C 语言小游戏《保卫清华》设计报告

1. 简介

本游戏《保卫清华》由小组“清华保卫处”独立制作

游戏的内容及玩法主要借鉴了经典塔防游戏《保卫萝卜》，并融入了清华元素致敬清华大学，制作耗时 12 天。

玩家需花费金钱放置炮塔攻击怪物，阻止它们推倒二校门(即萝卜)，并在此基础上加入随机事件、无尽关卡等玩法。游戏源代码由基础 C 语言+Easyx 图形库组成，采用图形化界面。画风简约且富有幽默感，音乐或舒缓或激昂，音效或搞怪或震撼。也存在后期变卡顿(不影响基本操作)、无法存档等小缺点，但总之作为一个休闲小游戏还是可以玩的。

组内成员及分工

张赫阳(2022012816):游戏源代码及注释的作者、图片及音乐素材的制作者，debug 的主要人员，本设计报告的作者

何宇晖(2022012813): 游戏选题的提出者，游戏功能、炮塔类型、数值平衡、随机事件的策划者

傅家俊(2022013367):游戏主要玩法、经济系统的策划者，并撰写了游戏帮助和出怪部分的代码

赵思奇(2021010268):游戏内容的体验者，由于其他事务较忙，对本组的大作业无暇顾及

2. 设计思想

游戏设计方面主要有操作界面设计、地图设计、玩法策划、数值策划、音乐和图片、源代码各函数设计等方面，基本思路就是逐帧绘制画面并修改参数，形成外表生动又活泼，内在运算无差错的小游戏。

具体设计流程：首先是初始界面和按钮，接着是选择 6 个地图的界面和按钮，然后就是关卡内的部分，包括炮塔、怪物结构体数组的设计，炮塔、怪物的生成和交互，怪物行走路线的参数化，各数值参数的配置及显示(波数、金钱数等)，逐帧绘制画面，鼠标操作 UI，随机事件的触发和消除等等，最后是游戏输赢的判断及结算动画

3. 系统框架

模块分解：

开始模块：选关、查看游戏帮助、退出游戏

计时模块：炮塔内置时钟、波数计算时钟、事件生效时钟、帧率计时

怪物模块：怪物种类、位置、速度、血量，出怪，重置怪(包括血量为负和位置越界两种可能)

炮塔模块：炮塔种类、位置、伤害、范围、攻频、debuff，炮塔放置、升级和拆除

交互模块：炮塔索敌单伤、范围群伤，怪物减血减速，二校门扣血，金钱消耗和生产，返回主菜单，暂停

鼠标模块：光标移动、网格坐标、高亮按钮、左右键光标变颜色，鼠标放置、升级或拆除炮塔

绘制模块：绘制菜单，绘制地图，绘制按钮，绘制炮塔，绘制怪物，绘制 UI，绘制结算动画等

事件模块：“网课掉线”随机事件的产生与消除

结束模块：判断是否 break 游戏主循环，并判断成功还是失败，再绘制/播放结算界面

4. 游戏功能

攻方

怪物种类：

1.FDU:移速慢，血量一般的小怪，死亡后掉落 13 金币

2.SJTU:移速快，血量少的小怪，死亡后掉落 6 金币

3.PKU:移速慢，血量巨厚的精英怪，死亡后掉落 48 金币

出怪逻辑：

电脑出怪，普通关卡一共 20 波，每波 20 个怪物，前 19 个是随机小怪，最后一个是精英怪，无尽关卡没有最高波数限制，进入下一波当且仅当一波所有怪物被重置(死亡或越界)

攻击方式：

怪物按照地图路线行走，攻击对象是路线末端的 2 校门，当怪物触碰到二校门(越界)时，二校门血量-1，怪物重置，否则，当怪物血量为非正数时，怪物立即重置。

守方

炮塔种类：

炮塔共有 10 种，其中有 8 种攻击型炮塔和 2 种功能型炮塔。

攻击方式：

黑笔：单体攻击型炮塔，攻频 4 次/秒，攻击力较低，花费便宜

红笔：单体攻击型炮塔，攻频为帧伤，攻击力中等，花费较便宜

铅笔：单体攻击型炮塔，攻频 2 次/秒，攻击力低，有概率击退怪物，概率随等级提升，花费适中

直尺：单体攻击型炮塔，攻频 1 次/秒，攻击力高，对精英怪伤害翻倍，花费昂贵

三角板：群伤溅射攻击型炮塔，攻频 1 次/秒，攻击力中等，攻击范围超模，对锁定目标造成正常伤害，对目标周围怪物造成 20%溅射伤害，花费适中

量角器：群伤溅射攻击型炮塔，攻频 1 次/秒，攻击力中等，对锁定目标造成正常伤害，对目标周围怪物造成 20%溅射伤害，并附加减速，花费较贵

圆规：群伤攻击型炮塔，攻频 1 次/秒，攻击力中等，对圆形范围内所有怪物造成同等伤害，花费较贵

橡皮擦：单体攻击型炮塔，攻频 1 次/秒，攻击力极低，对单体怪物造成巨量减速，花费较便宜

咖啡机：功能型炮塔，可产金币，根据等级，每秒产金币在 8 到 40 不等，花费天价，不过是后期的生力军

wifi：一次性功能型炮塔，可在放置后产生覆盖全屏的 5G 信号，拯救因网络波动而暂停攻击的炮塔们，放置 1s 后网络信号恢复正常，wifi 炮塔自动消失，花费较昂贵

鼠标操作方式：

炮塔和鼠标操作密切联系，由玩家主动放置、升级和拆除。

鼠标左键单击空格子，可以跳出选择炮塔的 UI，玩家可点击选择放置炮塔，若此时再次点击空格子则取消放置，若金钱不足则无法放置。

鼠标左键单击未满级的炮塔，可以选择花费金币升级，若点击该格以外的格子则取消升级，若金币不足则无法升级

鼠标右键单击炮塔，可以选择拆除，若点击该格以外的格子则取消拆除，拆除后返还放置/升级所花费的 60%的金币

其他功能

游戏主界面可查看游戏帮助，帮助新手玩家快速了解游戏主要玩法

游戏六个地图均有背景音乐，而且各不相同，炮塔有攻击音效

网课模式和无尽模式 10 波后会随机出现“网课掉线”事件，持续 10 秒，期间所有炮塔停止攻击，10 秒后自动恢复，若玩家放置 wifi 则放置后 1s 恢复

怪物路线末尾的二校门血量为 10(无尽为 20)，血量非正数时游戏失败

游戏失败会播放一段视频，其实是逐帧显示图片>_<

玩家在关卡内可随时选择暂停和返回主菜单，体现游戏的休闲性

5. 调试和解决方法

下面列举几个编写代码中遇到的困难

“0”or“1”的陷阱：网上总有人说，程序员数数都是从 0 开始数而不是从 1 开始数的。通过本次大作业确实切身体会到为什么这么说了。刚开始写鼠标界面时，发现第 10 列和第 7 行(即最后一列和最后一行)的格子始终点不动，检查了 inroute，drawchoosetrt，settrt 等函数好多遍都没有用，费了几乎一上午。最后用 printf 大法在多个节点调试才发现，第 10 列和第 7 行点不动是因为 node 的起始数是 1，而传参的时候从 0 开始，实际上第 1 列和第 1 行应该是第 0 列和第 0 行，第 10 列和第 7 行应该是第 9 列和第 6 行，掉进“0”和“1”的陷阱里了。

绘图时怎么找像素坐标：图形化界面离不开像素坐标，我最初用微信截

图工具，从左上角截图得出坐标，但是不见绘制的图形——原来微信截图工具的像素比 easyx 图形窗口的像素小，这意味着坐标不是很好得到的，写绘图函数的时候用到的坐标的都是无数次试验的结果。

inroute 函数难题(鼠标不能在怪物行走路线上放置炮塔)：每个地图的路线各不相同，最初我们都不知道怎么让怪物沿着路线走，后来很快想到了曲线参数方程的方法。然而这与数学上学的简单曲线不同，地图路线是分段曲线，而且有 5 种不同的路线，有横平竖直的线也有斜线甚至有圆弧，经过反复的数学计算，终于得出了每个地图路线的方程，每个地图的都分了很多段，然后把这些方程用代码表现出来也是试了好多次，这只能打表。最后再把不能放置炮塔的格子一个一个挑出来，也是只能打表封成函数，这个函数应该是全部函数中最考验数学功底和耐心的函数了。

attack 函数难题(炮塔对怪物进行攻击的函数)：8 种攻击型炮塔，攻击力，攻击方式，攻击范围各不相同，有的还有 debuff，部分数值只能用 if-else 结构打表封成函数，这还算只是时间的问题。在执行攻击的主体部分，刚开始就被单体攻击难住了——怎么写锁定敌人的 AI？按照一般思想来说，最有威胁的怪物是位置最靠前的怪物，因此需要比较怪物的位置参数，也就是说怪物不只需要遍历一遍。刚开始写的时候，发现绘制出的子弹会异常跳动，有时会无缘无故停止攻击，试了好久才最后发现，原来忘记考虑了攻击范围，当初的写法完全是错的。于是只能把函数推翻，删了重写。后来想到了好多方案，包括先扣数值再比较然后恢复部分数值，最后选择了还是先比较一遍再扣数值，又定义了一个数组 mstin 来解决范围问题，试了多次，终于能够锁定敌人了。然后紧接着就是炮塔的攻击问题，刚开始我们是没有考虑到子弹动画和造成伤害的问题的，后来冥思苦想才想出了在炮塔结构体中加个 clock 内置时钟来计时，巧妙解决了不连续攻击的问题，还有溅射伤害的问题等等，都是想了好多方案，试了很多次才测出最可行的方案，这个函数调试了 3 天左右，是最考验逻辑思维的函数。

鼠标卡顿问题：游戏主循环采用了精确帧率控制，在画面准备完成后，批量绘图前，有一步等待循环，鼠标操作就放在里面，结果到了后期鼠标直接失灵了，无法操作，这是不能接受的漏洞，关键是当初不知道怎么解决，

后来发现是运算量过大，拾取鼠标消息的步骤被跳过了，只好把失去鼠标消息的步骤加几次到准备画面的代码中，又大幅增加了传指针的占比，最后虽然还会有卡顿，但不影响操作了，这也是我们能想到的力所能及的解决卡顿的方法。

……

其实敲代码的过程中遇到的 bug 数不胜数，包括数组越界，逻辑漏洞甚至错别字等等，好在我们准备的还算早，时间还足够解决问题。虽然最终成品可能还会有隐藏的 bug，但是不会有影响游戏体验的恶性 bug 出现了。

6. 心得体会

本学期的程设大作业是我大一学年最难忘的经历。作为一个零编程基础的小白，第一次课上看到大作业十分惶恐，中间又经历了转组，五一期间还遭受了组长退课的打击，组内鸦雀无声。离 ddl 半个月时，我决定不再等下去，开始反复翻看复习刘老师和陈助教的 PPT，又斥 128r 巨资买了 easyx 图形界面的课程，马不停蹄学了一星期左右，敲了一个简陋的主菜单，然后一敲就停不下来。我把成果发到了小组群，组员们也开始浮出水面，群里渐渐开始有人说话了。我朋友圈的好友们也非常期待这个游戏，期待能够当上“内测玩家”。于是我有了坚持的动力，开始日复一日赶工，在宿舍、图书馆~~，甚至微积分课上~~苦干硬干了 12 天。中间大大小小的 bug 真的很折磨人，在 5 月 23 号凌晨 2 点我一时兴起，从床上爬起来盯着屏幕盯到 4 点，只为把核心函数的 bug 给 de 掉。虽然代码“一镜到底”全是我写的，但少不了组员在逻辑链条、函数设计、数值平衡、代码构思乃至心理上的帮助。当 2500 多行代码写完的那一刻，我获得了一种前所未有的成就感，看似遥不可及的目标最终真的变成了现实！这就是这门课带给我的成长，老师教会我的不仅是代码，还有毅力和自信心。

