enemy[i].x;

m[0].y=enemy[i].y;

m[1].x=enemy[i].x+role.w;

m[1].y=enemy[i].y;

if((y-m[1].y)/role.h==0&&x>m[0].x&&x<m[1].x)

return 1;

}

}

return 0;

}

else //id==3表示敌人是否碰到地图的左边

{

int j;

for(j=0;j<20;j++)

{

if(enemy[j].id!=0)

{

x=enemy[j].x;

y=enemy[j].y;

for(i=0;i<350;i++)

{

if(map[i].id!=0&&map[i].id<4)

{

POINT m[2];

m[0].x=map[i].x;

m[0].y=map[i].y;

m[1].x=map[i].x+role.w;

m[1].y=map[i].y;

if(map[i].id==3)

{

if(((y-m[1].y)/role.h==0||(y-m[1].y-role.h)/role.h==0)&&x>m[1].x&&x<m[1].x+role.w)

return 1;

}

else

{

if((y-m[1].y)/role.h==0&&x>m[0].x&&x<m[1].x)

return 1;

}

}

}

}

}

return 0;

}

}

int game::is\_r\_touch(int id)

{

int x,y;

int i;

if(id==1)

{

x=-xmap+role.x+role.w;

y=role.y;

for(i=0;i<350;i++)

{

if(map[i].id!=0&&map[i].id<4)

{

POINT m[2];

m[0].x=map[i].x;

m[0].y=map[i].y;

m[1].x=map[i].x+role.w;

m[1].y=map[i].y;

if(map[i].id==3)

{

if(((y-m[0].y)/role.h==0||(y-m[0].y-role.h)/role.h==0)&&x>m[0].x&&x<m[1].x)

return 1;

}

else

{

if((y-m[0].y)/role.h==0&&x>m[0].x&&x<m[1].x)

return 1;

}

}

}

return 0;

}

else if(id==2)

{

x=-xmap+role.x+role.w;

y=role.y;

for(i=0;i<20;i++)

{

if(enemy[i].id!=0)

{

POINT m[2];

m[0].x=enemy[i].x;

m[0].y=enemy[i].y;

m[1].x=enemy[i].x+role.w;

m[1].y=enemy[i].y;

if((y-m[0].y)/role.h==0&&x>m[0].x&&x<m[1].x)

return 1;

}

}

return 0;

}

else

{

int j;

for(j=0;j<20;j++)

{

if(enemy[j].id!=0)

{

x=enemy[j].x+role.w;

y=enemy[j].y;

for(i=0;i<350;i++)

{

if(map[i].id!=0&&map[i].id<4)

{

POINT m[2];

m[0].x=map[i].x;

m[0].y=map[i].y;

m[1].x=map[i].x+role.w;

m[1].y=map[i].y;

if(map[i].id==3)

{

if(((y-m[0].y)/role.h==0||(y-m[0].y-role.h)/role.h==0)&&x>m[0].x&&x<m[1].x)

return 1;

}

else

{

if((y-m[0].y)/role.h==0&&x>m[0].x&&x<m[1].x)

return 1;

}

}

}

}

}

return 0;

}

}

int game::is\_t\_touch()

{

int x,y;

x=-xmap+role.x;

y=role.y;

for(int i=0;i<350;i++)

{

if(map[i].id!=0&&map[i].id<4)

{

POINT m[2];

m[0].x=map[i].x;

m[0].y=map[i].y;

m[1].x=map[i].x;

m[1].y=map[i].y+role.h;

if((x-m[1].x)/role.w==0&&y>m[0].y&&y<m[1].y)

return 1;

}

}

return 0;

}

int game::is\_b\_touch(int id)

{

if(id==1)

{

int x,y;

x=-xmap+role.x;

y=role.y+role.h;

for(int i=0;i<350;i++)

{

if(map[i].id!=0&&map[i].id<4)

{

POINT m[2];

m[0].x=map[i].x;

m[0].y=map[i].y;

m[1].x=map[i].x;

m[1].y=map[i].y+role.h;

if(map[i].id==3)

{

if(((x-m[0].x)/role.w==0||(x+role.w-m[0].x-2\*role.w)/role.w==0)&&y>=m[0].y&&y<m[1].y)

return 1;

}

else

{

if((x-m[0].x)/role.w==0&&y>=m[0].y&&y<m[1].y)

return 1;

}

}

}

return 0;

