

夺岛军演需求分析报告

C 语言课程设计

专业班级：自动化 2201 班

小组成员：刘政孜 U202214933

李鹏程 U202214930

指导老师：周纯杰、何顶新、汪国有、左峥嵘、

周凯波、彭刚、高常鑫、陈忠

上交时间：二〇二三年 三月 七 日

目录

第一部分 前言.....	3
一、 编写背景.....	3
二、 软件功能.....	4
三、 参考资料.....	4
四、 参考软件.....	4
五、 编者的话.....	4
第二部分 任务概述.....	5
一、 目标功能.....	5
二、 编写规范.....	5
第三部分 运行环境和配置.....	6
一、 硬件接口.....	6
二、 软件接口.....	6
三、 控制.....	6
第四部分 需求分析与系统设计.....	7
一、 需求分析.....	7
二、 系统设计.....	8
第五部分 数据结构设计.....	9
第六部分 界面设计.....	11
一、 鼠标设计.....	11
二、 界面设计.....	11
第七部分 主要函数说明及函数原型.....	16

第一部分 前言

一、编写背景

台湾问题事关中国主权和领土完整，涉及中国核心利益。台湾问题纯属中国内政，事关中方核心利益，不容任何外部势力以任何方式进行干涉。

台湾自古以来是中国的一部分，台湾问题不容任何外来干涉。中国人民解放军东部战区在台岛周边海空域组织多军兵种联合战备警巡和实战化演练，就是针对美台勾连、根据维护国家主权和领土完整需要采取的必要行动。中国人民解放军枕戈待旦，采取一切必要措施，坚决挫败外部势力干涉和“台独”分裂图谋。

解决台湾问题，实现祖国完全统一，是全体中华儿女共同愿望，是中华民族根本利益所在，也是中国共产党和中国政府的神圣历史使命。

中国政府一直以来坚定不移地坚持“和平统一、一国两制”的基本方针，全面贯彻现阶段发展两岸关系、推进祖国和平统一进程的八项主张。将继续以最大的诚意、尽最大的努力，争取和平统一的前景，但绝不允许任何人、以任何方式把台湾从中国分割出去。

但是近年以来，以美国为首的部分西方国家屡屡触碰台湾问题红线，使得台海局势空前紧张。尽管中国政府一直以来坚定不移地坚持“和平统一、一国两制”的基本方针，但也从未放弃武力统一台湾的准备。

以此为背景，我们小组借此 c 课设机会制作了“夺岛军演”游戏。

二、 软件功能

利用 c 语言结构及相关函数实现一个攻岛模拟塔防回合游戏。在本游戏中，玩家通过操纵我方单位（士兵、飞机、坦克、运输船、支援兵等）来攻打一个模拟的岛屿，岛屿中存在敌方单位（主城、炮台、城墙、叛军等），敌方通过特定的优先级来搜索并攻击我方单位，而我方也可通过一定的排兵布阵以及策略来攻击敌方，最终，当我方攻破敌方主城时，玩家赢得游戏胜利，而当我方在一定回合内未攻破敌方主城时，则判定玩家游戏失败。

三、 参考资料

1. 王士元. C 高级实用程序设计. 北京: 清华大学出版社. 1996 年
2. 周纯杰, 何顶新等. 程序设计与应用 (用 C/C++ 编程) 北京: 机械工业出版社. 2008 年[美] Prata. C Primer Plus (第六版) 北京: 人民邮电出版社. 2016 年
严蔚敏, 吴伟民编著.
3. 数据结构 (C 语言版) 北京: 清华大学出版社. 2018 年

四、 参考软件

文明 6

五、 编者的话

在第一眼看到选题表之后，我们小组就确定了要选择“夺岛军演游戏”这个主题。

随着智能化时代一步步到来，很多原本只能在线下进行的军事力量的

演练对抗可以在线上模拟，这样不仅可以减少实际军演时带来的资源浪费，在实际作战时也能帮助指挥官更为有效地指挥战斗。因此，尽管我们制作的仅仅是一款模拟游戏，但是我们相信这款游戏代表着一种信息化战争的趋势。

