

# 华中科技大学

人工智能与自动化学院

C 语言课程设计

红蓝军对战系统

需求分析与系统设计报告



专业班级：自动化 1904

小组成员：刘云笛 学号 U201914665

陈旭桐 学号 U201914650

指导老师：周纯杰、何顶新、彭刚、周凯波、桑农、  
左峥嵘、高常鑫、汪国有、陈忠

上交时间：2020 年 9 月 26 日

# 目录

<b>1 引言</b>	<b>3</b>
1.1 编写目的	3
1.2 背景	3
1.3 参考资料	4
<b>2 可行性分析</b>	<b>4</b>
2.1 目标	4
2.2 要求	5
2.3 用户特点	6
<b>3 需求分析</b>	<b>7</b>
<b>4 总体设计</b>	<b>9</b>
4.1 总体系统架构	9
4.2 接口设计	10
4.3 数据流分析	11
4.4 功能分析	12
<b>5 详细设计</b>	<b>19</b>
5.1 算法设计	19
5.2 数据结构设计	21
5.3 界面设计	24
5.4 界面初步设计	25
5.5 函数设计	30
<b>6 运行环境规定</b>	<b>37</b>
6.1 设备	37
6.2 支持软件	37
6.3 接口	37
6.4 控制	37
<b>7 程序流程图</b>	<b>39</b>
7.1 系统流程图	39
7.2 对战流程图	40
<b>8 组员分工及时间安排</b>	<b>41</b>
8.1 组员分工	41
8.2 时间安排	41

# 1 引言

## 1.1 编写目的

在经过为期近一个月的系统开发，且了解了市面上多款军事策略模拟游戏和军演模拟软件之后，我们编写了本报告，旨在确定软件开发目的和大局方针，以及对我们已经完成的工作进行总结和后续更多开发的总体规划。

本报告中，将会详细提到本系统开发目的和背景、用户的需求分析、系统开发的可行性分析、系统功能设计，还包括本系统的具体信息，运行环境要求、界面设计、数据结构设计、总体程序架构、功能实现、流程结构、以及组员的分工安排和时间安排。本报告将会给出实现功能的函数，大部分源代码也将在本报告中给出。

## 1.2 背景

在如今的国际局势下，军事人才的培养是必须且亟需的，但是很难要求青年人主动系统地学习军事知识、培养军事素养。

经由调查，军事模拟以及策略战斗类的软件现在广受欢迎。开发一款快节奏、高仿真的系统，在吸引用户使用的同时，也能起到一定程度的军事素养提高作用。相较传统的象棋和军棋，这类软件具有精美的画面以及更加复杂的机制，更加贴近现实或具有天马行空的想象力。与棋类的抽象意识相比，显然此类软件更加友善，能直观有效地提升用户对现代战争的了解，提升自己的运筹能力。除此之外，用户能在使用的过程中，体验博弈的快感，还可以在与朋友对战时增进友谊。这也是此类软件吸引用户的主要方式。

现在中国的经济飞速发展，人们的生活水平在不断提高，科技的进步与城市化为人们带来了更快地生活节奏，对于各种软件的需求也在发生改变。人们更加青睐内容简单、趣味性强、节奏快、可以随时离开的软件。哪款软件能更好的利用用户的碎片化时间，哪款软件就能获得成功。除此之外，软件编写技术快速进步，使编写出一款更加具有趣味性的模拟军事对战软件变得日益简

单。图形化界面编程的不断进步，也可以让程序界面可以变得更加精美、吸引人。

我们参考了网络上几款广受好评的软件，结合它们各自的优点以及人们日益变换的需求，设计出了本软件。本软件具有能够吸引人的精美画面，加速战斗进程的机制设定，可以随时保存退出的回合制设计，能够快速上手的简单机制，以及巨大的可能决策空间。希望能够给用户提供一个良好的使用体验。

### 1.3 参考资料

王士元. C 高级实用程序设计. 北京: 清华大学出版社. 1996

周纯杰, 刘正林等. 标准 C 语言程序及应用. 武汉: 华中科技大学出版社. 2008

六边形网格设计方法

<https://www.redblobgames.com/grids/hexagons/>

CSDN 最短路径算法

[https://blog.csdn.net/weixin\\_39956356/article/details/80620667](https://blog.csdn.net/weixin_39956356/article/details/80620667)

## 2 可行性分析

### 2.1 目标

本系统旨在给用户提供一个进行模拟军事对战的平台，提供图形化界面，实时操作兵种进行指挥决策，本系统尽量模拟真实战场的对战场景，地形、兵种克制、天气、排兵布阵，各种因素都会影响到用户的决定和战斗的结果，在使用过程中，用户可以亲身体会到作为一名指挥官，在紧张又激烈的战场上如履薄冰地进行每一次决策，调兵遣将，排兵布阵，争夺并合理分配使用资源，以彻底摧毁敌方大本营为战斗胜利目标。

本系统尽量模拟真实战场的对战情景，给用户真实体验的同时，将现代战争复杂难清的战斗优化为可以轻松掌控的模拟战。通过本系统，一方面可以锻

炼军事思维能力、培养一定的军事策略水平，另一方面可以给现代战争作为一定的参考，同时，对于军事策略爱好者来说，也可以作为一种不错的休闲娱乐软件。

本产品可实现红蓝军模拟对战的基本功能。产品用户可以在进入的主界面进行登录注册，并可以根据设置的密保问题重置密码，设置的游客登录模式可以让用户快速上手体验。本产品对用户的密码作了良好的加密保护，仅能快速检验密码正确性而不能计算真正密码。

进入主菜单后，用户可以选择进入之前的存档继续对战，或者建立新的存档。本产品具有单人模式（人机对抗），以及两位用户同屏对战的双人模式，可以在不同的模式中体验不同的乐趣。对于首次使用本产品的用户，我们还设置了教学模式，方便新用户快速熟悉各个界面和操作，还可以深入了解各项对战机制。

进入对战界面，可以在此界面进行战斗的所有功能。本产品一方面提供了精美的画质，制作了较为美观的动画效果，又通过优良的算法，尽可能减少了对系统配置的要求以及更新屏幕造成的卡顿，以提升用户使用体验。地图采用六边形网格，为更加复杂多变的移动方式提供可能，采用迪杰斯特拉最短路算法实现每个兵种快速准确地移动。地图上设有多种地形，兵种也具有丰富的多样性，还有争夺资源以获取优势的机制，既能让用户快速上手，又能创造出一个复杂多变的战场。回合制的设定让用户可以体验纯粹的策略比拼，不受时间的限制，让用户可以更加仔细地分析场面形势，降低了使用的门槛。还设置了快速存档的功能，可以实现快速保存退出。

## 2.2 要求

### 2.2.1 基本要求

实现红蓝军模拟对战，提升用户的军事素养和运筹能力。

### 2.2.2 具体要求

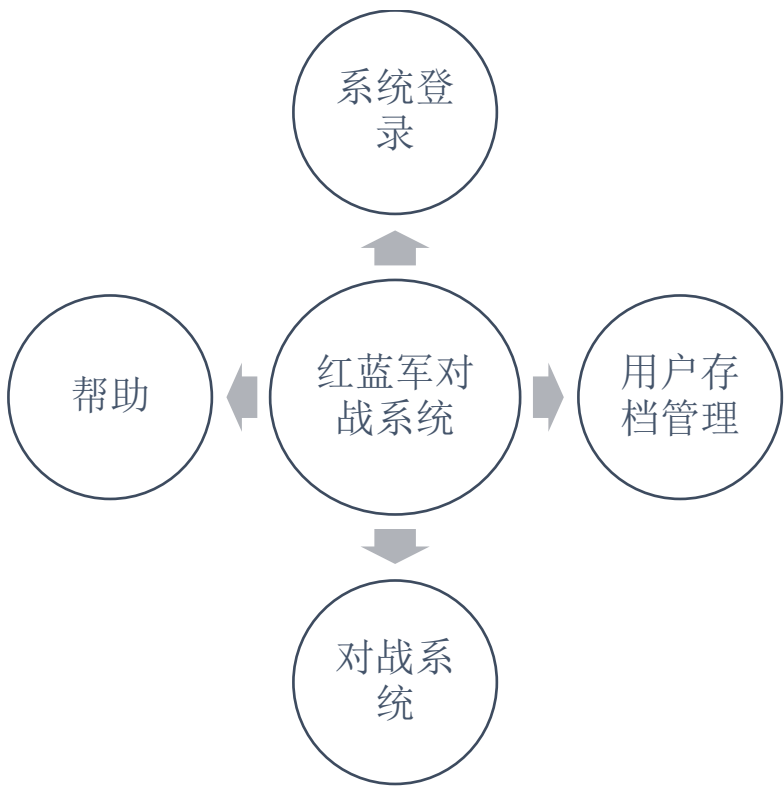
- 1) 使用图形化界面并实现鼠标和键盘操作。

- 2) 实现各单位的移动、攻击等操作
- 3) 屏幕上实时显示战场变化

## 2.3 用户特点

本产品适用于任何对军事、运筹感兴趣的用户，上手简单，只需观看简短的教学即可快速上手，程序中也会有大量的提示引导用户操作。对于维护人员需要一定 C 语言编程基础，但设有报错功能，可以方便维护人员快速发现问题所在。

### 3 需求分析



流程图 1：整体需求分析

本系统的目的是模拟尽可能真实的战场进行红蓝军对战演习，同时，为了降低真实战争的复杂度，使绝大多数用户可以轻松使用，本系统对现代战争的战斗进行了简单化、娱乐化、艺术化，给广大的军事爱好者和策略对战爱好者做出一款快节奏、可以轻松上手、使用时间点灵活、趣味性强的软件，同时，也能给现代战争做出一定程度的参考，用户可以从提升自己的军事素养和决策能力。