}

else if(id==2)

{

int x,y;

x=-xmap+role.x;

y=role.y+role.h;

for(int i=0;i<20;i++)

{

if(enemy[i].id!=0)

{

POINT m[2];

m[0].x=enemy[i].x;

m[0].y=enemy[i].y;

m[1].x=enemy[i].x;

m[1].y=enemy[i].y+role.h;

if((x-m[0].x)/role.w==0&&y>m[0].y&&y<m[1].y)

{

mciSendString("play mymusic7 from 0", NULL, 0, NULL);

score+=10;

bomb[i].x=enemy[i].x;

bomb[i].y=enemy[i].y;

enemy[i].id=0;

enemy[i].iframe=-1;

enemy[i].turn=1;

enemy[i].x=-1;

enemy[i].y=-1;

return 1;

}

}

}

return 0;

}

return 0;

}

int game::is\_touch()

{

int i,j;

POINT r[2];

r[0].x=-xmap+role.x;

r[0].y=role.y;

r[1].x=-xmap+role.x+role.w;

r[1].y=role.y+role.h;

for(i=0;i<50;i++)

{

if(coint[i].x!=-1||coint[i].y!=-1)

{

POINT c[4];

c[0].x=coint[i].x;

c[0].y=coint[i].y;

c[1].x=coint[i].x+role.w;

c[1].y=coint[i].y;

c[2].x=coint[i].x;

c[2].y=coint[i].y+role.h;

c[3].x=coint[i].x+role.w;

c[3].y=coint[i].y+role.h;

for(j=0;j<4;j++)

{

if(c[j].x>=r[0].x&&c[j].y>=r[0].y&&c[j].x<=r[1].x&&c[j].y<=r[1].y)

{

mciSendString("play mymusic3 from 0", NULL, 0, NULL);

score+=20;

icoint.x=coint[i].x;

icoint.y=coint[i].y;

coint[i].x=-1;

coint[i].y=-1;

coint[i].iframe=1;

return 1;

}

}

}

}

return 0;

}

int game::is\_land(ENEMY e)

{

POINT r[2];

r[0].x=e.x;

r[0].y=e.y+role.h;

r[1].x=e.x+role.h;

r[1].y=e.y+role.h;

for(int i=0;i<350;i++)

{

if(map[i].id!=0&&map[i].id<4)

{

POINT m[3];

m[0].x=map[i].x;

m[0].y=map[i].y;

m[1].x=map[i].x+role.w;

m[1].y=map[i].y;

m[2].x=map[i].x;

m[2].y=map[i].y+role.h;

if(e.turn==1)

{

if((r[1].x-m[0].x)/role.w==0&&r[1].y>=m[0].y&&r[1].y<m[2].y)

return 1;

}

else

{

if((r[0].x-m[1].x)/role.w==0&&r[0].y>=m[0].y&&r[0].y<m[2].y)

return 1;

}

}

}

return 0;

}

void game::getbullet()

{

int i;

POINT r[2];

r[0].x=-xmap+role.x;

r[0].y=role.y;

r[1].x=-xmap+role.x+role.w;

r[1].y=role.y+role.h;

POINT b[4];

b[0].x=xbullet;

b[0].y=ybullet;

b[1].x=xbullet+52;

b[1].y=ybullet;

b[2].x=xbullet;

b[2].y=ybullet+25;

b[3].x=xbullet+52;

b[3].y=ybullet+25;

for(i=0;i<4;i++)

{

if(b[i].x>=r[0].x&&b[i].y>=r[0].y&&b[i].x<=r[1].x&&b[i].y<=r[1].y)

{

mciSendString("play mymusic8 from 0", NULL, 0, NULL);

get\_bullet=1;

xbullet=0;

ybullet=0;

}

}

}

void game::fall()

{

h=v0\*t+G\*pow(t,2)/2;

role.y+=(int)(h+0.5);

if(v0>=0) //自由落体

{

if(isdie()==1)

return;

if(is\_b\_touch(1)==1)

{

v0=0;

role.y=role.y/role.h\*role.h;

role.jump=0;

}

}

else //向上跳跃

{

if(v0>=0)

h=0;

else

role.y+=(int)(h+0.5);

if(is\_t\_touch()==1)

{

v0=0;

h=0;

role.y=role.y/role.h\*role.h+role.h;

}

}

v0=v0+G\*t;

}

void game::shoot()

