南开大学 计算机科学与技术

张祯颉

2213919

计算机科学卓越班

2023年5月10日

高级语言程序设计

实验报告

高级语言程序设计大作业实验报告

**目录**

**一、作业题目**

**二、开发软件**

**三、课题要求**

**四、主要流程**

**1、运行流程**

**2、开发流程**

a.立项与初期规划

b.控制台代码编写

c.图形化代码编写

d.其他功能的图形化编写

e.音乐播放器

f.小修补

**五、代码介绍**

**1、Tools.cpp（外部引入的经典代码模板）**

**2、common.cpp（常量定义和窗口加载函数）**

**3、UI.cpp（按钮结构体定义）**

**4、main.cpp（主函数）**

a.常量和全局变量定义

b.游戏规则判定

c.操作相关

d.AI（电脑玩家）相关

e.游戏交互

f.调试和人类玩家全体失败后的函数

g.游戏的附属功能

h.音乐播放器

**六、算法或公式**

**1、并查集（Disjoint Set Union, DSU）**

**2、随机游走（Random Walk）**

**七、项目测试**

**1、测试项目**

**2、测试结果**

**八、收获**

**九、目前存在的缺陷**

**作业题目**

图形化小游戏——一款类似Slay-Antiyoy的战棋小游戏

**开发软件**

Visual Studio 2022，EasyX库

**课题要求**

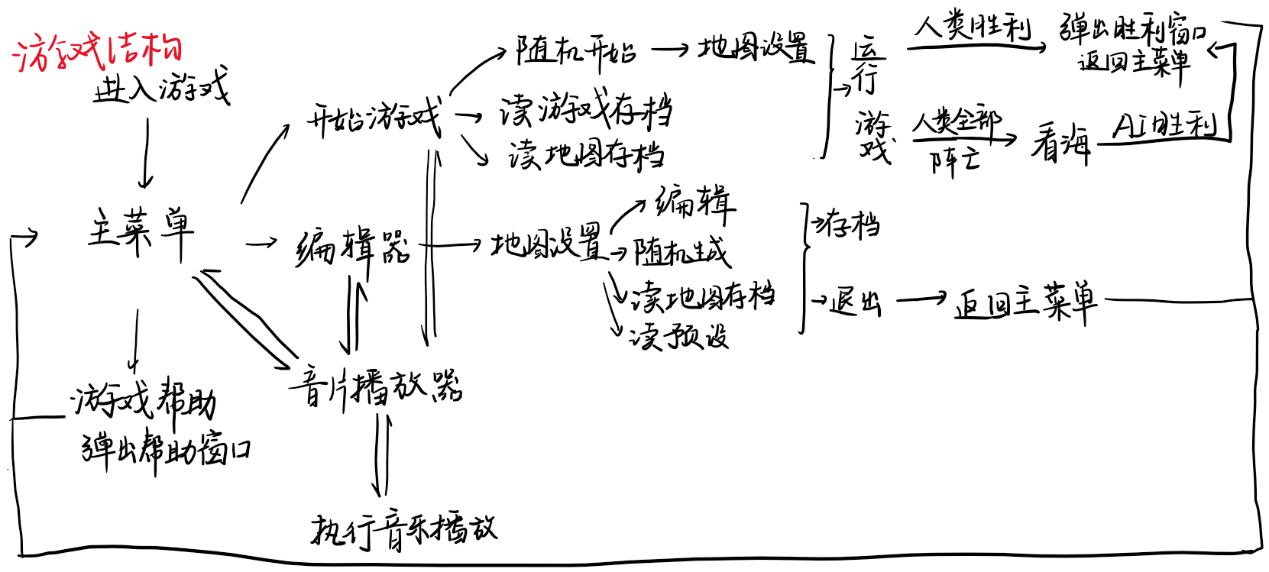
1）基于C++的图形化

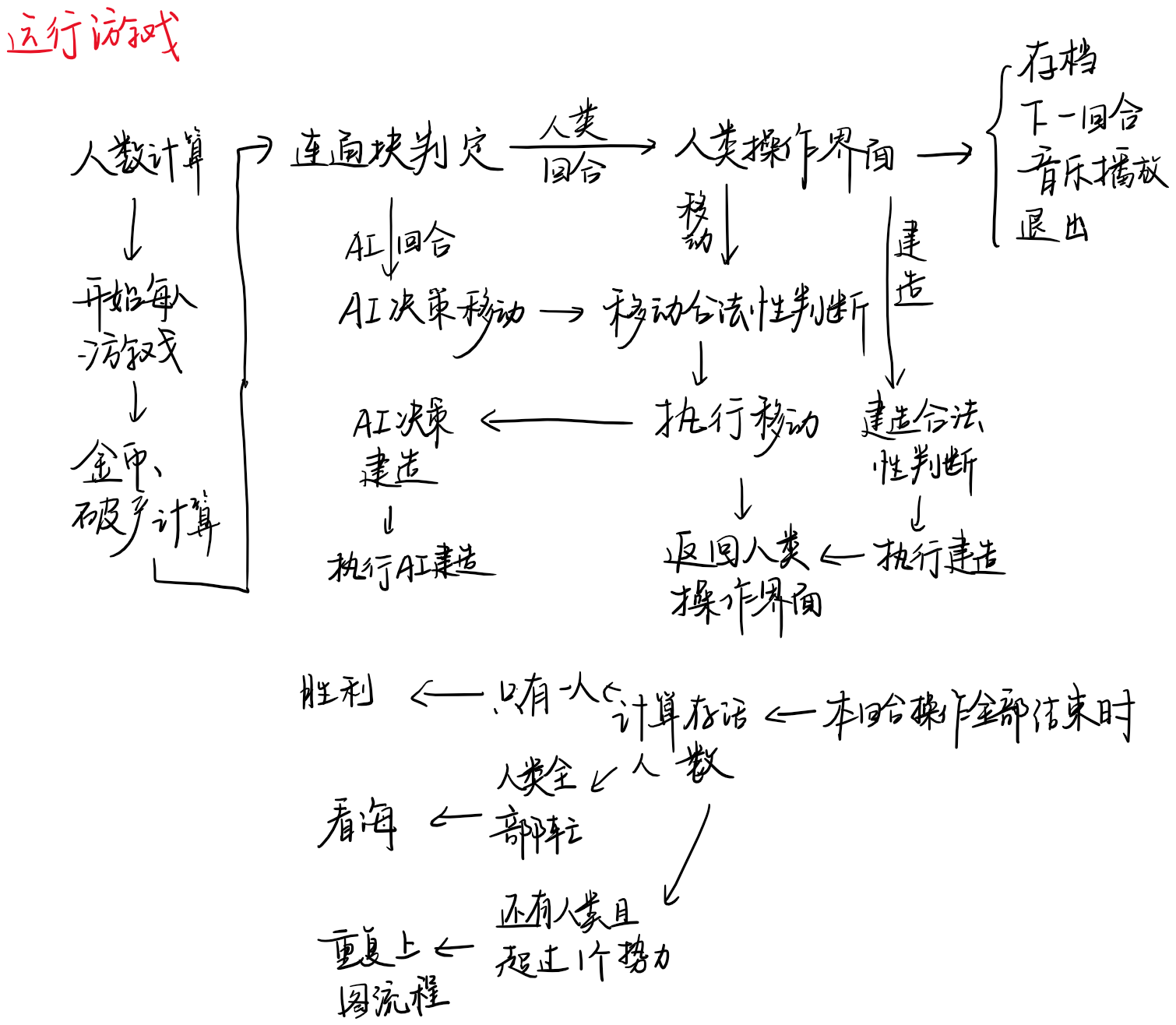
2）其他要要求不限

**主要流程**

**运行流程：**

游戏由四大部分构成：进行游戏、地图编辑器、音乐播放器、游戏帮助，三部分并行运行在进行游戏和地图编辑器大类中还有子类。流程图如下：

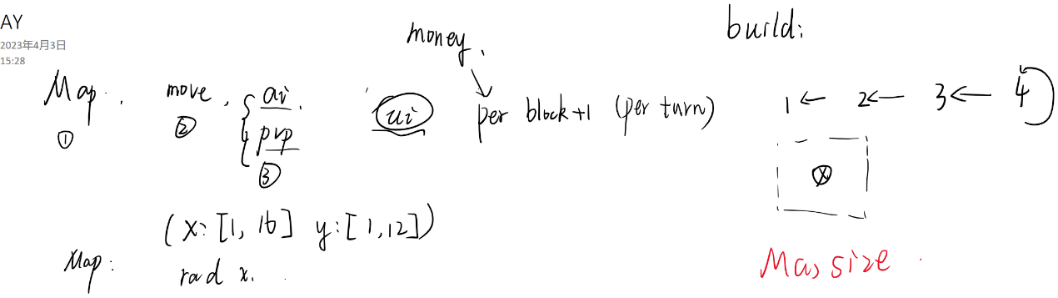


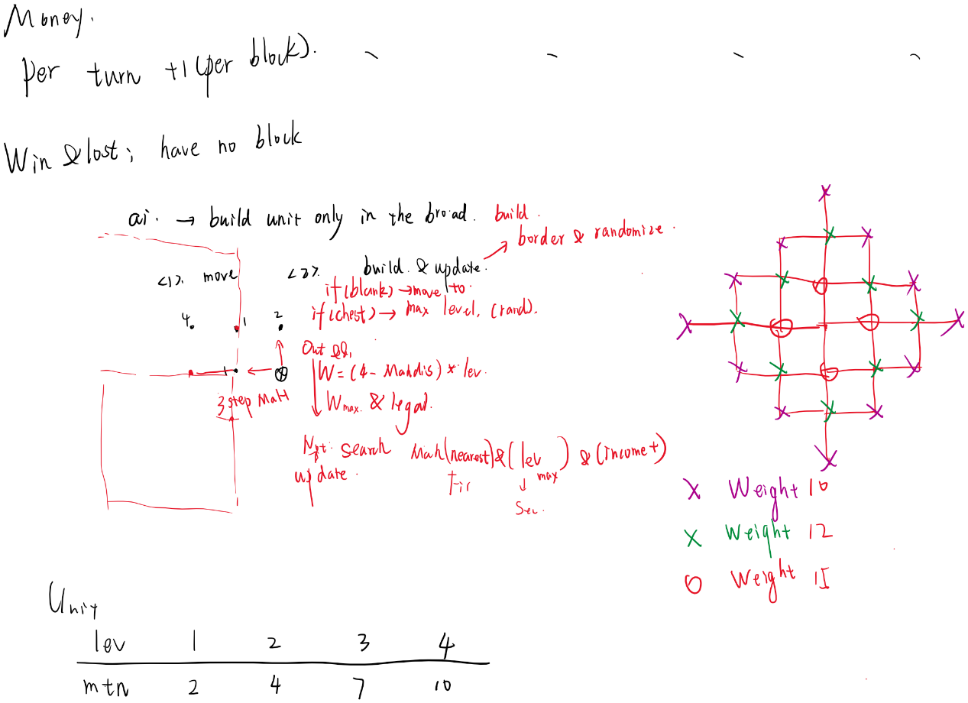


**开发流程：**

**立项和初期规划：**

确定制作战棋游戏大概于2023年4月初。