为了使本产品可以达成上述目的，主要需要四个板块，即系统登录、用户存档管理、帮助和对战系统。

系统登录模块可以让多个用户在同一台设备上使用，也可以保护用户信息，后续还可以用于服务器登录，进行线上对战。

存档管理模块是为了让用户更加便捷地进行对战，可以随时暂停、随时保存，还可以保留多个存档并进行快捷管理。

帮助模块包含丰富的信息，可以带新用户快速入门、了解对战机制。对于战斗有着更大追求的老用户也可以在这里了解对战相关的所有数值，增加对机制的理解。

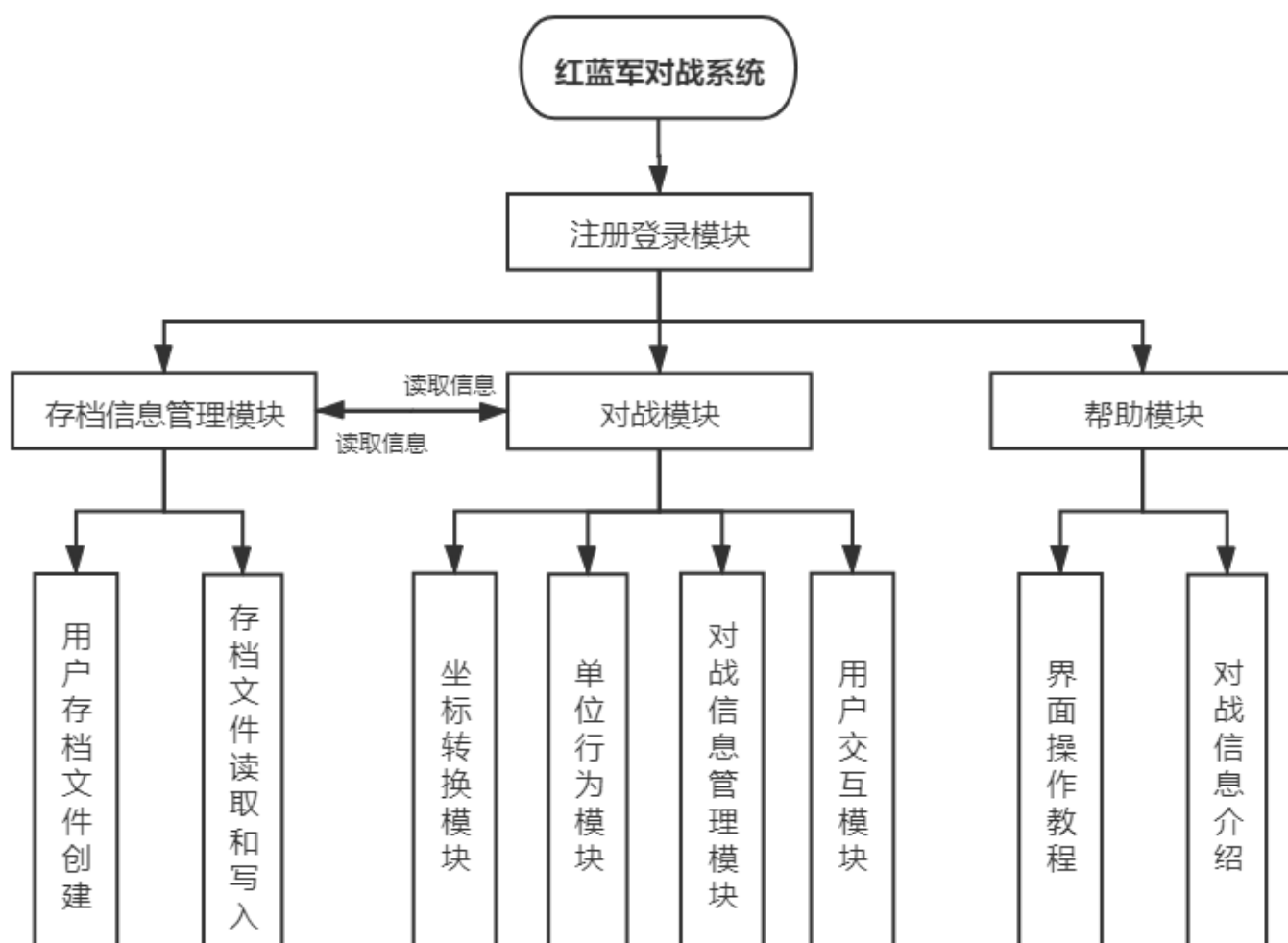
对战系统是本软件的核心部分，工程量最大，代码最复杂，只为了给用户提供一个最好的使用体检。在对战中，用户（或电脑）扮演指挥官，在双方信息互相公开的战场上对弈，以攻破对方大本营为目的。对战分为人机对决和双人对决，用户一个人的时候可以与电脑斗智斗勇，也可以在双人模式下和朋友同屏对战。对战中，需要实现所有单位最基本的移动、攻击、驻扎等操作，并且能够判断移动和攻击能否进行。将来还要实现移动时的最短路径算法。为了增加对战真实性，还增加了地形系统，不同地形有不同的特性（移动难度等）。为加快对战节奏，同时增强策略性，我们还设置了资源系统，并在地图中央设置了高级资源，让战斗从对战开始就一触即发。

