#include<stdio.h>

#include<easyx.h>

#include<graphics.h>

#include<windows.h>

#include<conio.h>

#define WIDTH 980 //窗口大小

#define HEIGHT 460

#define M 83 //金币的数量

#define N 37 //墙体数量

#define Vwall -10 //墙体速度

#define Vperson 10 //小人y方向速度

#define Vcoin 20 //金币被拾取后下落的速度

IMAGE backimg; //定义背景图

int m = 0; //金币数目

int init[N + 1]; //用数组记录各个墙体的初始位置

int length[N + 1]; //用数组记录各个墙体的x长度

int height[N + 1]; //用数组记录各个墙体的y高度

int coinnum[M + 1]; //用数组记录各个硬币的初始位置

int coiny[M + 1]; //记录各硬币的初始y坐标

char str1[40]; //游戏结束时输出的字符串

char str2[20]; //显示金币数量的字符串

unsigned int t1, t2; //定义两个时间用于定时器

//定义小人数据

struct Person

{

IMAGE img[2][8]; //共2\*8=16张小人图

int x, y;

int vx, vy[2];

int frame1; //定义帧frame1(0或1)来判断小人y方向向下或向上移动

int frame2; //定义帧frame2(0,1,2,3,4,5,6,7)来加载小人图像

}person;

//定义墙体数据

struct Wall

{

IMAGE img;

int x, y;

int vx;

}wall[N + 1]; //N堵墙体

//定义金币

struct coin

{

IMAGE inmg;

int x, y;

int vx;

int vy = 500;

}coin[M + 1];

//函数：初始化墙体

void GameInitWall()

{

//设置墙体y高度

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

{

height[i] = 40;

}

height[4] = height[5] = height[34] = height[35] = 140;

//设置墙体x长度

length[0] = 800; length[1] = 800; length[2] = 800; length[3] = 800; length[4] = 40;

length[5] = 40; length[6] = 200; length[7] = 200; length[8] = 200; length[9] = 500;

length[10] = 200; length[11] = 200; length[12] = 200; length[13] = 800; length[14] = 500;

length[16] = 200; length[15] = 200; length[17] = 500; length[18] = 800; length[19] = 500;

length[20] = 200; length[21] = 200; length[22] = 500; length[23] = 500; length[24] = 200;

length[25] = 200; length[26] = 500; length[27] = 500; length[28] = 200; length[29] = 200;

length[30] = 200; length[31] = 200; length[32] = 200; length[33] = 500; length[34] = 40;

length[35] = 40; length[36] = 800; length[37] = 800;

loadimage(&wall[0].img, "images/wall1.png"); wall[0].x = -800; wall[0].y = 320;

loadimage(&wall[1].img, "images/wall1.png"); wall[1].x = 0; wall[1].y = 320;

loadimage(&wall[2].img, "images/wall1.png"); wall[2].x = 400; wall[2].y = 100;

loadimage(&wall[3].img, "images/wall1.png"); wall[3].x = 1080; wall[3].y = 320;

loadimage(&wall[4].img, "images/column.png"); wall[4].x = 800; wall[4].y = 320;

loadimage(&wall[5].img, "images/column.png"); wall[5].x = 1040; wall[5].y = 320;

loadimage(&wall[6].img, "images/wall3.png"); wall[6].x = 1980; wall[6].y = 120;

loadimage(&wall[7].img, "images/wall3.png"); wall[7].x = 2180; wall[7].y = 100;

loadimage(&wall[8].img, "images/wall3.png"); wall[8].x = 2380; wall[8].y = 80;

loadimage(&wall[9].img, "images/wall2.png"); wall[9].x = 2580; wall[9].y = 60;

loadimage(&wall[10].img, "images/wall3.png"); wall[10].x = 3080; wall[10].y = 240;

loadimage(&wall[11].img, "images/wall3.png"); wall[11].x = 3280; wall[11].y = 260;

loadimage(&wall[12].img, "images/wall3.png"); wall[12].x = 3480; wall[12].y = 280;

loadimage(&wall[13].img, "images/wall1.png"); wall[13].x = 3680; wall[13].y = 300;

loadimage(&wall[14].img, "images/wall2.png"); wall[14].x = 4280; wall[14].y = 100;