由于Yiotro在游戏Slay基础上开发的Antiyoy（几乎）是仅有的兼具看海模式（即没有玩家，全程ai对决，人类观看）、地图编辑器、自由选择人数单机多人模式于一体的战棋游戏，且其的游戏规则极其简单，熟练掌握却很难（尤其是高手对局时），用作者Yiotro的话来说就是“Easy to learn, hard to master.”同时，这款游戏的大小仅有7Mibs，在使用并不高配的手机的高中时代，这款游戏成为高中宿舍集体生活中少有的多人同时和单人兼具的娱乐方式，遂决定以此为蓝本，制作一款类似的战棋游戏。





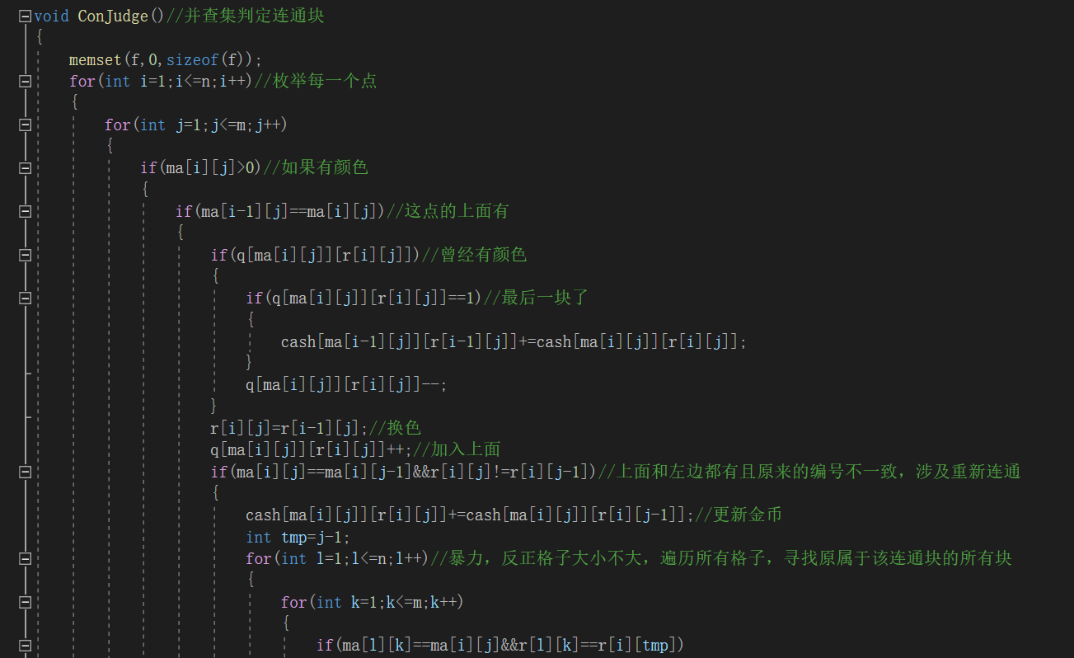
早期规划的截图（使用OneNote所做的草稿）

第一张为游戏流程的设计，第二张是ai权值计算的初稿。

**控制台代码编写：**

由于此前几乎没有接触过可视化编程，初高中长期使用基于控制台的编程方式，遂决定先编写游戏规则和游戏运行模块的代码再进行图形化。

游戏规则和游戏运行是项目最核心的部分，也是项目出现bug最多的部分，测试和修复bug消耗大量时间。在4月23日才完成基本的控制台代码，可以使用txtprintChessBoard()打印的棋盘进行一次正常的游戏了。