具体功能分析见 4.4。



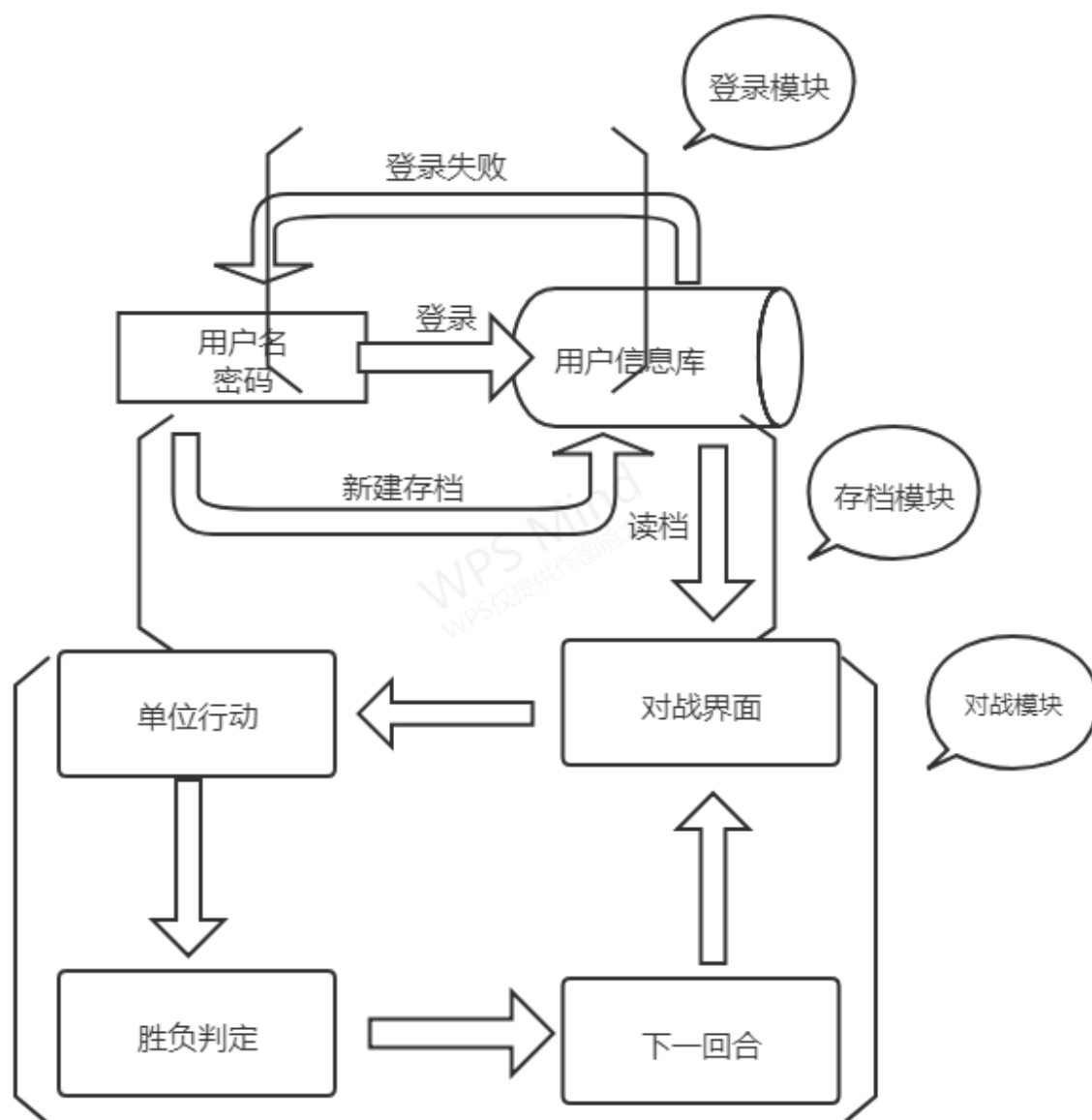
## 4 总体设计

### 4.1 总体系统架构



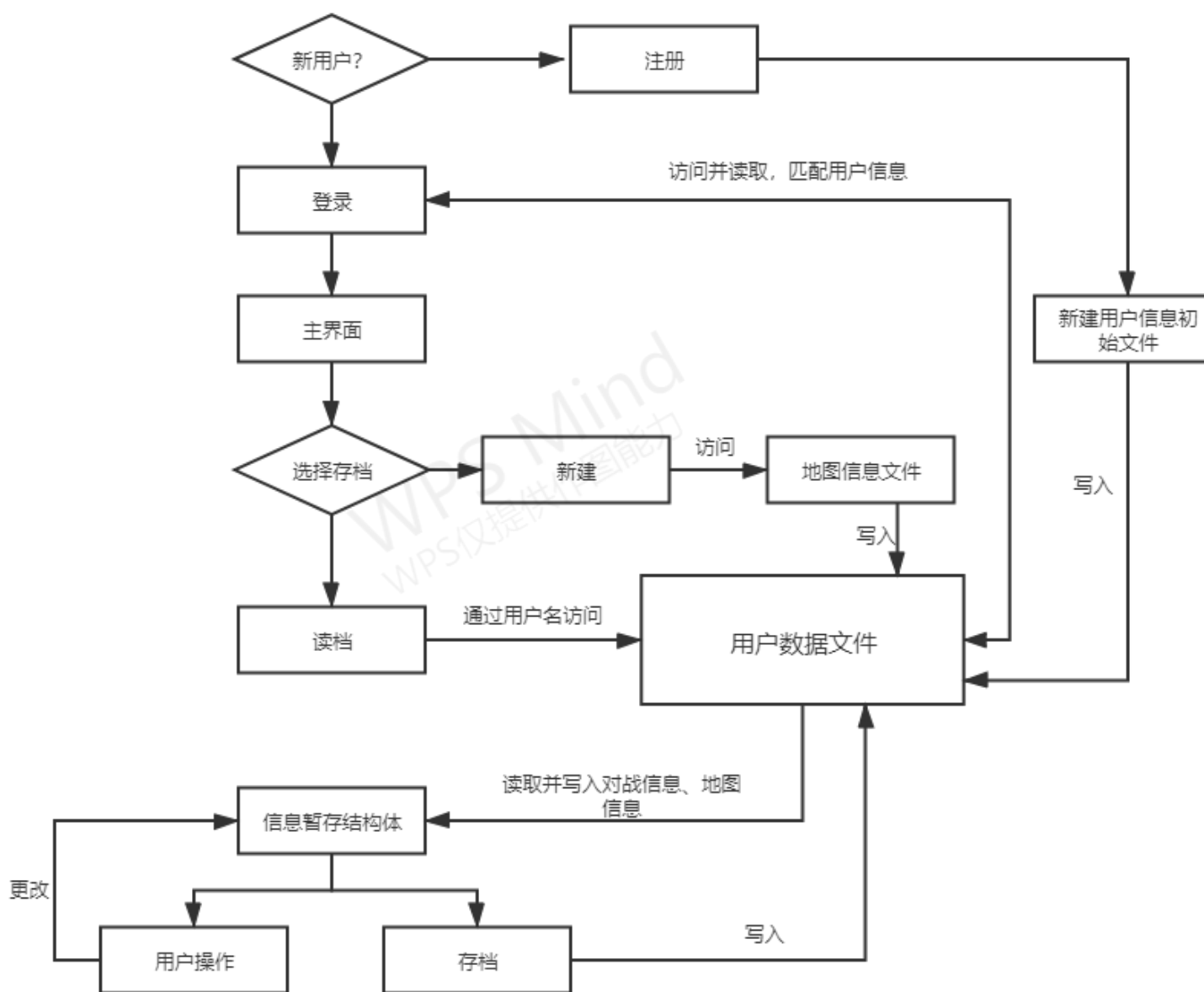
流程图 2：系统总体架构

## 4.2 接口设计



流程图 3：接口设计

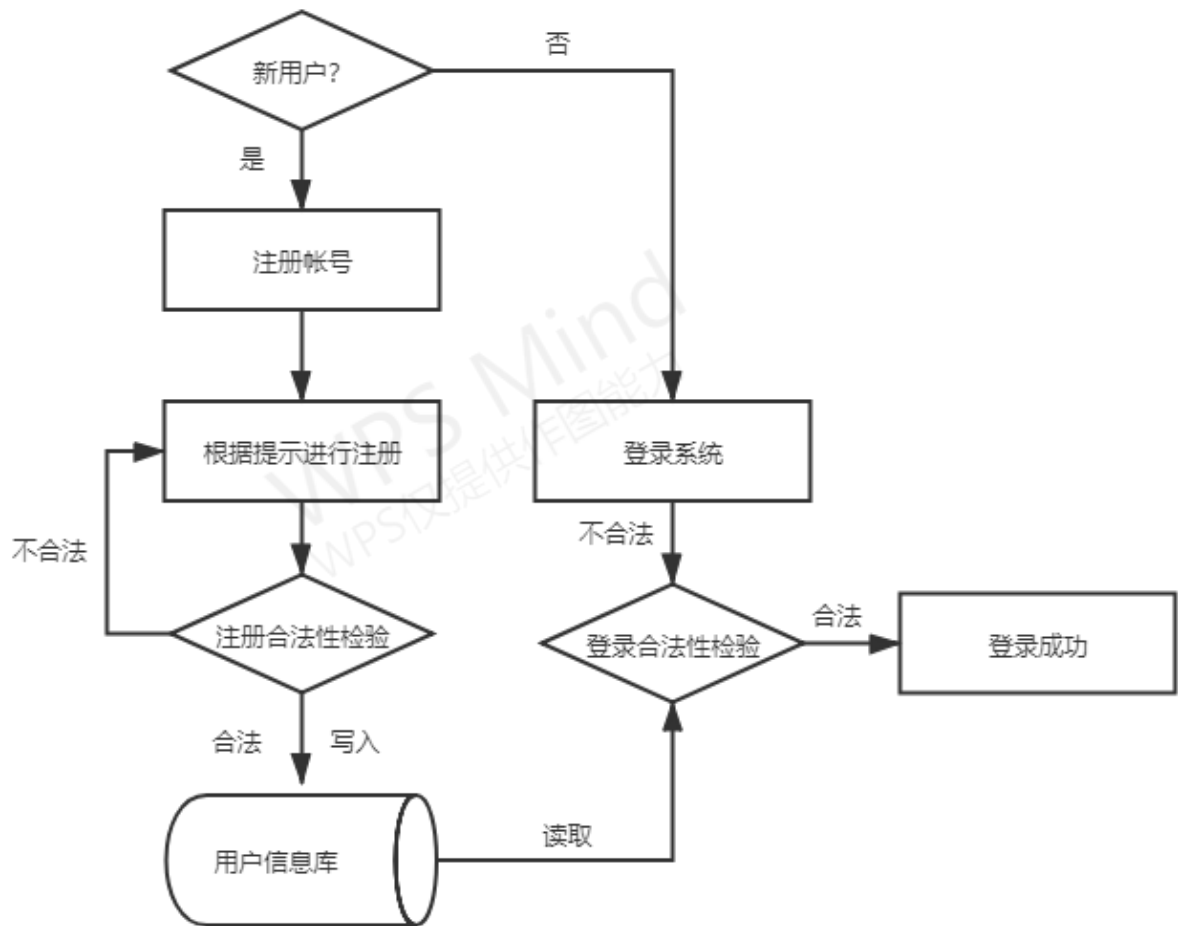
### 4.3 数据流分析



流程图 4：数据流分析

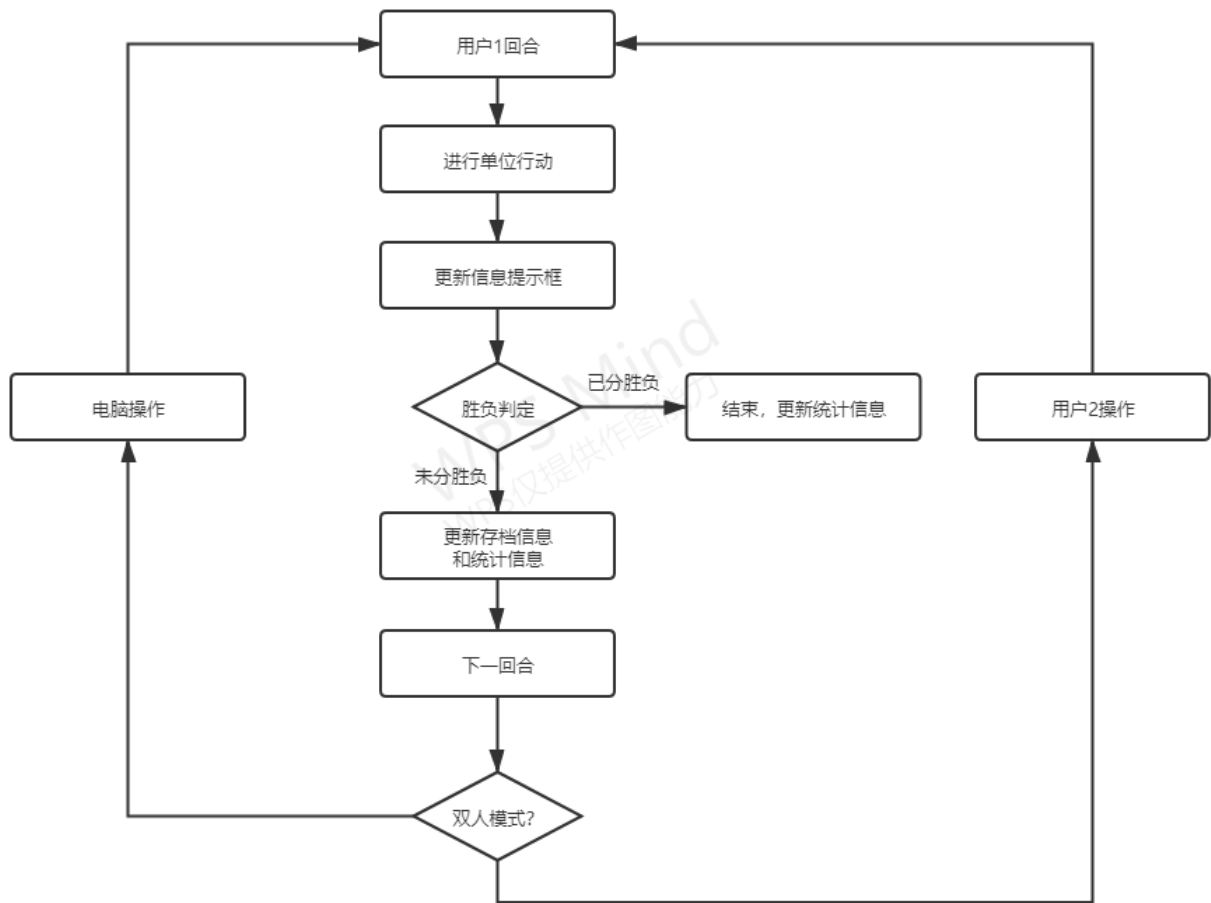