张赫阳　2023 年 5 月 29 日

通过这次大作业，我对于开发游戏的难度有了更加深刻的了解。刚开始做这项作业，我还以为只是定义几个变量，写几个函数而已。真正开始做后才发现有许多的问题需要解决，如何初始化图形界面，如何获取鼠标信息，如何控制帧数……一堆问题扑面而来，但是最后把这些问题一个个解决，做出一个有模有样的游戏的感觉真的好。

何宇晖　2023 年 5 月 29 日

这次写大作业是我第一次参与到一款游戏的制作里面，在得知我们小组要做的游戏原型是保卫萝卜这款塔防游戏时，我还特意去把它下载回来玩了一遍，去探索游戏策划是怎么做到保证一款游戏的平衡性的。

我发现只有真正去做一款游戏，去读这个游戏写的代码时，才会觉得一个能跑起来而且没有明显 bug 的游戏背后原来需要付出这么多心血。

这次写大作业的过程中，虽然有很多部分代码不是我写的，但是在读代码的过程中还是感觉到自己之前学的内容有一个深度融合，能够更好地理解它们的含义与用法。

傅家俊　2023 年 5 月 30 日

本学期我头脑发热，同时学着 3 种语言（C，python 和必修的 MATLAB），真正体会到了程序员的不易。当我把三种语言串烧使用然后出现 compilation error 时，脑子里蹦出来的第一句话总是“我真傻，真的，我单单知道 python 可以这么写，我不知道 C 语言不能”。在 debug 的时候找了半天没找到这样的 bug，也是一件十分搞心态的事情。在此感谢全世界的程序员们，你们辛苦了！衷心祝愿所有的程序员 debug 顺利，不要脱发(bushi)