loadimage(&wall[15].img, "images/wall3.png"); wall[15].x = 4780; wall[15].y = 80;

loadimage(&wall[16].img, "images/wall3.png"); wall[16].x = 4980; wall[16].y = 60;

loadimage(&wall[17].img, "images/wall2.png"); wall[17].x = 5180; wall[17].y = 40;

loadimage(&wall[18].img, "images/wall1.png"); wall[18].x = 5480; wall[18].y = 300;

loadimage(&wall[19].img, "images/wall2.png"); wall[19].x = 5780; wall[19].y = 120;

loadimage(&wall[20].img, "images/wall3.png"); wall[20].x = 6280; wall[20].y = 100;

loadimage(&wall[21].img, "images/wall3.png"); wall[21].x = 6280; wall[21].y = 280;

loadimage(&wall[22].img, "images/wall2.png"); wall[22].x = 6480; wall[22].y = 80;

loadimage(&wall[23].img, "images/wall2.png"); wall[23].x = 6480; wall[23].y = 260;

loadimage(&wall[24].img, "images/wall3.png"); wall[24].x = 6980; wall[24].y = 100;

loadimage(&wall[25].img, "images/wall3.png"); wall[25].x = 6980; wall[25].y = 280;

loadimage(&wall[26].img, "images/wall2.png"); wall[26].x = 7180; wall[26].y = 120;

loadimage(&wall[27].img, "images/wall2.png"); wall[27].x = 7180; wall[27].y = 300;

loadimage(&wall[28].img, "images/wall3.png"); wall[28].x = 7680; wall[28].y = 100;

loadimage(&wall[29].img, "images/wall3.png"); wall[29].x = 7680; wall[29].y = 280;

loadimage(&wall[30].img, "images/wall3.png"); wall[30].x = 7880; wall[30].y = 80;

loadimage(&wall[31].img, "images/wall3.png"); wall[31].x = 7880; wall[31].y = 260;

loadimage(&wall[32].img, "images/wall3.png"); wall[32].x = 8080; wall[32].y = 60;

loadimage(&wall[33].img, "images/wall2.png"); wall[33].x = 8280; wall[33].y = 40;

loadimage(&wall[34].img, "images/column.png"); wall[34].x = 8980; wall[34].y = 0;

loadimage(&wall[35].img, "images/column.png"); wall[35].x = 8980; wall[35].y = 320;

loadimage(&wall[36].img, "images/wall1.png"); wall[36].x = 9020; wall[36].y = 320;

loadimage(&wall[37].img, "images/wall1.png"); wall[37].x = 9820; wall[37].y = 320;

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

{

init[i] = wall[i].x; //记录墙体初始位置

wall[i].vx = Vwall; //所有墙体的移动速度均为Vwall

}

}

//初始化金币

void Gamecoin()

{

loadimage(&coin[1].inmg, "images/coin.png"); coin[1].x = 1080; coin[1].y = 300;

loadimage(&coin[2].inmg, "images/coin.png"); coin[2].x = 1110; coin[2].y = 300;

loadimage(&coin[3].inmg, "images/coin.png"); coin[3].x = 1140; coin[3].y = 300;

loadimage(&coin[4].inmg, "images/coin.png"); coin[4].x = 1170; coin[4].y = 300;

loadimage(&coin[5].inmg, "images/coin.png"); coin[5].x = 1200; coin[5].y = 300;

loadimage(&coin[6].inmg, "images/coin.png"); coin[6].x = 1230; coin[6].y = 300;

loadimage(&coin[7].inmg, "images/coin.png"); coin[7].x = 1260; coin[7].y = 300;

loadimage(&coin[8].inmg, "images/coin.png"); coin[8].x = 1290; coin[8].y = 300;

loadimage(&coin[9].inmg, "images/coin.png"); coin[9].x = 1500; coin[9].y = 300;

loadimage(&coin[10].inmg, "images/coin.png"); coin[10].x = 1530; coin[10].y = 300;

loadimage(&coin[11].inmg, "images/coin.png"); coin[11].x = 1560; coin[11].y = 300;

loadimage(&coin[12].inmg, "images/coin.png"); coin[12].x = 1590; coin[12].y = 300;