#### 4.4 功能分析

##### 1) 登录注册模块



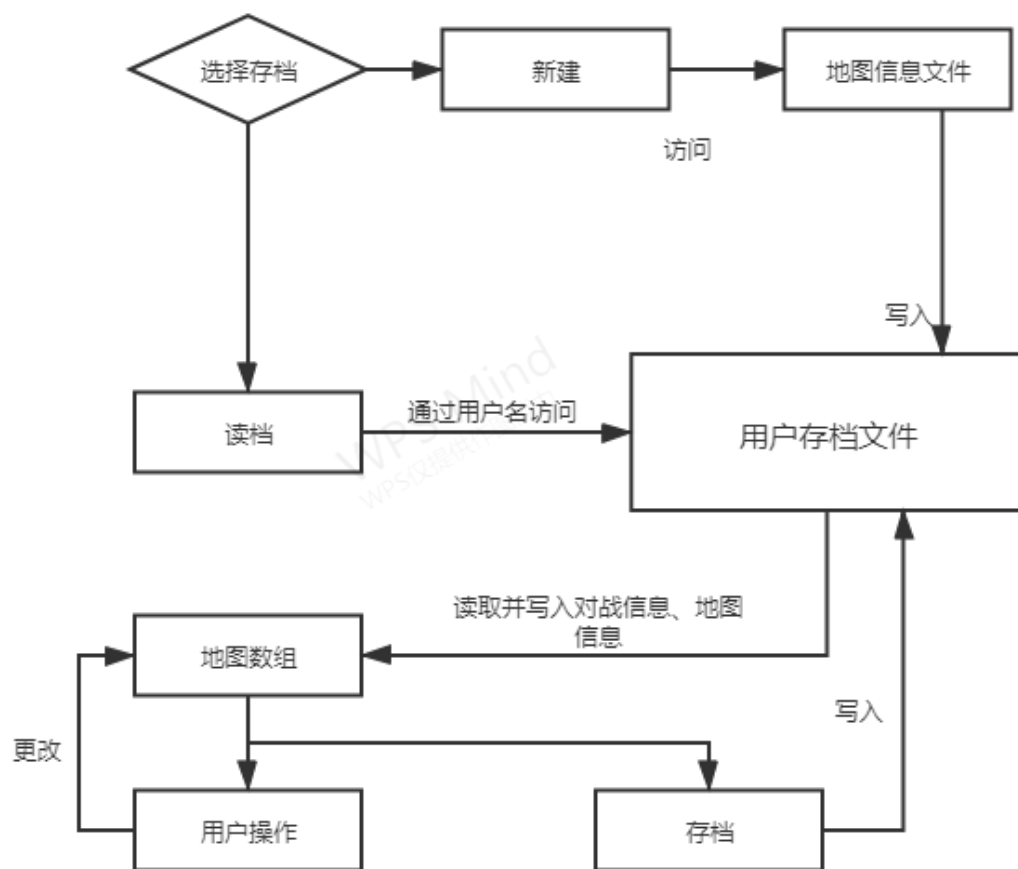
流程图 5：登录注册

## 2) 对战系统



流程图 6：对战系统

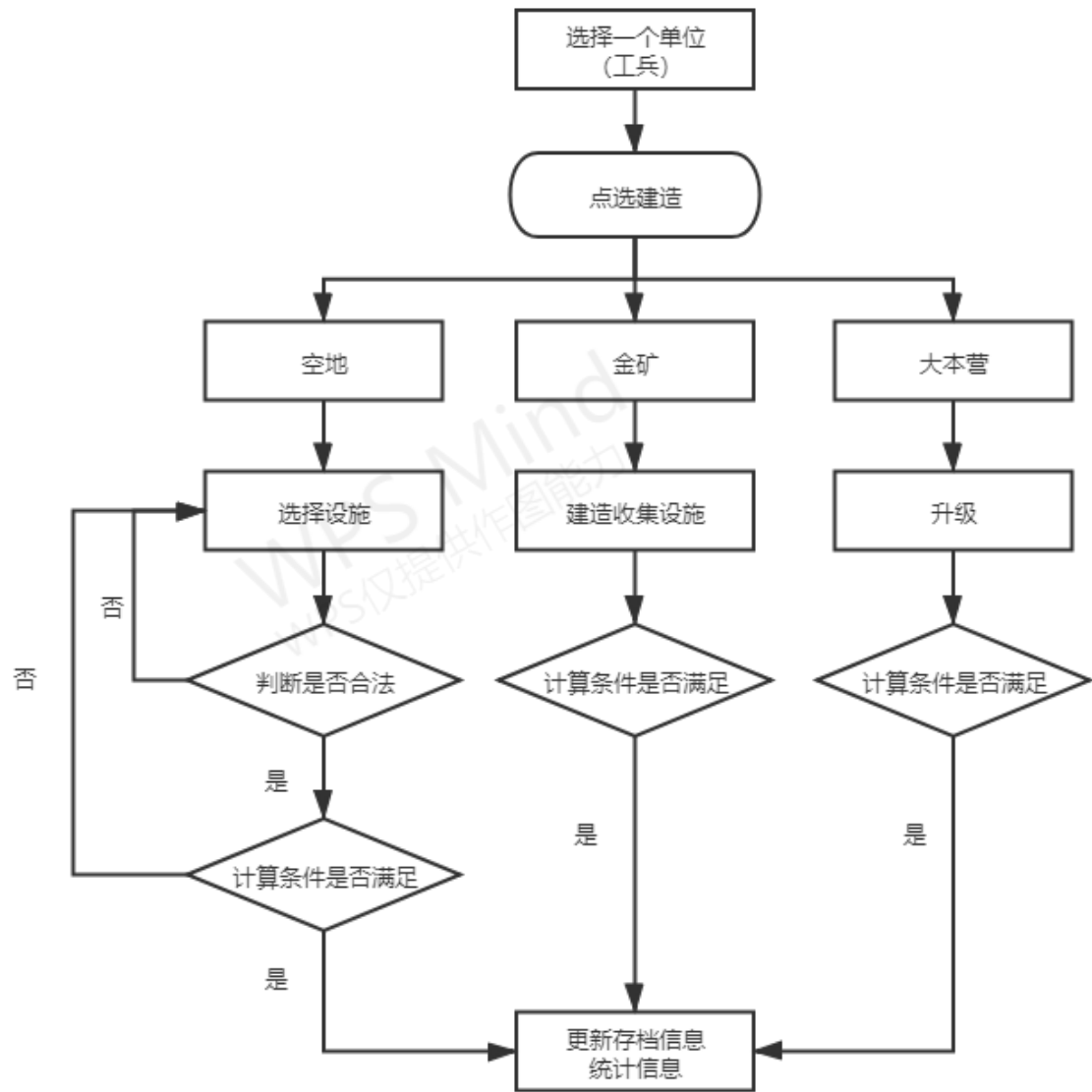
### 3) 用户存档管理



流程图 7：存档读档模块

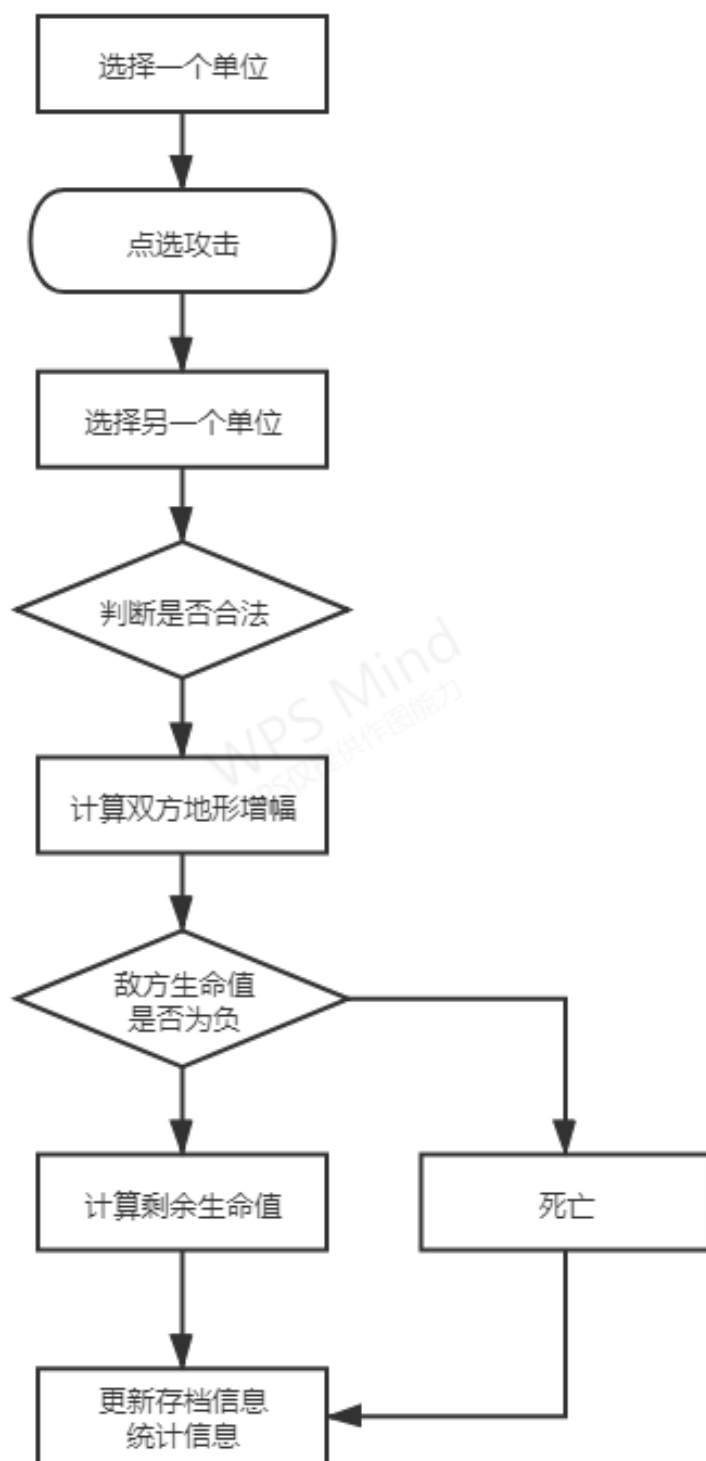
#### 4) 单位行为模块

##### 1. 建造



流程图 8-1：单位行为模块-建造