同时编写这一游戏的过程也提高了我们小组的程序设计能力，对我们今后的学习和未来进行相关的工作时积累了一定的经验。

第二部分 任务概述

一、 目标功能、

1. 我方单位移动功能：通过 BFS 算法，根据我方单位规定所能移动的最大格数，在地图上显示所有能到达的最大为位置，点击能到达的位置，便可使我方单位移动到预定位置。
2. 我方单位攻击功能：根据规定的最大攻击范围，在地图上显示可攻击的敌方单位，点击要攻击的敌方单位，即可对敌方单位发动攻击。
3. 敌方单位自动攻击功能：根据规定的最大攻击范围，并通过我方对敌方主城的威胁程度，以一定的优先级进行排序，自动对我方单位发出攻击。
4. 排行榜功能：对玩家胜利的回合数进行排名，回合数越少，排名越靠前。

二、 编写规范

1. 命名规范

变量命名，应该尽量用英文或拼音表达其准确定义。较为重要或不常见的变

量名应配合注释说明其主要功能。

2. 函数命名应该用英文或拼音表达其确定含义。

文件命名都用小写，并且表达出该文件所包含函数的主要功能。

涉及数据结构的命名应参考数据结构（C 语言版），并进行适当修改。

3. 注释

在每种函数上方使用`/**`注释，内容前后空一格。注释内描述该函数。

使用`//`注释时，添加在一行代码之后，跟代码之间空一格。

部分令测试者比较难以理解的算法和流程应该给出相应的注释。

第三部分 运行环境和配置

一、硬件接口

处理器：Intel Pentium 166 MX 或以上

硬盘：空间 500MB 以上

屏幕适配器：VGA 接口

系统运行内存：要求 32MB 以上

二、软件接口

开发软件工具：Borland C++ 3.1

文字编辑工具：Vscodc、Dev C++、sublime、Notepad++

数据储存方式：文本存储（记事本）

操作系统：DOS

三、控制

该游戏通过鼠标和键盘控制。玩家通过将鼠标移至开始游戏进行点击后进入游戏，通过键盘完成用户名的输入功能。操作完成后正式进入游戏内容，玩家通过鼠标完成我方单位的各种功能（移动、攻击、结束等），同时也可通过鼠标完成对敌方信息的查看，最终游戏通关或失败

第四部分 需求分析与系统设计

一、需求分析

本游戏意在模拟攻占岛屿的军演作战系统。因此，我们需要对现实中的战场进行抽象与简化，形成一个回合制攻防游戏。经分析，我们认为本模拟游戏应至少包含以下这 4 个要素：地图场景设定，我方兵力调度，我方后勤补给，敌方反击方式。

地图场景设定用于模拟还原真实战场中的地形情况。

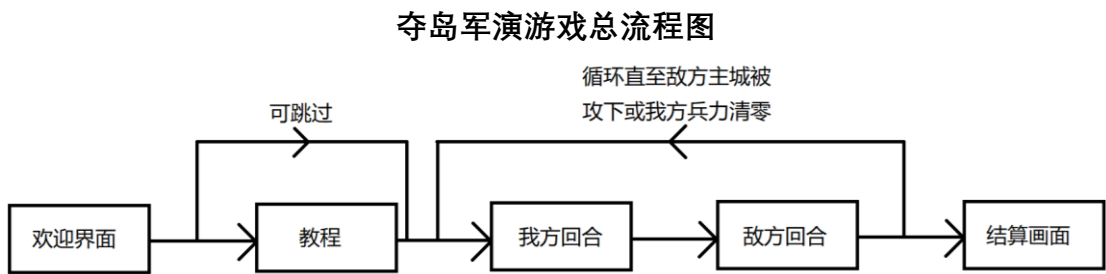
我方兵力调度用于模拟真实战场中指挥官调度作战单位进行移动、攻击的过程。

后勤补给同样是影响战场局势的重要因素，当某作战单位所携带的资源（燃油、军粮等）耗尽时同样会失去作战能力。因此我们的模拟游戏应具有剩余资源值和支援兵来对这种情况进行模拟。

