

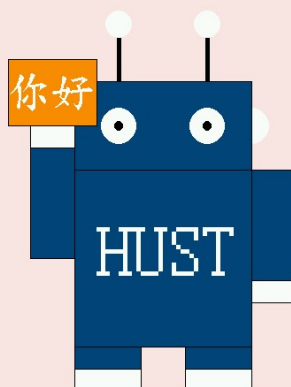
家居机器人模拟系统 结题验收报告

——C 语言课程作业

欢迎使用家居机器人模拟系统

按任意键继续

本程序由华中科技大学自动化学院
一九零五班孟繁鹏，黄瀚文共同开发



课题名称：家居机器人模拟系统

专业班级：自动化 1905

小组成员：孟繁鹏 U201914689 黄瀚文 U201914680

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 10 月 31 日

目录	
1 前言	4
1.1 项目背景	4
1.2 项目内容	5
1.3 项目目标	5
2 项目概述	5
2.1 开发环境	5
2.2.1 硬件接口	5
2.2.1 软件接口	5
2.2 编写规范	6
2.2.1 工程目录规范	6
2.2.2 项目管理规范	6
2.2.3 命名规范	6
2.2.4 注释规范	6
2.2.5 版权规范	6
2.3 开源情况	7
3 需求分析	7
3.1 软件对象	7
3.2 软件维护要求	7
3.3 软件推广与使用	8
3.4 功能设计	8
3.4.1 登录注册功能	8
3.4.2 核心功能	8
3.4.3 控制功能	9
3.4.4 电量与充电功能	9
3.4.5 环境与清洁功能	10
3.4.6 舒适度功能	10
3.4.7 人机交互功能	10
3.5 模块设计	11
3.5.1 注册登陆逻辑	11
3.5.2 核心功能逻辑	15
4 界面设计	19
4.1 登录注册界面	19
4.1.1 加载界面	19
4.1.2 欢迎界面	20
4.1.3 登陆界面	20
4.1.4 注册界面	22
4.1.5 找回密码界面	24
4.2 核心程序界面	26
4.2.1 核心框架	26
4.2.2 清洁界面	26
4.2.3 电量界面	27
4.2.4 舒适度界面	28
4.2.5 控制界面	29

4.2.6 人机交互界面	30
5 数据结构与算法	31
5.1 数据结构	31
5.2 高级算法	32
6 源代码	36
6.1 系统和主函数	36
6.1.1 svgc.c *	36
6.1.2 mouse.c *	36
6.1.3 hzk.c *	36
6.1.4 pinyin.c	36
6.1.5 main.c	36
6.1.6 mysys.c	37
6.2 登陆注册函数	42
6.2.1 interf.c	42
6.2.2 login.c	47
6.2.3 register.c	56
6.3 核心程序函数	62
6.3.1 drawrob.c	62
6.3.2 mainwin.c	66
6.3.3 board.c	73
6.3.4 drawmap.c	83
6.3.5 electr.c	93
6.3.6 rubbish.c	96
6.3.7 plan.c	99
6.3.8 move.c	102
6.3.9 comfort.c	111
6.3.10 chat.c	113
7 时间与工作安排	125
7.1 时间安排	125
7.2 分工情况	125
8 总结与感想	126
8.1 程序亮点	126
8.2 程序不足	126
8.3 组员感想	126
9 参考资料	128

1 前言

1.1 项目背景

随着人工智能技术的迅速发展以及人民生活水平的提高，人工智能产业化以及智能家居在最近几年的得到了蓬勃发展。正如未来学家沃尔夫·伦森所说：“人类在经过农耕、工业、电气化等时代后，将进入关注梦想、精神和生活情趣的新社会。”智能家居就是这样一个关注生活情趣和生活质量的话题，智能家居以住宅为平台，利用综合布线技术、网络通信协议技术、安全防范技术、自动控制技术、视音频技术将家居有关的设施，集成构建高效的住宅设施与家庭日程事务的管理系统，提升家居安全性、便利性、舒适性、艺术性，并实现环保节能的居住环境。智能家居能够给人们生活带来很多的便捷跟体验，也引来很多企业争相在智能化市场分一杯羹。

而在整个智能家居设计中，出镜率最高的当属智能机器人。日本索尼在放弃机器人产业 12 年后，在 2016 年 6 月宣布重新涉足机器人业务。索尼社长平井一夫曾表示，“包括工厂自动化等业用途在内，将广泛开展机器人业务”。索尼日前发布消 “包括工厂自动化等业用途在内，将广泛开展机器人业务”。被中国美的集团收购德国机器人巨头库卡也十分关注机器人市场。国库卡集团总部研究中心负责人莱纳·比绍夫博士在接受采访时表明，库卡与美的合作重点领域是家用服务和医疗机器人。比绍夫博士说，在社会老龄化趋势下，库卡希望为工厂的老人、家庭提供协助机器人，实现如移动帮助老人行走的功能。在 2020 年的工博会上，总部设在瑞士的老牌机器人巨头企业 ABB 机器人将以未来医院、未来汽车、未来工厂为主题，分设三大展区，全方位展现机器人在各个场景、领域的自动化柔性解决方案。并且，工博会首日，ABB 机器人还将举行新品 IRB 1300 机器人的全球首发揭幕仪式。根据预测，机器人的应用量和普及度将会持续增长。随着扫地机的全球销量实现新突破，家庭服务机器人的全球销量将高达 122 亿美元，超过 2015 年市场规模的 5 倍。目前中国市场非常庞大，但是新兴产品的渗透率与日本、美国等国家相比还非常低，目前的销售额也很小。因此中国家用机器人未来市场空间巨大，一旦相关产品取得新的突破，将获得巨大的利润和市场认可。

机器人产业之所以成为智能化与自动化产业的核心，是因为机器人是智能化与自动化实现从规划算法到对环境产生影响，进行实际操作最直接的硬件载体，所以很多智能家居开发的核心也放在了智能机器人的设计制造，功能分析，控制系统，算法优化等方面的工作。目前已经有发达地区的家庭中开始出现智能机器人的身影，我们也有理由相信，随着技术的发展和人们生活水平的提高，智能家居机器人一定会像电视，电脑一样，普及到每一个普通的家庭中，为人们所用。

开发者正是本着抓住时代脉搏的想法，选择了这个题目，开发一套家居机器人模拟系统。由于环境和编译器版本的限制，本产品能够实现的功能有限，将重心放在了模拟和仿真。旨在为未来智能家居开发描绘一个可能发生的图景，进行多角度的模拟和仿真，并提出一些有创造性的新思路和新构想。

1.2 项目内容

本项目以家庭服务机器人为出发点，结合智能手的广泛使用实现远程操控服务机器人的功能。基于此，我们希望通过该产品将多项功能集成化，主要需要实现的功能包括：

1. 机器人自身功能：如：移动，充电等；
2. 控制功能：如：远程遥控，巡逻等；
3. 清洁功能：如：捡垃圾，倒垃圾，除草等；
4. 舒适度显示及调节功能：如：各房间空调温度，空气湿度，pm2.5 等；
5. 房间操作功能，如：开门关门，开窗关窗等；
6. 交互功能，主要实现人机交互，如：聊天等。

1.3 项目目标

该家具机器人模拟系统主要目标是实现以下功能：

1. 在欢迎界面点击之后，系统用户可以在最初界面实现登陆、注册、找回密码等操作。
2. 用户可以实现基本的家居机器人的功能模拟。并且可以观察其动态的效果模拟。
3. 使用高级算法进行距离计算和路径规划，并尝试进行优化以降低时间和空间成本。
4. 尝试开发此前并未提出过的家具机器人可能的功能，并进行模拟仿真。

2 项目概述

2.1 开发环境

2.2.1 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：SVGA 接口

系统运行内存：要求 32MB 以上

2.2.1 软件接口

操作系统：DOS WINDOWS 9X/ME/2000/XP/WINDOWS 7

开发软件工具：Borland C 3.0

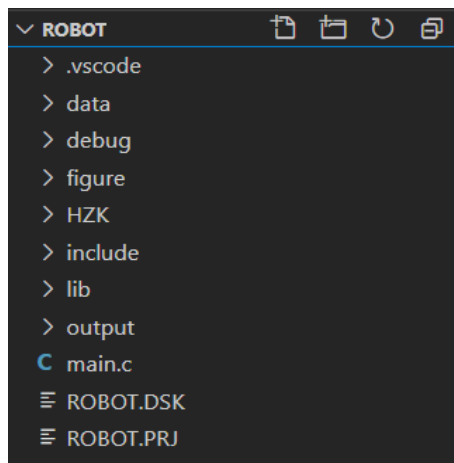
文本编辑工具：Visual Studio Code, Typora

项目管理工具：git 终端+github

数据库：txt 文件

2.2 编写规范

2.2.1 工程目录规范



工程目录如文件夹所示，头文件储存在 include 文件夹，头文件的源文件储存在 lib 文件夹，编译输出文件储存在 output 文件夹，用户数据储存在 data 文件夹，图片源文件储存在 figure 文件夹，调试过程中使用的输出文件储存在 data 文件夹。

2.2.2 项目管理规范

本项目使用 git 管理，开发时需要注意以下规范：

- 每次开始工作前一定要提前 pull 更新本地库，以跟进工作进度。
- 每次 commit 必须标记清楚：日期，开发者，本次更新完成的工作，如下图：



2.2.3 命名规范

- 文件命名应用小写英文体现其具体功能。
- 变量命名，函数命名应用英文表达其确定含义。
- 文件 headers.h 中包括了所有需要调用的头文件，必要的颜色和变量的宏定义，每个文件直接调用该文件即可。

2.2.4 注释规范

- 函数功能应在函数原型后注明。
- 复杂算法和流程应给出相应的注释。

2.2.5 版权规范

- 每个文件的开头必须注明：文件名，创建时间，最后更新时间，开发者。如直接使用前人代码活部分改编前人代码应注明：“(Partly) Adapted by pervious version”。如下图：

```

/*
    main.c

    Initial version: 2020.7.12
    Lastest update: 2020.7.19
    Author: mfp & hhws
*/

```

• 添加新函数后应更新 log.md 文件，并标明开发者以及函数功能。未完成调试需注明。
如下图：

login.c 开始界面逻辑文件 ---mfp&hhw编写

```

mfp:

void start_func(); 开始页面的逻辑

void input(int x1,int x2,int y1,int y2, char *s, int max_len, int fp, int bgcolor); 输入函数

hhw:

void login_func(); 登陆界面逻辑 *未完成, 需要调试

void user_register(); 注册界面逻辑 *未完成, 需要调试

void user_findback(); 找回界面逻辑 *未完成, 需要调试

void exit_pro(); 推出窗口逻辑 *未完成, 需要调试

```

2.3 开源情况

为了方便同学们进行交流讨论，借鉴学习。开发者选择将此代码开源，包括 SVGA 框架及其扩展，鼠标库，汉字库等文件均可以借用，并注明引用来源。本项目在 github 上开源，开源地址：https://github.com/mfp0610/HUATAIA_Cproject-robot/

3 需求分析

3.1 软件对象

1. 家用机器人产品研发人员：主要可用于家用机器人产品的仿真和模拟，以调试产品。
2. 欲购买家用机器人的潜在消费者：主要可用于选购家用机器人时对性能进行相关的仿真与模拟，以供了解产品的性能，功能，进行选购。

3.2 软件维护要求

1. 具有良好的 C 语言（C89）编程技术，并能熟练使用 BC 环境。

2. 充分了解**面向过程**的程序设计思想，并熟悉底层编程工作。
3. 可以在使用低级环境，无法使用 STL 等高级集成 API 的情况下完成数据结构的构建与调试和算法的设计与调试。
4. 了解本软件的相关模块，并能够调试相关参数。

3.3 软件推广与使用

由于环境和语言的限制，本软件可以实现的功能有限，且推广性差，在如今常用的大多数操作系统（win10 / linux / macOS）中均无法使用。预计无法进行大面积推广，也无法实现在生产生活中的应用，相关功能和内容仅供参考。

3.4 功能设计

3.4.1 登录注册功能

任务分配：孟繁鹏负责前端工作（interface.c），黄瀚文负责后端工作（login.c）。

用户进入登录页面后，如果曾注册过，可以直接输入账号和密码，并点击“登录”按钮进行；如果用户名不存在或与密码不匹配，将提示登陆失败。

如果没有进行过注册，则可以点击“注册账户”按钮进行。按钮后，将跳转至注册页面用户输入账号，密码，手机号。手机号可以接收验证码，并作为密保用以找回密码。如果账号，密码，手机号格式均符合要求且该账号不曾被注册过，用户完成前三项输入后点击“发送”，可以收到验证码，输入验证码正确后，则注册成功。完成注册后，用户点击“返回”按钮即可返回登录界面。

如果忘记了已注册账号的密码，则可以点击“找回密码”按钮进行找回。点击“找回密码”按钮后，将跳转至身份验证界面，用户输入账号和电话号码进行身份验证，以上信息如输入错误，则会提示错误；如输入正确，将显示用户密码。

用户在登录界面输入账号密码，登录成功后则跳转至主界面。

用户在登陆界面点击退出程序，即可直接退出软件。

3.4.2 核心功能

1. 地图显示

任务分配：黄瀚文负责（paintmp.c）。

用二维数组储存地图，具体情况如下：

0 是地板，1 是机器人，2 是墙壁，3 是垃圾桶，4 30 是椅子，5 是桌子，6 是一般高度家具，7 8 12 13 14 是床，9 是竖门，10 是左门，11 是右门 21 是充电桩，22 是垃圾，23 24 25 26 是开门状态，27 是床头柜，28 是电视，29 是书柜

根据二维数组绘制地图，然后每次操作后调用 paintmp()函数重新绘制地图。

当前工作情况：（已完成，hhw 负责）

2. 状态显示功能

任务分配：孟繁鹏负责（mainwin.c）。

总共需要显示的状态：电量，垃圾桶内的垃圾数，地面垃圾数，温度，空气湿度，空气质量。

1. 电量是以图像动态显示；
2. 温度在波动区间内态刷新；
3. 地面垃圾数以数字显示；
4. 温度以摄氏度形式显示；
5. 空气湿度以百分比形式显示；
6. 污染程度以百分比显示。

按照传入变量 st 值的不同来显示不同的信息。

3. 移动控制与显示

任务分配：孟繁鹏负责（mainwin.c）。

移动时利用 move()函数进行移动，传入当前坐标并进行相应的加减操作，完成移动后重新绘制地图和机器人，绘制时按照不同的移动方向绘制朝向不同的机器人，具体情况大致如下：

- 向右移动，机器人右侧面朝向用户；
- 向左移动，机器人左侧面朝向用户；
- 向前移动，机器人正面朝向用户；
- 向后移动，机器人背面朝向用户。

机器人共有三种移动方式：

- 输入 WASD 键移动机器人。
- 点击控制面板上的上下左右摇杆移动机器人
- 点击地图任意位置移动机器人（定向巡逻）

3.4.3 控制功能

任务分配：孟繁鹏负责（move.c）。

1. 定向巡逻

点击图片任意位置后，如果机器人能够走到该位置，则移动机器人，计划采用 A star 算法（手动实现 STL 优先队列）规划路径。如不能则输出一行提示。

2. 自由巡逻

提前规划自由巡逻路线，遍历到所有的屋子，尽可能覆盖所有区域，如果屋子门没开则自动跳到下一个房间。

3. 开关门功能

地图上和显示面板上都会显示当前门开关情况，用户可以选择开或闭。接收到用户的该命令后，门会自动打开关闭。

3.4.4 电量与充电功能

任务分配：黄瀚文负责（electr.c）。

1. 电量耗损

电量随时间流逝减少，同时会在移动的时候减少。

2. 电量增加

1. 手动充电：点击充电选项，会自动前往充电桩，并进行充电，
2. 自动充电：机器人电量少 10%时会触发自动充电选项，触发后会自动前往充电桩，并进行充电，然后返回原位。

3.4.5 环境与清洁功能

任务分配：黄瀚文负责（rubbish.c）。

当用户点击控制面板上的“清洁”命令时，可以进入互动界面。然后点击不同的按钮选择功能。

2. 生成垃圾功能

用户可以点击控制面板上的“生成垃圾”功能按钮，当用户发起该命令时，房间中随机生成一个垃圾，房间中同时最多生成三个垃圾。

3. 拾垃圾功能

用户可以点击控制面板上的“拾垃圾”功能按钮，向机器人发起该命令时，机器人将移动向当前房间地面上出现的垃圾，并将垃圾清除。首先清理最后一个生成的垃圾，每一次完成清理后寻找下一个垃圾清理。机器人完成全部垃圾清理后会将垃圾汇集到的小桶里。

前置功能：定向巡逻

3.4.6 舒适度功能

任务分配：孟繁鹏负责（comfort.c）。

当用户点击控制面板上的“环境”命令时，可以进入互动界面。然后点击不同的按钮选择功能。

1. 室内温度调节功能

点击控制面板“温度”右侧的开关进行温度调节。初始状态室内温度是 20 摄氏度，室外温度是 26 摄氏度，红色代表空调开启，黑色代表空调，开启空调后房间温度将向设定温度缓慢靠近。当空调关闭时，温度将逐渐向室外温度靠拢。

2. 空气湿度调节功能

点击控制面板“湿度调节”进行湿度调节。初始状态空气湿度是 50%，点击调节湿度后，湿度将迅速下降，当下降到 20%以下时自动停止下降。

3. 空气质量调节功能

点击控制面板“湿度调节”进行湿度调节。初始状态污染指数是 50%，点击调节湿度后，污染指数将迅速下降，当下降到 20%以下时自动停止下降。

3.4.7 人机交互功能

任务分配：孟繁鹏负责（chat.c）。

当用户点击控制面板上的“互动”命令时，可以进入互动界面。然后点击不同的按钮选择功能，结束聊天后清空上次聊天的内容，以保证内存充足。

1. 聊天功能

进入聊天程序后，用户可以通过控制面板上的输入框与机器人进行对话，机器人在收到用户留言后会给出回应。本功能采用关键词匹配方法，我们首先找到对应的关键词，我们认为出现次数最多的词语为关键词，使用 kmp 算法统计词语出现次数，根据关键词匹配答句。

前置功能：汉字输入法。

3. 猜谜语功能

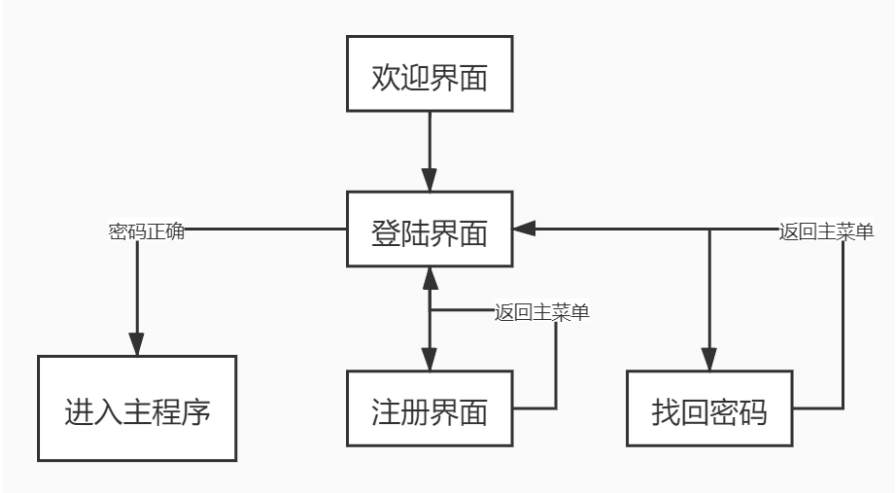
用户进入该功能后系统会给出相应的谜语，用户输入谜底，输入错误将得到提示，输入正确将自动触发下一条谜语。谜语数据库储存在 txt 文件夹中。在该功能中用户随时可以选择返回上一级界面。