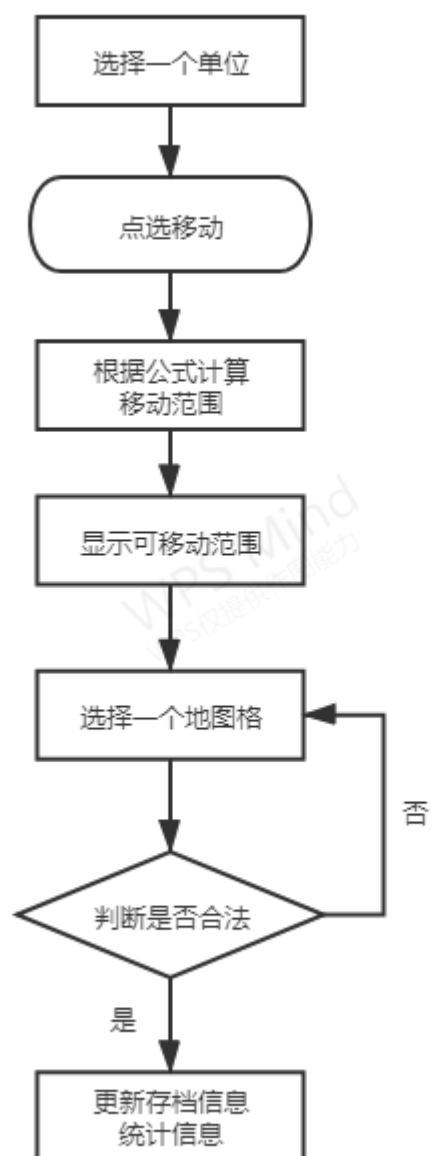
## 2. 攻击



流程图 8-2：单位行为模块-攻击

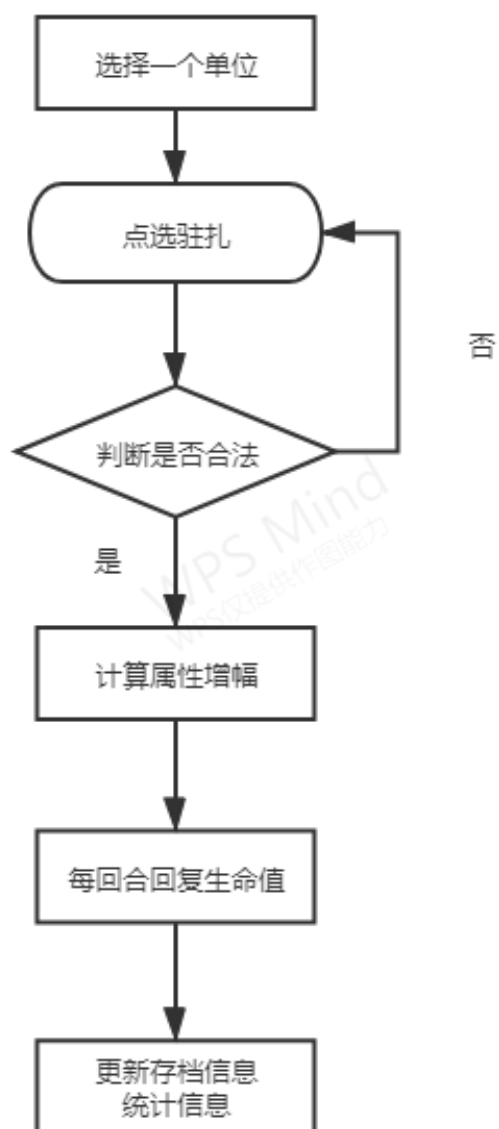


### 3. 移动



流程图 8-3：单位行为模块-移动

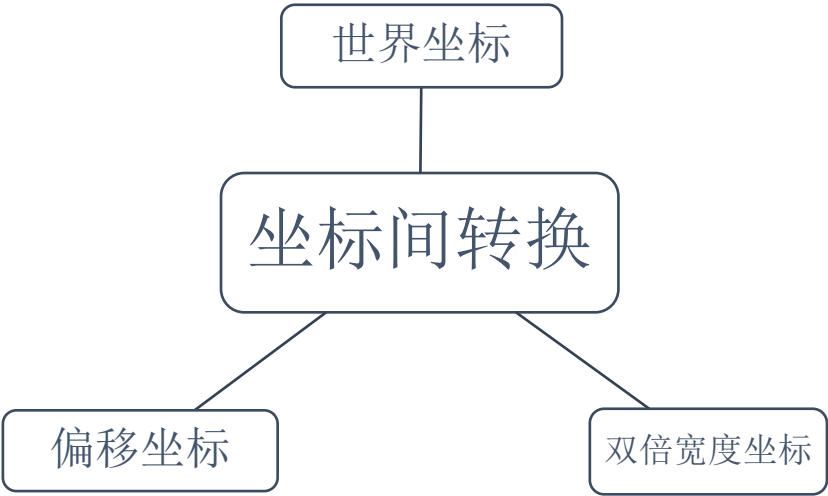
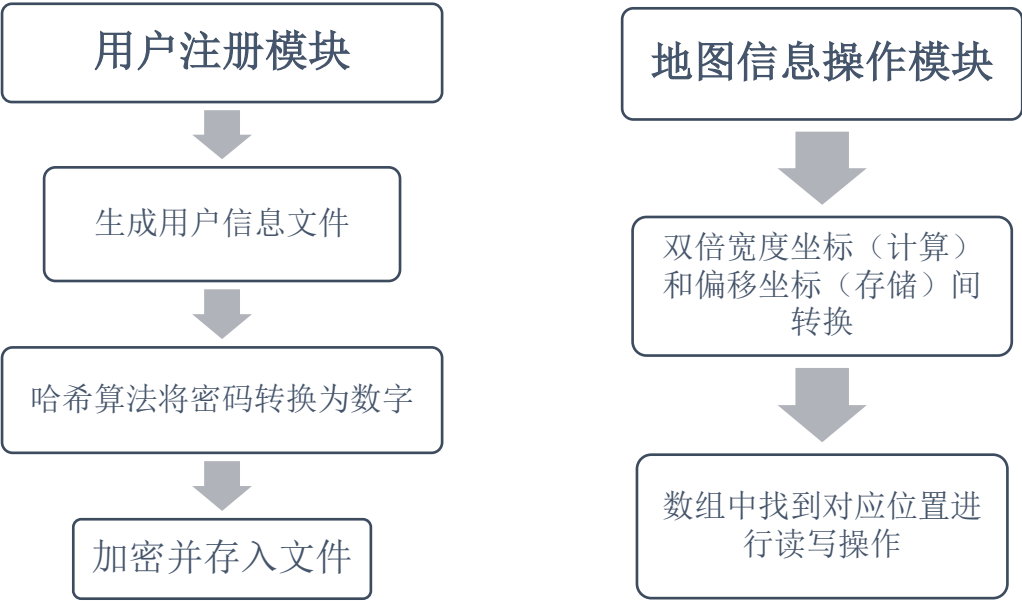
#### 4. 驻扎