赵思奇 2023 年 6 月 1 日

7. 源代码

```
//我们确认本程序完全由本组独立完成
//姓名（按首字母）:傅家俊、何宇晖、张赫阳、赵思奇
//学号: 2022013367，2022012813, 2022012816, 2021010268
//时间:2023 年 5 月 29 日

#include <stdio.h>//标准输入输出库
#include <easyx.h>//绘图库
#include <math.h>//数学函数库
#include <time.h>//计时函数库
```

```
#define sq2 1.414//根号2，计算怪物路线所需
#define pi 3.14//圆周率，计算与圆形有关的数据所需
#define f 30//(最高)帧率
#define inm 640//初始金钱

typedef struct { int a; int b; } NODE;//网格结构体(列，行)
typedef struct { int type;  double t; double v; double hp; } MST;//怪物结构体(种类，位置参数，速度，血量)
typedef struct { int type; int lv; int clock; }TRT;//炮塔结构体(种类，等级，钟表)

void diytext(char* s, int x, int y, int x1, int y1, int x2, int y2, COLORREF color) {
    setlinecolor(BLACK);
    setlinestyle(PS_SOLID, 4);
    setfillcolor(color);
    fillroundrect(x1, y1, x2, y2, 20, 20);

    settextcolor(BLACK);
    settextstyle(25, 0, "幼圆");
    setbkmode(TRANSPARENT);
    LOGFONT fontStyle;
    gettextstyle(&fontStyle);
    fontStyle.lfQuality = ANTIALIASED_QUALITY;
    settextstyle(&fontStyle);
    outtextxy(x, y, s);//绘制按钮类型0
}
```

```
    void diytext1(char* s, int x, int y, int x1, int y1, int x2,
int y2, COLORREF color) {
        setlinecolor(BLACK);
        setlinestyle(PS_SOLID, 4);
        setfillcolor(color);
        fillellipse(x1, y1, x2, y2);
        settextstyle(40, 0, "华文琥珀");
        setbkmode(TRANSPARENT);
        LOGFONT fontStyle;
        gettextstyle(&fontStyle);
        fontStyle.lfQuality = ANTIALIASED_QUALITY;
        settextstyle(&fontStyle);
        outtextxy(x, y, s);//绘制按钮类型 1
    }

    void diytext2(char* s, int x, int y, int x1, int y1, int x2,
int y2, COLORREF color) {
        setlinecolor(BLACK);
        setlinestyle(PS_SOLID, 4);
        setfillcolor(color);
        fillroundrect(x1, y1, x2, y2, 20, 20);

        settextcolor(BLACK);
        settextstyle(20, 0, "幼圆");
        setbkmode(TRANSPARENT);
        LOGFONT fontStyle;
        gettextstyle(&fontStyle);
        fontStyle.lfQuality = ANTIALIASED_QUALITY;
        settextstyle(&fontStyle);
```

```
        outtextxy(x, y, s);//绘制按钮类型 2
    }

    int menu() {
        mciSendString("open music/音乐 0.mp3", NULL, 0, NULL);
        mciSendString("play music/音乐 0.mp3 repeat", NULL, 0,
NULL);//播放初始页面音乐
        IMAGE Background0;
        loadimage(&Background0, "img/背景 0.jpg");
        putimage(0, 0, &Background0);//背景图片

        setlinecolor(BLACK);
        setlinestyle(PS_SOLID, 4);
        setfillcolor(GREEN);
        fillroundrect(398, 364, 592, 615, 30, 30);//菜单背景

        char defend[] = "保 卫 清 华", start[] = "开 始 游 戏",
help[] = "游 戏 帮 助", kick[] = "关 闭 游 戏";
        settextcolor(BLACK);
        settextstyle(100, 0, "方正舒体");
        setbkmode(TRANSPARENT);
        LOGFONT fontStyle;
        gettextstyle(&fontStyle);
        fontStyle.lfQuality = ANTIALIASED_QUALITY;
        settextstyle(&fontStyle);
        outtextxy(271, 130, defend);//显示游戏名
        diytext(start, 419, 399, 414, 393, 566, 434, RGB(255, 201,
14));
        diytext(help, 419, 481, 414, 475, 566, 516, RGB(255, 201,
```

```
14));
        diytext(kick, 419, 556, 414, 550, 566, 591, RGB(255, 201,
14));//生成按钮

        ExMessage msg;//鼠标交互
        BeginBatchDraw();
        while (1) {
            bool isok = peekmessage(&msg, EX_MOUSE);
            if (isok == true) {
                if (msg.x >= 414 && msg.x <= 566 && msg.y >= 393 &&
msg.y <= 434) {
                    diytext(start, 419, 399, 414, 393, 566, 434,
RGB(255, 255, 0));
                    FlushBatchDraw();
                    if (msg.message == WM_LBUTTONDOWN) {
                        EndBatchDraw();
                        return 0;
                    }
                }

                else {
                    diytext(start, 419, 399, 414, 393, 566, 434,
RGB(255, 201, 14));
                    FlushBatchDraw();
                }

                if (msg.x >= 414 && msg.x <= 566 && msg.y >= 475 &&
msg.y <= 516) {
                    diytext(help, 419, 481, 414, 475, 566, 516,
```

```
RGB(255, 255, 0));
                FlushBatchDraw();
                if (msg.message == WM_LBUTTONDOWN) {
                    EndBatchDraw();
                    return 1;
                }
            }

            else {
                diytext(help, 419, 481, 414, 475, 566, 516,
RGB(255, 201, 14));
                FlushBatchDraw();
            }

            if (msg.x >= 414 && msg.x <= 566 && msg.y >= 550 &&
msg.y <= 591) {
                diytext(kick, 419, 556, 414, 550, 566, 591,
RGB(255, 255, 0));
                FlushBatchDraw();
                if (msg.message == WM_LBUTTONDOWN) {
                    exit(0);
                }
            }

            else {
                diytext(kick, 419, 556, 414, 550, 566, 591,
RGB(255, 201, 14));
                FlushBatchDraw();
            }
```

```
        }
    }
}//显示主菜单

int map() {
    setlinecolor(BLACK);
    setlinestyle(PS_SOLID, 4);
    setfillcolor(GREEN);
    fillroundrect(200, 150, 800, 630, 30, 30);
    setfillcolor(RGB(255, 201, 14));
    int i, j;
    for (i = 0; i < 2; i++) {
        for (j = 0; j < 3; j++) {
            fillellipse(220 + i * 300, 160 + j * 150, 480 + i *
300, 290 + j * 150);
        }
    }
    char level1[3][20] = { "C     楼", "图  书  馆", "网 课 模
式" };
    char level2[3][20] = { "三    教", "六    教", "无 尽 模
式" };
    settextcolor(BLACK);
    settextstyle(40, 0, "华文琥珀");
    setbkmode(TRANSPARENT);
    LOGFONT fontStyle;
    gettextstyle(&fontStyle);
    fontStyle.lfQuality = ANTIALIASED_QUALITY;
    settextstyle(&fontStyle);
    for (i = 0; i < 2; i++) {
```

```
            for (j = 0; j < 3; j++) {
                if (i == 0) {
                    outtextxy(270, 210 + j * 150, level1[j]);
                }
                else {
                    outtextxy(570, 210 + j * 150, level2[j]);
                }
            }
        }
        char back[] = "  返    回";
        diytext(back, 5, 6, 0, 0, 152, 41, RGB(255, 201, 14));
        ExMessage msg;
        BeginBatchDraw();
        while (1) {
            bool isok = peekmessage(&msg, EX_MOUSE);
            if (isok == true) {
                if (msg.x >= 220 && msg.x <= 480 && msg.y >= 160 &&
msg.y <= 290) {
                    diytext1(level1[0], 270, 210, 220, 160, 480,
290, RGB(255, 255, 0));
                    FlushBatchDraw();
                    if (msg.message == WM_LBUTTONDOWN) {
                        EndBatchDraw();
                        return 1;//C楼
                    }
                }

                else {
                    diytext1(level1[0], 270, 210, 220, 160, 480,
```

```
290, RGB(255, 201, 14));
                FlushBatchDraw();
            }

            if (msg.x >= 220 && msg.x <= 480 && msg.y >= 310 &&
msg.y <= 440) {
                diytext1(level1[1], 270, 360, 220, 310, 480,
440, RGB(255, 255, 0));
                FlushBatchDraw();
                if (msg.message == WM_LBUTTONDOWN) {
                    EndBatchDraw();
                    return 2;//图书馆
                }
            }

            else {
                diytext1(level1[1], 270, 360, 220, 310, 480,
440, RGB(255, 201, 14));
                FlushBatchDraw();
            }

            if (msg.x >= 220 && msg.x <= 480 && msg.y >= 460 &&
msg.y <= 590) {
                diytext1(level1[2], 270, 510, 220, 460, 480,
590, RGB(255, 255, 0));
                FlushBatchDraw();
                if (msg.message == WM_LBUTTONDOWN) {
                    EndBatchDraw();
                    return 3;//网课模式
```

```
            }
        }

        else {
            diytext1(level1[2], 270, 510, 220, 460, 480,
590, RGB(255, 201, 14));
            FlushBatchDraw();
        }

        if (msg.x >= 520 && msg.x <= 780 && msg.y >= 160 &&
msg.y <= 290) {
            diytext1(level2[0], 570, 210, 520, 160, 780,
290, RGB(255, 255, 0));
            FlushBatchDraw();
            if (msg.message == WM_LBUTTONDOWN) {
                EndBatchDraw();
                return 4; // 三教
            }
        }

        else {
            diytext1(level2[0], 570, 210, 520, 160, 780,
290, RGB(255, 201, 14));
            FlushBatchDraw();
        }

        if (msg.x >= 520 && msg.x <= 780 && msg.y >= 310 &&
msg.y <= 440) {
            diytext1(level2[1], 570, 360, 520, 310, 780,
```

```
440, RGB(255, 255, 0));
                FlushBatchDraw();
                if (msg.message == WM_LBUTTONDOWN) {
                    EndBatchDraw();
                    return 5;//六教
                }
            }

            else {
                diytext1(level2[1], 570, 360, 520, 310, 780,
440, RGB(255, 201, 14));
                FlushBatchDraw();
            }

            if (msg.x >= 520 && msg.x <= 780 && msg.y >= 460 &&
msg.y <= 590) {
                diytext1(level2[2], 570, 510, 520, 460, 780,
590, RGB(255, 255, 0));
                FlushBatchDraw();
                if (msg.message == WM_LBUTTONDOWN) {
                    EndBatchDraw();
                    return 6;//无尽模式
                }
            }

            else {
                diytext1(level2[2], 570, 510, 520, 460, 780,
590, RGB(255, 201, 14));
                FlushBatchDraw();
```

```
            }

            if (msg.x >= 0 && msg.x <= 152 && msg.y >= 0 &&
msg.y <= 41) {
                diytext(back, 5, 6, 0, 0, 152, 41, RGB(255, 255,
0));
                FlushBatchDraw();
                if (msg.message == WM_LBUTTONDOWN) {
                    EndBatchDraw();
                    return 0;//map 函数返回 0，回到主界面
                }
            }

            else {
                diytext(back, 5, 6, 0, 0, 152, 41, RGB(255, 201,
14));
                FlushBatchDraw();
            }
        }
    }
}//选关界面

void initmst(MST mst[25]) {
    int i = 0;
    for (i = 0; i < 25; i++) {
        mst[i].type = 0;
        mst[i].t = 0;
        mst[i].v = 0;
        mst[i].hp = 0;
```

```
        }
    }//初始化所有怪物

    void inittrt(TRT trt[10][7]) {
        int i = 0, j = 0;
        for (i = 0; i < 10; i++) {
            for (j = 0; j < 7; j++) {
                trt[i][j].lv = 0;
                trt[i][j].type = 0;
                trt[i][j].clock = 0;
            }
        }
    }//初始化所有炮塔

    void PlayerHelp() {
        int page;
        char up[] = "上 一 页", down[] = "下 一 页", back[] = "返    回";
        while (1) {
            page = 1;
            if (page == 1) {
                setfillcolor(RGB(255, 201, 14));
                fillroundrect(160, 80, 840, 670, 30, 30);
                char helptext0[] = "                欢迎来到保卫清华";
                char helptext1[] = "1.放置各种防御塔阻止怪物到达二校门，保卫清華圜";
                char helptext2[] = "2.左键点击空格子，若金钱足够，可选择放置防御塔，";
```

```
        char helptext3[] = "3.左键点击防御塔，若金钱足够，可花费金钱升级";
        char helptext4[] = "4.右键点击防御塔，可选择拆除，拆除后返还部分金钱";
        char helptext5[] = "5.升级可以强化防御塔的能力，满级 2 级(无尽关卡为 3 级)";
        char helptext6[] = "6.可以通过击杀怪物获得资金，咖啡机可持续产生资金";
        char helptext7[] = "7.网课和无尽模式会出现掉线事件，防御塔将停止攻击";
        char helptext8[] = "8.怪物有三种不同的类型，每波 20 个，血量随波数递增";
        char helptext9[] = "9.退出关卡或重启游戏后不会存档，敬请谅解！";

        settextstyle(40, 0, "华文琥珀");
        setbkmode(TRANSPARENT);
        LOGFONT fontStyle;
        gettextstyle(&fontStyle);
        fontStyle.lfQuality = ANTIALIASED_QUALITY;
        settextstyle(&fontStyle);
        outtextxy(180, 90, helptext0);
        settextstyle(25, 0, "宋体");
        outtextxy(180, 150, helptext1);
        outtextxy(180, 200, helptext2);
        outtextxy(180, 250, helptext3);
        outtextxy(180, 300, helptext4);
        outtextxy(180, 350, helptext5);
        outtextxy(180, 400, helptext6);
```

```
                outtextxy(180, 450, helptext7);
                outtextxy(180, 500, helptext8);
                settextcolor(RED);
                outtextxy(180, 550, helptext9);

                diytext(down, 450, 590, 430, 580, 570, 625,
RGB(245, 215, 215));
                ExMessage msg;//鼠标交互
                BeginBatchDraw();
                while (1) {
                    bool isok = peekmessage(&msg, EX_MOUSE);
                    if (isok == true) {
                        if (msg.x >= 430 && msg.x <= 570 && msg.y >=
580 && msg.y <= 625) {
                            diytext(down, 450, 590, 430, 580, 570,
625, RGB(255, 255, 125));
                            FlushBatchDraw();
                            if (msg.message == WM_LBUTTONDOWN) {
                                EndBatchDraw();
                                page = 2;
                                break;
                            }
                        }

                        else {
                            diytext(down, 450, 590, 430, 580, 570,
625, RGB(245, 215, 215));
                            FlushBatchDraw();
                        }
```

```
                    if (msg.x >= 0 && msg.x <= 152 && msg.y >= 0
&& msg.y <= 41) {
                        diytext(back, 5, 6, 0, 0, 152, 41,