在实际的演习中，敌方不总是被动挨打的标靶，而是会针对我方行动进行反击，所以，本游戏会根据某些条件自动攻击我方单位，进而模拟真实战场上敌我双方的对抗。因此在我方回合结束后应当设立敌方回合来模拟敌方的反击方式。

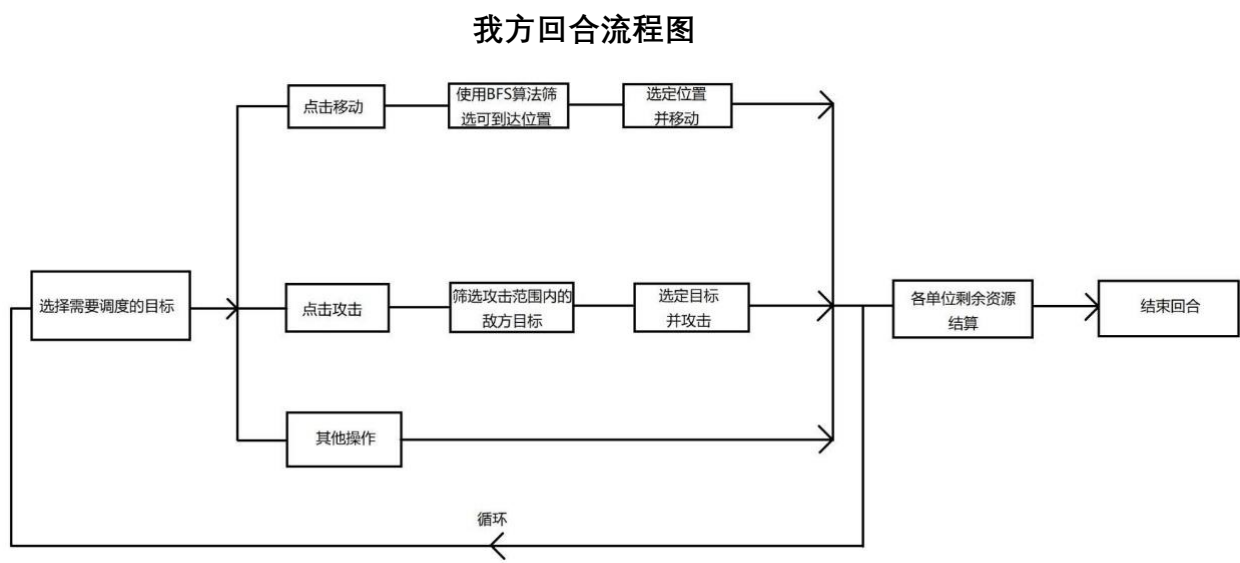
二、系统设计

总流程：



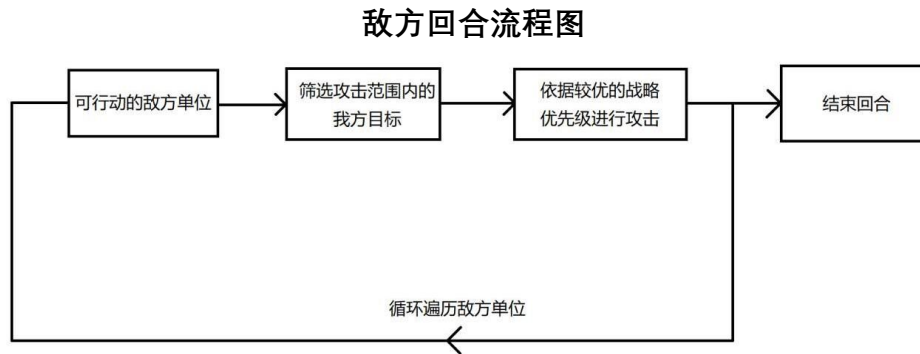
图一：夺岛军演游戏总流程图

我方回合流程：



图二：我方回合流程图

敌方回合流程：



图三：敌方回合流程图

第五部分 数据结构设计

本程序一共定义了三个结构体。

(1) unit 类型结构体

```

struct unit{
    char *name; //战斗单位名称（中文）
    int type; //战斗单位类型，统一编码，比如坦克是0等等
    int num; //战斗单位编号，指在数组里的下标
    int hp[15]; //剩余血量
    int oil[15]; //剩余资源
    int x[15],y[15]; //像素位置
    int a[15],b[15]; //地图位置
    int moved[15],atked[15]; //记录某个单位本回合是否移动过/攻击过
    int max_move; //最大移动距离
    int max_attack; //最大打击距离
    int atk; //攻击力（打一次对面掉多少血）
    //以下是一些关于运输船的参数
    int ported[15]; //表示单位是否被运输，0表示未被运输，1表示被运输，被运输时在地图上不显示
    int port_type[15]; //仅运输船会用到的参数，表示运输物的兵种