loadimage(&coin[13].inmg, "images/coin.png"); coin[13].x = 2620; coin[13].y = 100;

loadimage(&coin[14].inmg, "images/coin.png"); coin[14].x = 2650; coin[14].y = 100;

loadimage(&coin[15].inmg, "images/coin.png"); coin[15].x = 2680; coin[15].y = 100;

loadimage(&coin[16].inmg, "images/coin.png"); coin[16].x = 2710; coin[16].y = 100;

loadimage(&coin[17].inmg, "images/coin.png"); coin[17].x = 2740; coin[17].y = 100;

loadimage(&coin[18].inmg, "images/coin.png"); coin[18].x = 2770; coin[18].y = 100;

loadimage(&coin[19].inmg, "images/coin.png"); coin[19].x = 3080; coin[19].y = 220;

loadimage(&coin[20].inmg, "images/coin.png"); coin[20].x = 3110; coin[20].y = 220;

loadimage(&coin[21].inmg, "images/coin.png"); coin[21].x = 3140; coin[21].y = 220;

loadimage(&coin[22].inmg, "images/coin.png"); coin[22].x = 3170; coin[22].y = 220;

loadimage(&coin[23].inmg, "images/coin.png"); coin[23].x = 3200; coin[23].y = 220;

loadimage(&coin[24].inmg, "images/coin.png"); coin[24].x = 3230; coin[24].y = 220;

loadimage(&coin[25].inmg, "images/coin.png"); coin[25].x = 3500; coin[25].y = 260;

loadimage(&coin[26].inmg, "images/coin.png"); coin[26].x = 3530; coin[26].y = 260;

loadimage(&coin[27].inmg, "images/coin.png"); coin[27].x = 3560; coin[27].y = 260;

loadimage(&coin[28].inmg, "images/coin.png"); coin[28].x = 3590; coin[28].y = 260;

loadimage(&coin[29].inmg, "images/coin.png"); coin[29].x = 3620; coin[29].y = 260;

loadimage(&coin[30].inmg, "images/coin.png"); coin[30].x = 3650; coin[30].y = 260;

loadimage(&coin[31].inmg, "images/coin.png"); coin[31].x = 4800; coin[31].y = 120;

loadimage(&coin[32].inmg, "images/coin.png"); coin[32].x = 4830; coin[32].y = 120;

loadimage(&coin[33].inmg, "images/coin.png"); coin[33].x = 4860; coin[33].y = 120;

loadimage(&coin[34].inmg, "images/coin.png"); coin[34].x = 4890; coin[34].y = 120;

loadimage(&coin[35].inmg, "images/coin.png"); coin[35].x = 4920; coin[35].y = 120;

loadimage(&coin[36].inmg, "images/coin.png"); coin[36].x = 4950; coin[36].y = 120;

loadimage(&coin[37].inmg, "images/coin.png"); coin[37].x = 5200; coin[37].y = 80;

loadimage(&coin[38].inmg, "images/coin.png"); coin[38].x = 5230; coin[38].y = 80;

loadimage(&coin[39].inmg, "images/coin.png"); coin[39].x = 5260; coin[39].y = 80;

loadimage(&coin[40].inmg, "images/coin.png"); coin[40].x = 5290; coin[40].y = 80;

loadimage(&coin[41].inmg, "images/coin.png"); coin[41].x = 5320; coin[41].y = 80;

loadimage(&coin[42].inmg, "images/coin.png"); coin[42].x = 5350; coin[42].y = 80;

loadimage(&coin[43].inmg, "images/coin.png"); coin[43].x = 5380; coin[43].y = 80;

loadimage(&coin[44].inmg, "images/coin.png"); coin[44].x = 5410; coin[44].y = 80;

loadimage(&coin[45].inmg, "images/coin.png"); coin[45].x = 5800; coin[45].y = 160;

loadimage(&coin[46].inmg, "images/coin.png"); coin[46].x = 5830; coin[46].y = 160;

loadimage(&coin[47].inmg, "images/coin.png"); coin[47].x = 5860; coin[47].y = 160;

loadimage(&coin[48].inmg, "images/coin.png"); coin[48].x = 5890; coin[48].y = 160;