前置功能：汉字输入法。

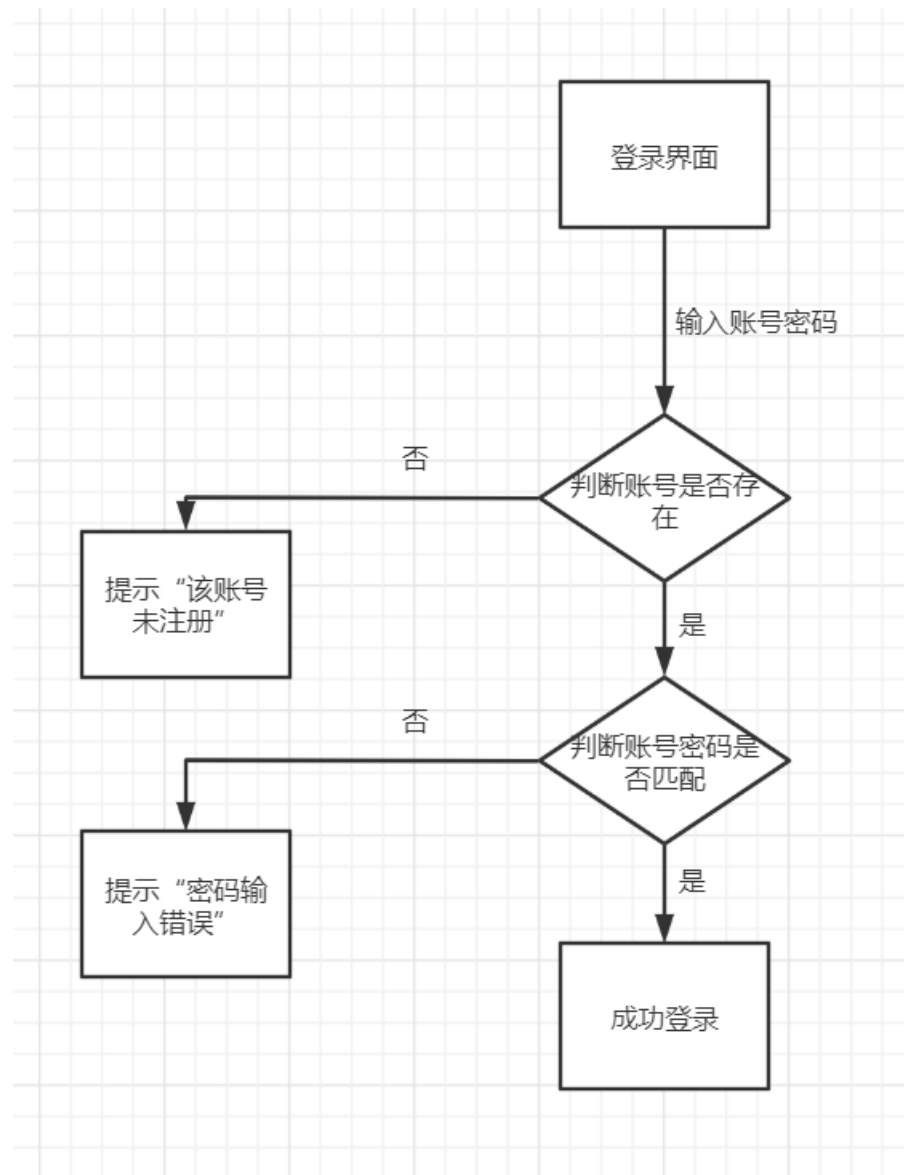
3.5 模块设计

3.5.1 注册登陆逻辑

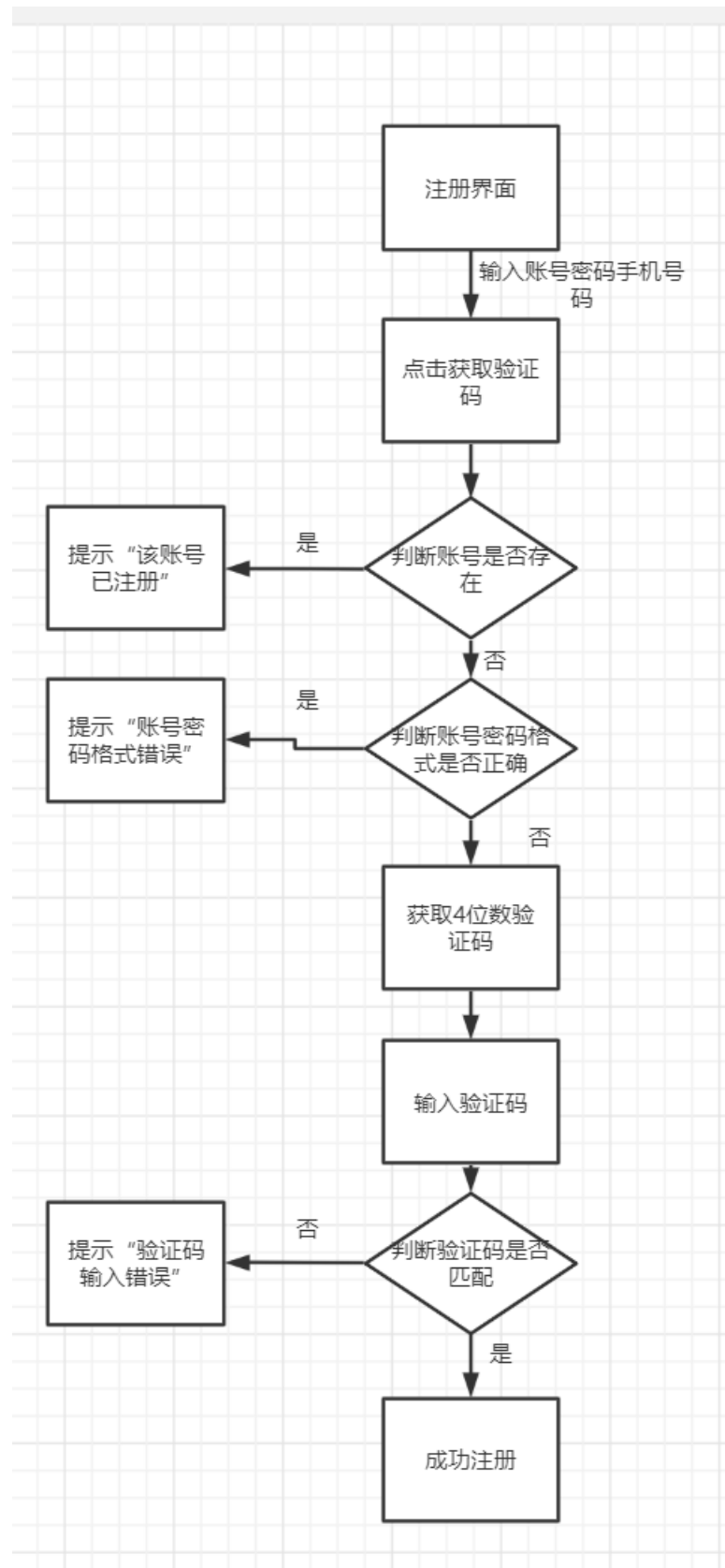
• 登录注册界面由孟繁鹏和黄韩文共同开发，采取前端-后端分工模式。孟繁鹏主要负责前端开发，黄瀚文主要负责后端开发。登录注册界面整体逻辑如下：