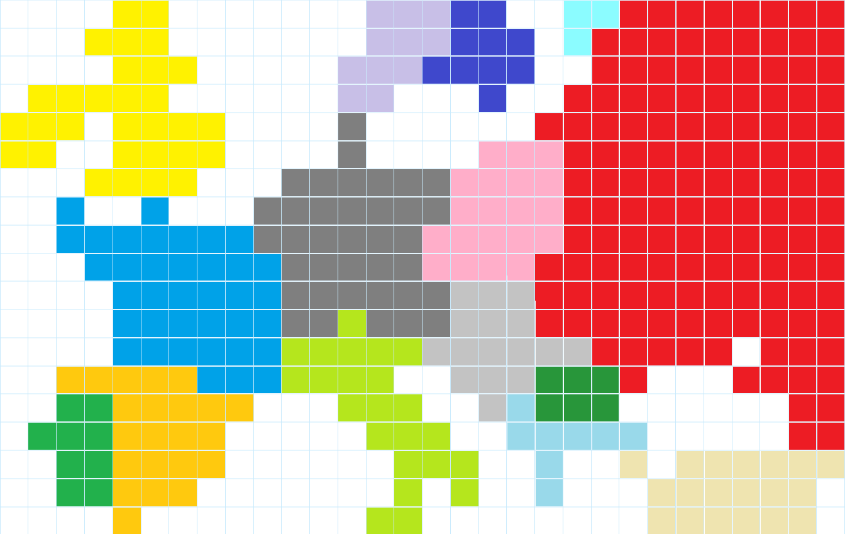


作为游戏运行核心的Conjudge()函数

大量的注释因为这里是bug的高 发区，该段代码光重构就重构了四次

**图形化代码编写：**

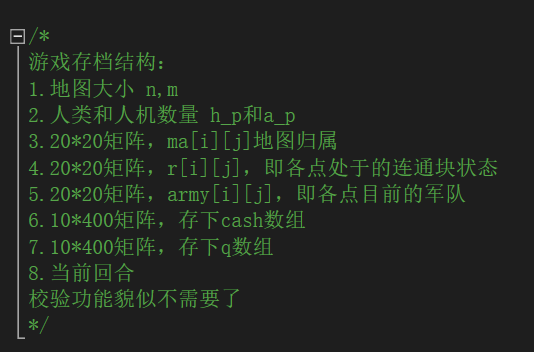
控制台代码完成后，需要查阅大量资料从头学习EasyX的使用。决定先编写按钮类，因为游戏中有大量按钮，使用统一的按钮结构体可以有效减轻代码长度和工作量。此后，先完成“毛坯版”的游戏菜单和游戏的编写，其他部分暂时保留控制台。此时游戏运行又出现了很多bug。修复完成大约在4月29日左右。



游戏的背景图，使用Excel和画图软件的填色桶画格子的。

图片是抽象化的西班牙内战结束至二战爆发前的欧洲局势。

**其他功能的图形化编写：**

最主要的就是编辑器的图形化编写了。因为编辑器的工作量相当于第二个游戏环节。但正是由于这个原因，可以大量复用游戏环节的代码，从而减轻工作量。此外，关于游戏存档，原本计划是类似Word等软件打开文件夹存读档，但是由于EasyX只能打开一个窗口只好作罢，此后创造性的想出了存档插槽这一折中办法，实现分别存档。

最新的存档结构，校验功能是最初进行任意文件存读档时准备的，

后来使用插槽，不需要了。

存档结构发生过多次改变，起初将n和m写反导致存读档错误，后来又因为cash和q数组的误判导致存读档后金币数据极其诡异，后面认为直接以文本形式存档即可，因为地图很小，文本形式也不会占用过大的空间。

**音乐播放器：**

EasyX使用的MciSendString的api，最开始由于网易云音乐下载的mp3格式是加密过的无法直接使用MciSendString(“play”)打开，后来只能将其转换为.wav格式，导致文件大小极其夸张。此外，在我的电脑上MciSendString(“repeat”)无法正常使用，所以只能写一个函数来控制音乐的随机播放。

编写完之后发现声音音量时空，故而重新转换了一次.wav压音量，导致文件体积更大。音乐包括玩梗和历史向音乐两种。下面是具体列表：

|  |  |
| --- | --- |
| 游戏内名称 | 音乐名称 |
| 脱粪の小曲 | 一般男性脱糞シリーズ |
| 1444の小曲 | The Voyage（欧陆风云4主题曲） |
| 1836の小曲 | Rule the World（维多利亚3主题曲） |
| 1936同盟国の小曲 | HOI4 Main Theme Alies（钢铁雄心4主题曲兼同盟国主题曲） |
| 1936苏维埃の小曲 | Comintern Theme（钢铁雄心4共产国际主题曲） |
| 1936洗头佬の小曲 | Axis Theme（钢铁雄心4某种意识形态的主题曲） |
| 凯南开大学の小曲 | 来源于一个b站视频“电棍：凯南开大学校歌”（只有补档） |
| 只因你太美の小曲 | 只因你太美 |

蓝色为历史向音乐，均为瑞典游戏公司Paradox Interactive制作的诸多大战略游戏的背景音乐。

棕色为玩梗音乐，包括日本现代年轻人的亚文化，中国的一些亚文化和著名网红的成名曲。

在获取音乐播放时长上，最初采用了笨方法：即手动建立一个数组来记录音乐播放时长，后来发现可以调用mciSendString的status中position和length两个api来完成，同时可以获取当前播放时长。