流程图 8-4：单位行为模块-驻扎

5 详细设计

5.1 算法设计



坐标说明：

通用坐标为双倍宽度坐标，由于两坐标轴均为直线并且互相垂直，便于算法实现，并且较符合人类看图习惯。

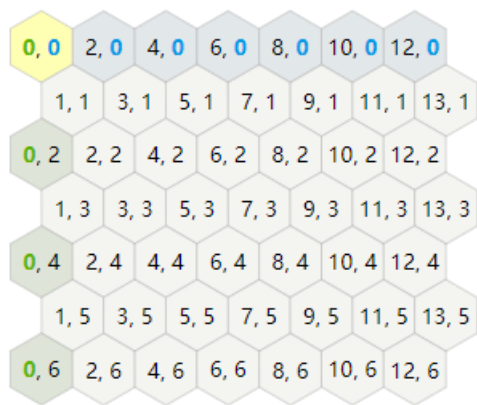


图 1 双倍宽度坐标

偏移坐标为储存时使用的坐标，更符合计算机二维数组的序号排布，可以更大程度地节省空间。并且可以很轻易地与双倍宽度坐标互相转换。

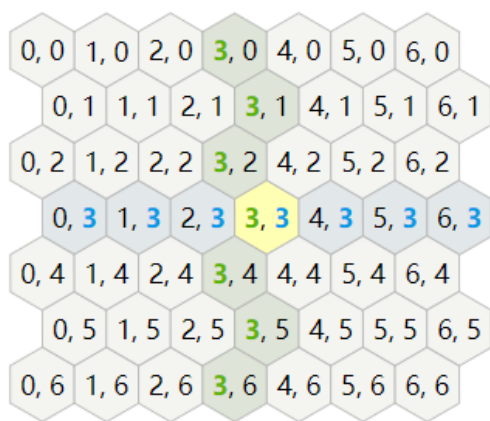
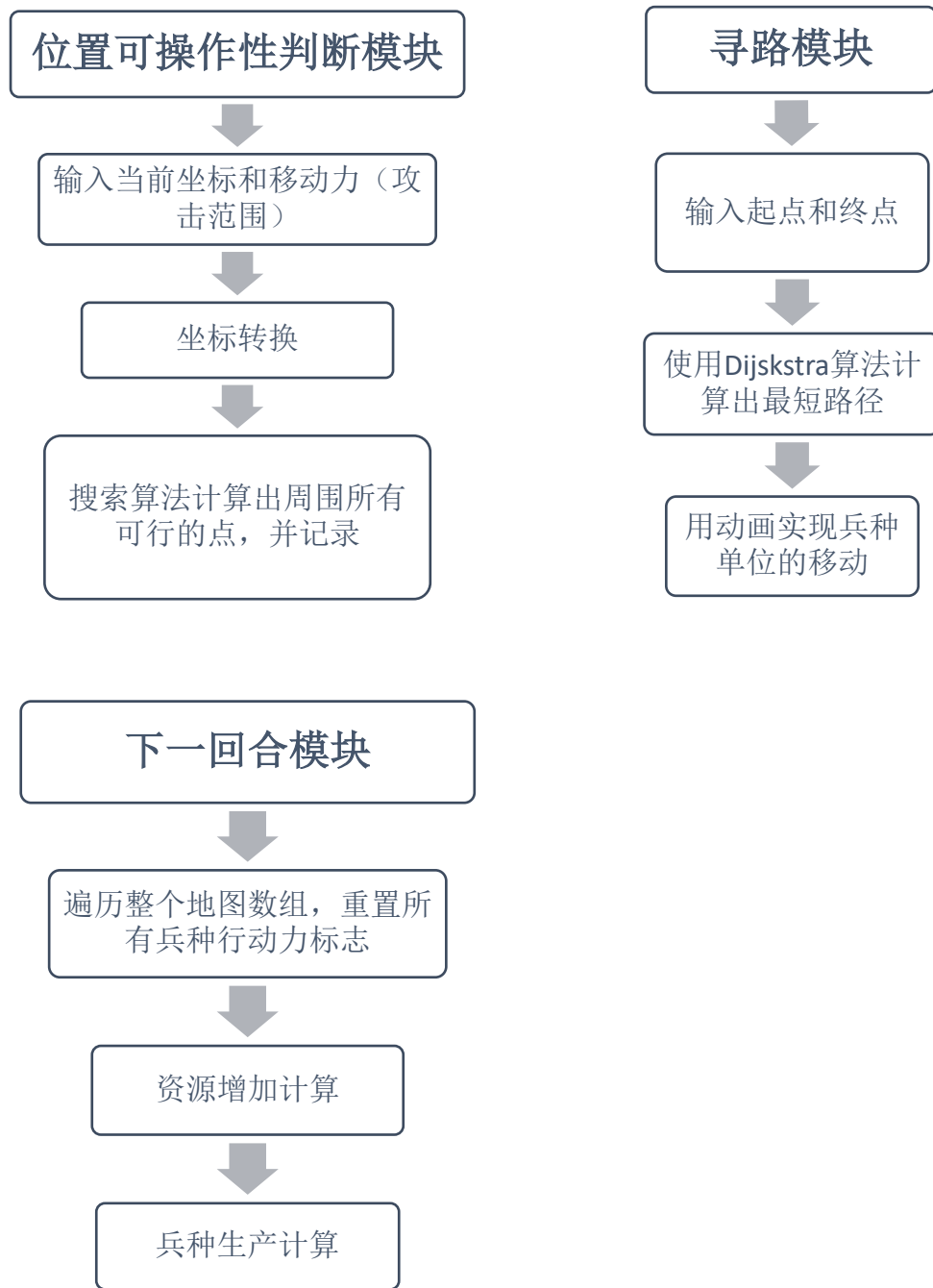


图 2 偏移坐标

世界坐标为图形界面像素点坐标，与其它两种坐标转换难度教导。我们在尽可能保证精度的情况下，使用几何知识编写了可以快速计算每个像素所在格子坐标的函数，并经过不断的调试，才得以完成坐标转换中这最关键的一环。



## 5.2 数据结构设计

### 1) 坐标结构

```
typedef struct _Position
{
    short x;
    short y;
} POS;
```

## 2) 地图格子信息位字段结构

```
typedef struct _cell
{
    unsigned geo : 3; //地形, 序号见_GEO
    unsigned cost : 3; //移动消耗
    unsigned side : 1; //阵营, 蓝0红1
    unsigned kind : 3; //兵种种类, 序号见_KIND, 大本营和资源无需此项
    unsigned health : 5; //兵种、大本营血量, 高级资源占领回合数
    unsigned flag : 1; //算法实现时标记用
}CELL;
```

(使用位字段结构, 节省空间)

## 3) 对战信息结构

```
typedef struct _Battleinfo
{
    unsigned round; //回合数, 此处为双倍
    unsigned r_source; //红方资源
    unsigned b_source; //蓝方资源
}Battleinfo;
```

## 4) 存档文件设计

```

/*****
save 文件设计:                                     位数表: 自开头偏移量 (字节)
  文件名: username
  1: 文件中存档个数, short*1                      共1

  以下重复多个存档块的设计 (注: short均为unsigned)
  -----
  存档序号short*1                                0
  日期 年unsigned*1, 月日unsigned*1, 时分unsigned*1 1
  回合数 unsigned*1                              7
  蓝方、红方 资源 unsigned*2                     9

  以下13行*13个unsigned大小 (位字段CELL) 为地图      13
  位字段结构设计见struct.h
  -----
  地图共2*SIZE*SIZE

  Author: 刘云笛
*****/
```

### 5.3 界面设计

(1) 用户进入系统后，首先进入主界面；

(2) 首界面可进行鼠标操作，当鼠标移至以下图标后，会标亮显示：

1) 登录，点击进入登录界面后，用户可根据提示登录系统，在 3 秒后进入主菜单，也可进行找回密码操作（尚未开发）；

2) 注册，点击进入注册界面后，用户可根据提示进行帐号注册，然后回到登录界面；

3) 关于，点击查看版权信息、产品介绍及其开发人员；

4) 退出，点击退出系统；

(3) 进入主菜单界面后，可进行鼠标操作，当鼠标移至以下图标后，会标亮显示且图标变为更容易理解的信息：

1) 红蓝对决：双人对战，点击后进入对战界面；

2) 决战智械：单人模式，点击后进入对战界面与电脑对战；

3) 重回往昔：载入存档；

4) 作战指导：进入教程界面；

5) 注销账号：回到登录界面；

(4) 教程界面（尚未开发）：操作说明和新手提示将在此显示；

(5) 对战界面（尚未开发），可进行鼠标操作；

1) 单位操作：对单位进行操作，详见功能模块。

2) 下一回合：结束当前回合，进入下一回合

3) 选项：进入选项界面，可进行存档、投降、退出等操作



## 5.4 界面初步设计

注：

所有界面按钮均实现鼠标悬浮其上变为手势型，并实现标亮。

目前所有界面均处于初步设计阶段，后续会重新进行美化

为了提升用户使用体验，两张背景大图，以及一张地图使用贴图，由于要求，其余均使用绘图函数完成。

### 1) 初始界面



图 3：初始界面

2) 登录界面



图 4：登录界面

### 3) 注册界面

DOSBox 0.72, Cpu Cycles: max, Frameskip 0, Program: BC

# WARCHES 通行证注册

**用户名**  八位以内，大写字母和数字

**密码**  密码区分大小写哦

**再次输入**  两次输入保持一致

**注册**



图 5：注册界面

4) 对战系统主界面按钮设计

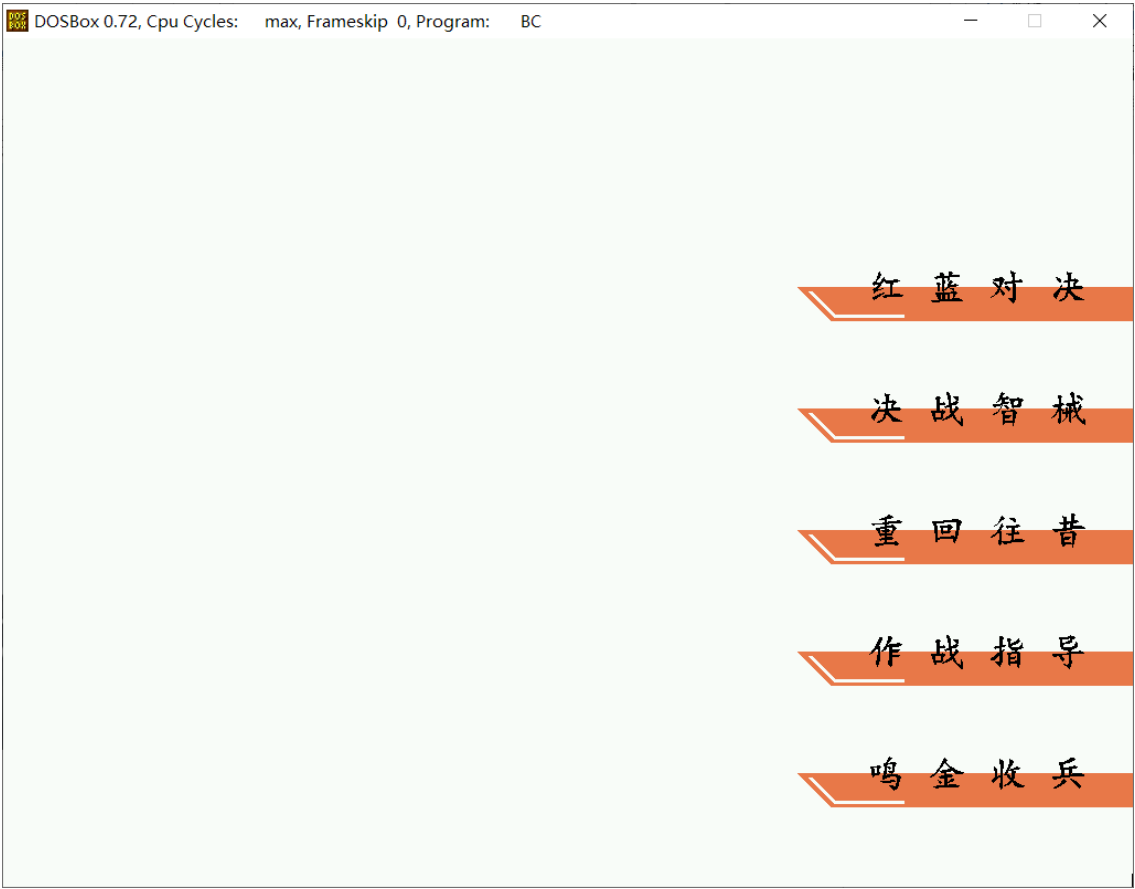


图 6：主菜单

所有按钮被标亮后：

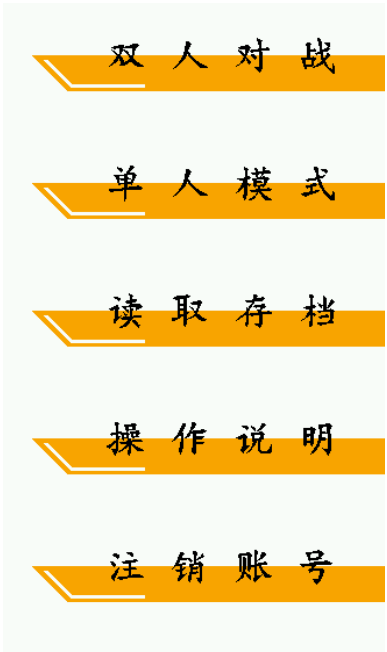


图 7：主菜单按钮效果

## 5) 对战地图



图 8：对战地图

## 5.5 函数设计

### 1) 屏幕文字输入模块

1. `void kbinput(int x1, int y1, int x2, int y2, char *s, int mode);`

描述：键盘输入并显示在屏幕上（已完成，共 64 行）

参数：前四个坐标为输入框坐标，s 为接收内容数组，mode 取 1 输出原文、取 0 输出 ‘\*’