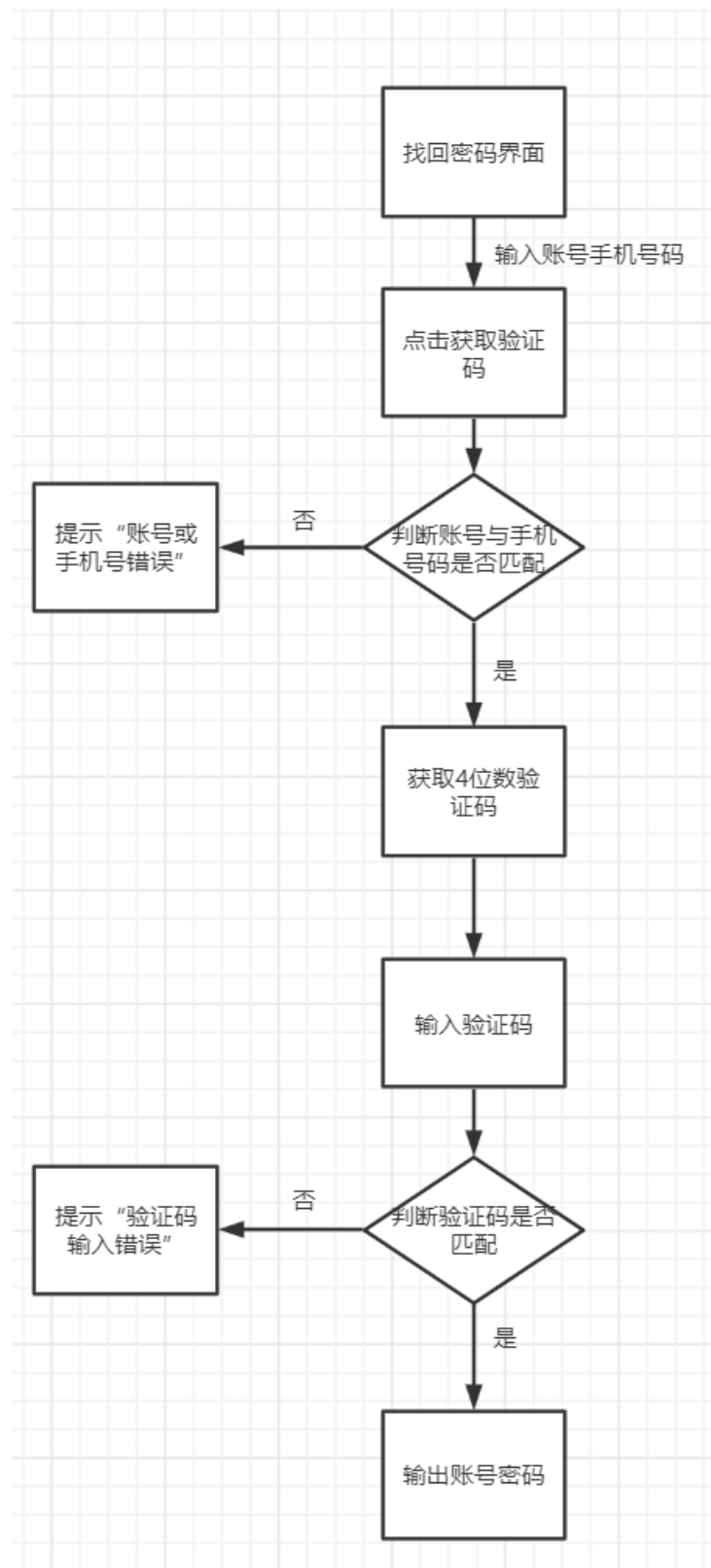
登录对比逻辑：



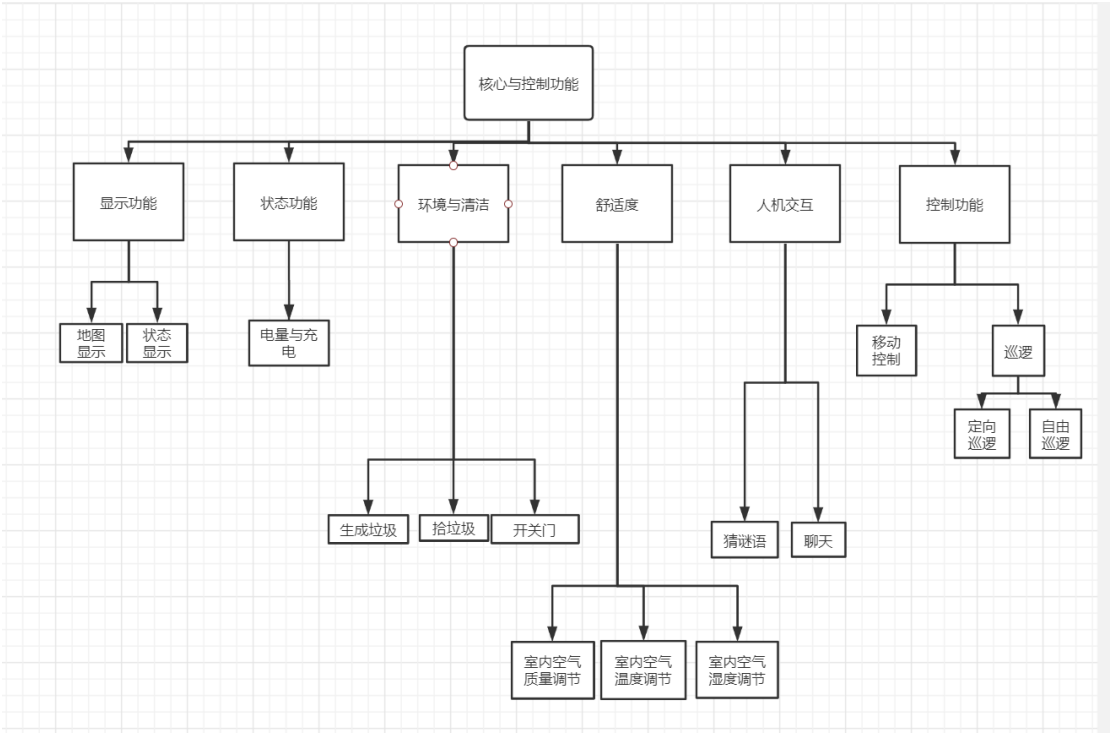
注册账号逻辑：



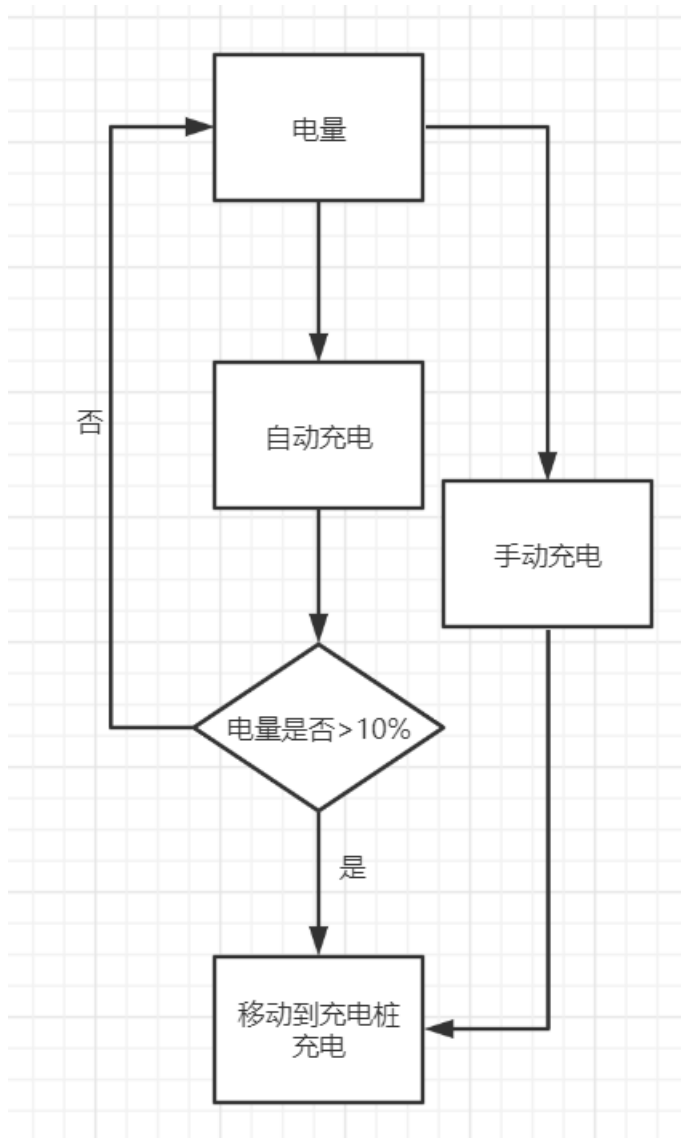
找回密码逻辑：



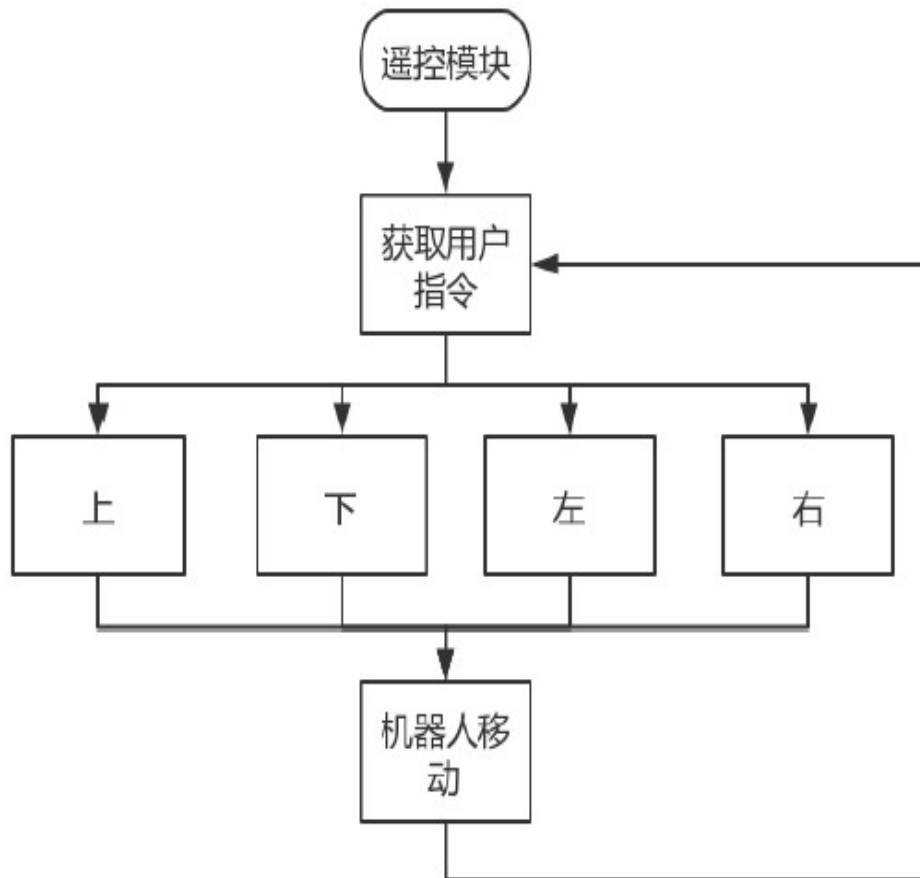
3.5.2 核心功能逻辑



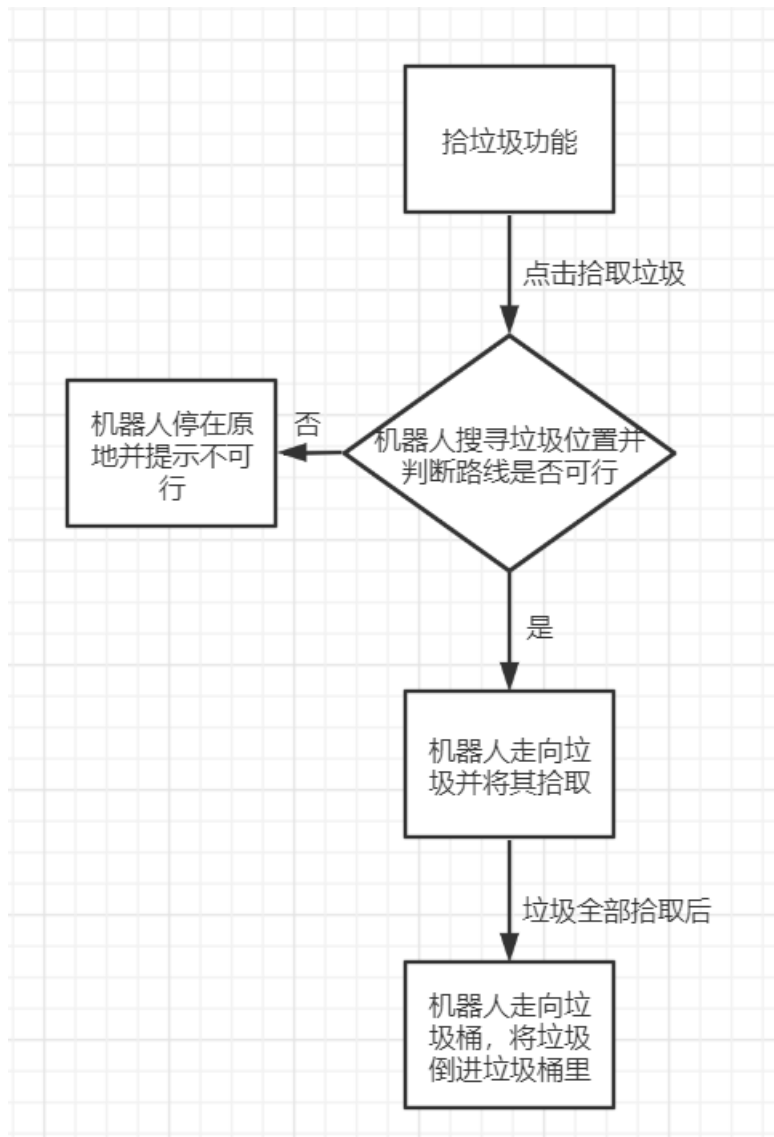
电量模块：



遥控模块：



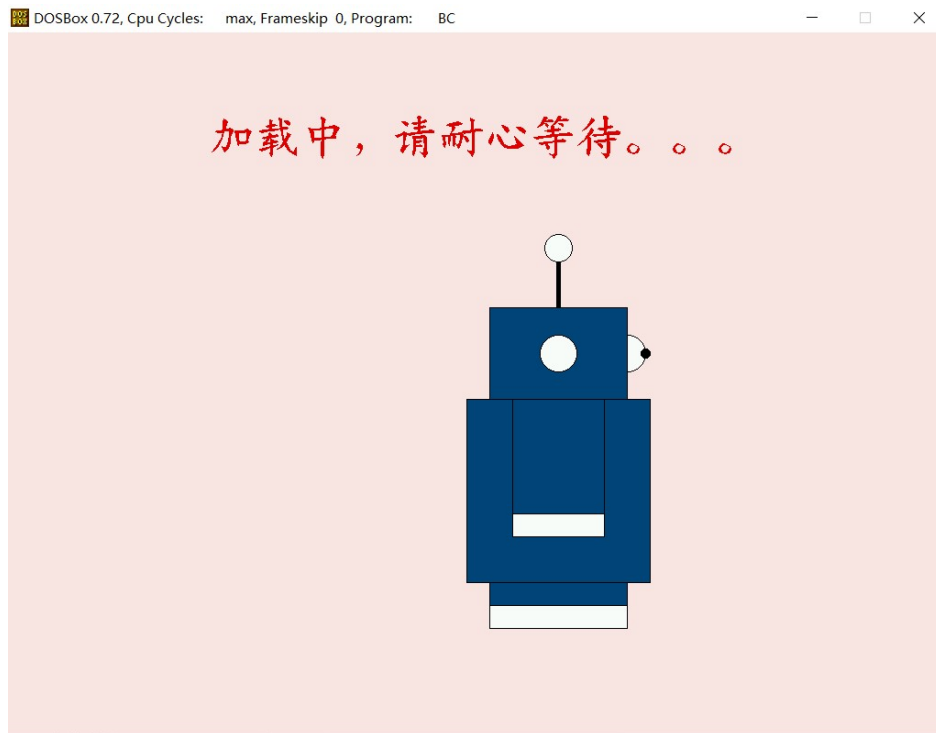
拾取垃圾模块：



4 界面设计

4.1 登录注册界面

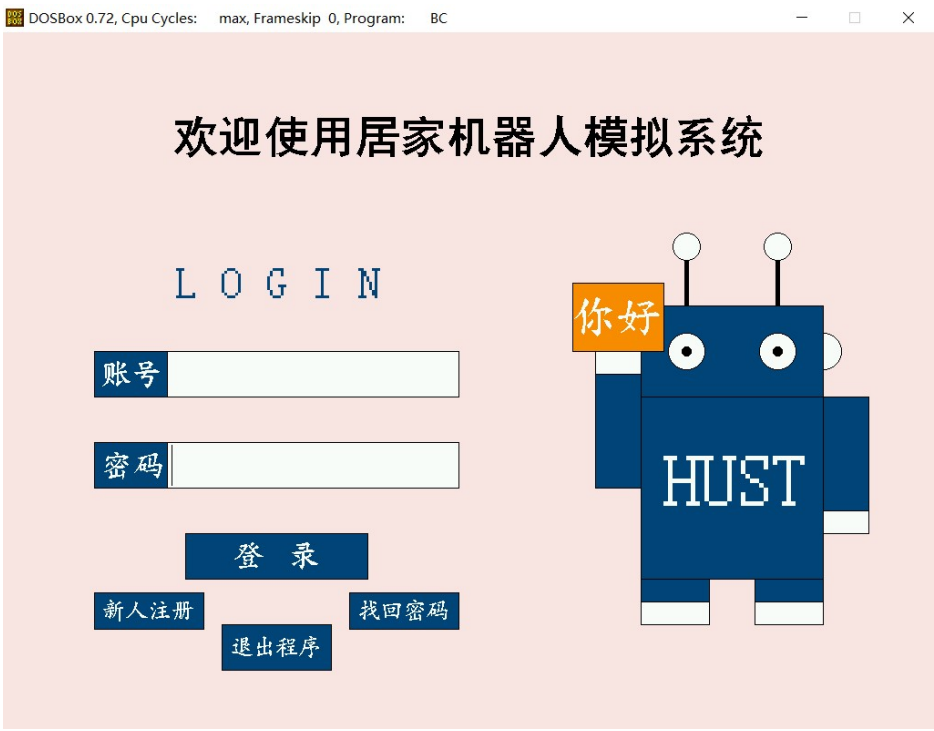
4.1.1 加载界面



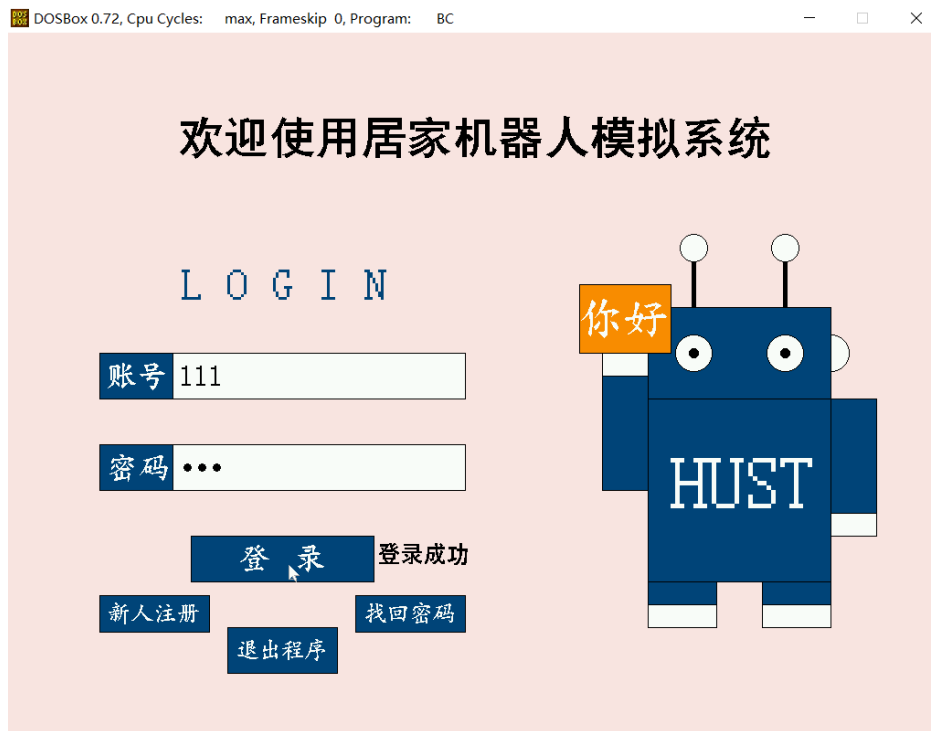
4.1.2 欢迎界面



4.1.3 登陆界面



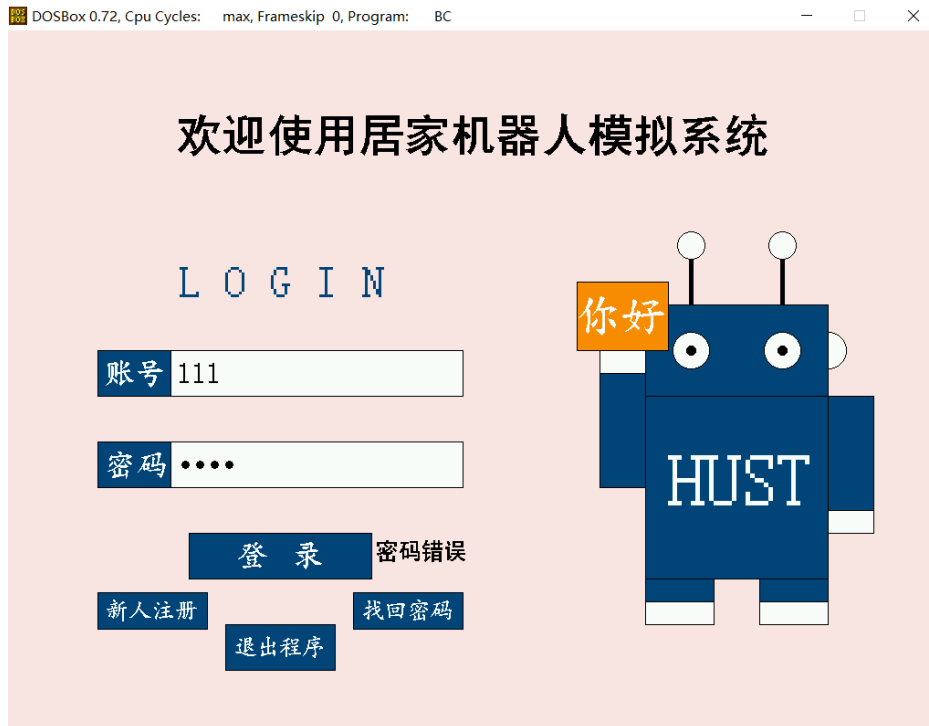
密码正确:



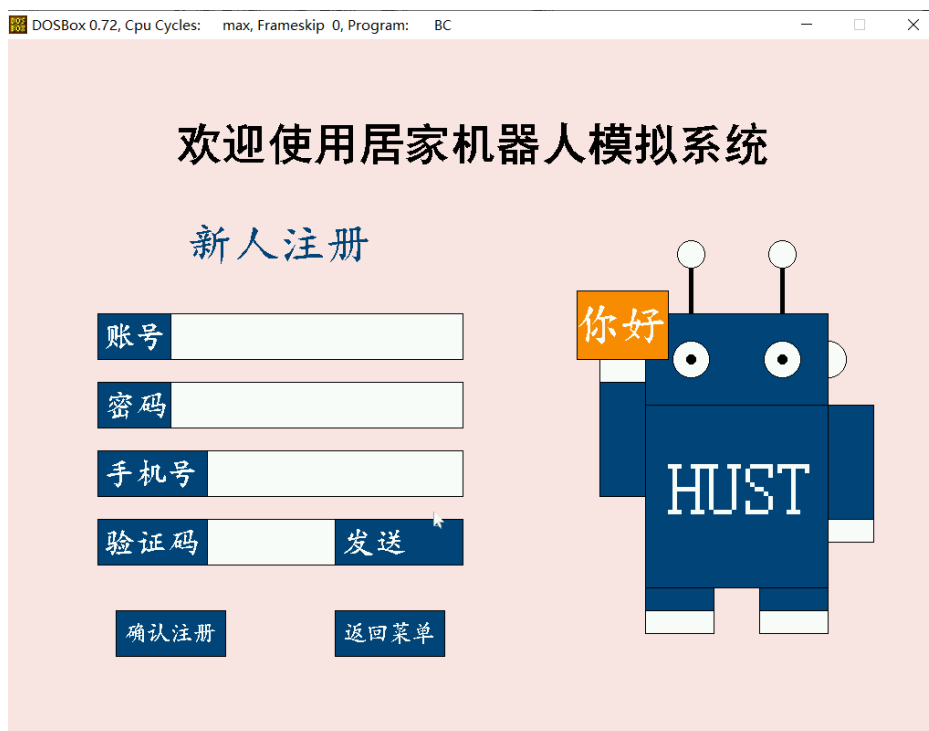
账号不存在:



密码错误:



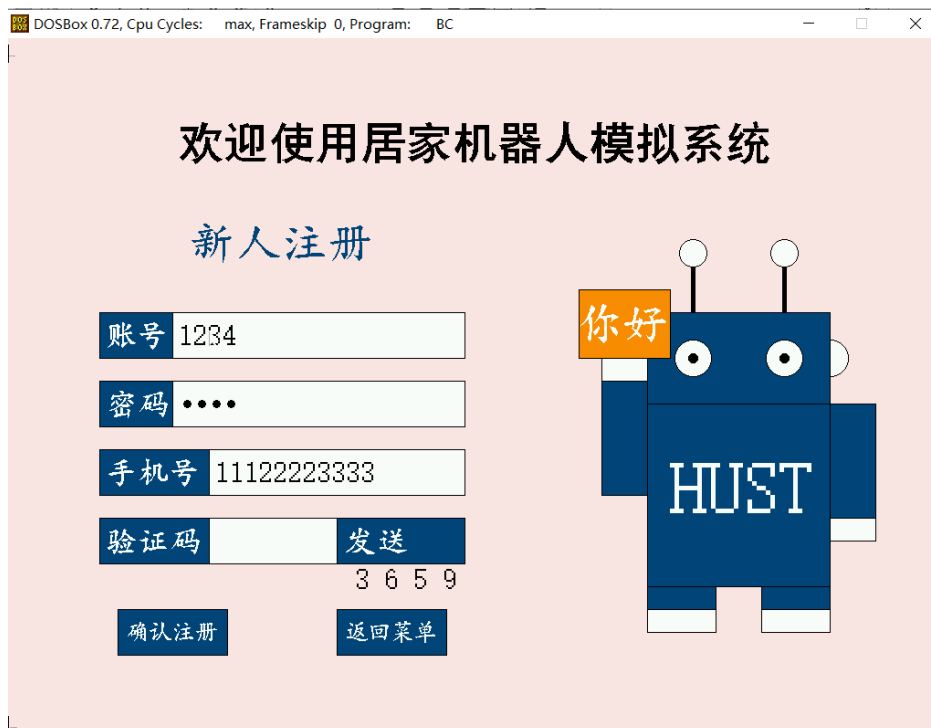
4.1.4 注册界面



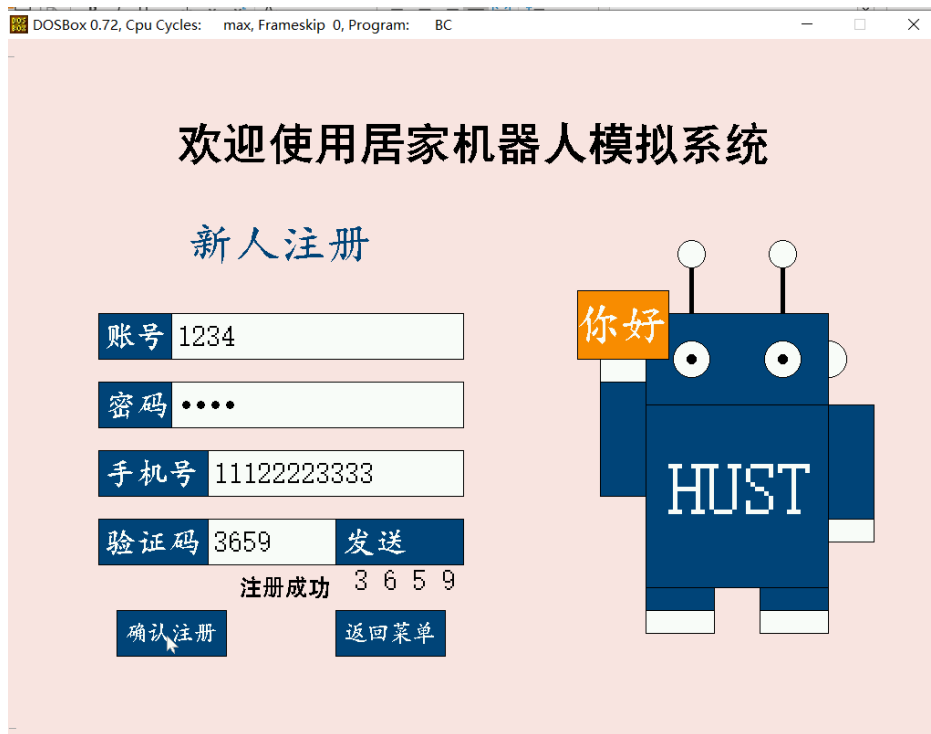
账号已注册:



信息正确:



注册成功:

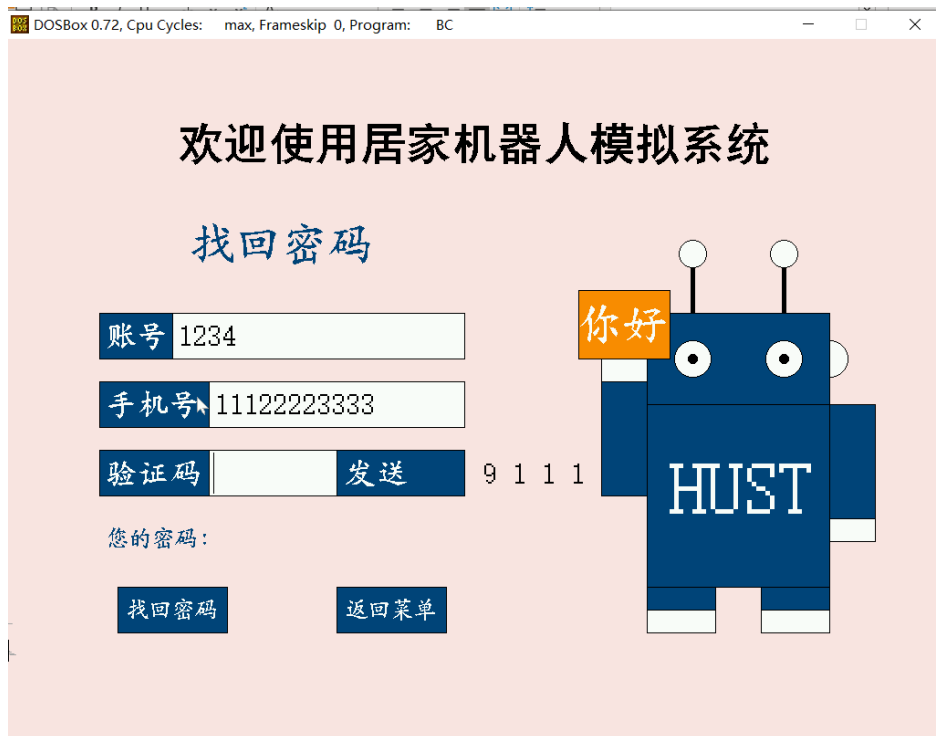


4.1.5 找回密码界面

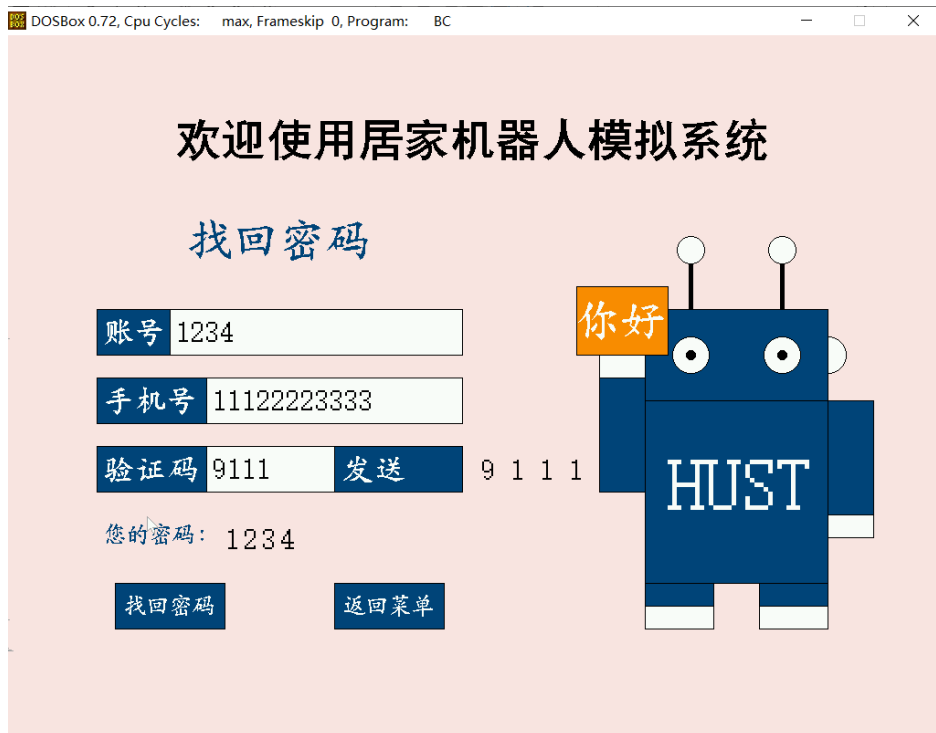
信息错误:



信息正确:



验证码正确，找回密码：



4.2 核心程序界面

4.2.1 核心框架



4.2.2 清洁界面

生成垃圾:



拾取垃圾:



4.2.3 电量界面

自动充电:



手动充电:



4.2.4 舒适度界面



4.2.5 控制界面



自动巡逻:



4.2.6 人机交互界面



聊天:



字谜:



5 数据结构与算法

5.1 数据结构

结构体 1：用于储存机器人基本信息。

```
typedef struct{  
    int electr; //机器人电量  
    int px, py; //机器人位置  
    char rt; //机器人朝向  
} ROBOT;
```

结构体 2：用于储存房间的基本信息。

```
typedef struct{  
    long int time; //程序运行时间  
    int set, setd, setc; //人工设定标签  
    int tempset; //温度设定温度  
    int tempout; //室外温度  
    int temp; //房间温度  
    int wet; //房间湿度  
    int pm25; //房间的 pm2.5 指数  
    int mp1[N][N]; //储存地图  
    int mpinit[N][N]; //储存初始地图
```

```
int dor[5];
int rubnum;
} HOUSE;
```

5.2 高级算法

本程序采用了两次高级算法，并将在此做呈现分析：

1. 使用 Astar 实现路径规划。

使用 heap 数组手动模拟优先队列，采用某一位置到起点的曼哈顿距离作为估价函数，最后得到启发式搜索找到的一条路径。

```
#include "headers.h"
```

```
/**heap 实现优先队列部分**/
```

```
void swapnode(NODE1 *a, NODE1 *b)
```

```
{
    NODE1 x=*a;
    *a=*b, *b=x;
}
```

```
void put(NODE1 *heap, int *size, NODE1 d)
```

```
{
    int now,next;
    heap[++(*size)]=d;
    now=(*size);
    while(now>1)
    {
        next=now>>1;
        if(heap[now].val>=heap[next].val) return;
        swapnode(&heap[now],&heap[next]);
        now=next;
    }
}
```

```
void del(NODE1 *heap, int *size)
```

```
{
    int now,next;
    heap[1]=heap[( *size)--];
    now=1;
    while(now*2<=(*size))
    {
        next=now*2;
```



```

        if(next<(*size)&&heap[next+1].val<heap[next].val) next++;
        if(heap[now].val<=heap[next].val) return;
        swapnode(&heap[now],&heap[next]);
        now=next;
    }
    return;
}
/****完成优先队列的实现****/

```

```

int val_func(int a,NODE beg,int x,int y) //估价函数
{
    return a+abs(beg.x-x)+abs(beg.y-y);
}

```

```

int BFS(NODE beg,NODE end,NODE1 *node,int sz,HOUSE *house)
{
    NODE1 heap[M];
    int hp_sz=0; //定义优先队列

    int vis_map[N][N]={0}; //地图访问情况
    int dir[4][2]={{0,1},{0,-1},{-1,0},{1,0}}; //方向数组
    int i,newx,newy;

    node[++sz].x=beg.x;
    node[sz].y=beg.y;
    node[sz].dis=0;
    node[sz].val=val_func(node[sz].dis,beg,node[sz].x,node[sz].y);
    node[sz].id=sz;
    node[sz].last=-1;
    put(heap,&hp_sz,node[sz]);
    while(hp_sz!=0)
    {
        NODE1 hed=heap[1];
        del(heap,&hp_sz);
        vis_map[hed.x][hed.y]=1;
        for(i=0;i<4;i++)
        {
            newx=dir[i][0]+hed.x;

```

```

newy=dir[i][1]+hed.y;
if(newx>=0 && newx<N && newy>=0 && newy<N
&& !vis_map[newx][newy] && pd_pass((*house).mp1[newx][newy]))
{
    vis_map[newx][newy]=1;
    node[++sz].x=newx;
    node[sz].y=newy;
    node[sz].dis=hed.dis+1;
    node[sz].last=hed.id;
    node[sz].id=sz;
    node[sz].val=val_func(node[sz].dis,beg,node[sz].x,node[sz].y);
    put(heap,&hp_sz,node[sz]);
    if(newx==end.x&&newy==end.y) return sz;
}
}
}
return -1;
}

```

```

void get_path(NODE *road, int *cnt, NODE1 *node, int x)
{
    if(node[x].last==-1)
    {
        road[++(*cnt)].x=node[x].x;
        road[*cnt].y=node[x].y;
        return;
    }
    get_path(road,cnt,node,node[x].last);
    road[++(*cnt)].x=node[x].x;
    road[*cnt].y=node[x].y;
}

```

```

int Astarmove(NODE beg, NODE end, ROBOT *robot, HOUSE *house)
{
    NODE1 node[M];
    int nd_sz=0; //定义结点记录数组
    NODE road[M];

```

```

int rd_sz=0; //定义路径记录数组

char dir;
int i,j;

nd_sz=BFS(beg,end,node,0,house); //宽度优先搜索得到路径
if(nd_sz==-1) return 0;
get_path(road,&rd_sz,node,nd_sz); //规范化路径
for(i=2;i<=rd_sz;i++)
{
    /*for(j=0;j<=100;j++)
        timepass(house,robot,1);*/
    if(i%3==0) (*robot).electr--;
    if(road[i].x-road[i-1].x==1) dir='s';
    if(road[i].x-road[i-1].x==-1) dir='w';
    if(road[i].y-road[i-1].y==1) dir='d';
    if(road[i].y-road[i-1].y==-1) dir='a';
    Delaytime(100);
    moveupdate(house,robot,dir);
}

return 1;
}