**小修补：**

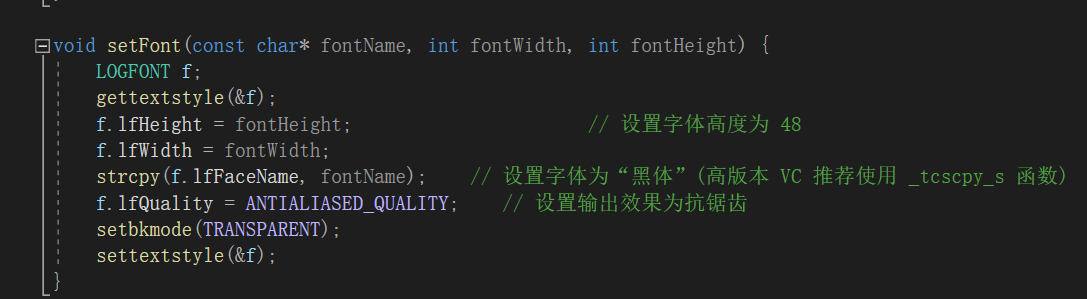
此后进行了一些小修补，比如返回键、音乐播放器的自定义功能等，由于mciSendString的api无法使用L”String”, \_T”String”, ch.c\_str()中的任何一种形式的字符串，无法加入播放自定义音乐名称和路径的功能。

并加入了几张地图预设。

**代码介绍**

代码部分分为tools.cpp，common.cpp，UI.cpp，main.cpp四个子cpp文件，其中主要功能都在main.cpp中。

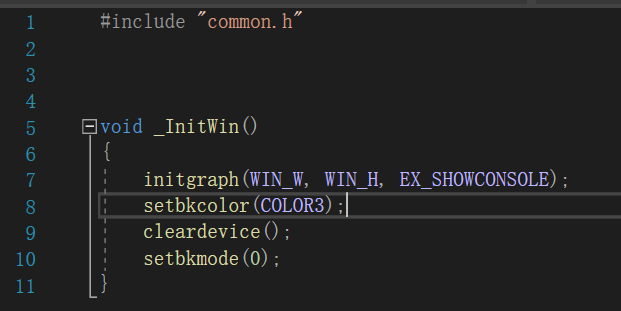
**tools.cpp（外部引入的经典代码模板）**

这个部分主要是一些常用工具，包括透明抠图，高质量文字等（其中的播放音乐功能本机不明原因无法使用。

部分代码截图

**common.cpp（常量定义和窗口加载函数）**

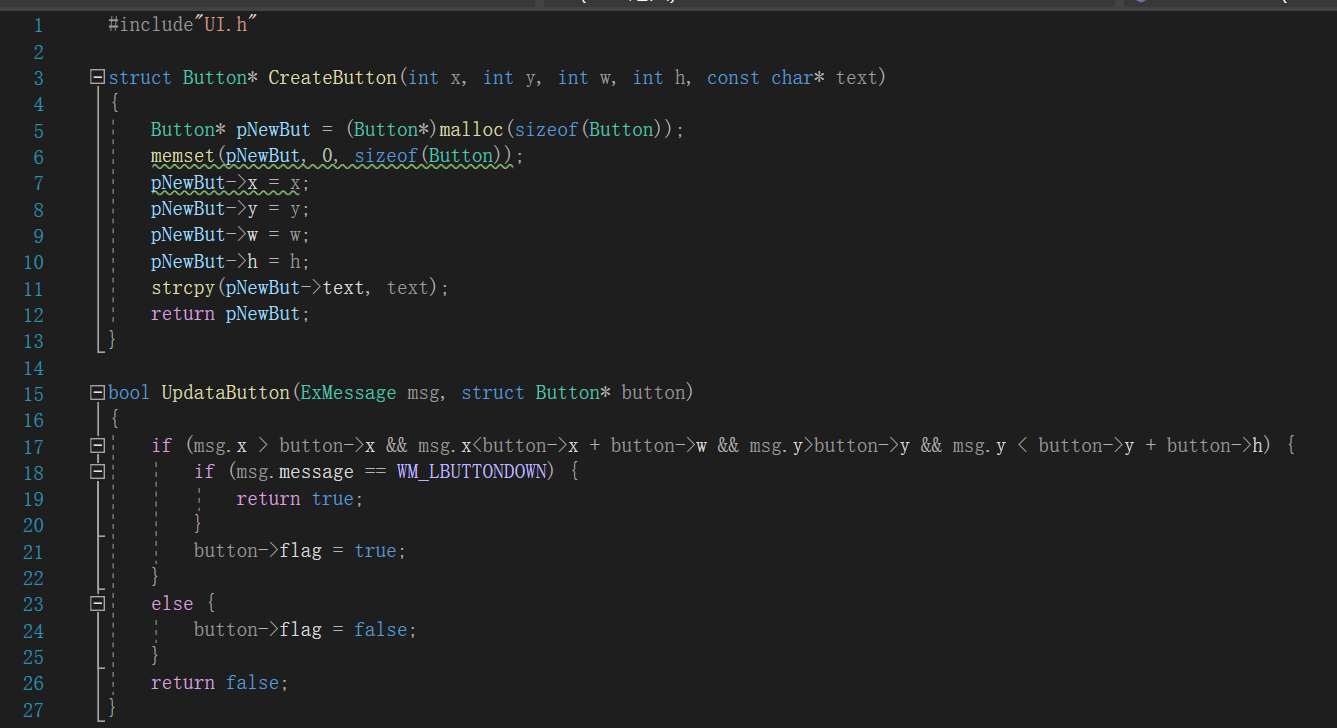
这个部分主要是一些常量的定义和窗口加载函数。后来因为头文件连接出错，导致常量定义移动到了主函数下。