RGB(255, 255, 0));
                        FlushBatchDraw();
                        if (msg.message == WM_LBUTTONDOWN) {
                            EndBatchDraw();
                            return;//PlayerHelp 函数返回，回到主界
面
                        }
                    }

                    else {
                        diytext(back, 5, 6, 0, 0, 152, 41,
RGB(255, 201, 14));
                        FlushBatchDraw();
                    }
                }
            }
        }

        if (page == 2) {
            setfillcolor(RGB(255, 201, 14));
            fillroundrect(160, 80, 840, 670, 30, 30);
            char helptext9[] = "                        防 御 塔
介 绍";
            char helptext10[] = "1.黑笔：最普通的防御塔，和清华
的同学们一样普通";
```

```
char helptext11[] = "2.红笔：背负着老师的怒火，可以点燃目标造成持续伤害";
char helptext12[] = "3.铅笔：有几率击退目标半格，击退几率随等级提升而增大";
char helptext13[] = "4.直尺：可以对怪物造成巨大单体伤害，对 PKU 伤害翻倍";
char helptext14[] = "5.三角尺：借助工程制图的神秘力量，对怪物造成溅射伤害";
char helptext15[] = "6.量角器：这是填空题的神器，可对怪物造成溅射减速伤害";
char helptext16[] = "7.圆规：对圆形范围内所有怪物造成同等 AOE 伤害";
char helptext17[] = "8.橡皮擦：可以改变过去的神器，借助时间力量使怪物减速";
char helptext18[] = "9.wifi：可以在网课模式和无尽模式拯救你的校园网";

settextstyle(40, 0, "华文琥珀");
setbkmode(TRANSPARENT);
LOGFONT fontStyle1;
gettextstyle(&fontStyle1);
fontStyle1.lfQuality = ANTIALIASED_QUALITY;
settextstyle(&fontStyle1);
outtextxy(180, 90, helptext9);
settextstyle(25, 0, "宋体");
outtextxy(180, 150, helptext10);
outtextxy(180, 200, helptext11);
outtextxy(180, 250, helptext12);
outtextxy(180, 300, helptext13);
```

```
                outtextxy(180, 350, helptext14);
                outtextxy(180, 400, helptext15);
                outtextxy(180, 450, helptext16);
                outtextxy(180, 500, helptext17);
                outtextxy(180, 550, helptext18);

                diytext(down, 450, 590, 430, 580, 570, 625,
RGB(245, 215, 215));
                ExMessage msg1;//鼠标交互
                BeginBatchDraw();
                while (1) {
                    bool isok1 = peekmessage(&msg1, EX_MOUSE);
                    if (isok1 == true) {
                        if (msg1.x >= 430 && msg1.x <= 570 &&
msg1.y >= 580 && msg1.y <= 625) {
                            diytext(up, 450, 590, 430, 580, 570, 625,
RGB(255, 255, 125));
                            FlushBatchDraw();
                            if (msg1.message == WM_LBUTTONDOWN) {
                                EndBatchDraw();
                                page = 1;
                                break;
                            }
                        }

                        else {
                            diytext(up, 450, 590, 430, 580, 570, 625,
RGB(245, 215, 215));
                            FlushBatchDraw();
```

```
                    }

                    if (msg1.x >= 0 && msg1.x <= 152 &&
msg1.y >= 0 && msg1.y <= 41) {
                        diytext(back, 5, 6, 0, 0, 152, 41,
RGB(255, 255, 0));
                        FlushBatchDraw();
                        if (msg1.message == WM_LBUTTONDOWN) {
                            EndBatchDraw();
                            return;//PlayerHelp 函数返回，回到主界
面
                        }
                    }

                    else {
                        diytext(back, 5, 6, 0, 0, 152, 41,
RGB(255, 201, 14));
                        FlushBatchDraw();
                    }
                }
            }
        }
    }//显示游戏帮助

    void playmusic(int *choice1) {
        if (*choice1 == 1) {
            mciSendString("open music/音乐 1.mp3", NULL, 0, NULL);
            mciSendString("play music/音乐 1.mp3 repeat", NULL, 0,
```

```
NULL);
        }
        else if (*choice1 == 2) {
            mciSendString("open music/音乐2.mp3", NULL, 0, NULL);
            mciSendString("play music/音乐2.mp3 repeat", NULL, 0,
NULL);
        }
        else if (*choice1 == 3) {
            mciSendString("open music/音乐3.mp3", NULL, 0, NULL);
            mciSendString("play music/音乐3.mp3 repeat", NULL, 0,
NULL);
        }
        else if (*choice1 == 4) {
            mciSendString("open music/音乐4.mp3", NULL, 0, NULL);
            mciSendString("play music/音乐4.mp3 repeat", NULL, 0,
NULL);
        }
        else if (*choice1 == 5) {
            mciSendString("open music/音乐5.mp3", NULL, 0, NULL);
            mciSendString("play music/音乐5.mp3 repeat", NULL, 0,
NULL);
        }
        else if (*choice1 == 6) {
            mciSendString("open music/音乐6.mp3", NULL, 0, NULL);
            mciSendString("play music/音乐6.mp3 repeat", NULL, 0,
NULL);
        }
    }//播放关卡背景音乐
```

```
    void closemusic(int *choice1) {
        if (*choice1 == 1) {
            mciSendString("close music/音乐1.mp3", NULL, 0, NULL);
        }
        else if (*choice1 == 2) {
            mciSendString("close music/音乐2.mp3", NULL, 0, NULL);
        }
        else if (*choice1 == 3) {
            mciSendString("close music/音乐3.mp3", NULL, 0, NULL);
        }
        else if (*choice1 == 4) {
            mciSendString("close music/音乐4.mp3", NULL, 0, NULL);
        }
        else if (*choice1 == 5) {
            mciSendString("close music/音乐5.mp3", NULL, 0, NULL);
        }
        else if (*choice1 == 6) {
            mciSendString("close music/音乐6.mp3", NULL, 0, NULL);
        }
    }//关闭关卡背景音乐

    void playtrtsound(TRT trt, int i, int j) {
        char soundname[40];
        sprintf_s(soundname, "open sound/attack%d/%d.mp3",
trt.type, i * 7 + j + 1);
        mciSendString(soundname, NULL, 0, NULL);
        sprintf_s(soundname, "play sound/attack%d/%d.mp3",
trt.type, i * 7 + j + 1);
        mciSendString(soundname, NULL, 0, NULL);
```

```
}//播放炮塔攻击音效

void closetrtsound(TRT trt, int i, int j) {
    char soundname[40];
    sprintf_s(soundname, "close sound/attack%d/%d.mp3",
trt.type, i * 7 + j + 1);
    mciSendString(soundname, NULL, 0, NULL);
    mciSendString(soundname, NULL, 0, NULL);
}//关闭炮塔攻击音效

NODE xytonode(int x, int y) {
    NODE node = {0, 0};
    if (y <= 700) {
        int x1 = floor(x / 100.0);
        int y1 = floor(y / 100.0);
        node.a = x1 + 1;
        node.b = y1 + 1;
    }
    else {
        node.a = 0;
        node.b = 0;
    }
    return node;
}//像素坐标转化为网格坐标

POINT nodetoxy(NODE node) {
    POINT p = {-100, -100};
    p.x = (node.a - 1) * 100;
    p.y = (node.b - 1) * 100;
```

```
    return p;
}//网格坐标转化为像素坐标（按网格左上角）

int lvmax(int *choice1) {
    if (*choice1 == 6) {
        return 3;
    }
    else {
        return 2;
    }
}//根据选择关卡的不同确定炮塔的满级

int msthp(int type, int* wave) {
    if (type == 1) {
        return (10 + 30 * *wave);
    }
    else if (type == 2) {
        return(5 + *wave * 15);
    }
    else if (type == 3) {
        return(2000 + *wave * 500);
    }
}//根据怪物种类及波数返回血量（满血）

double tmax(int* choice1) {
    switch (*choice1) {
    case 1:
        return 2130;
        break;
```

```
    case 2:
        return 2100;
        break;
    case 3:
        return 2000;
        break;
    case 4:
        return 2220;
        break;
    case 5:
        return 1100 * sq2 + 500;
        break;
    case 6:
        return 2000;
        break;
    default:
        return 0;
        break;
    }
}//每个地图 t 的最大值

int cost(int type, int lv) {
    if (type == 0) {
        return 0;
    }

    else if (type == 1) {
        if (lv == 1) {
            return 100;
```

```
    }
    else if (lv == 2) {
        return 180;
    }
    else if (lv == 3) {
        return 260;
    }
    else {
        return 10000;
    }
}

else if (type == 2) {
    if (lv == 1) {
        return 160;
    }
    else if (lv == 2) {
        return 260;
    }
    else if (lv == 3) {
        return 320;
    }
    else {
        return 10000;
    }
}

else if (type == 3 || type == 6 || type == 7) {
    if (lv == 1) {
```

```
            return 180;
        }
        else if (lv == 2) {
            return 260;
        }
        else if (lv == 3) {
            return 320;
        }
        else {
            return 10000;
        }
    }

    else if (type == 4) {
        if (lv == 1) {
            return 420;
        }
        else if (lv == 2) {
            return 560;
        }
        else if (lv == 3) {
            return 680;
        }
        else {
            return 10000;
        }
    }

    else if (type == 5) {
```

```
    if (lv == 1) {
        return 160;
    }
    else if (lv == 2) {
        return 220;
    }
    else if (lv == 3) {
        return 260;
    }
    else {
        return 10000;
    }
}

else if (type == 8) {
    if (lv == 1) {
        return 120;
    }
    else if (lv == 2) {
        return 220;
    }
    else if (lv == 3) {
        return 260;
    }
    else {
        return 10000;
    }
}
```

```
else if (type == 9) {
    if (lv == 1) {
        return 600;
    }
    else if (lv == 2) {
        return 800;
    }
    else if (lv == 3) {
        return 1000;
    }
    else {
        return 10000;
    }
}

else if (type == 10) {
    if (lv == 1) {
        return 260;
    }
    else if (lv == 2) {
        return 0;
    }
    else if (lv == 3) {
        return 0;
    }
    else {
        return 10000;
    }
}
```

```
        else {
            return 0;
        }
    }//根据炮塔等级返回放置/升级所需金钱

    POINT txy1(int t) {
        int x = -100, y = -100;
        if (t < 278) {
            x = 900 - t;
            y = 101;
        }
        else if (t >= 278 && t < 1530) {
            x = 484 + 277 * cos((t - 278) * pi / 4.83 / 180 + pi /
3);
            y = 341 - 277 * sin((t - 278) * pi / 4.83 / 180 + pi /
3);
        }
        else if (t >= 1530 && t < 1630) {
            x = 693 + (t - 1530);
            y = 522 - (t - 1530) / 5;
        }
        else if (t >= 1630 && t < 1830) {
            x = 793 + (t - 1630) * 7 / 200;
            y = 503 - (t - 1630) * 3 / 200 - (t - 1630);
        }
        else if (t >= 1830 && t <= 2130) {
            x = 800 - (t - 1830);
            y = 300;
```

```
    }
    else {
        x = -1000;
        y = -1000;
    }
    POINT P = { -100, -100 };
    P.x = x;
    P.y = y;
    return P;
}   //C楼地图的参数方程

POINT txy2(int t) {
    int x = -100, y = -100;
    if (t < 300) {
        x = 200 + t;
        y = 600;
    }
    else if (t >= 300 && t < 500) {
        x = 500;
        y = 600 - (t - 300);
    }
    else if (t >= 500 && t < 800) {
        x = 500 - (t - 500);
        y = 400;
    }
    else if (t >= 800 && t < 1000) {
        x = 200;
        y = 400 - (t - 800);
    }
```

```
    else if (t >= 1000 && t < 1600) {
        x = 200 + (t - 1000);
        y = 200;
    }
    else if (t >= 1600 && t < 1800) {
        x = 800;
        y = 200 + (t - 1600);
    }
    else if (t >= 1800 && t < 2000) {
        x = 800 - (t - 1800);
        y = 400;
    }
    else if (t >= 2000 && t < 2100) {
        x = 600;
        y = 400 + (t - 2000);
    }
    else {
        x = -1000;
        y = -1000;
    }
    POINT P = { -100, -100 };
    P.x = x;
    P.y = y;
    return P;
}//图书馆地图的参数方程

POINT txy3(int t) {
    int x = -100, y = -100;
    if (t < 500) {
```

```
    x = 700 - t;
    y = 600;
}
else if (t >= 500 && t < 900) {
    x = 200;
    y = 600 - (t - 500);
}
else if (t >= 900 && t < 1000) {
    x = 200 + (t - 900);
    y = 200;
}
else if (t >= 1000 && t < 1100) {
    x = 300;
    y = 200 - (t - 1000);
}
else if (t >= 1100 && t < 1500) {
    x = 300 + (t - 1100);
    y = 100;
}
else if (t >= 1500 && t < 1800) {
    x = 700;
    y = 100 + (t - 1500);
}
else if (t >= 1800 && t <= 2000) {
    x = 700 - (t - 1800);
    y = 400;
}
else {
    x = -1000;
```

```
        y = -1000;
    }
    POINT P = { -100, -100 };
    P.x = x;
    P.y = y;
    return P;
}//网课/无尽地图的参数方程

POINT txy4(int t) {
    int x = -100, y = -100;
    if (t < 400) {
        x = 800 - t;
        y = 500;
    }
    else if (t >= 400 && t < 600) {
        x = 400;
        y = 500 - (t - 400);
    }
    else if (t >= 600 && t < 1000) {
        x = 400 + (t - 600);
        y = 300;
    }
    else if (t >= 1000 && t < 1300) {
        x = 800;
        y = 300 - (t - 1000);
    }
    else if (t >= 1300 && t < 1800) {
        x = 800 - (t - 1300);
        y = 0;
```

```
    }
    else if (t >= 1800 && t < 2000) {
        x = 300;
        y = t - 1800;
    }
    else if (t >= 2000 && t < 2200) {
        x = 300 - (t - 2000);
        y = 200;
    }
    else {
        x = -1000;
        y = -1000;
    }
    POINT P = { -100, -100 };
    P.x = x;
    P.y = y;
    return P;
}//三教地图的参数方程

POINT txy5(int t) {
    int x = -100, y = -100;
    if (t < 200 * sq2) {
        x = 700 - t / sq2;
        y = t / sq2;
    }
    else if (t >= (200 * sq2) && t < (400 * sq2)) {
        x = 500 + (t / sq2 - 200);
        y = 200 + (t / sq2 - 200);
    }
```

```
        else if (t >= (400 * sq2) && t < (600 * sq2)) {
            x = 700 - (t / sq2 - 400);
            y = 400 + (t / sq2 - 400);
        }
        else if (t >= (600 * sq2) && t < (600 * sq2 + 100)) {