```

2. kmp 统计输入串中某一子串的个数

```

int count_str(char *str,char *tar)
{
    int num=0,ne[40]={0},l1,l2;
    int i,j;
    l1=strlen(str);
    l2=strlen(tar);
    str--;
    tar--;
    j=0,ne[0]=-1;
    for(i=2;i<=l2;i++)
    {
        while(j&&tar[i]!=tar[j+1]) j=ne[j];
        if(tar[j+1]==tar[i]) j++;
        ne[i]=j;
    }
}

```

```

    }
    j=0;
    for(i=1;i<=l1;i++)
    {
        while(j>0&&tar[j+1]!=str[i]) j=ne[j];
        if(tar[j+1]==str[i]) j++;
        if(j==l2) num++;
    }
    return num;
}

```

6 源代码

6.1 系统和主函数

6.1.1 svga.c *

svga 头文件（使用前人代码）。

6.1.2 mouse.c *

鼠标文件（mfp 改编前人代码使用）。

新功能：可以判断鼠标在一个圆内的状态。w

6.1.3 hzk.c *

汉字库文件(mfp 改编前人代码使用)。

新功能：

1. 可使用 4 种不同的字号（16，24，32，48）和 4 种字体（宋，楷，黑，仿宋）。
2. 可以在某个小范围内限定输入汉字，使汉字在超过小框后自动换行

6.1.4 pinyin.c

拼音输入法文件

6.1.5 main.c

主函数。（黄瀚文负责）

```
#include "headers.h"
```

```

int main()
{
    SetSVGA64k();

```

```

        start_func();
        nocombo();
        return 0;
    }

```

6.1.6 mysys.c

自己编写的系统功能函数。（孟繁鹏负责）

```
#include "headers.h"
```

```

void Delaytime(int time)
{
    int i,j,k;
    for(i=0;i<time;i++)
    {for(j=0;j<1100;j++)
    for(k=0;k<100;k++);}
}

```

```

void swap(int *x,int *y)
{
    int t=0;
    t=*x, *x=*y, *y=t;
}

```

```

void nocombo()
{
    Delaytime(300);
    //Delaytime(100);
}

```

```

int sign(int x)
{
    if(x>0) return 1;
    if(x<0) return -1;
    return 0;
}

```

```

int randin(int x)
{
    int t;

```

```

        srand((unsigned)time(NULL));
        t=rand()%x;
        return t;
    }

```

• drawbas.c

自己编写的系统画图函数。（孟繁鹏负责）

```
#include "headers.h"
```

```

void setbkcol(int color)
{
    bar(0,0,1024,768,color);
}

```

```

void line(int x1,int y1,int x2,int y2,int color)
{
    unsigned int  far *const video_buffer=(unsigned int  far *)0xa0000000L;
    unsigned char new_page = 0;
    unsigned long int page;
    int i,wid,yt;
    double k;
    if(x1==x2)
    {
        if(y1>y2) swap(&y1,&y2);
        for(i=y1;i<=y2;i++)
            Putpixel64k(x1,i,color);
        return;
    }
    if(y1==y2)
    {
        if(x1>x2) swap(&x1,&x2);
        page=((unsigned long int)y1<<10)+x1;
        new_page=page>>15;
        SelectPage(new_page);
        for(i=0;i<x2-x1;i++)
            *(video_buffer+page+i)=color;
        return;
    }
    if(x1>x2)

```

```

{
    swap(&x1,&x2);
    swap(&y1,&y2);
}
k=(y2-y1)*1.00/(x2-x1);
for(i=x1;i<=x2;i++)
    Putpixel64k(i,(int) y1+(i-x1)*k,color);
return;
}

```

```

void thick_line(int x1,int y1,int x2,int y2,int thick,int color)
{
    unsigned int  far *const video_buffer=(unsigned int  far *)0xa0000000L;
    unsigned char new_page = 0;
    unsigned long int page;
    int i,j,wid,thi=thick/2;
    double k;
    if(x1==x2)
    {
        if(y1>y2) swap(&y1,&y2);
        for(i=y1;i<=y2;i++)
            for(j=-thi;j<=thi;j++)
                Putpixel64k(x1+j,i,color);
        return;
    }
    if(y1==y2)
    {
        if(x1>x2) swap(&x1,&x2);
        for(i=x1;i<=x2;i++)
            for(j=-thi;j<=thi;j++)
                Putpixel64k(i,y1+j,color);
        return;
    }
    if(x1>x2)
    {
        swap(&x1,&x2);
        swap(&y1,&y2);
    }

```

```

        k=(y2-y1)*1.00/(x2-x1);
        for(j=-thi;j<=thi;j++)
        for(i=x1;i<=x2;i++)
            Putpixel64k(i,(int)y1+(i-x1)*k+j,color);
        return;
    }

```

```

void bar(int x1,int y1,int x2,int y2,int color)
{
    int i;
    if(x1>x2) swap(&x1,&x2);
    if(y1>y2) swap(&y1,&y2);
    for(i=y1;i<=y2;i++)
        line(x1,i,x2,i,color);
}

```

```

void rect(int x1,int y1,int x2,int y2,int color)
{
    if(x1>x2) swap(&x1,&x2);
    if(y1>y2) swap(&y1,&y2);
    line(x1,y1,x1,y2,color);
    line(x1,y1,x2,y1,color);
    line(x1,y2,x2,y2,color);
    line(x2,y1,x2,y2,color);
}

```

```

void drcircle(int xc,int yc,int r,int color)
{
    int x=0, y=r, dx=3, dy=2-2*r, d=1-r;
    if(r<=0) return;

    //运用 Bresenham 算法生成空心圆。
    while(x<=y)
    {
        Putpixel64k(xc-x,yc-y,color);
        Putpixel64k(xc+x,yc-y,color);
        Putpixel64k(xc-y,yc-x,color);
        Putpixel64k(xc+y,yc-x,color);
    }
}

```



```

        Putpixel64k(xc-y,yc+x,color);
        Putpixel64k(xc+y,yc+x,color);
        Putpixel64k(xc-x,yc+y,color);
        Putpixel64k(xc+x,yc+y,color);

        if(d<0) d+=dx, dx+=2;
        else d+=(dx+dy), dx+=2, dy+=2, y--;
        x++;
    }
}

void fill_rect(int x1,int y1,int x2,int y2,int color,int edgecolor)
{
    bar(x1,y1,x2,y2,color);
    rect(x1,y1,x2,y2,edgecolor);
}

void fill_circle(int xc,int yc,int r,int color,int edgecolor)
{
    int x=0, y=r, dx=3, dy=2-2*r, d=1-r;
    if(r<=0) return;

    //运用 Bresenham 算法生成实心圆。
    while(x<=y)
    {
        line(xc-x,yc-y,xc+x,yc-y,color);
        line(xc-y,yc-x,xc+y,yc-x,color);
        line(xc-y,yc+x,xc+y,yc+x,color);
        line(xc-x,yc+y,xc+x,yc+y,color);

        if(d<0) d+=dx, dx+=2;
        else d+=(dx+dy), dx+=2, dy+=2, y--;
        x++;
    }

    drcircle(xc,yc,r,edgecolor);
}

```

6.2 登陆注册函数

6.2.1 interf.c

登陆注册的前端函数。（孟繁鹏负责）

```
#include "headers.h"
```

```
void start_func()
```

```
{
    USER *usr;
    start_page();//画出开始界面

    while(1)
    {
        login_page(); //画出登录界面
        if(login_func(usr)) //进入登录界面
            mainWindow(usr); //程序主界面进入接
        else return;
    }
    return;
}
```

```
void input(int x1,int y1,int x2,int y2, char *s, int max_len, int fp, int fn,int backcolor)
```

```
{
    char value;
    int length=strlen(s);
    int width=16; //字符宽度
    int flag;
    line(x1+5+width*length, y1+3, x1+5+width*length, y2-3, BLACK);//绘制光标
    while(1)
    {
        int flag;
        newmouse(&MouseX, &MouseY, &press);
        if(mouse_press_out(x1,y1,x2,y2)==1)
        {
            s[length]='\0';
            line(x1+5+width*length, y1+3, x1+5+width*length, y2-3, backcolor); //
            覆盖光标
            break;
        }
    }
}
```

```

    }
    if(kbhit())
    {
        value=getch();
        if(value=='\n' || value=='\r') //判断到按下回车
        {
            s[length]='\0';
            line(x1+5+width*length, y1+3, x1+5+width*length, y2-3, bgcolor);
//覆盖光标
            break;
        }
        else if(value=='\b')
        {
            clrmouse(MouseX, MouseY);
            newmouse(&MouseX, &MouseY, &press);
            if(length==0) continue;
            line(x1+5+width*length, y1+3, x1+5+width*length, y2-3, BLACK);
            bar(x1+5+width*(length-1), y1+3, x2, y2-3, bgcolor); //覆盖光标
和最后一个字符
            s[--length]='\0';
            if(length+1<max_len) s[length+1]='\0';
            line(x1+5+width*length, y1+3, x1+5+width*length, y2-3, BLACK); //
重新绘制光标
        }
        //尝试添加可通过方向键动态修改功能，待完善
        /*else if(value==224)
        {
            value1=getch();
            if(value1==72 || value1==80) continue;
            else if(value1==75)
            else if(value1==77)
        }*/
        else if(length<max_len)
        {
            if(fn&&(value>'9' || value<'0')) continue; //判断是否要求全数字
            clrmouse(MouseX, MouseY);
            newmouse(&MouseX, &MouseY, &press);
            line(x1+5+width*length, y1+3, x1+5+width*length, y2-3, bgcolor);

```

```

        //覆盖光标
        //判断是否为密码，如果是，则需要保密
        if(!fp) putEngletter(x1-4+width*length, y1+8, (int)(value), 2,2,BLACK);
//显示字符
        else fill_circle(x1+16+width*length, (y1+y2)/2, 4,BLACK,BLACK); //显示
小圆点
        s[length++]=value;
        line(x1+5+width*length, y1+3, x1+5+width*length, y2-3, BLACK); //
重新绘制光标
    }
}
}
}
}

```

```

void start_page()
{
    int i,f1=1;
    setbkcol(MISTY_ROSE);
    for(i=15;i<=80;i++)
    {
        drawrobot_right(i*10,500,5);
        Delaytime(100);
        clearrobot(i*10,500,5);
        if(i%10<5) puthz(220,90,"加载中，请耐心等待。。。",48,50,'K',BLACK);
        else puthz(220,90,"加载中，请耐心等待。。。",48,50,'K',STRONG_RED);
    }
    setbkcol(MISTY_ROSE);
    drawrobot_front(800,500,5);
    Delaytime(1000);
    clearrobot(800,500,5);
    drawrobot_front_hello(800,500,5);
    puthz(187,90,"欢迎使用家居机器人模拟系统",48,50,'H',BLACK);
    puthz(120,500,"本程序由华中科技大学自动化学院",24,28,'K',BLACK);
    puthz(120,550,"一九零五班孟繁鹏，黄瀚文共同开发",24,28,'K',BLACK);
    puthz(200,300,"按任意键继续",32,35,'K',BLACK);
    while(1)
    {
        Delaytime(500);
    }
}

```

```

        if(f1) puthz(200,300,"按任意键继续",32,35,'K',BLACK), f1=0;
        else puthz(200,300,"按任意键继续",32,35,'K',STRONG_RED), f1=1;
        if(bioskey(1)) break;
    }
}

```

```

void login_page()

```

```

{
    setbkcol(MISTY_ROSE);
    drawrobot_front_hello(800,500,5);
    puthz(187,90,"欢迎使用居家机器人模拟系统",48,50,'H',BLACK);
    outtextxy(175,250,"LOGIN",3,3,50,MARINE_BLUE);

    fill_rect(100,350,500,400,WHITE,BLACK);
    fill_rect(100,450,500,500,WHITE,BLACK);
    fill_rect(100,350,180,400,MARINE_BLUE,BLACK);
    fill_rect(100,450,180,500,MARINE_BLUE,BLACK);
    puthz(108,359,"账号",32,35,'K',WHITE);
    puthz(108,459,"密码",32,35,'K',WHITE);

    fill_rect(200,550,400,600,MARINE_BLUE,BLACK);
    puthz(254,558,"登录",32,60,'K',WHITE); //画登录框图

    fill_rect(100,615,220,655,MARINE_BLUE,BLACK);
    fill_rect(380,615,500,655,MARINE_BLUE,BLACK);
    puthz(110,623,"新人注册",24,25,'K',WHITE);
    puthz(390,623,"找回密码",24,25,'K',WHITE);

    fill_rect(240,650,360,700,MARINE_BLUE,BLACK);
    puthz(250,663,"退出程序",24,25,'K',WHITE); //画找回和注册界面
}

```

```

void register_page()

```

```

{
    int begposi=300;

    setbkcol(MISTY_ROSE);

```

```

drawrobot_front_hello(800,500,5);
puthz(187,90,"欢迎使用居家机器人模拟系统",48,50,'H',BLACK);
puthz(200,200,"新人注册",48,50,'K',MARINE_BLUE);

fill_rect(100,begposi,500,begposi+50,WHITE,BLACK);
fill_rect(100,begposi+75,500,begposi+125,WHITE,BLACK);
fill_rect(100,begposi+150,500,begposi+200,WHITE,BLACK);
fill_rect(100,begposi+225,500,begposi+275,WHITE,BLACK);
fill_rect(100,begposi,180,begposi+50,MARINE_BLUE,BLACK);
fill_rect(100,begposi+75,180,begposi+125,MARINE_BLUE,BLACK);
fill_rect(100,begposi+150,220,begposi+200,MARINE_BLUE,BLACK);
fill_rect(100,begposi+225,220,begposi+275,MARINE_BLUE,BLACK);
fill_rect(360,begposi+225,500,begposi+275,MARINE_BLUE,BLACK);
puthz(108,begposi+9,"账号",32,35,'K',WHITE);
puthz(108,begposi+84,"密码",32,35,'K',WHITE);
puthz(108,begposi+159,"手机号",32,35,'K',WHITE);
puthz(108,begposi+234,"验证码",32,35,'K',WHITE);
puthz(370,begposi+234,"发送",32,35,'K',WHITE);//画注册框图

fill_rect(120,begposi+325,240,begposi+375,MARINE_BLUE,BLACK);
fill_rect(360,begposi+325,480,begposi+375,MARINE_BLUE,BLACK);
puthz(130,begposi+338,"确认注册",24,25,'K',WHITE);
puthz(370,begposi+338,"返回菜单",24,25,'K',WHITE); //画找回和注册界面
}

void findback_page()
{
    int begposi=300;

    setbkcol(MISTY_ROSE);
    drawrobot_front_hello(800,500,5);
    puthz(187,90,"欢迎使用居家机器人模拟系统",48,50,'H',BLACK);
    puthz(200,200,"找回密码",48,50,'K',MARINE_BLUE);

    fill_rect(100,begposi,500,begposi+50,WHITE,BLACK);
    fill_rect(100,begposi+75,500,begposi+125,WHITE,BLACK);
    fill_rect(100,begposi+150,500,begposi+200,WHITE,BLACK);
    fill_rect(100,begposi,180,begposi+50,MARINE_BLUE,BLACK);

```

```

fill_rect(100,begposi+75,220,begposi+125,MARINE_BLUE,BLACK);
fill_rect(100,begposi+150,220,begposi+200,MARINE_BLUE,BLACK);
fill_rect(360,begposi+150,500,begposi+200,MARINE_BLUE,BLACK);
puthz(108,begposi+9,"账号",32,35,'K',WHITE);
puthz(108,begposi+84,"手机号",32,35,'K',WHITE);
puthz(108,begposi+159,"验证码",32,35,'K',WHITE);
puthz(370,begposi+159,"发送",32,35,'K',WHITE); //画找回密码框图

puthz(108,begposi+235,"您的密码: ",24,25,'K',MARINE_BLUE);

fill_rect(120,begposi+300,240,begposi+350,MARINE_BLUE,BLACK);
fill_rect(360,begposi+300,480,begposi+350,MARINE_BLUE,BLACK);
puthz(130,begposi+313,"找回密码",24,25,'K',WHITE);
puthz(370,begposi+313,"返回菜单",24,25,'K',WHITE); //画找回和注册界面
}

void exit_window()
{
    int begposi=200;

    fill_rect(200,begposi,600,begposi+300,WHITE,BLACK);
    fill_rect(300,begposi+250,330,begposi+280,MARINE_BLUE,BLACK);
    fill_rect(470,begposi+250,500,begposi+280,MARINE_BLUE,BLACK);
    puthz(267,begposi+100,"确定退出本程序吗",32,35,'K',BLACK);
    puthz(305,begposi+255,"是",24,25,'K',WHITE);
    puthz(475,begposi+255,"否",24,25,'K',WHITE); //画退出界面
}

```

6.2.2 login.c

登录后端函数。(黄瀚文负责)

登陆#include "headers.h"

```

int login_func(USER *us)
{
    int flag;
    USER usr; //用户信息
    usr.user[0]='\0';
    usr.code[0]='\0';
    mouseinit();

```

```

while(1)
{
    newmouse(&MouseX, &MouseY, &press);
    if(mouse_press(180,350,500,400)==1) //输入账号
        input(180,350,500,400,usr.user,15,0,0,WHITE);
    if(mouse_press(180,450,500,500)==1) //输入密码
        input(180,450,500,500,usr.code,15,1,0,WHITE);

    if(mouse_press(200,550,400,600)==1)
    {
        if (login_comp(usr.user, usr.code)) //信息正确
        {
            clrmous(MouseX, MouseY);
            strcpy((*us).user,usr.user);
            return 1;
        }
        else continue;
    }
    if(mouse_press(100,615,220,655)==1)
    {
        clrmous(MouseX, MouseY);
        register_page();
        nocombo();
        user_register();
        nocombo();
        continue;
    }
    if(mouse_press(380,615,500,655)==1)
    {
        clrmous(MouseX, MouseY);
        findback_page();
        nocombo();
        user_findback();
        nocombo();
        continue;
    }
    if(mouse_press(240,650,360,700)==1)
    {

```



```

        clrmous(MouseX, MouseY);
        exit_window();
        if(exit_pro()==1) return 0;
        nocombo();
        continue;
    }
}
}

void user_register()
{
    int f1=1, f2=1;
    int len; //电话号码长度
    USER usr; //用户信息
    char ver_cod[5]; //输入验证码
    char ver_cod1[5]; //随机验证码
    usr.user[0]='\0';
    usr.code[0]='\0';
    usr.tel[0]='\0';
    ver_cod[0]='\0';
    while(1)
    {
        newmouse(&MouseX, &MouseY, &press);
        if(mouse_press(180,300,500,350)==1&&f2) //输入账号
            input(180,300,500,350,usr.user,15,0,0,WHITE);
        if(mouse_press(180,375,500,425)==1&&f2) //输入密码
            input(180,375,500,425,usr.code,15,1,0,WHITE);
        if(mouse_press(220,450,500,500)==1&&f2) //输入电话
            input(220,450,500,500,usr.tel,15,0,1,WHITE);
        if(mouse_press(220,525,360,575)==1&&f2) //输入验证码
            input(220,525,360,575,ver_cod,15,0,1,WHITE);
        len=strlen(usr.tel);
        if(mouse_press(360,525,500,575)==1&&f1&&f2)
        {
            if(check_user(usr.user)!=1)
            {
                if(usr.code[0]!='\0')

```

```

    {
        if(len==11)
        {
            f1=0;
            random_vc(ver_cod1); //生成随机数
            bar(170,588,370,615,MISTY_ROSE);
            outtextxy(370,575,ver_cod1,2,2,32,BLACK); //输出随机数
        }
        else
        {
            bar(170,588,370,615,MISTY_ROSE);
            puthz(170,588,"手机号码错误",24,25,'H',BLACK);
        }
    }
    else
    {
        bar(170,588,370,615,MISTY_ROSE);
        puthz(170,588,"请输入密码",24,25,'H',BLACK);
    }
}
else
{
    bar(170,588,370,615,MISTY_ROSE);
    puthz(170,588,"该账号已经注册",24,25,'H',BLACK);
}
}
if(mouse_press(120,625,240,675)==1&&f2) //注册判断
{
    if(register_func usr.user,usr.code,usr.tel,ver_cod,ver_cod1))
    {
        f2=0;
        continue;
    }
    else continue;
}
if(mouse_press(360,625,480,675)==1)
{
    clrmous(MouseX, MouseY);

```

```

        login_page();
        nocombo();
        return;
    }
}

void user_findback()
{
    int f1=1, f2=1;
    USER usr; //用户信息
    char ver_cod[5]; //验证码
    char ver_cod1[5]; //随机验证码
    char code[15]; //找回密码的密码
    usr.user[0]='\0';
    usr.code[0]='\0';
    usr.tel[0]='\0';
    ver_cod[0]='\0';
    while(1)
    {
        newmouse(&MouseX, &MouseY, &press);
        if(mouse_press(180,300,500,350)==1&&f2) //输入账号
            input(180,300,500,350,usr.user,15,0,0,WHITE);
        if(mouse_press(220,375,500,425)==1&&f2) //输入电话
            input(220,375,500,425,usr.tel,15,0,1,WHITE);
        if(mouse_press(220,450,360,500)==1&&f2) //输入验证码
            input(220,450,360,500,ver_cod,15,0,1,WHITE);
        if(mouse_press(360,450,500,500)==1&&f1&&f2)
        {
            if (findback_func(usr.user,code,usr.tel)) //验证账号和手机号
            {
                f1=0;
                random_vc(ver_cod1); //生成随机数
                bar(510,459,650,509,MISTY_ROSE);
                bar(230,510,450,590,MISTY_ROSE);
                outtextxy(510,459,ver_cod1,2,2,32,BLACK); //输出随机数
            }
        }
    }
}

```

```

if(mouse_press(120,600,240,650)==1&&f2) //找回密码判断
{
    if (strcmp(ver_cod,ver_cod1)==0) //判断验证码
    {
        bar(230,510,450,590,MISTY_ROSE);
        outtextxy(230,535,code,2,2,20,BLACK);
        f2=0;
        continue;
    }
    else
    {
        bar(230,510,450,590,MISTY_ROSE);
        puthz(230,535,"输入的验证码错误",24,25,'H',BLACK);
        /*outtextxy(230,535,"error",2,2,20,BLACK);*/
        continue;
    }
}
if(mouse_press(360,600,480,650)==1)
{
    clrmous(MouseX, MouseY);
    login_page();
    nocombo();
    return;
}
}

```

```

int exit_pro()
{
    while(1)
    {
        newmouse(&MouseX, &MouseY, &press);
        if(mouse_press(300,450,330,480)==1)
            return 1;
        if(mouse_press(470,450,500,480)==1)
        {
            clrmous(MouseX, MouseY);
            login_page();
        }
    }
}