### 2) 对战模块

1. `int battle(char* user, short save_num);`

描述：战斗界面主函数（部分完成）

参数：user 指向用户名，save\_num 为存档序号

返回值：页面值

2. `void battle_draw(void);`

描述：战斗界面绘制（部分完成）

3. `void Battle_init(FILE* fp, Battleinfo* info, CELL map[][13]);`

描述：战斗信息初始化，读取文件中战斗信息，存入内存中数组和结构。  
（已完成，16 行）

参数：fp 为指向用户存档的指针，info 为接收对战信息的结构体，map 为地图储存数组

4. `void save_battle(FILE* fp, Battleinfo* batinfo, CELL map[][13]);`

描述：将战斗信息保存至文件，并更新存档日期（已完成，30 行）

参数：fp 为指向用户存档的指针，batinfo 为对战信息结构体，map 为地图储存数组

### 3) 存档管理模块

1. `int savefile_creat(char *user);`

描述：创建一个新的用户存档并初始化地图和对战信息，并记录创建时间  
（向已有的用户存档文件中写入）（已完成，25 行）

参数：user 为用户名

返回值：当存档满（6 个）时，返回 1，否则返回 0

2. void seek\_savinfo(FILE\* fp, short n, int x, int y);

描述：将用户存档指针 fp 定位到需要的位置，若 x\*y=0 则定位到该存档首，否则定位到 (x,y) 对应地图数据位置（已完成，13 行）

参数：fp 为用户存档指针，n 为存档号，x,y 为坐标（注意：使用双倍宽度坐标）

3. void savefile\_init(FILE\* fp, short n);

描述：覆盖性初始化某一个存档

参数：fp 为指向用户存档文件的指针，n 为存档序号

#### 4) 登录注册模块

1. int register\_(void);

描述：注册界面主函数；

参数：无；

返回值：页面值；

2. void drawregi(void);

描述：注册界面绘制；

参数：无；

3. void user\_creat(char \*username, char \*password);

描述：用户信息创建，将用户输入的信息写入文件，以用户名为 txt 文件作为文件名，更加方便的查找用户信息；

参数：username 指向输入的用户名，password 指向输入的密码

4. unsigned long long password\_classified(char \*p, int radix);



描述：利用哈希算法对用户密码进行加密，使文件中只显示加密过后的数字无法直接得到用户密码；

参数：p 指向密码，密码转换 radix 进制数；

返回值：加密后的密码；

5. int regis\_check(char \*un, char \*pw, char \*pwr);

描述：登录信息检测，根据返回值确定能否成功登录；

参数：un 指向输入的用户名， pw 指向输入的密码， pwr 指向再次输入的密码；

返回值：检测的结果；

6. int login(char \*user);

描述：登录界面主函数

参数：user 指向用户名

返回值：页面值

7. void drawlogin(void);

描述：登录界面绘制；

参数：无；

8. int login\_check(char \*username, char \*password);

描述：登录信息检测，根据返回值确定能否成功登录；

参数：username 指向输入的用户名， password 指向输入的密码；

返回值：检测的结果；

## 5) 坐标转换模块

备注：此为本系统的基础功能模块，且实现较为复杂

1. POS xy2cell(int x, int y);



描述：世界坐标转换为格子坐标，利用中心线网格划定区域，以几何知识确定；

参数：x、y 为世界坐标，通常来讲会将鼠标点击的坐标传进函数；

返回值：POS 结构体，内含两个整型变量，分别为格子的横纵坐标

2. POS center\_xy(int row, int col);

描述：将格子坐标转换为格子中心点的世界坐标，方便单位操作

参数：row 为格子横坐标、col 为格子纵坐标

返回值：POS 结构体，内含两个整型变量，分别为中心点的横纵坐标

## 6) 自定义绘图函数模块

1. void shadow(int x1, int y1, int x2, int y2, int color1, int color2);

描述：绘制带阴影的矩形

参数：矩形对角线坐标，color1 为高亮颜色，color2 为矩形颜色

2. void shadow\_l(int x1, int y1, int x2, int y2, int color);

描述：绘制带大阴影的矩形

参数：矩形对角线坐标，color 为矩形颜色

3. void frame(int x1, int y1, int x2, int y2, int color);

描述：绘制带边框的文本框

参数：矩形对角线坐标，color 为矩形颜色

4. void file\_draw(int x1, int y1, int x2, int y2);

描述：绘制一个文档的图形

参数：矩形对角线坐标

5. int Outtextxx(int x1, int y, int x2, char \*s, int flag, int color);

描述：根据所需要输出汉字的长度来输出汉字，和原函数相比，不需要计算字间距

参数：将文字视作水平矩形，x1 和 y 为矩形左上角坐标，x2 为矩形右侧横坐标，s 指向输出的内容，flag 为字号，color 为字体颜色

返回值：将计算出的字间距返回，通常来说不会用到，仅在设计界面时辅助计算使用

6. void GetBackground(int left, int top, int right, int bottom, short \*buffer);

描述：截取屏幕某一部分的图像，并保存（用于实现动画）

参数：矩形对角线坐标，buffer 指向存储的数组

7. void PutBackground(int left, int top, int right, int bottom, short \*buffer);

描述：搭配上函数实现背景还原

参数：矩形对角线坐标，buffer 指向存储的数组

8. unsigned int Lightcolor(unsigned int colorbf, float drgb);

描述：将某种颜色增加亮度的函数

参数：colorbf 为原色，grab 为变白比例

返回值：增加亮度后的颜色值

9. void Lightpixel(int x, int y, float drgb);

描述：增加屏幕上一点的亮度

参数：点坐标和变白比例

10. void Lightline(int x1, int y1, int x2, int y2, float drgb);

描述：增加一条线的量度

参数：线的起止坐标和变白比例

11. void Lightbar(int x1, int y1, int x2, int y2, float drgb);

描述：增加一个矩形的亮度

参数：矩形的对角线坐标和变白比例

12. void Bar64k\_radial\_re(int x1, int y1, int x2, int y2, unsigned int color, int fill\_time);

描述：反向放射状绘出矩形，使绘图过程更加美观

参数：矩形对角线坐标，color 为矩形颜色，fill\_time 为延迟时间

13. void Bar64k\_radial(int x1, int y1, int x2, int y2, unsigned int color, int fill\_time);

描述：放射状绘出矩形

参数：矩形对角线坐标，color 为矩形颜色，fill\_time 为延迟时间

14. void Button(int y1, char\* s, int color, int color2);

描述：绘制一个好看的按钮，因为必须贴屏幕右侧边界使用，故无横坐标

参数：按钮纵坐标，s 指向按钮上输出的文本，color 和 color2 为按钮配色

15. void Line45(int x1, int y1, int x2, int y2, unsigned int color);

描述：绘制 45 度斜线

参数：斜线起止坐标，color 为斜线颜色

16. void Icon\_draw(POS pos, int side);

描述：绘制一个图标的边框

参数：通过 pos 结构体传进入此图标的中心点坐标，通常为六边形格子中心点坐标，side 为阵营

17. void Icon\_builder(POS pos, int side);

描述：绘制一个工兵图标

参数：通过 pos 结构体传进入此图标的中心点坐标，通常为六边形格子中心点坐标，side 为阵营

```
18. void diamond(int x1, int y1, int x2, int y2, int x3, int y3,
    int color);
```

描述：绘制水平的菱形，实际上为四个直角三角形，此函数非常不稳定，需要重新编写

参数：三个坐标分别为，菱形左端点，菱形中心点，菱形上端点，color 为菱形颜色

## 7) 其他函数

```
1. void show_error(char* msg, int mode);
```

描述：报错函数，在屏幕上显示错误框和错误信息

参数：msg 为错误信息，mode 1 为巨大错误关闭程序，0 为小错误、仅返回

```
2. int Sharp_button(int y0, char* s, char* s_change, int color,
    int color2);
```

描述：主界面按钮函数，用于实现按钮标亮、按钮点亮功能，只需要在循环中调用该函数，接受返回值即可实现按钮功能。

参数：y0 为按钮上边界 y 坐标（该按钮仅在界面右端），s 为按钮上原来的文字，s\_change 为标亮后现实的文字，color 为按钮颜色，color2 为花纹颜色。

```
3. void auto_battle(void);
```

描述：人机对战，实现电脑自动操作函数。

## 6 运行环境规定

### 6.1 设备

本系统是对“红蓝军对抗系统”的基本模拟，因此，对此系统的运行环境将是在计算机上面进行，即在电脑上模拟实现，系统运行的设备需求如下：

- (1) 一台 586 以上的微机及兼容
- (2) 内存 16MB 及其以上
- (3) 彩色显示器一台

### 6.2 支持软件

根据“红蓝军对抗系统”设计环境要求，即在 DOS 环境下，用 C 语言编译实现，使用 TC 或 BC 开发软件，因此，此模拟系统可在绝大多数现形计算机系统上运行，包括最常见的 Windows XP 操作系统以及 Windows 7 32 位机上运行，但由于兼容性问题，此“红蓝军对抗系统”在含 windows 7 64 位操作系统的计算机上运行可能出现意想不到的问题或根本不能运行，因而，不推荐在包含 64 位操作系统的计算机上测试本系统。