            x = 500 - (t - 600 * sq2);
            y = 600;
        }
        else if (t >= (600 * sq2 + 100) && t < (800 * sq2 + 100))
{
            x = 400 - ((t - 100) / sq2 - 600);
            y = 600 - ((t - 100) / sq2 - 600);
        }
        else if (t >= (800 * sq2 + 100) && t < (1100 * sq2 + 100))
{
            x = 200 + ((t - 100) / sq2 - 800);
            y = 400 - ((t - 100) / sq2 - 800);
        }
        else if (t >= (1100 * sq2 + 100) && t < (1100 * sq2 +
200)) {
            x = 500;
            y = 100 - (t - (1100 * sq2 + 100));
        }
        else if (t >= (1100 * sq2 + 200) && t < (1100 * sq2 +
500)) {
            x = 500 - (t - (1100 * sq2 + 200));
            y = 0;
        }
        else {
```

```
            x = -1000;
            y = -1000;
        }
        POINT P = { -100, -100 };
        P.x = x;
        P.y = y;
        return P;
    }//六教地图的参数方程

    int RG(int type, int lv) {
        if (type == 0 || lv == 0) {
            return 0;
        }

        else if (type == 1) {
            if (lv == 1) {
                return 210;
            }
            else if (lv == 2 || lv == 3) {
                return 250;
            }
            else {
                return 0;
            }
        }

        else if (type == 2 || type == 3 || type == 5 || type == 6)
{
            if (lv == 1) {
```

```
            return 160;
        }
        else if (lv == 2) {
            return 210;
        }
        else if (lv == 3) {
            return 250;
        }
        else {
            return 0;
        }
    }

    else if (type == 4) {
        if (lv == 1) {
            return 250;
        }
        else if (lv == 2) {
            return 290;
        }
        else if (lv == 3) {
            return 390;
        }
        else {
            return 0;
        }
    }

    else if (type == 7) {
```

```
        if (lv == 1) {
            return 160;
        }
        else if (lv == 2) {
            return 210;
        }
        else if (lv == 3) {
            return 210;
        }
        else {
            return 0;
        }
    }

    else if (type == 8) {
        if (lv == 1) {
            return 180;
        }
        else if (lv == 2) {
            return 220;
        }
        else if (lv == 3) {
            return 280;
        }
        else {
            return 0;
        }
    }
}//根据炮塔类型和等级计算炮塔射程
```

```
double DMG(int type, int lv, int spt) {
    if (type == 0) {
        return 0;
    }

    else if (type == 1) {
        if (lv == 1) {
            return 4.0;
        }
        else if (lv == 2) {
            return 6.0;
        }
        else if (lv == 3) {
            return 9.0;
        }
        else {
            return 0;
        }
    }

    else if (type == 2) {
        if (lv == 1) {
            return 24.0 / f;
        }
        else if (lv == 2) {
            return 47.0 / f;
        }
        else if (lv == 3) {
```

```
        return 54.0 / f;
    }
    else {
        return 0;
    }
}

else if (type == 3) {
    if (lv == 1) {
        return 12;
    }
    else if (lv == 2) {
        return 14;
    }
    else if (lv == 3) {
        return 17;
    }
    else {
        return 0;
    }
}

else if (type == 4) {
    if (lv == 1) {
        return 48.0;
    }
    else if (lv == 2) {
        return 74.0;
    }
```

```
    else if (lv == 3) {
        return 92.0;
    }
    else {
        return 0;
    }
}

else if (type == 5 && spt == 0) {//spt为0表示不是溅射
    if (lv == 1) {
        return 21.0;
    }
    else if (lv == 2) {
        return 36.0;
    }
    else if (lv == 3) {
        return 48.0;
    }
    else {
        return 0;
    }
}

else if (type == 5 && spt == 1) {//spt为1表示溅射
    if (lv == 1) {
        return 21.0 * 0.2;
    }
    else if (lv == 2) {
        return 36.0 * 0.3;
```

```
    }
    else if (lv == 3) {
        return 48.0 * 0.4;
    }
    else {
        return 0;
    }
}

else if (type == 6 && spt == 0) {
    if (lv == 1) {
        return 15.0;
    }
    else if (lv == 2) {
        return 22.0;
    }
    else if (lv == 3) {
        return 36.0;
    }
    else {
        return 0;
    }
}

else if (type == 6 && spt == 1) {
    if (lv == 1) {
        return 15.0 * 0.2;
    }
    else if (lv == 2) {
```

```
        return 22.0 * 0.3;
    }
    else if (lv == 3) {
        return 36.0 * 0.4;
    }
    else {
        return 0;
    }
}

else if (type == 7) {
    if (lv == 1) {
        return 16.0;
    }
    else if (lv == 2) {
        return 23.0;
    }
    else if (lv == 3) {
        return 31.0;
    }
    else {
        return 0;
    }
}

else if (type == 8) {
    if (lv == 1) {
        return 1.0;
    }
```

```
        else if (lv == 2) {
            return 3.0;
        }
        else if (lv == 3) {
            return 5.0;
        }
        else {
            return 0;
        }
    }
    else {
        return 0;
    }
}//根据炮塔类型和等级计算炮塔伤害

POINT PM(MST mst, int* choice1) {
    POINT pm = { 0,0 };
    if (mst.t != 0) {
        if (*choice1 == 1) {
            pm = { txy1(mst.t).x + 50, txy1(mst.t).y + 50 };
        }
        else if (*choice1 == 2) {
            pm = { txy2(mst.t).x + 50, txy2(mst.t).y + 50 };
        }
        else if (*choice1 == 3 || *choice1 == 6) {
            pm = { txy3(mst.t).x + 50, txy3(mst.t).y + 50 };
        }
        else if (*choice1 == 4) {
            pm = { txy4(mst.t).x + 50, txy4(mst.t).y + 50 };
```

```
        }
        else if (*choice1 == 5) {
            pm = { txy5(mst.t).x + 50, txy5(mst.t).y + 50 };
        }
    }
    return pm;
}//由怪物的时间参数算出怪物中心点坐标

int spd(TRT trt) {
    if (trt.type == 0) {
        return 2023;
    }
    else if (trt.type == 1) {
        return f / 4;
    }
    else if (trt.type == 2) {
        return 1;
    }
    else if (trt.type == 3) {
        return f / 2;
    }
    else {
        return f;
    }
}//根据炮塔类型算出攻频

double v(MST mst) {
    if (mst.type == 2) {
        return 5.33;
```

```
    }
    else if (mst.type == 1 || mst.type == 3) {
        return 2;
    }
    else {
        return 0;
    }
}//跟据怪物类型设置初速度

bool inroute(int choice1, NODE node) {
    if (choice1 == 1) {
        if (node.b == 1) {
            if (node.a >= 5 && node.a <= 7) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 2) {
            if (node.a >= 4 && node.a <= 10) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 3) {
            if (node.a == 3 || node.a == 4) {
```

```
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 4) {
            if (node.a == 3 || (node.a <= 9 && node.a >= 5)) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 5) {
            if (node.a == 3 || node.a == 9) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 6) {
            if (node.a == 3 || node.a == 4 || node.a == 8 ||
node.a == 9) {
                return true;
            }
            else {
                return false;
```

```
            }
        }
        else if (node.b == 7) {
            if (node.a <= 8 && node.a >= 4) {
                return true;
            }
            else {
                return false;
            }
        }
    }
    else if (choice1 == 2) {
        if (node.b == 1 || node.b == 2) {
            return false;
        }
        else if (node.b == 3) {
            if (node.a >= 3 && node.a <= 9) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 4) {
            if (node.a == 3 || node.a == 9) {
                return true;
            }
            else {
                return false;
```

```
            }
        }
        else if (node.b == 5) {
            if (node.a >= 3 && node.a <= 9) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 6) {
            if (node.a == 6 || node.a == 7) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 7) {
            if (node.a >= 3 && node.a <= 7) {
                return true;
            }
            else {
                return false;
            }
        }
    }
    else if (choice1 == 3 || choice1 == 6) {
        if (node.b == 1) {
```

```
    return false;
}
else if (node.b == 2) {
    if (node.a >= 4 && node.a <= 8) {
        return true;
    }
    else {
        return false;
    }
}
else if (node.b == 3) {
    if (node.a == 3 || node.a == 4 || node.a == 8) {
        return true;
    }
    else {
        return false;
    }
}
else if (node.b == 4) {
    if (node.a == 3 || node.a == 8) {
        return true;
    }
    else {
        return false;
    }
}
else if (node.b == 5) {
    if (node.a == 3 || (node.a >= 5 && node.a <= 8)) {
        return true;
```

```
            }
            else {
                return false;
            }
        }
        else if (node.b == 6) {
            if (node.a == 3) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 7) {
            if (node.a >= 3 && node.a <= 8) {
                return true;
            }
            else {
                return false;
            }
        }
    }
    else if (choice1 == 4) {
        if (node.b == 1) {
            if (node.a >= 4 && node.a <= 9) {
                return true;
            }
            else {
                return false;
```

```
    }
}
else if (node.b == 2) {
    if (node.a == 4 || node.a == 9) {
        return true;
    }
    else {
        return false;
    }
}
else if (node.b == 3) {
    if ((node.a >= 1 && node.a <= 4) || node.a == 9) {
        return true;
    }
    else {
        return false;
    }
}
else if (node.b == 4) {
    if (node.a >= 5 && node.a <= 8) {
        return true;
    }
    else {
        return false;
    }
}
else if (node.b == 5) {
    if (node.a == 5) {
        return true;
```

```
            }
            else {
                return false;
            }
        }
        else if (node.b == 6) {
            if (node.a >= 5 && node.a <= 9) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 7) {
            return false;
        }
    }
    else if (choice1 == 5) {
        if (node.b == 1) {
            if ((node.a >= 2 && node.a <= 6) || node.a == 8) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 2) {
            if (node.a >= 5 && node.a <= 7) {
                return true;
```

```
            }
            else {
                return false;
            }
        }
        else if (node.b == 3) {
            if (node.a >= 4 && node.a <= 6) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 4) {
            if (node.a == 3 || node.a == 4 || node.a == 6 ||
node.a == 7) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 5) {
            if (node.a == 3 || node.a == 4 || node.a == 7 ||
node.a == 8) {
                return true;
            }
            else {
                return false;
```

```
            }
        }
        else if (node.b == 6) {
            if (node.a >= 4 && node.a <= 7) {
                return true;
            }
            else {
                return false;
            }
        }
        else if (node.b == 7) {
            if (node.a >= 5 && node.a <= 7) {
                return true;
            }
            else {
                return false;
            }
        }
    }
}//判断鼠标是否在怪物行走路线上

bool allmstdie(MST mst[25]) {
    int i = 0, j = 0;
    for (i = 0; i < 20; i++) {
        if (mst[i].hp <= 0) {
            j += 1;
        }
    }
    if (j >= 20) {
```

```
        return true;
    }
    else {
        return false;
    }
}//判断怪物是否全部死亡

void drawmap(int HP, int *choice1, int *money, int *wave) {
    IMAGE map, carrot0, carrot1;
    int x = 0, y = 0;
    loadimage(&carrot0, "img/二校门0.png");
    loadimage(&carrot1, "img/二校门1.png");
    if (*choice1 == 1) {
        x = 400;
        y = 309;
        loadimage(&map, "img/map1.png");
        putimage(0, 0, &map);
    }
    else if (*choice1 == 2) {
        x = 600;
        y = 609;
        loadimage(&map, "img/map2.png");
        putimage(0, 0, &map);
    }
    else if (*choice1 == 3 || *choice1 == 6) {
        x = 400;
        y = 409;
        loadimage(&map, "img/map5.png");
        putimage(0, 0, &map);
```

```
        }
        else if (*choice1 == 4) {
            x = 0;
            y = 209;
            loadimage(&map, "img/map3.png");
            putimage(0, 0, &map);
        }
        else if (*choice1 == 5) {
            x = 100;
            y = 9;
            loadimage(&map, "img/map4.png");
            putimage(0, 0, &map);
        }
        putimage(x, y, &carrot0, SRCPAINT);
        putimage(x, y, &carrot1, SRCAND);

        setlinestyle(PS_SOLID, 2);
        setlinecolor(BLACK);
        setfillcolor(RED);
        fillrectangle(x, y - 9, x + 100, y - 4);
        setlinestyle(PS_SOLID, 1);
        setlinecolor(WHITE);
        setfillcolor(WHITE);
        double HPmax = (*choice1 == 6) ? 20 : 10;
        if (HP != HPmax) {
            fillrectangle(x + 100.0 * (HP / HPmax) - 1, y - 8, x +
99, y - 6);
        }
```

```
        int i = 0;
        setlinestyle(PS_SOLID, 2);