代码截图

**UI.cpp（按钮结构体定义）**

这个部分主要定义了按钮的创建、绘制、触发等函数，节约了编写代码的时间。





代码截图

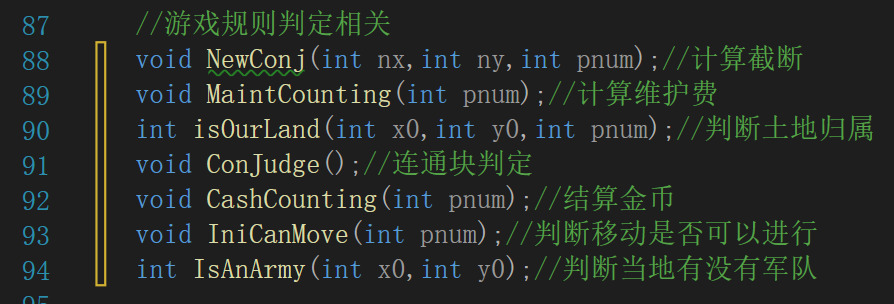
**main.cpp（主函数）**

这个部分由几类共47个函数组成，构成了程序的大部分内容，实现了程序的大部分功能。

1. 常量和全局变量定义。

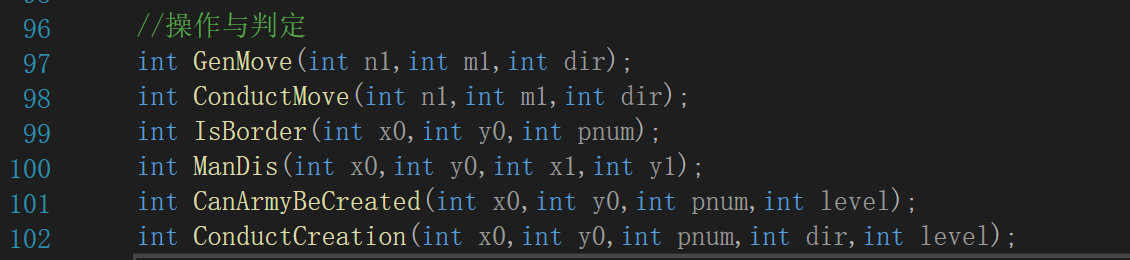
本部分定义了大部分全局变量和常量，如地图颜色，地图归属，兵种位置，音乐名称等。

2、游戏规则判定



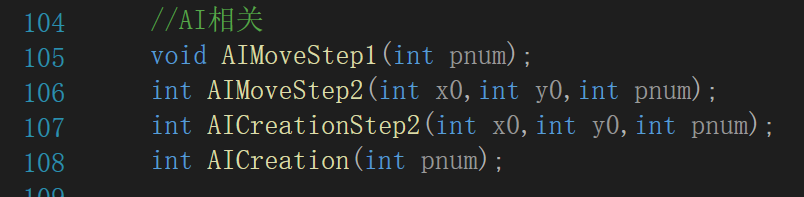
游戏的基础规则判定部分，其中的Conjudge()函数是整个游戏运行过程中的基石，它使用朴素的并查集（DSU）算法进行连通块判定，也是bug的高发区。绝大部分函数都与之相关。

1. 操作相关



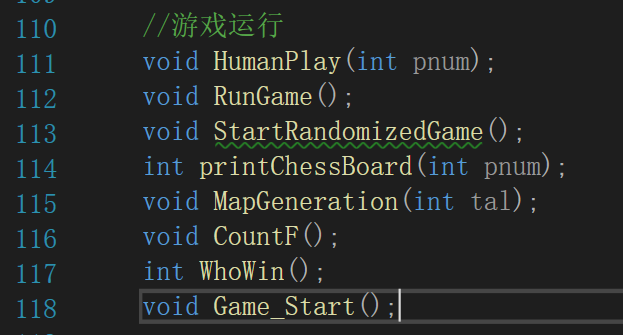
分为两类，造兵和移动。GenMove()和CanArmyBeCreated()是判断是否可以移动或建造军队，ConductMove()和ConductCreation()是执行移动和建造操作。Mandis()是计算曼哈顿距离，IsBorder()返回与己方领土颜色不同的敌方领土个数。