{

int i;

for(i=0;i<20;i++)

{

if(bullet[i].id==1)

{

if(bullet[i].turn==1)

{

bullet[i].x+=2\*STEP;

}

else

{

bullet[i].x-=2\*STEP;

}

if((bullet[i].x<(-3\*role.w))||(bullet[i].x>XSIZE))

{

bullet[i].id=0;

bullet[i].x=-1;

bullet[i].y=-1;

bullet[i].iframe=1;

bullet[i].turn=1;

}

if(eat(bullet[i])==1)

{

bullet[i].id=0;

bullet[i].x=-1;

bullet[i].y=-1;

bullet[i].iframe=1;

bullet[i].turn=1;

bomb[i].x=temp.x;

bomb[i].y=temp.y;

}

bullet[i].iframe\*=-1;

}

}

}

int game::eat(BULLET b)

{

POINT r[4];

r[0].x=-xmap+b.x+role.w/2;

r[0].y=b.y;

r[1].x=-xmap+b.x+role.w;

r[1].y=b.y;

r[2].x=-xmap+b.x+role.w/2;

r[2].y=b.y+role.h/2;

r[3].x=-xmap+b.x+role.w;

r[3].y=b.y+role.h/2;

int i;

for(i=0;i<350;i++)

{

if(map[i].id!=0&&map[i].id<4)

{

POINT m[2];

m[0].x=map[i].x;

m[0].y=map[i].y;

if(map[i].id==3)

{

m[1].x=map[i].x+2\*role.w;

m[1].y=map[i].y+2\*role.h;

}

else

{

m[1].x=map[i].x+role.w;

m[1].y=map[i].y+role.h;

}

for(int j=0;j<4;j++)

{

if(r[j].x>m[0].x&&r[j].x<m[1].x&&r[j].y>m[0].y&&r[j].y<m[1].y)

{

mciSendString("play mymusic6 from 0", NULL, 0, NULL);

temp.x=r[0].x-role.w/4;

temp.y=r[0].y-role.w/4;

return 1;

}

}

}

if(i<20)

{

if(enemy[i].id==1)

{

POINT e[2];

e[0].x=enemy[i].x;

e[0].y=enemy[i].y;

e[1].x=enemy[i].x+role.w;

e[1].y=enemy[i].y+role.h;

for(int j=0;j<4;j++)

{

if(r[j].x>e[0].x&&r[j].x<e[1].x&&r[j].y>e[0].y&&r[j].y<e[1].y)

{

mciSendString("play mymusic5 from 0", NULL, 0, NULL);

score+=10;

temp.x=enemy[i].x;

temp.y=enemy[i].y;

enemy[i].id=0;

enemy[i].iframe=-1;

enemy[i].turn=1;

enemy[i].x=-1;

enemy[i].y=-1;

return 1;

}

}

}

}

}

return 0;

}

void game::end()

{

MyTimer tt;

EndBatchDraw();

if(isdie()==1||win==1)

pause=0;

if(pause==1)

return;

if(win==1)

tt.Sleep(5000);

else

tt.Sleep(2700);

mciSendString("close all", NULL, 0, NULL);

tt.Sleep(1000);

if(win==1)

{

pause=0;

score=0;

life=0;

mciSendString("open 通关.mp3 alias mymusic13", NULL, 0, NULL);

mciSendString("play mymusic13", NULL, 0, NULL);

putimage(0,-3\*YSIZE,&img\_home);

tt.Sleep(7000);

mciSendString("close mymusic13", NULL, 0, NULL);

}

else

{

score=0;

if(life==0)

{

mciSendString("open 游戏结束.mp3 alias mymusic12", NULL, 0, NULL);

mciSendString("play mymusic12", NULL, 0, NULL);

putimage(0,-YSIZE,&img\_home);

tt.Sleep(5500);

mciSendString("close mymusic12", NULL, 0, NULL);

}

else

{

cleardevice();

outtextxy(XSIZE/2-43,YSIZE/3,"生命还剩下:");

if(life==1)

outtextxy(XSIZE/2,YSIZE/2-20,"1");

else if(life==2)

outtextxy(XSIZE/2,YSIZE/2-20,"2");

tt.Sleep(2000);

}

}

cleardevice();

}

void main()

{

game g;

while(true)

{

g.start();

g.init();

mciSendString("play mymusic1 repeat", NULL, 0, NULL);

g.show();

g.move();

g.end();

}

}