        setlinecolor(BLUE);
        for (i = 0; i < 8; i++) {
            line(1, 100 * i, 999, 100 * i);
        }
        for (i = 0; i < 11; i++) {
            line(100 * i, 1, 100 * i, 699);
        }

        char back[] = "返回主菜单", stop[] = "暂    停";
        diytext(back, 5, 715, 0, 750, 152, 709, RGB(255, 201,
14));
        diytext(stop, 157, 715, 152, 709, 304, 750, RGB(255, 255,
0));
        char q[20];
        int moneyint = floor(*money);
        sprintf_s(q, "资产: %d", moneyint);
        diytext(q, 853, 715, 848, 750, 1000, 709, RGB(255, 201,
14));
        char w[20];
        if (*choice1 != 6) {
            sprintf_s(w, "第 %d / 20 波", *wave);
        }
        else {
            sprintf_s(w, " 第  %d  波", *wave);
        }
        settextcolor(BLACK);
        settextstyle(50, 0, "方正舒体");
```

```
    setbkmode(TRANSPARENT);
    LOGFONT fontStyle;
    gettextstyle(&fontStyle);
    fontStyle.lfQuality = ANTIALIASED_QUALITY;
    settextstyle(&fontStyle);
    outtextxy(515, 700, w);
}//绘制地图，按钮，波数，二校门及其血量

void drawmst(MST mst, int* choice1, int* wave) {
    POINT P = { -1000, -1000 };
    int WAVE = *wave;
    if (mst.t != 0) {
        if (*choice1 == 1) {
            P = txy1(mst.t);
        }
        else if (*choice1 == 2) {
            P = txy2(mst.t);
        }
        else if (*choice1 == 3 || *choice1 == 6) {
            P = txy3(mst.t);
        }
        else if (*choice1 == 4) {
            P = txy4(mst.t);
        }
        else if (*choice1 == 5) {
            P = txy5(mst.t);
        }
        IMAGE s1, t1;
        loadimage(&s1, "img/shade.jpg");
```

```
            if (mst.type == 1) {
                loadimage(&t1, "img/mst1.jpg");
            }
            else if (mst.type == 2) {
                loadimage(&t1, "img/mst2.jpg");
            }
            else if (mst.type == 3) {
                loadimage(&t1, "img/mst3.jpg");
            }
            putimage(P.x, P.y, &s1, SRCPAINT);
            putimage(P.x, P.y, &t1, SRCAND);
            setlinestyle(PS_SOLID, 2);
            setlinecolor(BLACK);
            setfillcolor(RED);
            fillrectangle(P.x, P.y, P.x + 100, P.y + 5);
            setlinestyle(PS_SOLID, 1);
            setlinecolor(WHITE);
            setfillcolor(WHITE);
            if (mst.hp != msthp(mst.type, &WAVE)) {
                fillrectangle(P.x + 100 * (mst.hp / msthp(mst.type,
&WAVE)) - 1, P.y + 1, P.x + 99, P.y + 3);
            }
        }
    }//绘制怪物及其血条

    void drawtrt(int type, int lv, NODE node) {
        POINT p = nodetoxy(node);
        IMAGE s1, t1, s2, t2;
        if (type == 1) {
```

```
    loadimage(&s1, "img/pen0.png");
    loadimage(&t1, "img/pen1.png");
}
else if (type == 2) {
    loadimage(&s1, "img/redpen0.png");
    loadimage(&t1, "img/redpen1.png");
}
else if (type == 3) {
    loadimage(&s1, "img/pencil0.png");
    loadimage(&t1, "img/pencil1.png");
}
else if (type == 4) {
    loadimage(&s1, "img/ruler0.png");
    loadimage(&t1, "img/ruler1.png");
}
else if (type == 5) {
    loadimage(&s1, "img/triangle0.png");
    loadimage(&t1, "img/triangle1.png");
}
else if (type == 6) {
    loadimage(&s1, "img/protractor0.png");
    loadimage(&t1, "img/protractor1.png");
}
else if (type == 7) {
    loadimage(&s1, "img/compass0.png");
    loadimage(&t1, "img/compass1.png");
}
else if (type == 8) {
    loadimage(&s1, "img/eraser0.png");
```

```
        loadimage(&t1, "img/eraser1.png");
    }
    else if (type == 9) {
        loadimage(&s1, "img/coffee0.png");
        loadimage(&t1, "img/coffee1.png");
    }
    else if (type == 10) {
        loadimage(&s1, "img/wifi0.png");
        loadimage(&t1, "img/wifi1.png");
    }
    putimage(p.x, p.y, &s1, SRCPAINT);
    putimage(p.x, p.y, &t1, SRCAND);

    if (lv == 1) {
        loadimage(&s2, "img/1star0.png");
        loadimage(&t2, "img/1star1.png");
    }
    else if (lv == 2) {
        loadimage(&s2, "img/2star0.png");
        loadimage(&t2, "img/2star1.png");
    }
    else if (lv == 3) {
        loadimage(&s2, "img/3star0.png");
        loadimage(&t2, "img/3star1.png");
    }
    putimage(p.x + 25, p.y + 71, &s2, SRCPAINT);
    putimage(p.x + 25, p.y + 71, &t2, SRCAND);
}//绘制炮塔(带星级)
```

```
void drawchoosetrt(NODE node, NODE *nodehold) {
    POINT P = nodetoxy(node);
    int i = 0, delta;
    setlinecolor(RED);
    setlinestyle(PS_SOLID, 4);
    setfillcolor(RGB(255, 201, 14));
    if (nodehold->b != 7) {
        delta = 1;
    }
    else {
        delta = -1;
    }
    if (nodehold->b != 0 && nodehold->a != 0) {
        for (i = 0; i < 10; i++) {
            fillrectangle(i * 100, P.y + delta * 100, i * 100 +
100, P.y + 100 + delta * 100);
            rectangle(P.x, P.y, P.x + 100, P.y + 100);
        }

        for (i = 0; i < 10; i++) {
            drawtrt(i + 1, 0, { i + 1, node.b + delta });
        }
        char c1[] = "100";
        char c2[] = "160";
        char c3[] = "180";
        char c4[] = "420";
        char c5[] = "160";
        char c6[] = "180";
        char c7[] = "180";
```

```
            char c8[] = "120";
            char c9[] = "600";
            char c10[] = "260";
            settextstyle(15, 0, "宋体");
            setbkmode(TRANSPARENT);
            LOGFONT fontStyle;
            gettextstyle(&fontStyle);
            fontStyle.lfQuality = ANTIALIASED_QUALITY;
            settextstyle(&fontStyle);
            outtextxy(5, P.y + delta * 100 + 5, c1);
            outtextxy(105, P.y + delta * 100 + 5, c2);
            outtextxy(205, P.y + delta * 100 + 5, c3);
            outtextxy(305, P.y + delta * 100 + 5, c4);
            outtextxy(405, P.y + delta * 100 + 5, c5);
            outtextxy(505, P.y + delta * 100 + 5, c6);
            outtextxy(605, P.y + delta * 100 + 5, c7);
            outtextxy(705, P.y + delta * 100 + 5, c8);
            outtextxy(805, P.y + delta * 100 + 5, c9);
            outtextxy(905, P.y + delta * 100 + 5, c10);
            *nodehold = node;
        }
    }//绘制选择炮塔UI

    void drawuptrt(NODE node, TRT trt[10][7], int *condition, NODE
*nodehold, int *choice1) {
        POINT P = nodetoxy(node);
        int level = *choice1;
        setlinecolor(RED);
        setlinestyle(PS_SOLID, 4);
```

```
        setfillcolor(RGB(255, 201, 14));
        if (nodehold->b != 0 && nodehold->a != 0 &&
trt[nodehold->a - 1][nodehold->b - 1].lv != lvmax(&level)) {
            char d[10];
            sprintf_s(d, "-%d升级", cost(trt[nodehold->a -
1][nodehold->b - 1].type, trt[nodehold->a - 1][nodehold->b - 1].lv
+ 1));
            diytext2(d, P.x + 10, P.y + 10, P.x, P.y, P.x + 100,
P.y + 100, RGB(255, 201, 14));
            *nodehold = node;
        }
        else if (nodehold->b != 0 && nodehold->a != 0 &&
trt[nodehold->a - 1][nodehold->b - 1].lv == lvmax(&level)) {
            *nodehold = node;
            *condition = 0;
        }
    }//绘制升级炮塔UI

    void drawdeltrt(NODE node, TRT trt[10][7], NODE *nodehold) {
        POINT P = nodetoxy(node);
        setlinecolor(RED);
        setlinestyle(PS_SOLID, 4);
        setfillcolor(RGB(255, 201, 14));
        if (nodehold -> b != 0 && nodehold -> a != 0) {
            char c[] = "拆  除";
            diytext(c, P.x + 10, P.y + 10, P.x, P.y, P.x + 100, P.y
+ 100, RGB(255, 201, 14));
            *nodehold = node;
        }
```

```
}//绘制拆除炮塔UI

void drawbul(POINT pt, POINT pm, TRT trt) {
    int u = 0;
    if (trt.type == 2) {
        setlinecolor(RED);
        setlinestyle(PS_SOLID, 2 + trt.lv);
        line(pt.x, pt.y, pm.x, pm.y);
    }
    for (u = 0; u < 5; u++) {
        if (trt.clock % spd(trt) == u) {
            if (trt.type == 1) {
                setlinecolor(BLACK);
                setlinestyle(PS_SOLID, 3);
                setfillcolor(BLACK);
                fillcircle(pt.x - u * (pt.x - pm.x) / 5, pt.y -
u * (pt.y - pm.y) / 5, 4 + 2 * trt.lv);
            }
            else if (trt.type == 3) {
                setlinecolor(RGB(148, 148, 148));
                setfillcolor(RGB(148, 148, 148));
                setlinestyle(PS_SOLID, 3);
                fillcircle(pt.x - u * (pt.x - pm.x) / 5, pt.y -
u * (pt.y - pm.y) / 5, 4 + 2 * trt.lv);
            }
            else if (trt.type == 4) {
                setlinecolor(BLUE);
                setlinestyle(PS_SOLID, 2 + trt.lv);
                line(pt.x, pt.y, pm.x, pm.y);
```

```
                }
                else if (trt.type == 5) {
                    setlinecolor(BLACK);
                    setlinestyle(PS_SOLID, 2 + trt.lv);
                    line(pt.x - u * (pt.x - pm.x) / 5 - 15, pt.y - u
* (pt.y - pm.y) / 5 - 15,
                        pt.x - u * (pt.x - pm.x) / 5 - 15, pt.y - u
* (pt.y - pm.y) / 5 + 15);
                    line(pt.x - u * (pt.x - pm.x) / 5 + 15, pt.y - u
* (pt.y - pm.y) / 5 - 15,
                        pt.x - u * (pt.x - pm.x) / 5 - 15, pt.y - u
* (pt.y - pm.y) / 5 - 15);
                    line(pt.x - u * (pt.x - pm.x) / 5 - 15, pt.y - u
* (pt.y - pm.y) / 5 + 15,
                        pt.x - u * (pt.x - pm.x) / 5 + 15, pt.y - u
* (pt.y - pm.y) / 5 - 15);
                    if (u == 3 || u == 4) {
                        setlinecolor(BLACK);
                        setlinestyle(PS_DASH, 3 + trt.lv);
                        circle(pm.x, pm.y, RG(trt.type, trt.lv) *
0.8);
                    }
                }
                else if (trt.type == 6) {
                    setlinecolor(BLUE);
                    setlinestyle(PS_SOLID, 2 + trt.lv);
                    arc(pt.x - u * (pt.x - pm.x) / 5 - 15, pt.y - u
* (pt.y - pm.y) / 5 - 15,
                        pt.x - u * (pt.x - pm.x) / 5 + 15, pt.y - u
```

```
* (pt.y - pm.y) / 5 + 15,
                    0.0, pi);
                line(pt.x - u * (pt.x - pm.x) / 5 - 15, pt.y - u
* (pt.y - pm.y) / 5 + 1,
                    pt.x - u * (pt.x - pm.x) / 5 + 15, pt.y - u
* (pt.y - pm.y) / 5 + 1);
                if (u == 3 || u == 4) {
                    setlinecolor(BLUE);
                    setlinestyle(PS_DASH, 3 + trt.lv);
                    circle(pm.x, pm.y, RG(trt.type, trt.lv) *
0.75);
                }
            }
            else if (trt.type == 7) {
                setlinecolor(RED);
                setlinestyle(PS_SOLID, 3);
                circle(pt.x, pt.y, RG(7, trt.lv) * u / 5);

            }
            else if (trt.type == 8) {
                setlinecolor(GREEN);
                setfillcolor(WHITE);
                setlinestyle(PS_SOLID, 3);
                fillcircle(pt.x - u * (pt.x - pm.x) / 5, pt.y -
u * (pt.y - pm.y) / 5, 4 + 2 * trt.lv);
            }
        }
    }
}//绘制子弹动画
```

```
    void drawwifi(const bool* is_signal_fail) {
        IMAGE wifi0, wifi1;
        if (*is_signal_fail == false) {
            loadimage(&wifi0, "img/wifilogo0.png");
            loadimage(&wifi1, "img/wifilogo1.png");
        }
        else {
            loadimage(&wifi0, "img/wififail0.png");
            loadimage(&wifi1, "img/wififail1.png");
        }
        putimage(350, 700, &wifi0, SRCPAINT);
        putimage(350, 700, &wifi1, SRCAND);
    }//绘制 wifi 状况

    void drawbackorstop(int x, int y) {
        char back[] = "返回主菜单", stop[] = "暂    停";
        if (x >= 0 && x <= 152 && y >= 709 && y <= 750) {
            diytext(back, 5, 715, 0, 750, 152, 709, RGB(255, 255,
0));
        }
        else {
            diytext(back, 5, 715, 0, 750, 152, 709, RGB(255, 201,
14));
        }
        if (x >= 152 && x <= 304 && y >= 709 && y <= 750) {
            diytext(stop, 157, 715, 152, 709, 304, 750, RGB(255,
255, 0));
        }
```

```
        else {
            diytext(stop, 157, 715, 152, 709, 304, 750, RGB(255,
201, 14));
        }
    }//绘制按钮高光

    MST generatemst(int i, int* wave) {
        MST mst = { 0,0,0,0 };
        if (*wave <= 5) {
            mst.type = rand() % 2 + 1;
        }
        else {
            if (i < 19) {
                mst.type = rand() % 2 + 1;
            }
            else if (i == 19) {
                mst.type = 3;
            }
        }
        int WAVE = *wave;
        mst.v = v(mst);
        mst.hp = msthp(mst.type, &WAVE);
        mst.t = 1;

        return mst;
    }//出怪函数

    void settrt(NODE node, TRT trt[10][7], NODE* nodehold, int*
money) {
```

```
        int delta;
        if (nodehold->b != 7) {
            delta = 1;
        }
        else {
            delta = -1;
        }
        if (node.b == nodehold->b + delta) {
            trt[nodehold->a - 1][nodehold->b - 1].type = node.a;
            trt[nodehold->a - 1][nodehold->b - 1].lv = 1;
        }
        *money -= cost(trt[nodehold->a - 1][nodehold->b - 1].type,
1);
        if (*money < 0) {
            *money += cost(trt[nodehold->a - 1][nodehold->b -
1].type, 1);
            trt[nodehold->a - 1][nodehold->b - 1] = { 0, 0, 0 };
        }
        *nodehold = { 0, 0 };
    }//放置炮塔

    void uptrt(NODE node, TRT trt[10][7], NODE *nodehold, int