    int port_num[15]; //仅运输船会用到的参数，表示运输物在该兵种中的标号
};
  
```

图四：unit 结构体

如图，unit 结构体记录了一类军事单位的所有信息。结构体中各字段表示的意义如下：

char *name: 表示该类战斗单位的名称

int type: 表示该类单位的类型编号

int num: 表示该类单位中共有多少个单兵

int hp[]: 表示每个单兵的剩余血量

int oil[]: 表示每个单兵的剩余资源

int x[],y[]: 表示每个单兵所在的像素位置

int a[],b[]: 表示每个单兵所在的格子坐标位置

int moved[],atked[]: 表示每个单兵在本轮是否已经移动/攻击过

int max_move: 表示该兵种每次的最大移动距离

int max_attack: 表示该兵种每次的最大攻击距离

int atk: 表示该兵种的攻击力

int ported[]: 表示该单位是否被运输

int port_type[]: 仅运输船会用到的参数，表示运输物的兵种

int port_num[]: 仅运输船会用到的参数，表示运输物在该兵种中的标号

(2) node 类型数据结构，配合 BFS 算法使用，用于记录目前搜索队列里点的信息

```
struct node{  
    int x,y,rest;  
};
```

图五：node 结构体

int x,y: 表示入队的点的坐标

int rest:表示剩余的可用步数

(3) abk_atk 型结构体，用于记录敌方搜索到可以被攻击的我方单位信息

```
struct ab_atk {  
    int x;  
    int y;  
    int code;  
    int type;  
};
```