```

```

        nocombo();
        return 0;
    }
}
}

int login_comp(char *user, char *code)
{
    int state1 = 1; //是否输入的变量
    int state2 = 1;
    while (1)
    {
        judge(user, &state1);
        judge(code, &state2);
        if (state1 == 1 || state2 == 1)
            break;
        else if (state1 == 0 && state2 == 0)
        {
            if (judge_rightpassword(user, code))
                return 1;
            else
                break;
        }
    }
    return 0;
}

```

```

int judge_rightpassword(char *user, char *code)
{
    int len;
    int i;
    FILE *fp;
    USER *u;
    if ((fp = fopen("data\\user.dat", "rb+")) == NULL)
    {
        printf("cannot open file UserData.dat");
        Delaytime(3000);
        exit(1);
    }
}

```

```

}
fseek(fp, 0, SEEK_END);
len = ftell(fp) / sizeof(USER); // 比对文件长度
for (i = 0; i < len; i++)
{
    if ((u = (USER*)malloc(sizeof(USER))) == NULL)
    {
        printf("memoryallocation runs wrong in lgfunc.c");
        Delaytime(3000);
        exit(1);
    }
    fseek(fp, i * sizeof(USER), SEEK_SET);
    fread(u, sizeof(USER), 1, fp);
    if (strcmp(user, u->user) == 0) // 用户名匹配
    {
        if (strcmp(code, u->code) != 0) // 密码不配
        {
            bar(405,558,505,608,MISTY_ROSE);
            puthz(405,558,"密码错误",24,25,'H',BLACK);
            if (u != NULL)
            {
                free(u);
                u = NULL;
            }
            break;
        }
        else if (strcmp(code, u->code) == 0) // 密码匹配
        {
            bar(405,558,505,608,MISTY_ROSE);
            puthz(405,558,"登录成功",24,25,'H',BLACK);
            if (u != NULL)
            {
                free(u);
                u = NULL;
            }
            Delaytime(1000);
            if (fclose(fp) != 0)
            {

```

```

        printf("\n cannot close Database");
        Delaytime(3000);
        exit(1);
    }
    return 1;
    //切换界面
}
}
if (u != NULL)
{
    free(u);
    u = NULL;
}
}
if (i == len)
{
    puthz(405,558,"该账号不存在",24,25,'H',BLACK);//走到最后一位
}
if (u != NULL)
{
    free(u);
    u = NULL;
}
if (fclose(fp) != 0)
{
    printf("\n cannot close Database");
    Delaytime(3000);
    exit(1);
}
return 0;
}

void judge(char *str, int *p)
{
    if (strlen(str) != 0)
    {
        *p = 0;
    }
}

```

}的后端逻辑函数。（黄瀚文负责）

6.2.3 register.c

注册的后端逻辑函数。（黄瀚文负责）

```
void random_vc(char *vc)
{
    int a;
    int i;
    int x[4];
    srand((unsigned)(time(NULL)));
    a=rand()%9000+1000;
    x[0]=a/1000;
    x[1]=a/100%10;
    x[2]=a/10%10;
    x[3]=a%10;
    for ( i = 0; i < 4; i++)
    {
        vc[i]=x[i]+'0';
    }
    vc[4]='\0';
}

int register_func(char *user,char *code,char *tel,char *vc,char *vc1)
{
    int len;
    len=strlen(tel);
    if(check_user(user)!=1)
    {
        if(code[0]!='\0')
        {
            if(len==11)
            {
                if (strcmp(vc,vc1)==0)
                {
                    input_database(user,code,tel); //写入数据库
                    bar(170,588,370,615,MISTY_ROSE);
                    puthz(255,588,"注册成功",24,25,'H',BLACK);
                }
            }
        }
    }
}
```



```

        Delaytime(2000);
        return 1;
    }
    else
    {
        bar(170,588,370,615,MISTY_ROSE);
        puthz(170,588,"输入的验证码错误",24,25,'H',BLACK);
        return 0;
    }
}
else
{
    bar(170,588,370,615,MISTY_ROSE);
    puthz(170,588,"手机号码错误",24,25,'H',BLACK);
    return 0;
}
}
else
{
    bar(170,588,370,615,MISTY_ROSE);
    puthz(170,588,"请输入密码",24,25,'H',BLACK);
    return 0;
}
}
else
{
    bar(170,588,370,615,MISTY_ROSE);
    puthz(170,588,"该账号已经注册",24,25,'H',BLACK);
    return 0;
}
}

```

```

int findback_func(char *user,char *code,char *tel)
{
    int f=0;
    output_database(user,code,tel,&f); //读取输出数据库数据
    if(f)
    {

```

```

        Delaytime(2000);
        return 1;
    }
    return 0;
}

void input_database(char *user1,char *code1,char *tel1)
{
    FILE *fp;
    USER *u;
    char n;
    if ((fp = fopen("data\\user.dat", "rb+" )) == NULL)//建立数据库
    {
        printf("cannot open file UserData.dat");
        Delaytime(3000);
        exit(1);
    }
    if ((u = (USER*)malloc(sizeof(USER))) == NULL)
    {
        printf("memoryallocation runs wrong in lgfunc.c");
        Delaytime(3000);
        exit(1);
    }
    memset(u,'\0',sizeof(USER));
    strcpy(u->user,user1);
    strcpy(u->code,code1);
    strcpy(u->tel,tel1);
    fseek(fp, 0, SEEK_END);//跳转至文件末尾
    fwrite(u, sizeof(USER), 1, fp);//把用户信息写入文件
    if (u != NULL)
    {
        free(u);
        u = NULL;
    }
    if (fclose(fp) != 0)
    {
        printf("\n cannot close Database.");
        Delaytime(3000);
    }
}

```

```

        exit(1);
    }
}

void output_database(char *user,char *code,char *tel,int *f)
{
    if (check_data(user,code,tel))
    {
        *f = 1;
    }
    else
    {
        bar(230,510,450,590,MISTY_ROSE);
        puthz(230,535,"账号或手机号错误",24,25,'K',BLACK);
        *f = 0;
    }
}

int check_data(char *user,char *code,char *tel)
{
    int len;
    int i;
    FILE* fp;
    USER* u = NULL;
    if ((fp = fopen("data\\user.dat", "rb+")) == NULL)//建立数据库
    {
        printf("cannot open file UserData.dat");
        Delaytime(3000);
        exit(1);
    }
    fseek(fp, 0, SEEK_END);
    len = ftell(fp) / sizeof(USER);
    for (i = 0; i < len; i++)
    {
        if ((u = (USER*)malloc(sizeof(USER))) == NULL)
        {
            printf("memoryallocation runs wrong in lgfunc.c");
            Delaytime(3000);

```

```

        exit(1);
    }
    fseek(fp, i * sizeof(USER), SEEK_SET);
    fread(u, sizeof(USER), 1, fp);
    if (strcmp(u->user, user)==0&&strcmp(u->tel, tel)==0)
    {
        strcpy(code,u->code);
        if (u != NULL)
        {
            free(u);
            u = NULL;
        }
        if (fclose(fp) != 0)
        {
            printf("\n cannot close Database.");
            Delaytime(3000);
            exit(1);
        }
        return 1;
    }
    free(u);
    u = NULL;
}
if (u != NULL)
{
    free(u);
    u = NULL;
}
if (fclose(fp) != 0)
{
    printf("\n cannot close Database.");
    Delaytime(3000);
    exit(1);
}
return 0;
}

```

```

int check_user(char *user)

```

```

{
    int len;
    int i;
    FILE* fp;
    USER* u = NULL;
    if ((fp = fopen("data\\user.dat", "rb+")) == NULL)//建立数据库
    {
        printf("cannot open file UserData.dat");
        Delaytime(3000);
        exit(1);
    }
    fseek(fp, 0, SEEK_END);
    len = ftell(fp) / sizeof(USER);
    for (i = 0; i < len; i++)
    {
        if ((u = (USER*)malloc(sizeof(USER))) == NULL)
        {
            printf("memoryallocation runs wrong in lgfunc.c");
            Delaytime(3000);
            exit(1);
        }
        fseek(fp, i * sizeof(USER), SEEK_SET);
        fread(u, sizeof(USER), 1, fp);
        if (strcmp(u->user, user)==0)
        {
            if (u != NULL)
            {
                free(u);
                u = NULL;
            }
            if (fclose(fp) != 0)
            {
                printf("\n cannot close Database.");
                Delaytime(3000);
                exit(1);
            }
            return 1;
        }
    }
}

```

```

        free(u);
        u = NULL;
    }
    if (u != NULL)
    {
        free(u);
        u = NULL;
    }
    if (fclose(fp) != 0)
    {
        printf("\n cannot close Database.");
        Delaytime(3000);
        exit(1);
    }
    return 0;
}

```

6.3 核心程序函数

6.3.1 drawrob.c

绘制机器人函数。（孟繁鹏负责）

```

void drawrobot_front(int x,int y,int size)
{
    int sb=ROBOT_SIZE*size;
    fill_rect(x-sb,y-sb,x+sb,y+sb,MARINE_BLUE,BLACK); //画机器人身子

    fill_rect(x+sb,y-sb,x+sb/2*3,y+sb/2,MARINE_BLUE,BLACK);
    fill_rect(x-sb,y-sb,x-sb/2*3,y+sb/2,MARINE_BLUE,BLACK);
    fill_rect(x+sb,y+sb/4,x+sb/2*3,y+sb/2,WHITE,BLACK);
    fill_rect(x-sb,y+sb/4,x-sb/2*3,y+sb/2,WHITE,BLACK); //画机器人胳膊

    fill_rect(x+sb,y+sb,x+sb/4,y+sb/2*3,MARINE_BLUE,BLACK);
    fill_rect(x-sb,y+sb,x-sb/4,y+sb/2*3,MARINE_BLUE,BLACK);
    fill_rect(x+sb,y+sb/4*5,x+sb/4,y+sb/2*3,WHITE,BLACK);
    fill_rect(x-sb,y+sb/4*5,x-sb/4,y+sb/2*3,WHITE,BLACK); //画机器人腿

    fill_circle(x+sb,y-sb/2*3,sb/5,WHITE,BLACK);
    fill_circle(x-sb,y-sb/2*3,sb/5,WHITE,BLACK);
}

```

```

fill_rect(x-sb,y-sb,x+sb,y-2*sb,MARINE_BLUE,BLACK);
fill_circle(x+sb/2,y-sb/2*3,sb/5,WHITE,BLACK);
fill_circle(x-sb/2,y-sb/2*3,sb/5,WHITE,BLACK);
fill_circle(x+sb/2,y-sb/2*3,sb/20,BLACK,BLACK);
fill_circle(x-sb/2,y-sb/2*3,sb/20,BLACK,BLACK);

if(y-sb/2*5>=0)
{
    thick_line(x+sb/2,y-sb/2*5,x+sb/2,y-2*sb,sb/20,BLACK);
    thick_line(x-sb/2,y-sb/2*5,x-sb/2,y-2*sb,sb/20,BLACK);
    fill_circle(x+sb/2,y-sb/2*5-sb/20*3,sb/20*3,WHITE,BLACK);
    fill_circle(x-sb/2,y-sb/2*5-sb/20*3,sb/20*3,WHITE,BLACK); //画机器人天线
}

outtextxy(x-sb,y-sb/2,"HUST",sb/20,sb/20,sb/10*4,WHITE); //胸前写字

return;
}

void drawrobot_front_hello(int x,int y,int size)
{
    int sb=ROBOT_SIZE*size;
    fill_rect(x-sb,y-sb,x+sb,y+sb,MARINE_BLUE,BLACK); //画机器人身子

    fill_rect(x+sb,y+sb,x+sb/4,y+sb/2*3,MARINE_BLUE,BLACK);
    fill_rect(x-sb,y+sb,x-sb/4,y+sb/2*3,MARINE_BLUE,BLACK);
    fill_rect(x+sb,y+sb/4*5,x+sb/4,y+sb/2*3,WHITE,BLACK);
    fill_rect(x-sb,y+sb/4*5,x-sb/4,y+sb/2*3,WHITE,BLACK); //画机器人腿

    fill_circle(x+sb,y-sb/2*3,sb/5,WHITE,BLACK);
    fill_circle(x-sb,y-sb/2*3,sb/5,WHITE,BLACK);
    fill_rect(x-sb,y-sb,x+sb,y-2*sb,MARINE_BLUE,BLACK);
    fill_circle(x+sb/2,y-sb/2*3,sb/5,WHITE,BLACK);
    fill_circle(x-sb/2,y-sb/2*3,sb/5,WHITE,BLACK);
    fill_circle(x+sb/2,y-sb/2*3,sb/20,BLACK,BLACK);
    fill_circle(x-sb/2,y-sb/2*3,sb/20,BLACK,BLACK);

    thick_line(x+sb/2,y-sb/2*5,x+sb/2,y-2*sb,sb/20,BLACK);

```

```

thick_line(x-sb/2,y-sb/2*5,x-sb/2,y-2*sb,sb/20,BLACK);
fill_circle(x+sb/2,y-sb/2*5-sb/20*3,sb/20*3,WHITE,BLACK);
fill_circle(x-sb/2,y-sb/2*5-sb/20*3,sb/20*3,WHITE,BLACK); //画机器人天线

outtextxy(x-sb,y-sb/2,"HUST",sb/20,sb/20,sb/10*4,WHITE); //胸前写字

fill_rect(x+sb,y-sb,x+sb/2*3,y+sb/2,MARINE_BLUE,BLACK);
fill_rect(x-sb,y,x-sb/2*3,y-sb*3/2,MARINE_BLUE,BLACK);
fill_rect(x+sb,y+sb/4,x+sb/2*3,y+sb/2,WHITE,BLACK);
fill_rect(x-sb,y-sb*5/4,x-sb/2*3,y-sb*3/2,WHITE,BLACK); //画机器人胳膊

if(y-sb/2*5>=0)
{
    fill_rect(x-sb*7/4,y-sb*3/2,x-sb*3/4,y-sb*9/4,DARK_ORANGE,BLACK);
    puthz(x-sb*7/4,y-sb*17/8,"你好",48,48,'K',WHITE); //画牌子
}

return;
}

void drawrobot_left(int x,int y,int size)
{
    int sb=ROBOT_SIZE*size;
    fill_rect(x-sb,y-sb,x+sb,y+sb,MARINE_BLUE,BLACK); //画机器人身子

    fill_rect(x-sb/2,y-sb,x+sb/2,y+sb/2,MARINE_BLUE,BLACK);
    fill_rect(x-sb/2,y+sb/4,x+sb/2,y+sb/2,WHITE,BLACK); //画机器人胳膊

    fill_rect(x-sb/4*3,y+sb,x+sb/4*3,y+sb/2*3,MARINE_BLUE,BLACK);
    fill_rect(x-sb/4*3,y+sb/4*5,x+sb/4*3,y+sb/2*3,WHITE,BLACK); //画机器人腿

    fill_circle(x-sb/4*3,y-sb/2*3,sb/5,WHITE,BLACK);
    fill_circle(x-sb/4*3-sb/5,y-sb/2*3,sb/20,BLACK,BLACK);
    fill_rect(x-sb/4*3,y-sb,x+sb/4*3,y-2*sb,MARINE_BLUE,BLACK);
    fill_circle(x,y-sb/2*3,sb/5,WHITE,BLACK); //画机器人头

    if(y-sb/2*5>=0)
    {

```



```

        thick_line(x,y-sb/2*5,x,y-2*sb,sb/20,BLACK);
        fill_circle(x,y-sb/2*5-sb/20*3,sb/20*3,WHITE,BLACK); //画机器人天线
    }

    return;
}

void drawrobot_right(int x,int y,int size)
{
    int sb=ROBOT_SIZE*size;
    fill_rect(x-sb,y-sb,x+sb,y+sb,MARINE_BLUE,BLACK); //画机器人身子

    fill_rect(x-sb/2,y-sb,x+sb/2,y+sb/2,MARINE_BLUE,BLACK);
    fill_rect(x-sb/2,y+sb/4,x+sb/2,y+sb/2,WHITE,BLACK); //画机器人胳膊

    fill_rect(x-sb/4*3,y+sb,x+sb/4*3,y+sb/2*3,MARINE_BLUE,BLACK);
    fill_rect(x-sb/4*3,y+sb/4*5,x+sb/4*3,y+sb/2*3,WHITE,BLACK); //画机器人腿

    fill_circle(x+sb/4*3,y-sb/2*3,sb/5,WHITE,BLACK);
    fill_circle(x+sb/4*3+sb/5,y-sb/2*3,sb/20,BLACK,BLACK);
    fill_rect(x-sb/4*3,y-sb,x+sb/4*3,y-2*sb,MARINE_BLUE,BLACK);
    fill_circle(x,y-sb/2*3,sb/5,WHITE,BLACK); //画机器人头

    if(y-sb/2*5>=0)
    {
        thick_line(x,y-sb/2*5,x,y-2*sb,sb/20,BLACK);
        fill_circle(x,y-sb/2*5-sb/20*3,sb/20*3,WHITE,BLACK); //画机器人天线
    }

    return;
}

void drawrobot_back(int x,int y,int size)
{
    int sb=ROBOT_SIZE*size;
    fill_rect(x-sb,y-sb,x+sb,y+sb,MARINE_BLUE,BLACK); //画机器人身子

    fill_rect(x+sb,y-sb,x+sb/2*3,y+sb/2,MARINE_BLUE,BLACK);

```

```

fill_rect(x-sb,y-sb,x-sb/2*3,y+sb/2,MARINE_BLUE,BLACK);
fill_rect(x+sb,y+sb/4,x+sb/2*3,y+sb/2,WHITE,BLACK);
fill_rect(x-sb,y+sb/4,x-sb/2*3,y+sb/2,WHITE,BLACK); //画机器人胳膊

fill_rect(x+sb,y+sb,x+sb/4,y+sb/2*3,MARINE_BLUE,BLACK);
fill_rect(x-sb,y+sb,x-sb/4,y+sb/2*3,MARINE_BLUE,BLACK);
fill_rect(x+sb,y+sb/4*5,x+sb/4,y+sb/2*3,WHITE,BLACK);
fill_rect(x-sb,y+sb/4*5,x-sb/4,y+sb/2*3,WHITE,BLACK); //画机器人腿

fill_circle(x+sb,y-sb/2*3,sb/5,WHITE,BLACK);
fill_circle(x-sb,y-sb/2*3,sb/5,WHITE,BLACK);
fill_rect(x-sb,y-sb,x+sb,y-2*sb,MARINE_BLUE,BLACK); //画机器人头

if(y-sb/2*5>=0)
{
    thick_line(x+sb/2,y-sb/2*5,x+sb/2,y-2*sb,sb/20,BLACK);
    thick_line(x-sb/2,y-sb/2*5,x-sb/2,y-2*sb,sb/20,BLACK);
    fill_circle(x+sb/2,y-sb/2*5-sb/20*3,sb/20*3,WHITE,BLACK);
    fill_circle(x-sb/2,y-sb/2*5-sb/20*3,sb/20*3,WHITE,BLACK); //画机器人天线
}
return;
}

void clearrobot(int x,int y,int size)
{
    int sb=ROBOT_SIZE*size*3;
    bar(x-sb,y-sb,x+sb,y+sb,MISTY_ROSE);
}

```

6.3.2 mainwin.c

核心程序主框架和时间流逝等重要函数。（孟繁鹏负责）

```

#include "headers.h"

#define timeupdate 100000 //更新界面时间
#define timeset 50000 //污染湿度按照要求更新
#define timedirt 500000 //污染，湿度程度更新时间
#define timetmpset 500000 //设定温度更新时间
#define timetmp 1000000 //温度更新时间

```

```

#define timeele 500000 //电量更新时间
#define timecut 10000000 //时间计数器请清零
#define setclean 20
#define setwet 20

void mainWindow(USER *usr)
{
    HOUSE *house;
    ROBOT *robot;
    NODE rubbish[4];
    char value, value1;
    int i,j;
    int poscode;
    NODE mp,mto; //鼠标点击后行动坐标

    drawbasic(); //画出画面底板
    clrmous(MouseX, MouseY);

    house=(HOUSE *)malloc(sizeof(HOUSE));
    robot=(ROBOT *)malloc(sizeof(ROBOT));
    maininit(house,robot); //定义并初始化房间和机器人

    paintmp(house,robot);
    draw_control(usr);
    draw_bactr(robot);
    write_statu(house,robot,1);
    house->rubnum=0;

    while(1)
    {
        newmouse(&MouseX, &MouseY, &press);
        timepass(house,robot,usr,1);
        if(mouse_press(LB+37,UB+350,LB+127,UB+390)==1) //进入电量界面
        {
            clrmous(MouseX, MouseY);
            draw_electr(usr);
            nocombo();
            if(func_electr(house,robot,usr))

```

```

        return ;
    draw_control(usr);
    draw_bactr(robot);
    write_statu(house,robot,1);
    continue;
}
if(mouse_press(LB+147,UB+350,LB+237,UB+390)==1) //进入舒适度界面
{
    clrmous(MouseX, MouseY);
    draw_comfort(usr);
    nocombo();
    if(func_comfort(house,robot,usr))
        return ;
    draw_control(usr);
    draw_bactr(robot);
    write_statu(house,robot,1);
    continue;
}
if(mouse_press(LB+37,UB+410,LB+127,UB+450)==1) //进入环境界面
{
    clrmous(MouseX, MouseY);
    draw_clean(usr);
    nocombo();
    if(func_clean(rubbish,house,robot,usr))
        return ;
    draw_control(usr);
    draw_bactr(robot);
    write_statu(house,robot,1);
    continue;
}
if(mouse_press(LB+147,UB+410,LB+237,UB+450)==1) //进入控制界面
{
    clrmous(MouseX, MouseY);
    draw_move(usr);
    nocombo();
    if(func_move(house,robot,usr))
        return ;
    draw_control(usr);