*choice1, int *money) {
        if (node.a == nodehold->a && node.b == nodehold->b) {
            if ((trt[nodehold->a - 1][nodehold->b - 1].lv <= 1 &&
*choice1 != 3 && *choice1 != 6)
                || (trt[nodehold->a - 1][nodehold->b - 1].lv <= 2
&& (*choice1 == 3 || *choice1 == 6))) {
                trt[nodehold->a - 1][nodehold->b - 1].lv += 1;
```

```
                *money -= cost(trt[nodehold->a - 1][nodehold->b -
1].type, trt[nodehold->a - 1][nodehold->b - 1].lv);
                if (*money < 0) {
                    *money += cost(trt[nodehold->a - 1][nodehold->b
- 1].type, trt[nodehold->a - 1][nodehold->b - 1].lv);
                    trt[nodehold->a - 1][nodehold->b - 1].lv -= 1;
                }
            }
        }
    }//升级炮塔

    void deltrt(NODE node, TRT trt[10][7], int *money, NODE
*nodehold) {
        if (node.a == nodehold->a && node.b == nodehold->b) {
            double m = cost(trt[nodehold->a - 1][nodehold->b -
1].type, trt[nodehold->a - 1][nodehold->b - 1].lv) * 0.6;
            double m1 = m - floor(m);
            double m2 = ceil(m) - m;
            *money += m1 < m2 ? floor(m) : ceil(m);
            trt[nodehold->a - 1][nodehold->b - 1] = { 0, 0, 0 };
        }
    }//拆除炮塔

    int moneyplus(MST mst) {
        switch (mst.type)
        {
        case 1:
            return 13;
            break;
```

```
        case 2:
            return 13;
            break;
        case 3:
            return 88;
            break;
        default:
            return 0;
            break;
        }
    }//击杀怪物奖励金钱

    void coffeeplus(TRT trt[10][7], int* money) {
        int i = 0, j = 0, k = 0;
        for (i = 0; i < 10; i++) {
            for (j = 0; j < 7; j++) {
                if (trt[i][j].type == 9 && trt[i][j].lv == 1) {
                    k++;
                }
                else if (trt[i][j].type == 9 && trt[i][j].lv == 2)
{
                    k += 3;
                }
                else if (trt[i][j].type == 9 && trt[i][j].lv == 3)
{
                    k += 5;
                }
            }
        }
```

```
        *money += k * 8;
    }//咖啡机产金币

    bool signalfailure(double second, bool *happenget, int
*happen, int *choice1, int *wave) {
        if (*choice1 == 3 || *choice1 == 6) {
            if (*wave > 9) {
                if (*happenget == false) {
                    *happen = rand() % 45;
                    *happenget = true;
                }
                int secondpass = floor(second - 1.0 / f);
                int secondnow = floor(second);
                if (secondpass != secondnow) {
                    if (secondnow % 60 == *happen) {
                        *happenget = false;
                        return true;
                    }
                }
            }
        }
        return false;
    }//生成"网课掉线"随机事件

    void wifi(TRT trt[10][7], bool *signal_fail_removed, int
*signal_fail_time) {
        int i = 0, j = 0, u = 0;
        for (i = 0; i < 10; i++) {
            for (j = 0; j < 7; j++) {
```

```
                if (trt[i][j].type == 10) {
                    POINT pt = { nodetoxy({ i + 1, j + 1 }).x + 50,
nodetoxy({ i + 1, j + 1 }).y + 50 };
                    if (trt[i][j].clock < 30) {
                        trt[i][j].clock++;
                    }
                    else {
                        trt[i][j] = { 0,0,0 };
                        *signal_fail_removed = true;
                        *signal_fail_time = 0;
                    }
                    for (u = 0; u < 30; u++) {
                        if (trt[i][j].clock % 30 == u) {
                            setlinecolor(GREEN);
                            setlinestyle(PS_SOLID, 3);
                            circle(pt.x, pt.y, 1250 * u / 30);
                        }
                    }
                }
            }
        }
    }//wifi 炮塔函数

    bool isgamewin(MST mst[25], int *choice1, int *wave) {
        if (*choice1 != 6 && *wave == 21 && allmstdie(mst) ==
true) {
            return true;
        }
        else {
```

```
            return false;
        }
    }//判断是否胜利

    void fail() {
        mciSendString("play music/fail.mp3 repeat", NULL, 0,
NULL);
        IMAGE image[107];
        char s[30];
        int i = 3, j = 0;
        for (i = 3; i < 110; i++) {
            sprintf_s(s, "img/fail/Image%d.jpg", i);
            loadimage(&image[i - 3], s);
            Sleep(50);
            putimage(0, 0, &image[i - 3]);
        }
        setlinecolor(BLACK);
        setlinestyle(PS_SOLID, 4);
        setfillcolor(GREEN);
        fillroundrect(200, 250, 800, 500, 30, 30);
        char failure[] = "你 失 败 了!", replay[] = "重    生";
        settextcolor(RED);
        settextstyle(100, 0, "方正舒体");
        setbkmode(TRANSPARENT);
        LOGFONT fontStyle;
        gettextstyle(&fontStyle);
        fontStyle.lfQuality = ANTIALIASED_QUALITY;
        settextstyle(&fontStyle);
        outtextxy(271, 250, failure);
```

```
        ExMessage msg;
        BeginBatchDraw();
        while (1) {
            putimage(0, 0, &image[i - 3]);
            diytext(replay, 445, 410, 430, 400, 570, 445, RGB(255,
201, 14));

            bool isok = peekmessage(&msg, EX_MOUSE);
            if (isok == true) {
                if (msg.x >= 430 && msg.x <= 570 && msg.y >= 400 &&
msg.y <= 445) {
                    diytext(replay, 445, 410, 430, 400, 570, 445,
RGB(255, 255, 0));
                    if (msg.message == WM_LBUTTONDOWN) {
                        EndBatchDraw();
                        mciSendString("close music/fail.mp3", NULL,
0, NULL);
                        return;
                    }
                }
                else {
                    diytext(replay, 445, 410, 430, 400, 570, 445,
RGB(255, 201, 14));
                }
                FlushBatchDraw();

            }
        }
```

```
}//游戏失败结算

void win(){
    mciSendString("open music/win.mp3", NULL, 0, NULL);
    mciSendString("play music/win.mp3", NULL, 0, NULL);
    setlinecolor(BLACK);
    setlinestyle(PS_SOLID, 4);
    setfillcolor(GREEN);
    fillroundrect(200, 250, 800, 500, 30, 30);
    char success[] = "保 卫 成 功!", replay[] = "返回主菜单";
    settextcolor(BLACK);
    settextstyle(100, 0, "方正舒体");
    setbkmode(TRANSPARENT);
    LOGFONT fontStyle;
    gettextstyle(&fontStyle);
    fontStyle.lfQuality = ANTIALIASED_QUALITY;
    settextstyle(&fontStyle);
    outtextxy(271, 250, success);

    ExMessage msg;
    BeginBatchDraw();
    while (1) {
        diytext(replay, 445, 410, 430, 400, 570, 445, RGB(255,
201, 14));

        bool isok = peekmessage(&msg, EX_MOUSE);
        if (isok == true) {
            if (msg.x >= 430 && msg.x <= 570 && msg.y >= 400 &&
msg.y <= 445) {
```

```
                diytext(replay, 438, 410, 430, 400, 570, 445,
RGB(255, 255, 0));
                if (msg.message == WM_LBUTTONDOWN) {
                    EndBatchDraw();
                    mciSendString("close music/win.mp3", NULL,
0, NULL);
                    return;
                }
            }
            else {
                diytext(replay, 438, 410, 430, 400, 570, 445,
RGB(255, 201, 14));
            }
            FlushBatchDraw();

        }
    }
}//游戏成功结算

void mouse_operation(bool isok, bool *ifexit, NODE *nodehold,
NODE node, int *condition, int *choice1, TRT trt[10][7], int*
money, ExMessage* msg) {
    NODE NODEHOLD = *nodehold;
    int MONEY = *money, CHOICE1 = *choice1;
    if (isok == true) {
        if (msg->lbutton != true && msg->rbutton != true) {
            setlinecolor(BLACK);
            setlinestyle(PS_SOLID, 4);
            setfillcolor(BLACK);
```

```
            fillcircle(msg->x, msg->y, 5);
        }
        if (msg->lbutton == true) {
            setlinecolor(RED);
            setlinestyle(PS_SOLID, 4);
            setfillcolor(RED);
            fillcircle(msg->x, msg->y, 5);

            if (msg->x >= 0 && msg->x <= 152 && msg->y >= 709
&& msg->y <= 750) {
                *ifexit = true;
                NODEHOLD = { 0, 0 };
                *condition = 0;
                return;
            }
            else if (msg->x >= 152 && msg->x <= 304 &&
msg->y >= 709 && msg->y <= 750) {
                while (1) {
                    ExMessage goon;
                    getmessage(&goon, EX_MOUSE);
                    if (goon.x >= 152 && goon.x <= 304
                        && goon.y >= 709 && goon.y <= 750
                        && goon.message == WM_LBUTTONDOWN) {
                        break;
                    }

                }
            }
            else if (*condition == 0) {
```

```
                    if (inroute(CHOICE1, node) == false) {
                        NODEHOLD = node;
                        if (trt[node.a - 1][node.b - 1].type == 0 &&
msg->message == WM_LBUTTONDOWN) {
                            *condition = 1;
                        }
                        else if (msg->message == WM_LBUTTONDOWN) {
                            *condition = 2;
                        }
                    }
                }
                else if (*condition == 1) {
                    settrt(node, trt, &NODEHOLD, &MONEY);
                    *condition = 0;
                }
                else if (*condition == 2) {
                    uptrt(node, trt, &NODEHOLD, &CHOICE1, &MONEY);
                    *condition = 0;
                }
                else if (*condition == 3) {
                    deltrt(node, trt, &MONEY, &NODEHOLD);
                    *condition = 0;
                }
            }
            if (msg->rbutton == true) {
                setlinecolor(BLUE);
                setlinestyle(PS_SOLID, 4);
                setfillcolor(BLUE);
                fillcircle(msg->x, msg->y, 5);
```

```
                if (*condition == 0 && msg->message ==
WM_RBUTTONDOWN) {
                    if (trt[node.a - 1][node.b - 1].type != 0) {
                        NODEHOLD = node;
                        *condition = 3;
                    }
                }
            }
        }
        else {
            if (msg->rbutton == true) {
                setlinecolor(BLUE);
                setlinestyle(PS_SOLID, 4);
                setfillcolor(BLUE);
                fillcircle(msg->x, msg->y, 5);
            }
            else if (msg->lbutton == true) {
                setlinecolor(RED);
                setlinestyle(PS_SOLID, 4);
                setfillcolor(RED);
                fillcircle(msg->x, msg->y, 5);
            }
            else {
                setlinecolor(BLACK);
                setlinestyle(PS_SOLID, 4);
                setfillcolor(BLACK);
                fillcircle(msg->x, msg->y, 5);
            }
        }
```

```
        *nodehold = NODEHOLD;
        *money = MONEY;
        *choice1 = CHOICE1;
    }

    void attack(TRT trt[10][7], MST mst[25], int* money, int*
choice1) {
        int i = 0, j = 0, k = 0, u = 0, w = 0, level = *choice1;
        double d = 0, d2 = 0, r = 0, dmg = 0;
        bool clockok = false;
        POINT pm = { 0,0 }, pm2 = { 0,0 };
        POINT pt = { 0,0 };
        MST hold = { 0,0,0,0 };
        MST mstin[25];
        for (i = 0; i < 10; i++) {
            for (j = 0; j < 7; j++) {
                if (trt[i][j].type != 0) {
                    clockok = false;
                    hold = { 0,0,0,0 };
                    for (u = 0; u < 25; u++) {
                        mstin[u] = { 0,0,0,0 };
                    }
                    for (k = 0; k < 25; k++) {
                        if (clockok == false) {
                            if (trt[i][j].clock < 1800) {
                                trt[i][j].clock++;
                            }
                            else {
                                trt[i][j].clock = 1;
```

```
                    }
                    clockok = true;
                }
                if (trt[i][j].clock % spd(trt[i][j]) ==
spd(trt[i][j]) - 1
                    && (trt[i][j].type == 8
                    || trt[i][j].type == 6)) {
                    mst[k].v = v(mst[k]);
                }
                if (mst[k].t != 0 && mst[k].hp > 0) {
                    r = RG(trt[i][j].type, trt[i][j].lv);//计
算炮塔射程半径
                    pm = PM(mst[k], &level);
                    pt = { nodetoxy({ i + 1, j + 1 }).x + 50,
nodetoxy({ i + 1, j + 1 }).y + 50 };
                    d2 = (pm.x - pt.x) * (pm.x - pt.x) +
(pm.y - pt.y) * (pm.y - pt.y);
                    d = sqrt(d2);//计算炮塔与怪物的间距
                    if (d > r) {
                        continue;
                    }
                    else if (d < r) {
                        dmg = DMG(trt[i][j].type,
trt[i][j].lv, 0);
                        if (trt[i][j].type == 7) {//圆规无需
锁定敌人
                            if (trt[i][j].clock %
spd(trt[i][j]) == 2) {
                                playtrtsound(trt[i][j], i, j);