### 6.3 接口

- (1) 用户接口:本系统采用一般性架构，界面尽可能符合真实软件。
- (2) 硬件接口:无特殊要求。
- (3) 软件接口:无特殊要求。
- (4) 通信接口:无特殊要求。

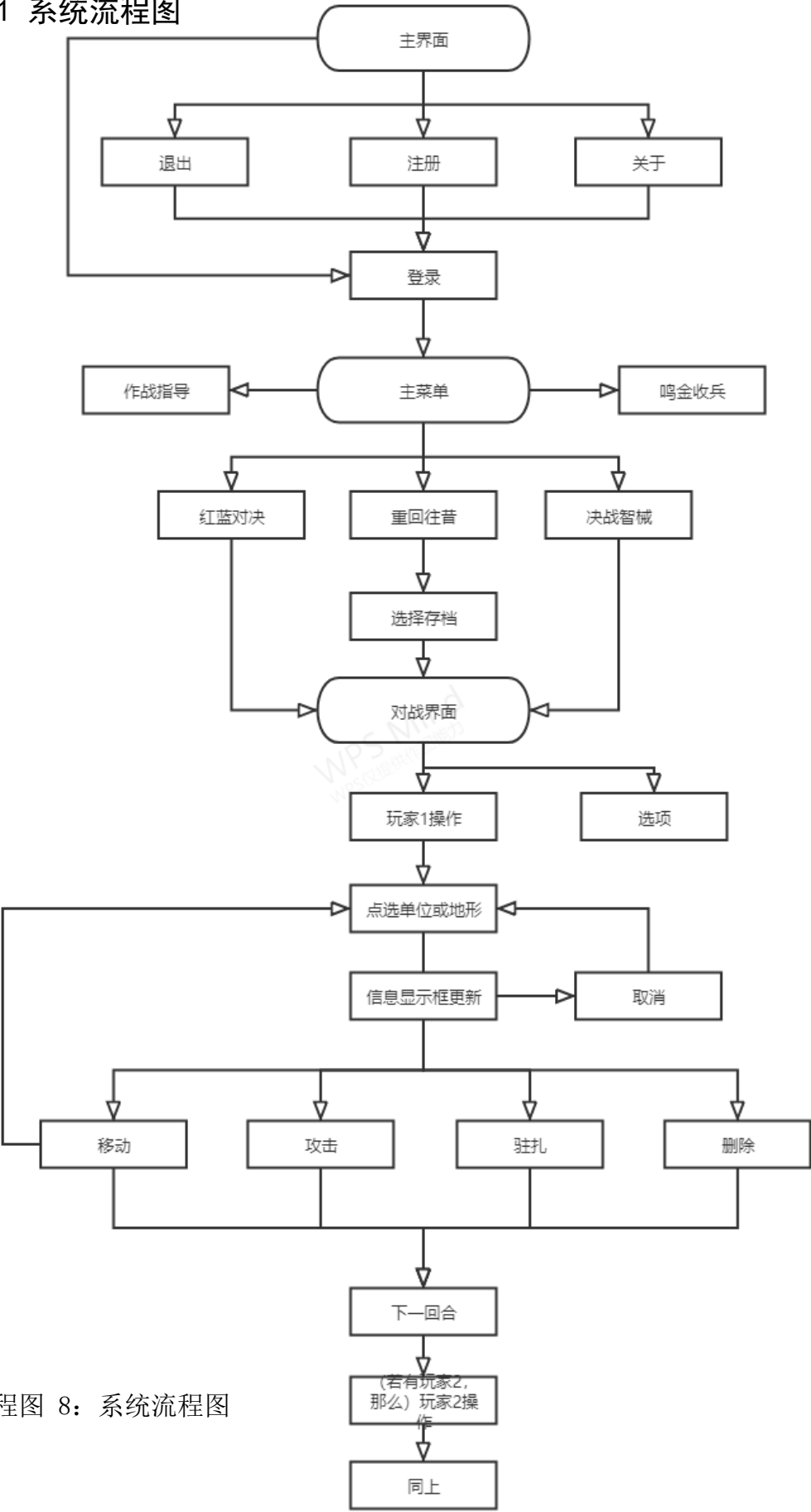
### 6.4 控制

本系统通过鼠标直接进行操控，人机交互界面设计简洁易懂，只需要简单的教程就可以操作。用户将鼠标移至需要操作的功能区点击，进行相应的操作，操作进行完后，屏幕将会显示用户操作的结果。控制信号来源于用户鼠标的位置，以及用户使用鼠标进行的操作（比如左键、右键）。当用户进入登录

注册界面可根据页面提示进行键盘操作，可以使用回车键完成输入、使用退格键删除内容

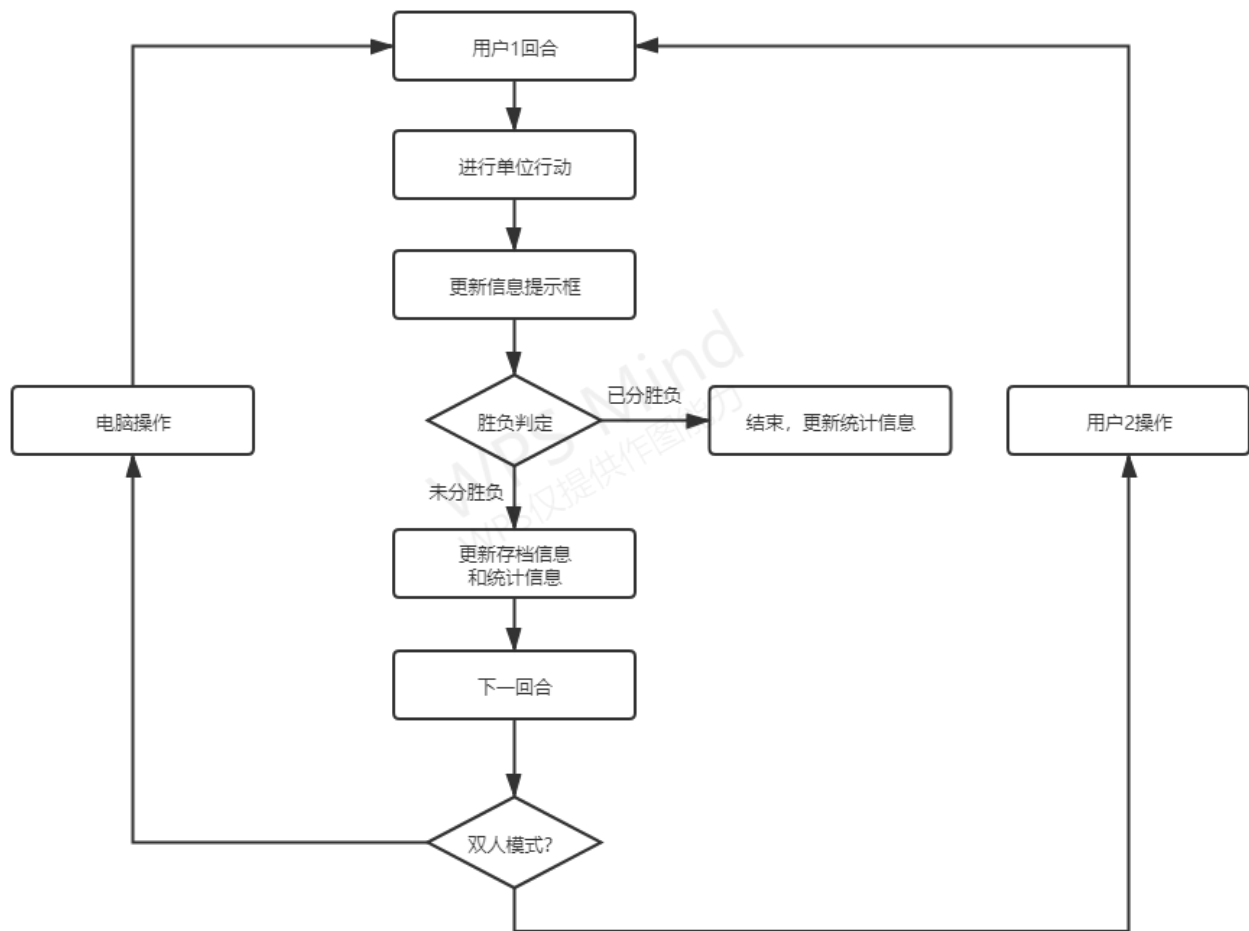
7 程序流程图

7.1 系统流程图



流程图 8：系统流程图

## 7.2 对战流程图



流程图 9：对战流程图



## 8 组员分工及时间安排

### 8.1 组员分工

姓名	分工	备注
刘云笛	大部分算法模块；数据结构设计；文件操作模块；动画实现基础；键盘、鼠标模块；功能性绘图函数；提示框和部分图标绘制。	大部分代码均由一人编写，另一人提出建议并做修改。
陈旭桐	所有界面模块； 登录注册（包括加密）模块；绘图模块；坐标转换模块；对战平衡性设计与兵种属性设计；兵种行为模块；	大部分代码均由一人编写，另一人提出建议并做修改。

表 1：合作分工表

### 8.2 时间安排

时间	任务	备注
第 1 周	系统需求分析与设计报告	
第 2 周	设计数据结构，界面及菜单设计	
第 3 周	设计数据结构以及初始化函数，欢迎退出动画	
第 4 周	算法设计，数据传输，鼠标设计	
第 5 周	输出函数设计，主函数设计	
第 6 周	编码实现	
第 7 周	程序合成并试运行、调试	
第 8 周	程序调试并写最终报告	

表 2：时间安排表