图六：ab_atk 结构体

int x: 表示我方单位的地图格子横坐标

int y: 表示我方单位的地图格子横坐标

int code: 表示我方单位的序号（同一军种）

int type: 表示我方单位军种类型

第六部分 界面设计

一、鼠标设计

我们直接采用了邓述民学长留下来的鼠标代码，节省了很多时间

二、界面设计



图七：游戏初始界面



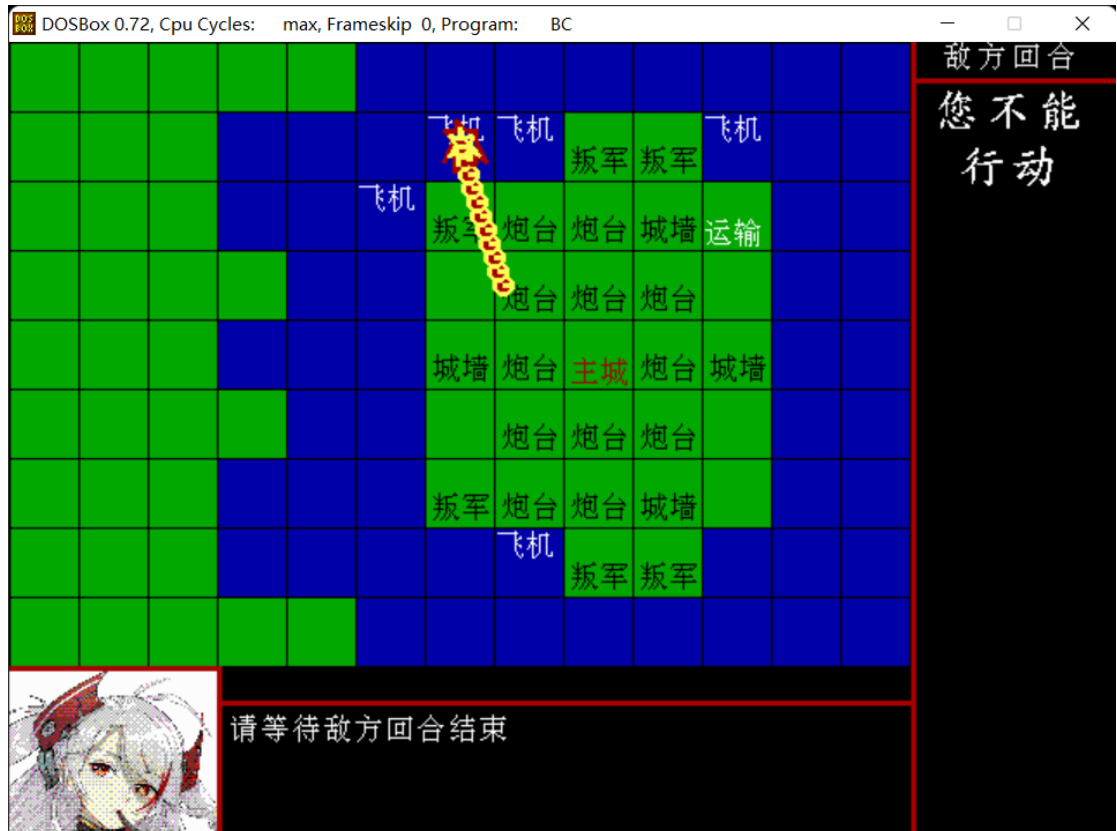
图十：查看作战单位信息界面



图十一：我方回合攻击界面



图十二：我方回合移动界面



图十三：敌方回合界面



图十四：结算界面一



图十五：结算界面二

第七部分 主要函数说明及函数原型

一、 common.h 库包含以下通用性较强的函数

1. `int get_dis(struct unit a,struct unit b);`

函数描述：用于获得两个单位之间的距离

输入参数：两个 unit 类型的作战单位变量 a,b

返回值：表示距离的整数值

2. `void nts(int x,char *s);`

函数描述：用于将数字 x 转换为字符串并储存在 s 中

输入参数：整型变量 x 和 char 型结构指针变量 s

返回值：无

3. `void print_map(int (*map_type)[13+3]);`

函数描述：用于根据 map_type 中的地形设定打印地图

输入参数：整型 9×13 数组指针 map_type

返回值：无

4. `void print_UI();`

函数描述：用于打印 UI，包括头像和边框

输入参数：无

返回值：无

5. `void init(int(*map_type)[13 + 3], struct unit my[], struct unit en[]);`

函数描述：用于游戏的初始化。包括设定地形、设定我方单位数值、设定

敌方单位数值和位置

输入参数：整型 9×13 数组指针 map_type, unit 型作战单位数组 my 和 unit 型作战单位数组 en

返回值：无

6. int get_block(int MouX,int MouY,int *x,int *y);

函数描述：用于根据鼠标的位置判断所在格子（其中形参中的 x, y 传递格子坐标位置，返回值是 0 表示点击的是该格子上部，返回值是 1 表示点击的是该格子下部）

输入参数：整型变量 MouX,MouY, 整型指针*x,*y

返回值：表示格子上下部的整型变量 half

7. void print_en(struct unit en[]);

函数描述：用于打印敌方单位

输入参数：unit 型作战单位数组 en

返回值：无

8. void print_my(struct unit my[],int num_my);

函数描述：用于打印我方单位

输入参数：unit 型作战单位数组 my

返回值：无

二、 welcome.h 中包含的函数

1. int welcome(char *s);

函数描述：用于展示欢迎界面。s 用于传递欢迎过程中用户输入的名字，

返回值表示是否需要教程

输入参数：char 型结构指针变量 s

返回值：表示是否需要教程的整型 op

2. void place(int(*map_type)[13+3], struct unit my[],struct unit en[],int num_my,int num_en);

函数描述：用于初始布置我方单位位置

输入参数：整型 9×13 数组指针 map_type，unit 型作战单位数组 my 和 unit 型作战单位数组 en，表示我方单位总类型数的整型变量 num_my，表示敌方单位总类型数的整型变量 num_en