loadimage(&coin[49].inmg, "images/coin.png"); coin[49].x = 5920; coin[49].y = 160;

loadimage(&coin[50].inmg, "images/coin.png"); coin[50].x = 5950; coin[50].y = 160;

loadimage(&coin[51].inmg, "images/coin.png"); coin[51].x = 5980; coin[51].y = 160;

loadimage(&coin[52].inmg, "images/coin.png"); coin[52].x = 6010; coin[52].y = 160;

loadimage(&coin[53].inmg, "images/coin.png"); coin[53].x = 6500; coin[53].y = 220;

loadimage(&coin[54].inmg, "images/coin.png"); coin[54].x = 6530; coin[54].y = 220;

loadimage(&coin[55].inmg, "images/coin.png"); coin[55].x = 6560; coin[55].y = 220;

loadimage(&coin[56].inmg, "images/coin.png"); coin[56].x = 6590; coin[56].y = 220;

loadimage(&coin[57].inmg, "images/coin.png"); coin[57].x = 6620; coin[57].y = 220;

loadimage(&coin[58].inmg, "images/coin.png"); coin[58].x = 6650; coin[58].y = 220;

loadimage(&coin[59].inmg, "images/coin.png"); coin[59].x = 6680; coin[59].y = 220;

loadimage(&coin[60].inmg, "images/coin.png"); coin[60].x = 7200; coin[60].y = 160;

loadimage(&coin[61].inmg, "images/coin.png"); coin[61].x = 7230; coin[61].y = 160;

loadimage(&coin[62].inmg, "images/coin.png"); coin[62].x = 7260; coin[62].y = 160;

loadimage(&coin[63].inmg, "images/coin.png"); coin[63].x = 7290; coin[63].y = 160;

loadimage(&coin[64].inmg, "images/coin.png"); coin[64].x = 7320; coin[64].y = 160;

loadimage(&coin[65].inmg, "images/coin.png"); coin[65].x = 7350; coin[65].y = 160;

loadimage(&coin[66].inmg, "images/coin.png"); coin[66].x = 7380; coin[66].y = 160;

loadimage(&coin[67].inmg, "images/coin.png"); coin[67].x = 7410; coin[67].y = 160;

loadimage(&coin[68].inmg, "images/coin.png"); coin[68].x = 7900; coin[68].y = 220;

loadimage(&coin[69].inmg, "images/coin.png"); coin[69].x = 7930; coin[69].y = 220;

loadimage(&coin[70].inmg, "images/coin.png"); coin[70].x = 7960; coin[70].y = 220;

loadimage(&coin[71].inmg, "images/coin.png"); coin[71].x = 7990; coin[71].y = 220;

loadimage(&coin[72].inmg, "images/coin.png"); coin[72].x = 8020; coin[72].y = 220;

loadimage(&coin[73].inmg, "images/coin.png"); coin[73].x = 8050; coin[73].y = 220;

loadimage(&coin[74].inmg, "images/coin.png"); coin[74].x = 8300; coin[74].y = 80;

loadimage(&coin[75].inmg, "images/coin.png"); coin[75].x = 8330; coin[75].y = 80;

loadimage(&coin[76].inmg, "images/coin.png"); coin[76].x = 8360; coin[76].y = 80;

loadimage(&coin[77].inmg, "images/coin.png"); coin[77].x = 8390; coin[77].y = 80;

loadimage(&coin[78].inmg, "images/coin.png"); coin[78].x = 8420; coin[78].y = 80;

loadimage(&coin[79].inmg, "images/coin.png"); coin[79].x = 8450; coin[79].y = 80;

loadimage(&coin[80].inmg, "images/coin.png"); coin[80].x = 8480; coin[80].y = 80;

loadimage(&coin[81].inmg, "images/coin.png"); coin[81].x = 8510; coin[81].y = 80;

loadimage(&coin[82].inmg, "images/coin.png"); coin[82].x = 8540; coin[82].y = 80;

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

{

coinnum[i] = coin[i].x;

coin[i].vx = Vwall;

}

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

{

coiny[j] = coin[j].y;

}

}

//函数：初始化小人

void GameInitPerson()