1. AI（电脑玩家）相关



都分为两个Step，分步解决电脑玩家的移动和建造行为。其中，AIMoveStep1()和AICreation()都是判定哪些地方应该被执行操作，然后下放到AIMoveStep2()和AICreationStep2()中，区别是，AIMovestep2()需要再调用一次ConductMove()函数，而AICreationStep2()直接完成操作。

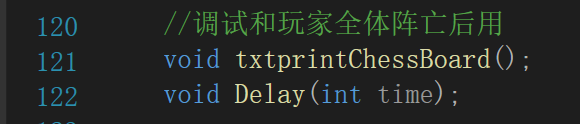
5、游戏交互



主要包括游戏和玩家的交互，HumanPlay()是玩家回合的界面绘制，RunGame()是游戏的运行全程，StartRandomizedGame()和MapGeneration()是随机地图游戏的两个步骤，前者负责绘制图形化选择界面，后者使用随机游走（Random Walk）算法生成一张保证联通的游戏地图。

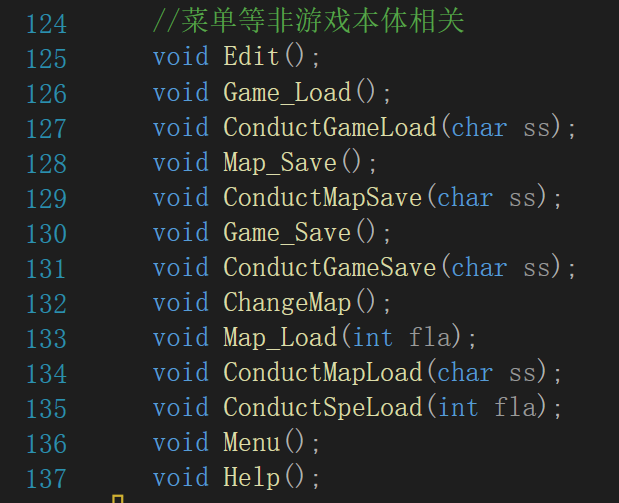
printChessBoard()绘制了一张游戏地图，CountF()，Whowin()是不同时期判断游戏结束与否的函数，Game\_Start()函数是执行不同游戏选项并开始游戏或返回错误信息的函数。

1. 调试和人类玩家全体失败后的函数



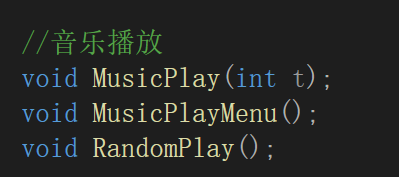
txtprintChessBoard()函数是命令行阶段的棋盘输出函数，后被printChessBoard()函数取代，成为调试工具。Delay()则是玩家全体阵亡后观看电脑操作（看海模式）的延时函数。

7、游戏的附属功能



主要包括地图编辑器、存读档、菜单界面和帮助程序。Edit()函数是编辑器界面，ChangeMap()是在编辑器内调整地图大小和游戏玩家情况。Help()是弹出一个帮助窗口。Game\_Load(), Game\_Save()等Load&Save族函数，是存读档的UI界面，而Conduct Load&Save族函数则执行存读档。Game.sav是游戏过程中存储的游戏存档，而Map.sav是预设或者编辑器的地图存档。其中，Map\_Load(int fla)较为特殊，因为其包括两种读取地图存档：开始游戏时读取编辑器存档和在编辑器中读取此前的编辑器存档和读取并修改预设存档进行游戏两种，所以传入一个参数进行区别。

1. 音乐播放器



MusicPlay(int t)函数是播放第t首乐曲，MusicPlayMenu()函数是音乐播放器的UI界面。RandomPlay()控制随机播放历史向音乐。

**算法或公式**

**并查集（Disjoint Set Union, DSU）**是一种主要用于解决一些元素分组的问题。它管理一系列不相交的集合，并支持两种操作：

合并（Union）：把两个不相交的集合合并为一个集合。

查询（Find）：查询两个元素是否在同一个集合中。

伪代码（不含路径优化和启发式合并）：

初始化：int fa[MAXN];

for i:1→n fa[i] = i;

查询：return (fa[x] == x)?x: find(fa[x]);

合并：fa[find(i)] = find(j);

本实验中主要用于解决地块归属和分类问题，即源码中的*r[i][j]*相当于*fa*数组，源码中的*ConJudge()*函数则是Union操作的体现。

**随机游走（Random Walk）算法**：从一个或一系列顶点开始遍历一张图。在任意一个顶点，遍历者将以一定概率游走到这个顶点的某个邻居顶点，每次游走后得出一张图，则这张图是连通图。

本实验中的体现：随机生成一张连通的地图。

伪代码：

int dis=rand()%4;

if(New node) position of Node[cnt++]=walk (from:now position by:dis);

**项目测试**

对游戏及代码进行多次测试。

**测试项目：**

包括但不限于是否能通过编译，控制台形式代码是否能正常运行，图形化游戏是否可以打开，游戏运行是否有超出规则的地方，游戏之外（编辑器、音乐播放器，存读档体系）的功能是否可以正常使用等。