返回值：无

3. void teach();

函数描述：用于展示教程

输入参数：无

返回值：无

三、 info.h 中包含的函数

1. void en_turn();

函数描述：用于在右侧信息栏打印敌方回合信息

输入参数：无

返回值：无

2. void my_turn();

函数描述：用于在右侧信息栏打印己方回合信息

输入参数：无

返回值：无

3. void info(struct unit x,int num,int op,struct unit my[],int butt);

函数描述：用于在右侧信息栏打印某一作战单位信息

输入参数：unit 型作战单位变量 x，表示输出是该兵种中第几个单位的整型变量 num，表示是否是己方单位的整型变量 op，unit 型作战单位数组 my，表示是否需要显示按钮的整形参数 butt

返回值：无

四、 anime.h 中包含的函数

1. void print_atk(int x,int y,int xx,int yy);

函数描述：用于显示攻击动画

输入参数：表示发起攻击方坐标的整型变量 x,y，表示被攻击方坐标的整型变量 xx,yy

返回值：无

五、 myturn.h 中包含的函数

1. void bfs(struct unit x,int num,int (*map_type)[13+3],int (*f)[13+3],struct unit my[],struct unit en[],int num_my,int num_en);

函数描述：用于搜索一个单位能走到哪些格子并将其标记到 f 里

输入参数：unit 型作战单位变量 x，整型 9×13 数组指针 map_type，整型 9×13 数组指针 f，unit 型作战单位数组 my，unit 型作战单位数组 en，表示我方单位总类型数的整型变量 num_my，表示敌方单位总类型数的整型变量 num_en

返回值：无

2. int myturn(int (*map_type)[13+3],struct unit my[],struct unit en[],int num_my,int num_en);

函数描述：用于执行我方行动轮

输入参数：整型 9×13 数组指针 map_type, unit 型作战单位数组 my, unit 型作战单位数组 en, 表示我方单位总类型数的整型变量 num_my, 表示敌方单位总类型数的整型变量 num_en

返回值：表示在本轮中主城是否被攻占的整型变量 op

六、 enturn.h 中包含的函数

1. int enturn(struct unit my[], struct unit en[]);

函数描述：用于执行敌方行动轮

输入参数：unit 型作战单位数组 my, unit 型作战单位数组 en。

返回值：0 或 1, 表示本回合内我方单位是否被完全消灭。

2. void en_p_search(struct unit my[], struct unit en[], int en_atk[], struct ab_atkable[]);

函数描述：用于执行敌方搜索我方可以被攻击的单位

输入参数：unit 型作战单位数组 my, unit 型作战单位数组 en。int 型数组 en_atk, 表示发出攻击的敌方炮台号码信息, ab_atk 型数组 able, 表示被攻击的我单位地图位置信息。

返回值：无。

3. void en_print_atk(struct unit my[], struct unit en[],int en_atk[], struct ab_atkable[]);

函数描述：用于打印敌方攻击动画。

输入参数：unit 型作战单位数组 my, unit 型作战单位数组 en。int 型数组 en_atk, 表示发出攻击的敌方炮台号码信息, ab_atk 型数组 able, 表

示被攻击的我单位地图位置信息。

返回值：无。

4. void en_p_dis_blood(struct unit my[], struct unit en[], struct ab_atk able[]);

函数描述：用于扣除被攻击的我方单位血量。

输入参数：unit 型作战单位数组 my，unit 型作战单位数组 en。int 型数组 en_atk，表示发出攻击的敌方炮台号码信息，ab_atk 型数组 able，表示被攻击的我单位地图位置信息。

返回值：无。

七、 data.c 中包含的函数

1. int exist(char *s);

函数描述：用于判断玩家名字之前是否存在

输入参数：char 型结构指针变量 s

返回值：表示玩家名字先前是否存在的整型变量 judge

2. void rank(char *s,int cnt);

函数描述：用于写入新的战斗数据并显示结算场景

输入参数：char 型结构指针变量 s，表示回合数的整型变量 cnt

返回值：无