{

//加载图像

loadimage(&person.img[0][0], "images/person11.jpg");

loadimage(&person.img[0][1], "images/person12.jpg");

loadimage(&person.img[0][2], "images/person13.jpg");

loadimage(&person.img[0][3], "images/person14.jpg");

loadimage(&person.img[0][4], "images/person15.jpg");

loadimage(&person.img[0][5], "images/person16.jpg");

loadimage(&person.img[0][6], "images/person17.jpg");

loadimage(&person.img[0][7], "images/person18.jpg");

loadimage(&person.img[1][0], "images/person21.jpg");

loadimage(&person.img[1][1], "images/person22.jpg");

loadimage(&person.img[1][2], "images/person23.jpg");

loadimage(&person.img[1][3], "images/person24.jpg");

loadimage(&person.img[1][4], "images/person25.jpg");

loadimage(&person.img[1][5], "images/person26.jpg");

loadimage(&person.img[1][6], "images/person27.jpg");

loadimage(&person.img[1][7], "images/person28.jpg");

//初始化数据

person.x = 300;

person.y = 200;

person.vx = 0;

person.vy[0] = Vperson;

person.vy[1] = -Vperson;

person.frame1 = 0;

person.frame2 = 0;

}

//函数：小人x坐标更新

void GameUpdatePersonX()

{

person.x += person.vx; //小人左右运动

/\*840~1040段在GameUpdatePersonY中设置\*/

//将特殊的33,34,35段墙体抽出

if (person.x + 40 >= wall[33].x + length[33] && person.x + 40 <= wall[34].x)

{

//如果小人右碰壁，则有向左的速度

if (person.x + 40 == wall[34].x &&

((person.y < wall[34].y + height[34] && person.y > wall[34].y) ||

(person.y + 40 > wall[35].y && person.y + 40 < wall[35].y + height[35])))

{

person.vx = Vwall;

}

}

//一般情况下

else

{

for (int i = 0; i <= N; i++) //遍历37个墙体

{

if (person.x + 40 == wall[i].x) //如果小人的右侧存在墙体

{

//只有在19~32墙体段会被卡住

if (i >= 19 && i <= 32)

{

//小人被右侧墙体卡住

if ((person.y < wall[i].y + height[i] && person.y > wall[i].y) ||

(person.y + 40 > wall[i].y && person.y + 40 < wall[i].y + height[i]) ||

//i+1和i-1用于上下均有墙体的情况

(person.y < wall[i + 1].y + height[i + 1] && person.y > wall[i + 1].y) ||

(person.y + 40 > wall[i + 1].y && person.y + 40 < wall[i + 1].y + height[i + 1]) ||

(person.y < wall[i - 1].y + height[i - 1] && person.y > wall[i - 1].y) ||

(person.y + 40 > wall[i - 1].y && person.y + 40 < wall[i - 1].y + height[i - 1]))

{

person.vx = Vwall; //小人跟随墙体向左移动

}

else

{

person.vx = 0; //恢复小人x方向的静止(此前碰到墙向左移动)

}

}

}

}

}

}

//函数：小人y坐标更新

void GameUpdatePersonY()

{

person.y += person.vy[person.frame1]; //小人上下运动

//将较为特殊的840~1040段抽出

if (person.x >= wall[4].x + length[4] && person.x + 40 <= wall[5].x)

{

if (person.y + 40 >= 320) //如果小人陷入坑内，则会一直掉落

{

person.vy[person.frame1] = Vperson;

if (person.x == wall[5].x - 40) //如果小人右碰壁，则有向左的速度

{

person.vx = Vwall;

}

}

else if (person.y == wall[2].y + 40) //如果小人碰到上壁，则vy为0

{

person.vy[person.frame1] = 0;

}

}

else

{

for (int i = 0; i <= N; i++) //遍历37个墙体

{

//找到与小人的x对应的墙体

if (person.x + 40 >= wall[i].x && person.x <= wall[i].x + length[i])

{

//上下均有墙体时

if (person.x + 40 >= wall[i + 1].x && person.x <= wall[i + 1].x + length[i + 1])

{

//若小人与墙体上下相碰，则vy变为0

if (person.y + 40 == wall[i].y || person.y == wall[i].y + 40 ||

//i+1用于上下均有墙体的情况

person.y + 40 == wall[i + 1].y || person.y == wall[i + 1].y + 40)