```

```

        draw_bactr(robot);
        write_statu(house,robot,1);
        continue;
    }
    if(mouse_press(LB+57,UB+470,LB+217,UB+510)==1) //进入互动界面
    {
        clrmous(MouseX, MouseY);
        draw_react(usr,0);
        nocombo();
        if(func_react(house,robot,usr))
            return ;
        draw_control(usr);
        nocombo();
        draw_bactr(robot);
        write_statu(house,robot,1);
        continue;
    }
    if(kbhit())
    {
        Delaytime(50);
        value=getch();
        moveupdate(house,robot,value);
    }
    get_conbot(house,robot);
    if(mouse_press(15,24,735,744)==1)
    {
        nocombo();
        poscode=getposition(MouseX, MouseY);
        mto.x=poscode/18;
        mto.y=poscode%18;
        mp.x=(*robot).px;
        mp.y=(*robot).py;
        if(!Astarmove(mp,mto,robot,house))
        {
            draw_cantgo();
        }
    }
    if(mouse_press(LB+140,UB+10,LB+250,UB+40)==1)

```

```

        return ;
    }
    return;
}

void drawbasic()
{
    setbkcol(MISTY_ROSE);
    bar(750,0,1024,768,MARINE_BLUE);
    bar(800,200,900,300,MISTY_ROSE);
}

void maininit(HOUSE *house, ROBOT *robot)
{
    int i,j;
    int mp1init[N][N]={
        {3,2,0,0,2,41,41,6,2,36,35,6,2,0,0,0,0,0},
        {0,9,0,0,2,0,0,6,2,39,36,6,2,5,5,0,0,0},
        {0,9,0,0,9,0,0,40,2,36,36,6,2,38,5,18,0,0},
        {0,2,0,0,9,0,0,6,2,36,36,37,2,38,5,19,0,0},
        {2,2,0,0,2,0,0,40,2,36,36,6,2,5,5,0,0,0},
        {0,0,0,0,2,6,6,6,2,36,36,6,2,0,0,0,0,0},
        {4,4,0,0,2,2,2,2,10,11,2,2,0,0,0,0,29},
        {5,5,0,0,0,0,0,0,0,0,0,0,0,9,0,0,29},
        {5,5,0,0,0,0,0,0,0,0,0,0,0,9,0,0,29},
        {30,30,0,0,0,0,0,2,10,11,2,2,2,2,2,2,2},
        {0,0,0,31,32,0,6,2,0,0,0,0,0,29,29,29,29},
        {28,0,0,0,0,20,17,2,0,0,0,0,0,0,0,0,0},
        {28,0,0,5,5,15,16,2,0,0,0,0,0,0,0,0,27},
        {28,0,0,5,5,20,17,2,28,0,0,0,0,7,7,7,8,12},
        {28,0,0,5,5,15,16,2,28,0,0,0,0,7,7,7,8,13},
        {28,0,0,0,0,20,17,2,28,0,0,0,0,7,7,7,8,14},
        {0,0,0,33,34,0,6,2,0,0,0,0,0,0,0,0,27},
        {21,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0},
    };
    //0 是地板，1 是机器人，2 是墙壁，3 是垃圾桶，4 30 是椅子，
    //5 是桌子，6 是一般高度家具，7 8 12 13 14 是床，9 是竖门，10 是左门，11 是右
    门

```

//21 是充电桩, 22 是垃圾, 23 24 25 26 是开门状态, 27 是床头柜, 28 是电视, 29 是书柜

//35 是马桶, 36 是瓷砖, 37 是洗手池, 39 是排水口, 40 是煤气灶

```
(*robot).electr=100; //初始化机器人信息
(*robot).px=10, (*robot).py=0;
(*robot).rt='d';
```

```
(*house).time=0;
```

```
(*house).set=0;
(*house).setd=0;
(*house).setc=0;
```

```
(*house).tempset=26;
(*house).tempout=20;
(*house).temp=26;
(*house).wet=50;
(*house).pm25=50;
```

```
for(i=0;i<5;i++)
    (*house).dor[i]=1;
//初始化房间信息
```

```
for(i=0;i<N;i++)
for(j=0;j<N;j++)
{
    (*house).mp1[i][j]=mp1init[i][j];
    (*house).mpinit[i][j]=mp1init[i][j];
}
}
```

```
void timepass(HOUSE *house, ROBOT *robot, USER *usr, int st)
{
    char *s;
    unsigned int *back[240][90];
    FILE *fpsv;
```

```

if((*house).time%timeupdate==0)
{
    draw_bactr(robot); //画 电池 电量
    write_statu(house,robot,st); //画出状态栏
}
if((*house).time%timeset==0)
{
    //pm25 变化
    if((*house).pm25>=setclean&&(*house).setc)
        (*house).pm25--;
    else if((*house).time%timedirt==0)
    {
        (*house).setc=0;
        (*house).pm25++;
    }
    //湿度变化
    if((*house).wet>=setwet&&(*house).setd)
        (*house).wet--;
    else if((*house).time%timedirt==0)
    {
        (*house).setd=0;
        (*house).wet++;
    }
}
if((*house).time%timetmpset==0)
{
    if((*house).set)
        (*house).temp-=sign((*house).temp-(*house).tempset);
    else if((*house).time%timetmp==0)
        (*house).temp-=sign((*house).temp-(*house).tempout);
}
if((*house).time%timeele==0)
    (*robot).electr--;
if((*robot).electr<=10) //触发自动充电模块
{
    fin(fpsv,"data\\mpsv.txt",LB+17,UB+60,LB+257,UB+510);

    draw_electr(usr);
}

```



```

        write_statu(house,robot,5);
        charge(house,robot,usr);

        fout(fpsv,"data\\mpsv.txt",LB+17,UB+60,LB+257,UB+510);
    }
    (*house).time%=timecut;
    ((*house).time)++;
}

```

```

void fin(FILE *fp, char *rd, int x1,int y1,int x2,int y2)
{
    int i,j;
    fp=fopen(rd,"w");
    for(i=x1;i<=x2;i++)
        for(j=y1;j<=y2;j++)
            fprintf(fp,"%u\n",Getpixel64k(i,j));
    fclose(fp);
    return ;
}

```

```

void fout(FILE *fp, char *rd, int x1,int y1,int x2,int y2)
{
    int i,j;
    unsigned int px;
    fp=fopen(rd,"r");
    for(i=x1;i<=x2;i++)
        for(j=y1;j<=y2;j++)
        {
            fscanf(fp,"%u",&px);
            Putpixel64k(i,j,px);
        }
    fclose(fp);
    return ;
}

```

6.3.3 board.c

核心程序前端函数。（孟繁鹏负责）

```

void draw_control(USER *usr)

```

```

{

    bar(LB,UB,1024,768,MARINE_BLUE);

    //画返回主菜单
    fill_rect(LB+140,UB+10,LB+250,UB+40,MISTY_ROSE,MISTY_ROSE);
    puthz(LB+145,UB+12,"返回菜单",24,25,'K',BLACK);

    //画状态显示栏
    fill_rect(LB+17,UB+60,LB+257,UB+320,MISTY_ROSE,MISTY_ROSE);
    puthz(LB+27,UB+95,"欢迎用户",32,32,'S',BLACK);
    outtextxy(LB+145,UB+95,(*usr).user,2,2,16,BLACK);
    puthz(LB+37,UB+160,"室内温度: ",24,25,'K',BLACK);
    puthz(LB+37,UB+195,"室外温度: ",24,25,'K',BLACK);
    puthz(LB+37,UB+230,"空气湿度: ",24,25,'K',BLACK);
    puthz(LB+37,UB+265,"污染指数: ",24,25,'K',BLACK);

    //画功能按钮
    fill_rect(LB+37,UB+350,LB+127,UB+390,MISTY_ROSE,MARINE_BLUE);
    puthz(LB+47,UB+354,"电量",32,35,'K',BLACK);
    fill_rect(LB+147,UB+350,LB+237,UB+390,MISTY_ROSE,MARINE_BLUE);
    puthz(LB+157,UB+354,"舒适",32,35,'K',BLACK);
    fill_rect(LB+37,UB+410,LB+127,UB+450,MISTY_ROSE,MARINE_BLUE);
    puthz(LB+47,UB+414,"环境",32,35,'K',BLACK);
    fill_rect(LB+147,UB+410,LB+237,UB+450,MISTY_ROSE,MARINE_BLUE);
    puthz(LB+157,UB+414,"控制",32,35,'K',BLACK);
    fill_rect(LB+57,UB+470,LB+217,UB+510,MISTY_ROSE,MARINE_BLUE);
    puthz(LB+71,UB+474,"人机互动",32,32,'K',BLACK);

    //画遥控按钮
    draw_conbot();
}

void draw_electr(USER *usr)
{
    bar(LB,UB,1024,768,MARINE_BLUE);

    //画返回主菜单

```

```

fill_rect(LB+140,UB+10,LB+250,UB+40,MISTY_ROSE,MISTY_ROSE);
puthz(LB+145,UB+12,"返回菜单",24,25,'K',BLACK);

//画状态显示栏
fill_rect(LB+17,UB+60,LB+257,UB+320,MISTY_ROSE,MISTY_ROSE);
puthz(LB+27,UB+95,"欢迎用户",32,32,'H',BLACK);
outtextxy(LB+145,UB+95,(*usr).user,2,2,16,BLACK);

//画功能按钮
fill_rect(LB+57,UB+350,LB+217,UB+390,MISTY_ROSE,MARINE_BLUE);
puthz(LB+71,UB+354,"手动充电",32,32,'K',BLACK);
fill_rect(LB+57,UB+410,LB+217,UB+450,MISTY_ROSE,MARINE_BLUE);
puthz(LB+71,UB+414,"停止充电",32,32,'K',BLACK);

//画返回按钮
fill_rect(LB+57,UB+470,LB+217,UB+510,MISTY_ROSE,MARINE_BLUE);
puthz(LB+71,UB+474,"返回主页",32,32,'K',BLACK);

//画遥控按钮
draw_conbot();
}

void draw_comfort(USER *usr)
{
    bar(LB,UB,1024,768,MARINE_BLUE);

    //画返回主菜单
    fill_rect(LB+140,UB+10,LB+250,UB+40,MISTY_ROSE,MISTY_ROSE);
    puthz(LB+145,UB+12,"返回菜单",24,25,'K',BLACK);

    //画状态显示栏
    fill_rect(LB+17,UB+60,LB+257,UB+320,MISTY_ROSE,MISTY_ROSE);
    puthz(LB+27,UB+95,"欢迎用户",32,32,'H',BLACK);
    outtextxy(LB+145,UB+95,(*usr).user,2,2,16,BLACK);

    puthz(LB+37,UB+160,"室内温度：",24,25,'K',BLACK);
    puthz(LB+37,UB+195,"室外温度：",24,25,'K',BLACK);
    puthz(LB+37,UB+230,"空气湿度：",24,25,'K',BLACK);

```

```

    puthz(LB+37,UB+265,"污染指数: ",24,25,'K',BLACK);

    //画功能按钮
    puthz(LB+27,UB+354,"温度",32,32,'K',WHITE);

    //画开关方框
    fill_rect(LB+110,UB+355,LB+140,UB+385,MISTY_ROSE,MARINE_BLUE);

    //画温度加减按钮
    fill_rect(LB+167,UB+360,LB+187,UB+380,MISTY_ROSE,MARINE_BLUE);
    fill_rect(LB+227,UB+360,LB+247,UB+380,MISTY_ROSE,MARINE_BLUE);
    outtextxy(LB+159,UB+353,"-",2,2,16,BLACK);
    outtextxy(LB+219,UB+353,"+",2,2,16,BLACK);

    //画功能按钮
    fill_rect(LB+37,UB+410,LB+127,UB+450,MISTY_ROSE,MARINE_BLUE);
    puthz(LB+47,UB+414,"除湿",32,35,'K',BLACK);
    fill_rect(LB+147,UB+410,LB+237,UB+450,MISTY_ROSE,MARINE_BLUE);
    puthz(LB+157,UB+414,"除尘",32,35,'K',BLACK);

    //画返回按钮
    fill_rect(LB+57,UB+470,LB+217,UB+510,MISTY_ROSE,MARINE_BLUE);
    puthz(LB+71,UB+474,"返回主页",32,32,'K',BLACK);

    //画遥控按钮
    draw_conbot();
}

void draw_move(USER *usr)
{
    bar(LB,UB,1024,768,MARINE_BLUE);

    //画返回主菜单
    fill_rect(LB+140,UB+10,LB+250,UB+40,MISTY_ROSE,MISTY_ROSE);
    puthz(LB+145,UB+12,"返回菜单",24,25,'K',BLACK);

    //画状态显示栏
    fill_rect(LB+17,UB+60,LB+257,UB+320,MISTY_ROSE,MISTY_ROSE);

```

```

puthz(LB+27,UB+95,"欢迎用户",32,32,'H',BLACK);
outtextxy(LB+145,UB+95,(*usr).user,2,2,16,BLACK);

//画功能按钮
fill_rect(LB+57,UB+350,LB+217,UB+390,MISTY_ROSE,MARINE_BLUE);
puthz(LB+71,UB+354,"自动巡逻",32,32,'K',BLACK);

//puthz(LB+47,UB+400,"门",24,24,'K',BLACK);
puthz(LB+47,UB+430,"门",24,24,'K',WHITE);

/*fill_rect(LB+80,UB+400,LB+100,UB+420,MISTY_ROSE,MARINE_BLUE);
fill_rect(LB+110,UB+400,LB+130,UB+420,MISTY_ROSE,MARINE_BLUE);
fill_rect(LB+140,UB+400,LB+160,UB+420,MISTY_ROSE,MARINE_BLUE);
fill_rect(LB+170,UB+400,LB+190,UB+420,MISTY_ROSE,MARINE_BLUE);
fill_rect(LB+200,UB+400,LB+220,UB+420,MISTY_ROSE,MARINE_BLUE);*/
outtextxy(LB+82,UB+410,"1",1,1,16,WHITE);
outtextxy(LB+112,UB+410,"2",1,1,16,WHITE);
outtextxy(LB+142,UB+410,"3",1,1,16,WHITE);
outtextxy(LB+172,UB+410,"4",1,1,16,WHITE);
outtextxy(LB+202,UB+410,"5",1,1,16,WHITE);

fill_rect(LB+80,UB+435,LB+100,UB+455,MISTY_ROSE,MARINE_BLUE);
fill_rect(LB+110,UB+435,LB+130,UB+455,MISTY_ROSE,MARINE_BLUE);
fill_rect(LB+140,UB+435,LB+160,UB+455,MISTY_ROSE,MARINE_BLUE);
fill_rect(LB+170,UB+435,LB+190,UB+455,MISTY_ROSE,MARINE_BLUE);
fill_rect(LB+200,UB+435,LB+220,UB+455,MISTY_ROSE,MARINE_BLUE);

//画返回按钮
fill_rect(LB+57,UB+470,LB+217,UB+510,MISTY_ROSE,MARINE_BLUE);
puthz(LB+71,UB+474,"返回主页",32,32,'K',BLACK);

//画遥控按钮
draw_conbot();
}

void draw_clean(USER *usr)
{
    bar(LB,UB,1024,768,MARINE_BLUE);

```

```

//画返回主菜单
fill_rect(LB+140,UB+10,LB+250,UB+40,MISTY_ROSE,MISTY_ROSE);
puthz(LB+145,UB+12,"返回菜单",24,25,'K',BLACK);

//画状态显示栏
fill_rect(LB+17,UB+60,LB+257,UB+320,MISTY_ROSE,MISTY_ROSE);
puthz(LB+27,UB+95,"欢迎用户",32,32,'H',BLACK);
outtextxy(LB+145,UB+95,(*usr).user,2,2,16,BLACK);

puthz(LB+37,UB+160,"地面垃圾： ",24,25,'K',BLACK);
//puthz(LB+37,UB+195,"桶内垃圾： ",24,25,'K',BLACK);

//画功能按钮
fill_rect(LB+57,UB+350,LB+217,UB+390,MISTY_ROSE,MARINE_BLUE);
puthz(LB+71,UB+354,"生成垃圾",32,32,'K',BLACK);
fill_rect(LB+57,UB+410,LB+217,UB+450,MISTY_ROSE,MARINE_BLUE);
puthz(LB+71,UB+414,"拾取垃圾",32,32,'K',BLACK);
fill_rect(LB+57,UB+470,LB+217,UB+510,MISTY_ROSE,MARINE_BLUE);
puthz(LB+71,UB+474,"返回主页",32,32,'K',BLACK);

//画遥控按钮
draw_conbot();
}

void draw_react(USER *usr,int mode)
{
    bar(LB,UB,1024,768,MARINE_BLUE);

    //画返回主菜单
    fill_rect(LB+140,UB+10,LB+250,UB+40,MISTY_ROSE,MISTY_ROSE);
    puthz(LB+145,UB+12,"返回菜单",24,25,'K',BLACK);

    //画状态显示栏
    fill_rect(LB+17,UB+60,LB+257,UB+380,MISTY_ROSE,MISTY_ROSE);
    if(mode==0)
    {
        puthz(LB+27,UB+95,"欢迎用户",32,32,'H',BLACK);
    }
}

```

```

        outtextxy(LB+145,UB+95,(*usr).user,2,2,16,BLACK);
    }

    //画功能按钮
    fill_rect(LB+37,UB+410,LB+127,UB+450,MISTY_ROSE,MARINE_BLUE);
    if(mode==1) puthz(LB+47,UB+414,"聊天",32,35,'K',STRONG_RED);
    else puthz(LB+47,UB+414,"聊天",32,35,'K',BLACK);
    fill_rect(LB+147,UB+410,LB+237,UB+450,MISTY_ROSE,MARINE_BLUE);
    if(mode==2) puthz(LB+157,UB+414,"字谜",32,35,'K',STRONG_RED);
    else puthz(LB+157,UB+414,"字谜",32,35,'K',BLACK);
    fill_rect(LB+57,UB+470,LB+217,UB+510,MISTY_ROSE,MARINE_BLUE);
    puthz(LB+71,UB+474,"返回主页",32,32,'K',BLACK);

    //画遥控按钮
    draw_conbot();
}

void draw_conbot()
{
    //midx-137 midy-700
    int cx,cy;

    fill_circle(LB+92,UB+630,30,MISTY_ROSE,MARINE_BLUE);
    fill_circle(LB+182,UB+630,30,MISTY_ROSE,MARINE_BLUE);
    fill_circle(LB+137,UB+585,30,MISTY_ROSE,MARINE_BLUE);
    fill_circle(LB+137,UB+675,30,MISTY_ROSE,MARINE_BLUE);

    cx=LB+137, cy=UB+585;
    thick_line(cx-20,cy-5,cx,cy-20,4,MARINE_BLUE);
    thick_line(cx,cy-20,cx+20,cy-5,4,MARINE_BLUE);
    thick_line(cx-5,cy-5,cx-20,cy-5,4,MARINE_BLUE);
    thick_line(cx+5,cy-5,cx+20,cy-5,4,MARINE_BLUE);
    thick_line(cx-5,cy-5,cx-5,cy+20,4,MARINE_BLUE);
    thick_line(cx+5,cy-5,cx+5,cy+20,4,MARINE_BLUE);
    thick_line(cx-5,cy+20,cx+5,cy+20,4,MARINE_BLUE);

    cx=LB+137, cy=UB+675;
    thick_line(cx-20,cy+5,cx,cy+20,4,MARINE_BLUE);

```

```

    thick_line(cx,cy+20,cx+20,cy+5,4,MARINE_BLUE);
    thick_line(cx-5,cy+5,cx-20,cy+5,4,MARINE_BLUE);
    thick_line(cx+5,cy+5,cx+20,cy+5,4,MARINE_BLUE);
    thick_line(cx-5,cy+5,cx-5,cy-20,4,MARINE_BLUE);
    thick_line(cx+5,cy+5,cx+5,cy-20,4,MARINE_BLUE);
    thick_line(cx-5,cy-20,cx+5,cy-20,4,MARINE_BLUE);

    cx=LB+92, cy=UB+630;
    thick_line(cx-20,cy,cx-5,cy+20,4,MARINE_BLUE);
    thick_line(cx-20,cy,cx-5,cy-20,4,MARINE_BLUE);
    thick_line(cx-5,cy+5,cx-5,cy+20,4,MARINE_BLUE);
    thick_line(cx-5,cy-5,cx-5,cy-20,4,MARINE_BLUE);
    thick_line(cx-5,cy-5,cx+20,cy-5,4,MARINE_BLUE);
    thick_line(cx-5,cy+5,cx+20,cy+5,4,MARINE_BLUE);
    thick_line(cx+20,cy-5,cx+20,cy+5,4,MARINE_BLUE);

    cx=LB+182, cy=UB+630;
    thick_line(cx+5,cy+20,cx+20,cy,4,MARINE_BLUE);
    thick_line(cx+5,cy-20,cx+20,cy,4,MARINE_BLUE);
    thick_line(cx+5,cy+5,cx+5,cy+20,4,MARINE_BLUE);
    thick_line(cx+5,cy-5,cx+5,cy-20,4,MARINE_BLUE);
    thick_line(cx+5,cy-5,cx-20,cy-5,4,MARINE_BLUE);
    thick_line(cx+5,cy+5,cx-20,cy+5,4,MARINE_BLUE);
    thick_line(cx-20,cy-5,cx-20,cy+5,4,MARINE_BLUE);

}

void draw_bactr(ROBOT *robot)
{
    char s[5];
    int len;
    bar(LB,UB,LB+130,UB+60,MARINE_BLUE);
    fill_rect(LB+21,UB+10,LB+70,UB+40,MISTY_ROSE,MARINE_BLUE);
    fill_rect(LB+70,UB+15,LB+75,UB+35,MISTY_ROSE,MISTY_ROSE);
    itoa((*robot).electr,s,10);
    len=strlen(s);
    s[len]='%';
    s[len+1]='\0';

```



```

    outtextxy(LB+80,UB+10,s,1,2,10,WHITE);
    if((*robot).electr<25)
fill_rect(LB+23,UB+12,LB+32,UB+38,STRONG_RED,BRIGHT_GREEN);
    else fill_rect(LB+23,UB+12,LB+32,UB+38,BRIGHT_GREEN,BRIGHT_GREEN);
    if((*robot).electr>=25)
fill_rect(LB+35,UB+12,LB+44,UB+38,BRIGHT_GREEN,BRIGHT_GREEN);
    if((*robot).electr>=50)
fill_rect(LB+47,UB+12,LB+56,UB+38,BRIGHT_GREEN,BRIGHT_GREEN);
    if((*robot).electr>=75)
fill_rect(LB+59,UB+12,LB+68,UB+38,BRIGHT_GREEN,BRIGHT_GREEN);
}