**测试结果：**

测试中出现过不少于100个通过编译后的bug，修复bug消耗了大量时间。例如：移动单位时一个m写成n导致的地图大小判定错误，音乐播放器的文件路径名使用“/”而不是“\”，许多数组没有进行初始化或开启新一局时没有清零导致的累计错误等等等等。

目前，绝大部分bug已经被修复。

**收获**

**EasyX的图形化可以定义按钮结构体，节约大量时间。**

struct Button\* CreateButton(int x, int y, int w, int h, const char\* text)//创建一个按钮

{

Button\* pNewBut = (Button\*)malloc(sizeof(Button));

memset(pNewBut, 0, sizeof(Button));

pNewBut->x = x;

pNewBut->y = y;

pNewBut->w = w;

pNewBut->h = h;

strcpy(pNewBut->text, text);

return pNewBut;

}

bool UpdataButton(ExMessage msg, struct Button\* button)//将按钮链接到信号槽msg

{

if (msg.x > button->x && msg.x<button->x + button->w && msg.y>button->y && msg.y < button->y + button->h) {

if (msg.message == WM\_LBUTTONDOWN) {

return true;

}

button->flag = true;

}

else {

button->flag = false;

}

return false;

}

void DrawButton(const struct Button\* button) {//显示按钮

setlinecolor(BLACK);

setlinestyle(0, 1);

if (button->flag) {

setfillcolor(COLOR2);

}

else {

setfillcolor(COLOR1);

}

settextcolor(LIGHTCYAN);

settextstyle(button->h \* 2 / 3, 0, "黑体");

fillrectangle(button->x, button->y, button->x + button->w, button->y + button->h);

outtextxy(button->x + (button->w - textwidth(button->text)) / 2, button->y + button->h / 6, button->text);

}

**MciSendString可以获取当前音乐的播放剩余时间**

s="当前播放：";

s+=ch[nmusic];

settextcolor(BLACK);

//统计长度

mciSendString("status bk position", sch, 255,0);//目前位置

lLength=atoi(sch);

lLength/=1000;

mciSendString("status bk length", sch, 255,0);//总长

len=atoi(sch);

len/=1000;

lLength=len-lLength;//总长-目前时长=剩余时长

s+=" ";

if(lLength/3600>0)

Musich=lLength/3600,lLength%=3600,s+=to\_string(Musich),s+="：";//通常情况下隐去小时分隔符

if(lLength/60<10) s+="0";

Musicm=lLength/60;

s+=to\_string(Musicm);

s+="：";

lLength%=60;

if (lLength % 60 < 10) s += "0";

Musics=lLength%60;

s+=to\_string(Musics);

注意：这段代码只有音乐文件是.wav格式时可以使用，其他文件格式（如.mp3格式）会出现获取总时长错误的bug。

**EasyX的绘图实际上并不清除此前图片**

使用outtextxy()函数时可以发现，此前输出的数字并不会被抹除。这是因为EasyX的绘图实际上是在已有基础上只覆盖需要被改变的像素点。

解决方法很简单，每次重新打印一张背景图来抹除所有像素点，再重新绘图。缺点是：地图较大时，可以出现明显的卡顿，并且没有很好的解决方法。

**善用string相关函数**

c.str(), to\_string, stringstream ss和sprintf()等函数大大减少了工作量，string的高度可操作性和衍生库的库函数使编程更方便，尤其是在接入需要char类型或Tchar()或Lchar()类型函数时。

**大规模使用Button和msg会导致严重卡顿**

当地图大小超过12×12后，游戏面临严重卡顿。原因是大规模调用msg信号和UpdataButton()函数导致的。同时，EasyX原生库仅支持一个窗口的绘制。故而，EasyX不适合进行游戏的编写，编写游戏还需要专门的游戏引擎来编写，而不是Qt或者EasyX。

**目前存在的缺陷**

**1、【严重】【稳定复现】【体验问题】卡顿问题**

【现象】前文已述。

【解决】无法解决，建议不要使用过大的地图。

**2、【严重】【偶发】【BUG】金币计算错误**

【现象】有时会发生某些地块有数万金币

【解决】已修复多次，虽然目前仍然可能出现此问题，但是可能性已经大大下降，若出现此类问题，建议弃档退出游戏重进。

**3、【轻微】【偶发】【BUG】部分地块无法向敌方境内造兵**

【现象】某些回合，无法向地方境内建造军队。

【解决】再过一回合或存档，或退出游戏重进即可。

**4、【轻微】【稳定复现】【体验问题】游戏体积太大**

【现象】整个文件过大。

【解决】无法解决，因为音乐文件只能使用.wav格式，而减少文件体积只能在降低音质上进行，但是测试后发现效果很差。

**5、【严重】【偶发】【BUG】进入游戏出现大量错误**

【现象】读档报错、读档出空白、音乐无法播放、金币持续错误计算、无法移动等同时出现多个。

【解决】原因不明，可能和系统内存有关。目前已经极少见，若出现请退出重进。