{

person.vy[person.frame1] = 0;

}

else //当进行类阶梯型运动时

{

person.vy[0] = Vperson;

person.vy[1] = -Vperson; //恢复小人的速度（此前碰到墙体被设为0）

}

break; //退出for循环防止速度错乱

}

//上下只有一个墙体时

else

{

//若小人与墙体上下相碰，则vy变为0

if (person.y + 40 == wall[i].y || person.y == wall[i].y + 40)

{

person.vy[person.frame1] = 0;

}

else //当进行类阶梯型运动时

{

person.vy[0] = Vperson;

person.vy[1] = -Vperson; //恢复小人的速度（此前碰到墙体被设为0）

}

break; //退出for循环防止速度错乱

}

}

}

}

}

//函数：初始化

void GameInit()

{

//初始化定时器

t1 = GetTickCount();

t2 = GetTickCount();

//初始化窗口

initgraph(WIDTH, HEIGHT);

//初始化背景

loadimage(&backimg, "images/bg.png");

//初始化墙体

GameInitWall();

//初始化金币

Gamecoin();

//初始化小人

GameInitPerson();

}

//函数：界面绘制

void GameDraw()

{

BeginBatchDraw(); //开始绘制

putimage(0, 0, &backimg); //绘制背景

for (int i = 0; i <= N; i++) //绘制墙体

{

putimage(wall[i].x, wall[i].y, &wall[i].img);

}

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

{

putimage(coin[j].x, coin[j].y, &coin[j].inmg);

}

putimage(person.x, person.y, &person.img[person.frame1][person.frame2]);//绘制小人

EndBatchDraw(); //结束绘制

}

//函数：数据更新

void GameUpdate()

{

//以下为手动更新

MOUSEMSG msg = { 0 }; //定义鼠标消息

if (MouseHit()) //判断是否有鼠标消息

{

msg = GetMouseMsg(); //获取鼠标消息

if (msg.uMsg == WM\_LBUTTONDOWN) //鼠标左键

{

if (person.vy[person.frame1] == 0) //当小人在墙体上时(vy为0)才能转换方向

{

person.vy[0] = Vperson;

person.vy[1] = -Vperson; //恢复小人的y速度(此前碰到墙体被设为0)

person.frame1++;

if (person.frame1 > 1)

{

person.frame1 = 0; //确保frame1在0到1之间循环

}

}

}

}

//以下为自动更新

t2 = GetTickCount(); //最外层为定时器

if (t2 - t1 > 20)

{

//墙体更新

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

{

wall[i].x += wall[i].vx; //所有墙体向左移动

}

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

{

if (init[i] - wall[i].x > 9020)

{

wall[i].x = init[i] + 580; //墙体返回初始位置右侧

}

}

//金币更新

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

{

coin[i].x += coin[i].vx;

}

//金币回到初始的x位置

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

{

if (coinnum[i] - coin[i].x > 9020)

{

coin[i].x = coinnum[i] + 580;

}

}

//小人图片更新

person.frame2++;

if (person.frame2 > 7)

{

person.frame2 = 0; //确保frame2在0到7之间循环

}

//小人吃金币

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

{

if (coin[i].x > person.x && coin[i].x < person.x + 40 && person.y + 40>coin[i].y && person.y < coin[i].y + 20)

{

coin[i].y = coin[i].y + coin[i].vy; m++;

}

for (int j = 0; j < M; j++) { if (coin[j].x < 0) { coin[j].y = coiny[j]; } }

}

//小人x坐标更新

GameUpdatePersonX();

//小人y坐标更新

GameUpdatePersonY();

//小人死亡

if (person.x + 40 <= 0 || person.y + 40 <= 0 || person.y >= 460)

{

settextcolor(WHITE);

settextstyle(40, 0, "宋体");

sprintf\_s(str1, "游戏结束，获得金币个数为%d", m);

outtextxy(450, 190, str1); // 在(450,190)位置输出文本

getchar();

}

t1 = t2;

}

}

//主函数

int main(void)

{

GameInit();

while (1)

{

GameDraw();

GameUpdate();

}

return 0;

}