```

```

void write_statu(HOUSE *house, ROBOT *robot,int st)
{
    int LBx=LB+110, LBy=UB+355;
    int color;
    char s1[5];
    if(st==1||st==2)
    {
        char s2[5],s3[5],s4[8];
        itoa((*house).temp,s1,10);
        itoa((*house).tempout,s2,10);
        itoa((*house).wet,s3,10);
        itoa((*house).pm25,s4,10);

        bar(LB+150,UB+150,LB+210,UB+300,MISTY_ROSE);
        outtextxy(LB+150,UB+155,s1,2,2,16,BLACK);
        outtextxy(LB+150,UB+190,s2,2,2,16,BLACK);
        outtextxy(LB+150,UB+225,s3,2,2,16,BLACK);
        outtextxy(LB+150,UB+260,s4,2,2,16,BLACK);
        outtextxy(LB+190,UB+155,"C",2,2,16,BLACK);
        outtextxy(LB+190,UB+190,"C",2,2,16,BLACK);
        outtextxy(LB+190,UB+225,"%",2,2,16,BLACK);
        outtextxy(LB+190,UB+260,"%",2,2,16,BLACK);
        drcircle(LB+197,UB+155,3,BLACK);
        drcircle(LB+197,UB+190,3,BLACK);
        if(st==2)
        {

```

```

        char s5[3];
        itoa((*house).tempset,s5,10);
        bar(LB+190,UB+360,LB+230,UB+390,MARINE_BLUE);

        if((*house).set==1) color=STRONG_RED;
        else color=BLACK;
        //LB+110,UB+355,LB+140,UB+385,
        bar(LBx+14,LBy+2,LBx+16,LBy+15,color);
        bar(LBx+5,LBy+5,LBx+12,LBy+7,color);
        bar(LBx+18,LBy+5,LBx+25,LBy+7,color);
        bar(LBx+5,LBy+5,LBx+7,LBy+25,color);
        bar(LBx+23,LBy+5,LBx+25,LBy+25,color);
        bar(LBx+5,LBy+23,LBx+25,LBy+25,color);

        outtextxy(LB+182,UB+354,s5,2,2,16,WHITE);
    }
    return ;
}
if(st==4)
{
    fill_rect(LB+80,UB+435,LB+100,UB+455,MISTY_ROSE,MARINE_BLUE);
    fill_rect(LB+110,UB+435,LB+130,UB+455,MISTY_ROSE,MARINE_BLUE);
    fill_rect(LB+140,UB+435,LB+160,UB+455,MISTY_ROSE,MARINE_BLUE);
    fill_rect(LB+170,UB+435,LB+190,UB+455,MISTY_ROSE,MARINE_BLUE);
    fill_rect(LB+200,UB+435,LB+220,UB+455,MISTY_ROSE,MARINE_BLUE);

    if(!(*house).dor[0]) puthz(LB+82,UB+437,"开",16,16,'K',BLACK);
    else puthz(LB+82,UB+437,"关",16,16,'K',BLACK);
    if(!(*house).dor[1]) puthz(LB+112,UB+437,"开",16,16,'K',BLACK);
    else puthz(LB+112,UB+437,"关",16,16,'K',BLACK);
    if(!(*house).dor[2]) puthz(LB+142,UB+437,"开",16,16,'K',BLACK);
    else puthz(LB+142,UB+437,"关",16,16,'K',BLACK);
    if(!(*house).dor[3]) puthz(LB+172,UB+437,"开",16,16,'K',BLACK);
    else puthz(LB+172,UB+437,"关",16,16,'K',BLACK);
    if(!(*house).dor[4]) puthz(LB+202,UB+437,"开",16,16,'K',BLACK);
    else puthz(LB+202,UB+437,"关",16,16,'K',BLACK);
}
if(st==5)

```

```

    {
        bar(LB+17,UB+180,LB+257,UB+320,MISTY_ROSE);
        bar(LB+77,UB+200,LB+197,UB+300,WHITE);
        bar(LB+107,UB+180,LB+167,UB+200,WHITE);

        if((*robot).electr>=25)
fill_rect(LB+87,UB+204,LB+187,UB+224,BRIGHT_GREEN,WHITE);
        if((*robot).electr>=50)
fill_rect(LB+87,UB+228,LB+187,UB+248,BRIGHT_GREEN,WHITE);
        if((*robot).electr>=75)
fill_rect(LB+87,UB+252,LB+187,UB+272,BRIGHT_GREEN,WHITE);
        if((*robot).electr<25)
fill_rect(LB+87,UB+276,LB+187,UB+296,STRONG_RED,WHITE);
        else fill_rect(LB+87,UB+276,LB+187,UB+296,BRIGHT_GREEN,WHITE);

    }
    if(st==6)
    {
        bar(LB+150,UB+155,LB+185,UB+190,MISTY_ROSE);
        itoa((*house).rubnum,s1,10);
        outtextxy(LB+150,UB+155,s1,2,2,16,BLACK);
    }
}

void draw_cantgo()
{
    unsigned int *back1[250][40];
    get_image(LB+17,UB+90,LB+257,UB+130,*back1);
    fill_rect(LB+17,UB+90,LB+257,UB+130,MISTY_ROSE,MISTY_ROSE);
    puthz(LB+67,UB+94,"此路不通",32,35,'K',STRONG_RED);
    Delaytime(3000);
    put_image(LB+17,UB+90,LB+257,UB+130,*back1);
    return ;
}

```

6.3.4 drawmap.c

绘制地图。（黄瀚文负责）

```
#include "headers.h"
```

```

#define SIZE 40
#define WIDE 2
#define WALL_WIDE 4

void paintmp(HOUSE *house, ROBOT *robot)
{
    int i,j;
    int flag_bed=0;
    int tx=24,ty=15,sz=40;
    int cx1,cy1,cx2,cy2;

    for(i=0;i<N;i++)
    for(j=0;j<N;j++)
    {
        cy1=tx+i*sz, cx1=ty+j*sz;
        cy2=cy1+sz, cx2=cx1+sz;

        switch((*house).mp1[i][j])
        {
            case 0:
                draw_floor_wood(cx1,cy1);
                break;
            case 2:
                bar(cx1,cy1,cx2,cy2,BLACK);
                break;
            case 3:
                draw_bin(cx1,cy1);
                break;
            case 4:
                draw_floor_wood(cx1,cy1);
                fill_circle((cx1+cx2)/2,(cy1+cy2)/2,20,PEACH_PUFF,BLACK);
                bar(cx1,cy1,cx2,cy1+8,DARK_GRAY);
                break;
            case 5:
                bar(cx1,cy1,cx2,cy2,WHEAT);
                break;
            case 6:

```

```

        bar(cx1,cy1,cx2,cy2,LIGHT_GRAY);
        break;
case 7:
    bar(cx1,cy1,cx2,cy2,PINK);
    break;
case 8:
    bar(cx1,cy1,cx1+2,cy2,BLACK);
    bar(cx1+2,cy1,cx2,cy2,PINK);
    break;
case 9:
    draw_floor_wood(cx1,cy1);
    draw_door_vertical(cx1,cy1,1);
    break;
case 10:
    draw_door_left(cx1,cy1);
    break;
case 11:
    draw_door_right(cx1,cy1);
    break;
case 12:
    bar(cx1,cy1,cx2,cy2,PINK);
    bar(cx1,cy1+8,cx2-8,cy2,WHEAT);
    break;
case 13:
    bar(cx1,cy1,cx2,cy2,PINK);
    bar(cx1,cy1,cx2-8,cy1+12,WHEAT);
    bar(cx1,cy1+28,cx2-8,cy2,WHEAT);
    break;
case 14:
    bar(cx1,cy1,cx2,cy2,PINK);
    bar(cx1,cy1,cx2-8,cy2-8,WHEAT);
    break;
case 15:
    bar(cx1,cy1,cx2,cy2,PINK);
    bar(cx1,cy1+19,cx2,cy2-19,BLACK);
    bar(cx1,cy1,cx1+2,cy2,BLACK);
    break;
case 16:

```

```

        bar(cx1,cy1,cx2,cy2,PINK);
        bar(cx1+21,cy1,cx2,cy2,WHEAT);
        bar(cx1,cy1+19,cx2,cy2-19,BLACK);
        bar(cx1+19,cy1,cx2-19,cy2,BLACK);
        break;
case 17:
        bar(cx1,cy1,cx2,cy2,PINK);
        bar(cx1+19,cy1,cx2-19,cy2,BLACK);
        bar(cx1+21,cy1,cx2,cy2,WHEAT);
        break;
case 18:
        bar(cx1,cy1,cx2,cy2,WHEAT);
        bar(cx1,cy1+8,cx2-8,cy1+10,BLACK);
        bar(cx2-10,cy1+10,cx2-8,cy2,BLACK);
        bar(cx1,cy1+10,cx2-10,cy2,PINK);
        bar(cx1,cy1,cx1+2,cy2,BLACK);
        break;
case 19:
        bar(cx1,cy1,cx2,cy2,WHEAT);
        bar(cx2-10,cy1,cx2-8,cy2-8,BLACK);
        bar(cx1,cy2-10,cx2-10,cy2-8,BLACK);
        bar(cx1,cy1,cx2-10,cy2-10,PINK);
        bar(cx1,cy1,cx1+2,cy2,BLACK);
        break;
case 20:
        bar(cx1,cy1,cx2,cy2,PINK);
        bar(cx1,cy1,cx1+2,cy2,BLACK);
        break;
case 21:
        draw_charge(cx1,cy1);
        break;
case 22:
        draw_floor_wood(cx1,cy1);
        draw_rub(cx1,cy1);
        break;
case 23:
        draw_floor_wood(cx1,cy1);
        open_up_door(cx1,cy1);

```

```

        break;
case 24:
    draw_floor_wood(cx1,cy1);
    open_down_door(cx1,cy1);
    break;
case 25:
    draw_floor_wood(cx1,cy1);
    open_left_door(cx1,cy1);
    break;
case 26:
    draw_floor_wood(cx1,cy1);
    open_right_door(cx1,cy1);
    break;
case 27:
    draw_floor_wood(cx1,cy1);
    fill_circle(cx1+20,cy1+20,20,WHEAT,BLACK);
    break;
case 28:
    draw_floor_wood(cx1,cy1);
    bar(cx1,cy1,cx1+10,cy2,DARK_GRAY);
    bar(cx1+10,cy1,cx1+20,cy2,BLACK);
    break;
case 29:
    bar(cx1,cy1,cx2,cy2,WHEAT);
    break;
case 30:
    draw_floor_wood(cx1,cy1);
    fill_circle((cx1+cx2)/2,(cy1+cy2)/2,20,PEACH_PUFF,BLACK);
    bar(cx1,cy2-8,cx2,cy2,DARK_GRAY);
    break;
case 31:
    bar(cx1,cy1,cx2,cy2,WHEAT);
    bar(cx1+8,cy1+8,cx1+10,cy2,BLACK);
    bar(cx1+10,cy1+8,cx2,cy1+10,BLACK);
    bar(cx1+10,cy1+10,cx2,cy2,PINK);
    bar(cx1,cy2-1,cx2,cy2,BLACK);
    break;
case 32:

```

```

        bar(cx1,cy1,cx2,cy2,WHEAT);
        bar(cx1,cy1+8,cx2-8,cy1+10,BLACK);
        bar(cx2-10,cy1+8,cx2-8,cy2,BLACK);
        bar(cx1,cy1+10,cx2-10,cy2,PINK);
        bar(cx1,cy2-1,cx2,cy2,BLACK);
        break;
case 33:
        bar(cx1,cy1,cx2,cy2,WHEAT);
        bar(cx1+8,cy1,cx1+10,cy2-8,BLACK);
        bar(cx1+8,cy2-10,cx2,cy2-8,BLACK);
        bar(cx1+10,cy1,cx2,cy2-10,PINK);
        bar(cx1,cy1,cx2,cy1+1,BLACK);
        break;
case 34:
        bar(cx1,cy1,cx2,cy2,WHEAT);
        bar(cx1,cy2-10,cx2-8,cy2-8,BLACK);
        bar(cx2-10,cy1,cx2-8,cy2-10,BLACK);
        bar(cx1,cy1,cx2-10,cy2-10,PINK);
        bar(cx1,cy1,cx2,cy1+1,BLACK);
        break;
case 35:
        draw_floor_ceramic(cx1,cy1);
        draw_toilet(cx1,cy1);
        break;
case 36:
        draw_floor_ceramic(cx1,cy1);
        break;
case 37:
        bar(cx1,cy1,cx2,cy2,LIGHT_GRAY);
        draw_wash(cx1,cy1);
        break;
case 38:
        bar(cx1,cy1,cx2,cy2,WHEAT);
        bar(cx1,cy1,cx1+10,cy2,DARK_GRAY);
        bar(cx1+10,cy1,cx1+20,cy2,BLACK);
        break;
case 39:
        draw_scupper(cx1,cy1);

```



```

        break;
    case 40:
        bar(cx1,cy1,cx2,cy2,LIGHT_GRAY);
        draw_cooker(cx1,cy1);
        break;
    case 41:
        bar(cx1,cy1,cx2,cy2,LIGHT_GRAY);
        fill_circle(cx1+20,cy1+20,18,WHITE,BLACK);
        fill_circle(cx1+20,cy1+15,10,LIGHT_GRAY,LIGHT_GRAY);
        fill_rect(cx1,cy1,cx2,cy1+7,WHITE,BLACK);
        fill_rect(cx1+16,cy1,cx2-16,cy1+20,DARK_GRAY,BLACK);
        break;
    default: break;
}

}

bar(0,0,15,768,MISTY_ROSE);
bar(735,0,750,768,MISTY_ROSE);
bar(0,0,750,24,MISTY_ROSE);
bar(0,744,750,768,MISTY_ROSE);
cy1=tx+(*robot).px*sz, cx1=ty+(*robot).py*sz;
cy2=cy1+sz, cx2=cx1+sz;
switch((*robot).rt)
{
    case 'u': drawrobot_back((cx1+cx2)/2,(cy1+cy2)/2,1); break;
    case 'd': drawrobot_front((cx1+cx2)/2,(cy1+cy2)/2,1); break;
    case 'l': drawrobot_left((cx1+cx2)/2,(cy1+cy2)/2,1); break;
    case 'r': drawrobot_right((cx1+cx2)/2,(cy1+cy2)/2,1); break;
    default: break;
}
}

void draw_floor_wood(int x,int y)
{
    bar(x,y,x+SIZE,y+SIZE,BROWN);
    bar(x,y+10,x+SIZE,y+20,DARK_RED);
    bar(x,y+30,x+SIZE,y+SIZE,DARK_RED);
}

```

```

void draw_floor_ceramic(int x,int y)
{
    fill_rect(x,y,x+SIZE,y+SIZE,LIGHT_BLUE,BULE);
}

void draw_table(int x,int y)
{
    bar(x,y,x+SIZE,y+SIZE,WHEAT);
}

void draw_chair(int x,int y)
{
    fill_circle(x+20,y+20,19,WHEAT,BLACK);
}

void draw_door_left(int x,int y)
{
    fill_rect(x,y,x+SIZE,y+SIZE,PINK,BLACK);
    fill_circle(x+30,y+20,4,BLACK,BLACK);
}

void draw_door_right(int x,int y)
{
    fill_rect(x,y,x+SIZE,y+SIZE,PINK,BLACK);
    fill_circle(x+8,y+20,4,BLACK,BLACK);
}

void draw_door_vertical(int x,int y,int n)
{
    fill_rect(x,y,x+2*WALL_WIDE,y+n*SIZE,PINK,BLACK);
}

void draw_else(int x,int y)
{
    bar(x,y,x+SIZE,y+SIZE,BLACK);
}

void draw_bin(int x,int y)

```

```

{
    bar(x,y,x+SIZE,y+SIZE,BLACK);
    bar(x+13,y,x+27,y+5,DARK_GRAY);
    bar(x,y+5,x+SIZE,y+15,DARK_GRAY);
    bar(x+5,y+15,x+35,y+SIZE,LIGHT_GRAY);
    bar(x+7,y+20,x+13,y+SIZE-5,DARK_GRAY);
    bar(x+17,y+20,x+23,y+SIZE-5,DARK_GRAY);
    bar(x+27,y+20,x+33,y+SIZE-5,DARK_GRAY);
    fill_circle(x+10,y+20,3,DARK_GRAY,DARK_GRAY);
    fill_circle(x+20,y+20,3,DARK_GRAY,DARK_GRAY);
    fill_circle(x+30,y+20,3,DARK_GRAY,DARK_GRAY);
}

```

```
void draw_charge(int x,int y)
```

```

{
    bar(x,y,x+SIZE,y+SIZE,LIGHT_GRAY);
    bar(x+18,y+5,x+22,y+15,BLACK);
    bar(x+5,y+27,x+13,y+30,BLACK);
    bar(x+27,y+27,x+35,y+30,BLACK);
}

```

```
void open_left_door(int x,int y)
```

```

{
    fill_rect(x,y,x+2*WALL_WIDE,y+SIZE,PINK,BLACK);
}

```

```
void open_right_door(int x,int y)
```

```

{
    fill_rect(x+SIZE-2*WALL_WIDE,y,x+SIZE,y+SIZE,PINK,BLACK);
}

```

```
void open_up_door(int x,int y)
```

```

{
    fill_rect(x,y,x+SIZE,y+2*WALL_WIDE,PINK,BLACK);
}

```

```
void open_down_door(int x,int y)
```

```

{

```

```

        fill_rect(x,y+SIZE-2*WALL_WIDE,x+SIZE,y+SIZE,PINK,BLACK);
    }

```

```

void draw_rub(int x,int y)

```

```

{
    fill_rect(x,y+10,x+SIZE,y+SIZE,MARINE_BLUE,BLACK);
    bar(x+17,y+7,x+22,y+9,BLACK);
    bar(x+16,y+6,x+23,y+6,MARINE_BLUE);
    bar(x+15,y+5,x+24,y+5,MARINE_BLUE);
    bar(x+14,y+4,x+25,y+4,MARINE_BLUE);
    bar(x+13,y+3,x+26,y+3,MARINE_BLUE);
    bar(x+12,y+2,x+27,y+2,MARINE_BLUE);
    bar(x+11,y+1,x+28,y+1,MARINE_BLUE);
    bar(x+10,y,x+29,y,MARINE_BLUE);
}

```

```

void draw_toilet(int x,int y)

```

```

{
    fill_circle(x+20,y+20,20,WHITE,BLACK);
    fill_circle(x+20,y+15,10,LIGHT_GRAY,LIGHT_GRAY);
    fill_rect(x,y,x+SIZE,y+10,WHITE,BLACK);
}

```

```

void draw_wash(int x,int y)

```

```

{
    fill_circle(x+20,y+20,20,WHITE,BLACK);
    fill_circle(x+25,y+20,10,LIGHT_GRAY,LIGHT_GRAY);
    fill_rect(x+SIZE-7,y,x+SIZE,y+SIZE,WHITE,BLACK);
    fill_rect(x+20,y+16,x+SIZE,y+24,DARK_GRAY,BLACK);
}

```

```

void draw_scupper(int x,int y)

```

```

{
    bar(x,y,x+SIZE,y+SIZE,BLACK);
    bar(x,y+8,x+SIZE,y+16,DARK_GRAY);
    bar(x,y+24,x+SIZE,y+32,DARK_GRAY);
    bar(x+8,y,x+16,y+SIZE,DARK_GRAY);
    bar(x+24,y,x+32,y+SIZE,DARK_GRAY);
}

```

```

}

void draw_cooker(int x,int y)
{
    fill_circle(x+20,y+20,20,BLACK,BLACK);
    fill_circle(x+20,y+20,17,DARK_GRAY,BLACK);
    fill_circle(x+20,y+20,10,BLACK,BLACK);
    fill_circle(x+20,y+20,7,STRONG_RED,BLACK);
    bar(x,y+17,x+10,y+23,BLACK);
    bar(x+SIZE-10,y+17,x+SIZE,y+23,BLACK);
    bar(x+17,y,x+23,y+10,BLACK);
    bar(x+17,y+SIZE-10,x+23,y+SIZE,BLACK);
}

```

6.3.5 electr.c

电量功能函数。（黄瀚文负责）

```
#include "headers.h"
```

```

int func_electr(HOUSE *house, ROBOT *robot, USER *usr)
{
    char value;
    int poscode;
    NODE mp,mto; //鼠标点击后行动坐标

    draw_bactr(robot);
    write_statu(house,robot,5);

    while(1)
    {
        newmouse(&MouseX, &MouseY, &press);
        timepass(house,robot,usr,5);
        if(mouse_press(LB+57,UB+350,LB+217,UB+390)==1) //手动充电
        {
            nocombo();
            charge(house,robot,usr);
            continue;
        }
        if(mouse_press(LB+57,UB+470,LB+217,UB+510)==1) //返回主菜单
    }
}

```

```

    {
        nocombo();
        return 0;
    }
    if(kbhit())
    {
        Delaytime(50);
        value=getch();
        moveupdate(house,robot,value);
    }
    get_conbot(house,robot);
    if(mouse_press(15,24,735,744)==1)
    {
        nocombo();
        poscode=getposition(MouseX, MouseY);
        mto.x=poscode/18;
        mto.y=poscode%18;
        mp.x=(*robot).px;
        mp.y=(*robot).py;
        if(!Astartmove(mp,mto,robot,house))
        {
            draw_cantgo();
        }
    }
    if(mouse_press(LB+140,UB+10,LB+250,UB+40)==1)
        return 1;
}
}

```

```

void charge(HOUSE *house, ROBOT *robot, USER *usr)
{
    NODE mp;
    NODE mp1;
    NODE charge_point;
    int i,t;

    //移动到对应位置
    mp.x=(*robot).px;

```

```

mp.y=(*robot).py;
charge_point.x=17;
charge_point.y=0;
Astartmove(mp,charge_point,robot,house);

t=5000/(100-(*robot).electr);
puthz(LB+87,UB+135,"充电中",32,32,'S',BLACK);
for(i=0;i<5000;i++)
{
    newmouse(&MouseX, &MouseY, &press);
    Delaytime(1);
    if(i%t==0)
    {
        (*robot).electr++;
        if((*robot).electr>100) (*robot).electr=100;
    }
    if(i%100==0) draw_bactr(robot);
    if(i%500==0)
    {
        bar(LB+77,UB+200,LB+197,UB+300,WHITE);
        if((i/500)%5>=4)
fill_rect(LB+87,UB+204,LB+187,UB+224,BRIGHT_GREEN,WHITE);
        if((i/500)%5>=3)
fill_rect(LB+87,UB+228,LB+187,UB+248,BRIGHT_GREEN,WHITE);
        if((i/500)%5>=2)
fill_rect(LB+87,UB+252,LB+187,UB+272,BRIGHT_GREEN,WHITE);
        if((i/500)%5>=1)
fill_rect(LB+87,UB+276,LB+187,UB+296,BRIGHT_GREEN,WHITE);
    }
    if(mouse_press(LB+57,UB+410,LB+217,UB+450)==1)
    {
        bar(LB+17,UB+130,LB+258,UB+170,MISTY_ROSE);
        (*robot).electr=100;
        write_statu(house,robot,5);
        return ;
    }
}
}