```

```
                                }
                                if (trt[i][j].clock %
spd(trt[i][j]) == 5) {
                                    mst[k].hp -= dmg;
                                    if (mst[k].hp <= 0) {
                                        *money +=
moneyplus(mst[k]);
                                    }
                                }
                                if (trt[i][j].clock %
spd(trt[i][j]) == spd(trt[i][j]) - 1) {
                                    closetrtsound(trt[i][j], i,
j);
                                }
                                drawbul(pt, pm, trt[i][j]);
                            }
                            else {
                                mstin[k] = mst[k];
                            }
                        }
                    }
                    if (k == 24 && trt[i][j].type != 7) {
                        for (u = 1; u < 25; u++) {
                            if (mstin[0].t < mstin[u].t) {
                                hold = mstin[0];
                                mstin[0] = mstin[u];
                                mstin[u] = hold;
                            }
                        }
```

```
                        for (u = 0; u < 25; u++) {
                            if (mst[u].t == mstin[0].t &&
mst[u].t != 0) {
                                pm2 = PM(mst[u], &level);
                                if (mst[u].t > 0 && pm.x != 0 &&
pt.x != 0) {
                                    drawbul(pt, pm2, trt[i][j]);
                                }
                                if (trt[i][j].clock %
spd(trt[i][j]) == 2 && mst[u].t > 0) {
                                    playtrtsound(trt[i][j], i, j);
                                }
                                if (((trt[i][j].clock %
spd(trt[i][j]) == 4 && trt[i][j].type != 2)
                                    || trt[i][j].type == 2) &&
mst[u].t > 0) {
                                    mst[u].hp -= (trt[i][j].type
== 4 && mst[u].type == 3) ? 2 * dmg : dmg;
                                    if (mst[u].hp <= 0) {
                                        *money +=
moneyplus(mst[u]);
                                    }
                                    if (trt[i][j].type == 6 &&
mst[u].v > v(mst[u]) * 0.6) {
                                        mst[u].v = v(mst[u]) * 0.6;
                                    }
                                    if (trt[i][j].type == 8) {
                                        if (trt[i][j].lv == 1 &&
mst[u].v > 0.2 * v(mst[u])) {
```

```
                                        mst[u].v = 0.2 *
v(mst[u]);
                                    }
                                    else if (trt[i][j].lv == 2
&& mst[u].v > 0.1 * v(mst[u])) {
                                        mst[u].v = 0.1 *
v(mst[u]);
                                    }
                                    else if (trt[i][j].lv == 3)
{
                                        mst[u].v = 0;
                                    }
                                }
                                if (trt[i][j].type == 3) {
                                    if (trt[i][j].lv == 1 &&
rand() % 20 == 0) {
                                        mst[u].t -= 50;
                                    }
                                    else if (trt[i][j].lv == 2
&& rand() % 10 == 0) {
                                        mst[u].t -= 50;
                                    }
                                    else if (trt[i][j].lv == 3
&& rand() % 5 == 0) {
                                        mst[u].t -= 50;
                                    }
                                    if (mst[u].t <= 0) {
                                        mst[u].t = 1;
                                    }
```

```
                                    }
                                    if (trt[i][j].type == 5 ||
trt[i][j].type == 6) {
                                        for (w = 0; w < 25; w++) {
                                            if (mst[w].t != 0 &&
mst[w].hp > 0) {
                                                r =
RG(trt[i][j].type, trt[i][j].lv) * 0.75;
                                                pm2 = PM(mst[w],
&level);
                                                pt = PM(mst[u],
&level);
                                                d2 = (pm2.x - pt.x)
* (pm2.x - pt.x) + (pm2.y - pt.y) * (pm2.y - pt.y);
                                                d = sqrt(d2);
                                                if (d < r && d != 0)
{
                                                    mst[w].hp -=
DMG(trt[i][j].type, trt[i][j].lv, 1);
                                                    if (mst[w].hp <=
0) {
                                                        *money +=
moneyplus(mst[w]);
                                                    }
                                                    if
(trt[i][j].type == 6 && mst[w].v > v(mst[w]) * 0.6) {
                                                        mst[w].v =
0.6 * v(mst[w]);
                                                    }
```

```
                                        }
                                    }
                                }
                            }
                            if (trt[i][j].clock %
spd(trt[i][j]) == spd(trt[i][j]) - 1) {
                                closetrtsound(trt[i][j], i,
j);
                            }
                            break;
                        }
                    }
                }
            }
        }
    }//核心函数:判定并执行攻击

    int main() {
        srand((unsigned)time(NULL));
        initgraph(1000, 750);//生成图形窗体
        while (1) {
            int choice0;//初始页面选择
            int choice1;//地图选择
            while (1) {
                choice0 = menu();//生成初始页面
                if (choice0 == 0) {
```

```
                choice1 = map();//获取选择的关卡代号
                if (choice1 == 0) {
                    continue;
                }
                else {
                    mciSendString("close music/音乐0.mp3", NULL,
0, NULL);
                    break;//进入关卡
                }
            }
            else {
                PlayerHelp();//查看游戏帮助
            }
        }

        playmusic(&choice1);//播放关卡音乐
        MST mst[25];//怪物数组
        initmst(mst);//将全部怪物初始化
        TRT trt[10][7];//炮塔数组
        inittrt(trt);//将全部炮塔初始化

        int money = inm;//金钱数
        int wave = 0;//波数
        int HP = (choice1 == 6) ? 20 : 10;//二校门的血量
        int condition = 0;//鼠标界面状态转换参数
        int i, j;//万能数字
        int tick = -120, happen = 0, signal_fail_time = 0;//控
制随机事件
        int fclock = 0;//控制咖啡机产金币
```

```
        double second = 1.0 / f;//控制波数增加
        bool ifexit = false;//判断是否主动退出游戏
        NODE nodehold = { 0, 0 }, node = { 0, 0 };//鼠标操作相关
        bool dieok = true, happenget = false, is_signal_fail = false, signal_fail_removed = true, isok = true;//状态转换参数
        bool ifstuck = false;//卡顿判断参数

        timeBeginPeriod(1);//提高计时器精度为1ms，便于控制帧率
        LARGE_INTEGER startCount, endCount, F;//初始化计时参数
        QueryPerformanceFrequency(&F);//计算计时频率

        BeginBatchDraw();//启动批量绘图
        ExMessage msg;//声明鼠标消息
        msg.x = 0;
        msg.y = 0;//初始化鼠标坐标，防止进图时卡住

        while (1) {
            QueryPerformanceCounter(&startCount);//开始计时
            if (fclock < 1800) {
                fclock++;
            }
            else {
                fclock = 1;
            }//关卡时钟

            drawmap(HP, &choice1, &money, &wave);//绘制地图，波数，交互按钮，二校门及其血条
```

```
            if (tick % f == 1 && tick >= 0) {
                i = tick / f;
                mst[i] = generatemst(i, &wave);
            }//出怪

            if (ifstuck == true) {
                isok = peekmessage(&msg, EX_MOUSE);
                mouse_operation(isok, &ifexit, &nodehold, node,
                    &condition, &choice1, trt, &money, &msg);
            }//防止鼠标过于卡顿

            if (wave > 10) {
                if (second < 18000) {
                    second += 1.0 / f;
                }
                else {
                    second = 1.0 / f;
                }
            }//改变波数

            if (signal_fail_removed == true) {
                is_signal_fail = signalfailure(second,
&happenget, &happen, &choice1, &wave);
                if (is_signal_fail == true) {
                    signal_fail_removed = false;
                }
            }
            else {
                signal_fail_time++;
```

```
            if (signal_fail_time >= 300) {
                signal_fail_removed = true;
                signal_fail_time = 0;
            }
        }//"网课掉线"随机事件触发

        if (choice1 == 3 || choice1 == 6) {
            drawwifi(&is_signal_fail);
        }
        wifi(trt, &signal_fail_removed,
&signal_fail_time);//wifi 炮塔生效及其绘图

        for (i = 0; i < 25; i++) {
            if (mst[i].hp <= 0){
                mst[i] = { 0, 0, 0, 0 };
            }
        }//清除死亡的怪物

        dieok = allmstdie(mst);
        if (tick < 600) {
            tick++;
            if (tick == 0) {
                wave++;
            }
        }
        else if (dieok == true) {
            tick = -120;
            dieok = false;
        }//判断一波是否结束
```

```
                for (i = 0; i < 20; i++) {
                    if (mst[i].t > tmax(&choice1)) {
                        HP -= 1;
                        mst[i] = { 0,0,0,0 };
                    }
                }//清除撞到二校门的怪物，并令二校门扣血

                for (i = 0; i < 10; i++) {
                    for (j = 0; j < 7; j++) {
                        drawtrt(trt[i][j].type, trt[i][j].lv, { i +
1,j + 1 });
                    }
                }//绘制炮塔

                if (ifstuck == true) {
                    isok = peekmessage(&msg, EX_MOUSE);
                    mouse_operation(isok, &ifexit, &nodehold, node,
                        &condition, &choice1, trt, &money, &msg);
                }

                for (i = 24; i >= 0; i--) {
                    if (mst[i].hp > 0) {
                        drawmst(mst[i], &choice1, &wave);
                        mst[i].t += mst[i].v;
                    }
                }//绘制并移动怪物

                if (ifstuck == true) {
```

```
            isok = peekmessage(&msg, EX_MOUSE);
            mouse_operation(isok, &ifexit, &nodehold, node,
                &condition, &choice1, trt, &money, &msg);
        }//防止鼠标过于卡顿

        if (is_signal_fail == false) {
            if (fclock % f == 0) {
                coffeeplus(trt, &money);//咖啡机产金币
            }
            attack(trt, mst, &money, &choice1);//全游戏的核心，炮塔攻击怪物
        }

        if (ifstuck == true) {
            isok = peekmessage(&msg, EX_MOUSE);
            mouse_operation(isok, &ifexit, &nodehold, node,
                &condition, &choice1, trt, &money, &msg);
        }//防止鼠标过于卡顿

        if (condition == 1) {
            drawchoosetrt(nodehold, &nodehold);
        }
        if (condition == 2) {
            drawuptrt(nodehold, trt, &condition, &nodehold,
&choice1);
        }
        if (condition == 3) {
            drawdeltrt(nodehold, trt, &nodehold);
        }//绘制鼠标UI
```

```
                node = xytonode(msg.x, msg.y);//获得鼠标的网格坐标
                 drawbackorstop(msg.x, msg.y);//绘制按钮高光及其恢
复

                QueryPerformanceCounter(&endCount);//停止计时
                long long elapse = (endCount.QuadPart -
startCount.QuadPart) * 1000000 / F.QuadPart;//计时

                if (ifstuck == true) {
                    isok = peekmessage(&msg, EX_MOUSE);
                    mouse_operation(isok, &ifexit, &nodehold, node,
                        &condition, &choicel, trt, &money, &msg);
                }//防止鼠标过于卡顿

                if (elapse >= 1000000 / f) {
                    ifstuck = true;
                }
                else {
                    ifstuck = false;
                }
                while (elapse < 1000000 / f) {
                    isok = peekmessage(&msg, EX_MOUSE);
                    mouse_operation(isok, &ifexit, &nodehold, node,
                        &condition, &choicel, trt, &money, &msg);
                    QueryPerformanceCounter(&endCount);
                    elapse = (endCount.QuadPart -
startCount.QuadPart) * 1000000 / F.QuadPart;
                }//控帧休眠循环, 拾取鼠标消息
```

```
            FlushBatchDraw();//加载批量绘图

            if (ifexit == true || isgamewin(mst, &choice1,
&wave) == true || HP <= 0){
                break;//判断是否结束游戏
            }
        }
        timeEndPeriod(1);
        EndBatchDraw();//结束批量绘图
        closemusic(&choice1);//关闭关卡音乐

        if (HP <= 0) {
            fail();
        }
        else if (isgamewin(mst, &choice1, &wave) == true) {
            win();
        }//判断输赢
    }//返回主菜单循环

    return 0;
}
```