```

```

    draw_bactr(robot);
    write_statu(house,robot,5);
    bar(LB+17,UB+130,LB+258,UB+170,MISTY_ROSE);
    (*robot).electr=100;
    mp1.x=(*robot).px;
    mp1.y=(*robot).py;
    Astarmove(mp1,mp,robot,house);
    return ;
}

```

6.3.6 rubbish.c

捡垃圾清洁功能函数。（黄瀚文负责）

```
#include "headers.h"
```

```

int func_clean(NODE *rubbish,HOUSE *house, ROBOT *robot, USER *usr)
{
    char value;
    int *f;
    int poscode;
    NODE mp,mto; //鼠标点击后行动坐标
    *f=1;

    draw_bactr(robot);
    write_statu(house,robot,6);

    while(1)
    {
        newmouse(&MouseX, &MouseY, &press);
        timepass(house,robot,usr,6);
        if(mouse_press(LB+57,UB+350,LB+217,UB+390)==1) //生成垃圾
        {
            nocombo();
            nocombo();
            if(house->rubnum<3)
            {
                (house->rubnum)++;
                set_rub(rubbish,house);
                paintmp(house,robot);
            }
        }
    }
}

```



```

    }
    continue;
}
if(mouse_press(LB+57,UB+410,LB+217,UB+450)==1) //拾倒垃圾
{
    nocombo();
    nocombo();
    while(1)
    {
        if(house->rubnum>0&&(*f)==1)
        {
            puthz(LB+37,UB+230,"正在拾取垃圾中。",24,25,'K',BLACK);
            col_rub(f,rubbish,house,robot);
            paintmp(house,robot);
            bar(LB+35,UB+230,LB+250,UB+260,MISTY_ROSE);
        }
        else
            break;
    }
    continue;
}
if(mouse_press(LB+57,UB+470,LB+217,UB+510)==1) //返回主界面
{
    nocombo();
    return 0;
}
if(kbhit())
{
    Delaytime(50);
    value=getch();
    moveupdate(house,robot,value);
}
get_conbot(house,robot);
if(mouse_press(15,24,735,744)==1)
{
    nocombo();
    poscode=getposition(MouseX, MouseY);
    mto.x=poscode/18;

```

```

        mto.y=poscode%18;
        mp.x=(*robot).px;
        mp.y=(*robot).py;
        if(!Astarmove(mp,mto,robot,house))
        {
            draw_cantgo();
        }
    }
    if(mouse_press(LB+140,UB+10,LB+250,UB+40)==1)
        return 1;
}
}

```

```

void set_rub(NODE *rubbish,HOUSE *house)

```

```

{
    int x,y;

    while(1)
    {
        x=randin(17);
        y=(randin(17)+23451)%17;
        if((*house).mp1[x][y]==0)
        {
            rubbish[house->rubnum].x=x;
            rubbish[house->rubnum].y=y;
            (*house).mp1[x][y]=22;
            (*house).mpinit[x][y]=22;

            /*draw_rub(pnum,rubbish);*/
            break;
        }
    }
}

```

```

}

```

```

void col_rub(int *f,NODE *rubbish,HOUSE *house,ROBOT *robot)

```

```

{

```

```

int x,y;
int i,j;
NODE mp;
NODE trash_bin;
trash_bin.x=0;
trash_bin.y=0;
x=rubbish[house->rubnum].x;
y=rubbish[house->rubnum].y;
mp.x=(*robot).px;
mp.y=(*robot).py;
*f=Astarmove(mp,rubbish[house->rubnum],robot,house);
if(*f==1)
{
    (*house).mp1[x][y]=0;
    (*house).mpinit[x][y]=0;
    mp.x=(*robot).px;
    mp.y=(*robot).py;
    if((house->rubnum)==1)
    {
        Astarmove(mp,trash_bin,robot,house);
    }
    (house->rubnum)--;
}
}

```

6.3.7 plan.c

实现 Astar 路径规划算法。（孟繁鹏负责）

```

/**heap 实现优先队列部分**/
void swapnode(NODE1 *a, NODE1 *b)
{
    NODE1 x=*a;
    *a=*b, *b=x;
}
void put(NODE1 *heap, int *size, NODE1 d)
{
    int now,next;
    heap[++(*size)]=d;
    now=(*size);
    while(now>1)

```

```

    {
        next=now>>1;
        if(heap[now].val>=heap[next].val) return;
        swapnode(&heap[now],&heap[next]);
        now=next;
    }
}
void del(NODE1 *heap, int *size)
{
    int now,next;
    heap[1]=heap[( *size)--];
    now=1;
    while(now*2<=( *size))
    {
        next=now*2;
        if(next<(*size)&&heap[next+1].val<heap[next].val) next++;
        if(heap[now].val<=heap[next].val) return;
        swapnode(&heap[now],&heap[next]);
        now=next;
    }
    return;
}
/****完成优先队列的实现****/

int val_func(int a,NODE beg,int x,int y) //估价函数
{
    return a+abs(beg.x-x)+abs(beg.y-y);
}

int BFS(NODE beg,NODE end,NODE1 *node,int sz,HOUSE *house)
{
    NODE1 heap[M];
    int hp_sz=0; //定义优先队列

    int vis_map[N][N]={0}; //地图访问情况
    int dir[4][2]={{0,1},{0,-1},{-1,0},{1,0}}; //方向数组
    int i,newx,newy;

    node[++sz].x=beg.x;
    node[sz].y=beg.y;
    node[sz].dis=0;
    node[sz].val=val_func(node[sz].dis,beg,node[sz].x,node[sz].y);
    node[sz].id=sz;
    node[sz].last=-1;

```

```

put(heap,&hp_sz,node[sz]);
while(hp_sz!=0)
{
    NODE1 hed=heap[1];
    del(heap,&hp_sz);
    vis_map[hed.x][hed.y]=1;
    for(i=0;i<4;i++)
    {
        newx=dir[i][0]+hed.x;
        newy=dir[i][1]+hed.y;
        if(newx>=0 && newx<N && newy>=0 && newy<N
        && !vis_map[newx][newy] && pd_pass((*house).mp1[newx][newy]))
        {
            vis_map[newx][newy]=1;
            node[++sz].x=newx;
            node[sz].y=newy;
            node[sz].dis=hed.dis+1;
            node[sz].last=hed.id;
            node[sz].id=sz;
            node[sz].val=val_func(node[sz].dis,beg,node[sz].x,node[sz].y);
            put(heap,&hp_sz,node[sz]);
            if(newx==end.x&&newy==end.y) return sz;
        }
    }
}
return -1;
}

```

```

void get_path(NODE *road, int *cnt, NODE1 *node, int x)
{
    if(node[x].last== -1)
    {
        road[++(*cnt)].x=node[x].x;
        road[*cnt].y=node[x].y;
        return;
    }
    get_path(road,cnt,node,node[x].last);
    road[++(*cnt)].x=node[x].x;
    road[*cnt].y=node[x].y;
}

```

```

int Astarmove(NODE beg, NODE end, ROBOT *robot, HOUSE *house)
{

```

```

NODE1 node[M];
int nd_sz=0; //定义结点记录数组
NODE road[M];
int rd_sz=0; //定义路径记录数组

char dir;
int i,j;

nd_sz=BFS(beg,end,node,0,house); //宽度优先搜索得到路径
if(nd_sz== -1) return 0;
get_path(road,&rd_sz,node,nd_sz); //规范化路径
for(i=2;i<=rd_sz;i++)
{
    /*for(j=0;j<=100;j++)
        timepass(house,robot,1);*/
    if(i%3==0) (*robot).electr--;
    if(road[i].x-road[i-1].x==1) dir='s';
    if(road[i].x-road[i-1].x==-1) dir='w';
    if(road[i].y-road[i-1].y==1) dir='d';
    if(road[i].y-road[i-1].y==-1) dir='a';
    Delaytime(100);
    moveupdate(house,robot,dir);
}

return 1;
}

```

6.3.8 move.c

移动与控制功能函数。（孟繁鹏负责）

```
#include "headers.h"
```

```

int func_move(HOUSE *house, ROBOT *robot, USER *usr)
{
    char value;
    int poscode;
    NODE mp,mto; //鼠标点击后行动坐标

    draw_bactr(robot);
    write_statu(house,robot,4);

    while(1)
    {

```

```

newmouse(&MouseX, &MouseY, &press);
timepass(house,robot,usr,4);
if(mouse_press(LB+57,UB+350,LB+217,UB+390)==1) //进入自由巡逻功能
{
    free_hang(house,robot);
    nocombo();
    continue;
}
ope_door(house,robot);
if(kbhit())
{
    Delaytime(50);
    value=getch();
    moveupdate(house,robot,value);
}
get_conbot(house,robot);
if(mouse_press(15,24,735,744)==1)
{
    nocombo();
    poscode=getposition(MouseX, MouseY);
    mto.x=poscode/18;
    mto.y=poscode%18;
    mp.x=(*robot).px;
    mp.y=(*robot).py;
    if(!Astarmove(mp,mto,robot,house))
    {
        draw_cantgo();
    }
}
if(mouse_press(LB+57,UB+470,LB+217,UB+510)==1) //返回主界面
{
    nocombo();
    return 0;
}
if(mouse_press(LB+140,UB+10,LB+250,UB+40)==1)
    return 1;
}
}

```

```

void move(HOUSE *house, ROBOT *robot,char dir) //移动机器人
{
    int dx,dy,nx,ny;
    switch(dir) //判断移动方向
    {
        case 'u': dx=-1; dy=0; break;
        case 'd': dx=1; dy=0; break;
        case 'l': dx=0; dy=-1; break;
        case 'r': dx=0; dy=1; break;
        default: break;
    }
    nx=(*robot).px+dx, ny=(*robot).py+dy;
    if(nx>=0 && nx<N && ny>=0 && ny<N &&
        pd_pass((*house).mp1[nx][ny]))
    {
        (*house).mp1[(*robot).px][(robot).py]=
            (*house).mpinit[(robot).px][(robot).py];
        (*robot).px=nx, (*robot).py=ny;
        (*house).mp1[(robot).px][(robot).py]=1;
    }
    (*robot).rt=dir;
}

void moveupdate(HOUSE *house,ROBOT *robot,char value) //移动更新函数
{
    if(value=='W'||value=='w') //向上运动
    {
        move(house,robot,'u'); //移动机器人
        paintmp(house,robot); //重新绘制地图
        clrmous(MouseX, MouseY);
        return;
    }
    else if(value=='S'||value=='s') //向下运动
    {
        move(house,robot,'d');
        paintmp(house,robot);
        clrmous(MouseX, MouseY);
    }
}

```



```

        return;
    }
    else if(value=='A' || value=='a') //向左运动
    {
        move(house,robot,'l');
        paintmp(house,robot);
        clrmous(MouseX, MouseY);
        return;
    }
    else if(value=='D' || value=='d') //向右运动
    {
        move(house,robot,'r');
        paintmp(house,robot);
        clrmous(MouseX, MouseY);
        return;
    }
}

```

```

int getposition(int x,int y)
{
    int by=24,bx=15,len=40,sn=18;
    int retcode;
    retcode=((y-by)/len)*sn+(x-bx)/len;
    return retcode;
}

```

```

int pd_pass(int mp)
{
    if(mp==0) return 1;
    if(mp==3) return 1;
    /*if(mp==9) return 1;
    if(mp==10) return 1;
    if(mp==11) return 1;*/
    if(mp==21) return 1;
    if(mp==22) return 1;
    if(mp==23) return 1;
    if(mp==24) return 1;
    if(mp==25) return 1;
}

```

```

    if(mp==26) return 1;
    if(mp==36) return 1;
    return 0;
}

void free_hang(HOUSE *house, ROBOT *robot)
{
    NODE pos[8];
    int n=8, beg=0, des=1;
    int flag=0;

    pos[0].x=(*robot).px, pos[0].y=(*robot).py;
    pos[1].x=1, pos[1].y=0;
    pos[2].x=7, pos[2].y=2;
    pos[3].x=15, pos[3].y=2;
    pos[4].x=3, pos[4].y=6;
    pos[5].x=7, pos[5].y=10;
    pos[6].x=1, pos[6].y=16;
    pos[7].x=15, pos[7].y=10;
    fill_rect(LB+17,UB+150,LB+257,UB+320,MISTY_ROSE,MISTY_ROSE);
    puthz(LB+37,UB+160,"自动巡逻中。。。",24,25,'K',BLACK);
    while(1)
    {
        if(des==1&&flag) break;
        if(!Astartmove(pos[beg],pos[des],robot,house))
        {
            des++;
            des%=n;
            continue;
        }
        Delaytime(1000);
        beg++, des++;
        flag=1;
    }
    write_statu(house,robot,3);
    fill_rect(LB+37,UB+160,LB+240,UB+190,MISTY_ROSE,MISTY_ROSE);
    return ;
}

```

```

void get_conbot(HOUSE *house, ROBOT *robot)
{
    char value;
    if(mouse_press_circle(LB+137,UB+585,30)==1)
    {
        Delaytime(50);
        value='W';
        moveupdate(house,robot,value);
        return ;
    }
    if(mouse_press_circle(LB+137,UB+675,30)==1)
    {
        Delaytime(50);
        value='S';
        moveupdate(house,robot,value);
        return ;
    }
    if(mouse_press_circle(LB+92,UB+630,30)==1)
    {
        Delaytime(50);
        value='A';
        moveupdate(house,robot,value);
        return ;
    }
    if(mouse_press_circle(LB+182,UB+630,30)==1)
    {
        Delaytime(50);
        value='D';
        moveupdate(house,robot,value);
        return ;
    }
    return ;
}

void ope_door(HOUSE *house, ROBOT *robot)
{
    if(mouse_press(LB+80,UB+435,LB+100,UB+455)==1)

```

```

{
    nocombo();
    if((*house).dor[0])
    {
        (*house).dor[0]=0;
        house->mp1[1][1]=23;
        house->mpinit[1][1]=23;
        house->mp1[2][1]=24;
        house->mpinit[2][1]=24;
        paintmp(house,robot);
    }
    else
    {
        (*house).dor[0]=1;
        house->mp1[1][1]=9;
        house->mpinit[1][1]=9;
        house->mp1[2][1]=9;
        house->mpinit[2][1]=9;
        paintmp(house,robot);
    }
    clrmous(MouseX, MouseY);
    write_statu(house,robot,4);
}
if(mouse_press(LB+110,UB+435,LB+130,UB+455)==1)
{
    nocombo();
    if((*house).dor[1])
    {
        (*house).dor[1]=0;
        house->mp1[2][4]=23;
        house->mpinit[2][4]=23;
        house->mp1[3][4]=24;
        house->mpinit[3][4]=24;
        paintmp(house,robot);
    }
    else
    {
        (*house).dor[1]=1;

```

```

        house->mp1[2][4]=9;
        house->mpinit[2][4]=9;
        house->mp1[3][4]=9;
        house->mpinit[3][4]=9;
        paintmp(house,robot);
    }
    clrmous(MouseX, MouseY);
    write_statu(house,robot,4);
}
if(mouse_press(LB+140,UB+435,LB+160,UB+455)==1)
{
    nocombo();
    if((*house).dor[2])
    {
        (*house).dor[2]=0;
        house->mp1[6][9]=25;
        house->mpinit[6][9]=25;
        house->mp1[6][10]=26;
        house->mpinit[6][10]=26;
        paintmp(house,robot);
    }
    else
    {
        (*house).dor[2]=1;
        house->mp1[6][9]=10;
        house->mpinit[6][9]=10;
        house->mp1[6][10]=11;
        house->mpinit[6][10]=11;
        paintmp(house,robot);
    }
    clrmous(MouseX, MouseY);
    write_statu(house,robot,4);
}
if(mouse_press(LB+170,UB+435,LB+190,UB+455)==1)
{
    nocombo();
    if((*house).dor[3])
    {

```

```

        (*house).dor[3]=0;
        house->mp1[7][13]=23;
        house->mpinit[7][13]=23;
        house->mp1[8][13]=24;
        house->mpinit[8][13]=24;
        paintmp(house,robot);
    }
    else
    {
        (*house).dor[3]=1;
        house->mp1[7][13]=9;
        house->mpinit[7][13]=9;
        house->mp1[8][13]=9;
        house->mpinit[8][13]=9;
        paintmp(house,robot);
    }
    clrmous(MouseX, MouseY);
    write_statu(house,robot,4);
}
if(mouse_press(LB+200,UB+435,LB+220,UB+455)==1)
{
    nocombo();
    if((*house).dor[4])
    {
        (*house).dor[4]=0;
        house->mp1[9][8]=25;
        house->mpinit[9][8]=25;
        house->mp1[9][9]=26;
        house->mpinit[9][9]=26;
        paintmp(house,robot);
    }
    else
    {
        (*house).dor[4]=1;
        house->mp1[9][8]=10;
        house->mpinit[9][8]=10;
        house->mp1[9][9]=11;
        house->mpinit[9][9]=11;
    }
}

```

```

        paintmp(house,robot);
    }
    clrmous(MouseX, MouseY);
    write_statu(house,robot,4);
}
}

```

• comfort.c

舒适度功能函数。（孟繁鹏负责）

6.3.9 comfort.c

舒适度功能函数。（孟繁鹏负责）

```
#include "headers.h"
```

```

int func_comfort(HOUSE *house, ROBOT *robot, USER *usr)
{
    char value;
    int poscode;
    NODE mp,mto; //鼠标点击后行动坐标

    draw_bactr(robot);
    write_statu(house,robot,2);

    while(1)
    {
        newmouse(&MouseX, &MouseY, &press);
        timepass(house,robot,usr,2);
        if(mouse_press(LB+110,UB+355,LB+140,UB+385)==1) //打开关闭空调开关
        {
            if((*house).set) (*house).set=0;
            else (*house).set=1;
            nocombo();
            write_statu(house,robot,2);
            continue;
        }
        if(mouse_press(LB+167,UB+360,LB+187,UB+380)==1) //降低设定温度
        {
            if((*house).tempset<=10) continue;

```

```

        com_settemp(house,robot,-1);
        nocombo();
        write_statu(house,robot,2);
        continue;
    }
    if(mouse_press(LB+227,UB+360,LB+247,UB+380)==1) //提高设定温度
    {
        if((*house).tempset>=40) continue;
        com_settemp(house,robot,1);
        nocombo();
        write_statu(house,robot,2);
        continue;
    }
    if(mouse_press(LB+37,UB+410,LB+127,UB+450)==1) //进入干燥
    {
        com_dry(house,robot);
        continue;
    }
    if(mouse_press(LB+147,UB+410,LB+237,UB+450)==1) //进行除尘
    {
        com_clean(house,robot);
        continue;
    }
    if(mouse_press(LB+57,UB+470,LB+217,UB+510)==1) //返回控制面板
    {
        nocombo();
        return 0;
    }
    if(kbhit())
    {
        Delaytime(50);
        value=getch();
        moveupdate(house,robot,value);
    }
    get_conbot(house,robot);
    if(mouse_press(15,24,735,744)==1)
    {
        nocombo();

```



```

        poscode=getposition(MouseX, MouseY);
        mto.x=poscode/18;
        mto.y=poscode%18;
        mp.x=(*robot).px;
        mp.y=(*robot).py;
        if(!Astarmove(mp,mto,robot,house))
        {
            draw_cantgo();
        }
    }
    if(mouse_press(LB+140,UB+10,LB+250,UB+40)==1)
        return 1;
}
}

```

```

void com_dry(HOUSE *house, ROBOT *robot)
{
    (*house).setd=1;
    //write_statu(house,robot,1);
}

```

```

void com_clean(HOUSE *house, ROBOT *robot)
{
    (*house).setc=1;
    //write_statu(house,robot,1);
}

```

```

void com_settemp(HOUSE *house, ROBOT *robot,int change)
{
    if(change==1) (*house).tempset++;
    else (*house).tempset--;
    write_statu(house,robot,2);
}

```

6.3.10 chat..c

互动功能函数。（孟繁鹏负责）

```
#include "headers.h"
```

```

int func_react(HOUSE *house, ROBOT *robot, USER *usr)
{
    char value;
    int poscode;
    NODE mp,mto; //鼠标点击后行动坐标

    Coordinate currentlu={IX+5,IY+WIDTH1+WIDTH2+5}; //文本输出的左上角坐标
    Area                txt_area={{IX+5,IY+WIDTH1+WIDTH2+5},{IX+LENGTH-
5,IY+WIDTH1+WIDTH2+WIDTH3-5}}; //文本输出的区域
    int *qhwh=(int*)malloc(sizeof(int)*MAXTXT); //区号位号
    char *incode=(char *)malloc(30*sizeof(char)); //标记输入内容
    char *quest=(char *)malloc(30*sizeof(char)); //标记谜语答案
    char *answer=(char *)malloc(30*sizeof(char)); //标记谜语答案
    int flag, f1=2;
    int mode=0;
    int show_y=IY-245;

    FILE *fpde;

    draw_bactr(robot);
    write_statu(house,robot,3);
    draw_react(usr,mode);
    clrmous(MouseX, MouseY);

    while(1)
    {
        newmouse(&MouseX, &MouseY, &press);
        timepass(house,robot,usr,3);
        if(mode==0)
        {
            puthz(IX+25,IY-140,"你好！ 我是你的",24,25,'K',BLACK);
            puthz(IX+25,IY-110,"家居机器人小科。 ",24,25,'K',BLACK);
            puthz(IX+10,IY-70,"想聊点什么呢？ ",32,32,'K',BLACK);
        }
        else if(mode==1)
        {
            bar(IX,IY,IX+LENGTH,IY+WIDTH1,LIGHT_GRAY); //拼音显示框
            bar(IX,IY+WIDTH1,IX+LENGTH,IY+WIDTH1+WIDTH2,LIGHT_GRAY); //输入

```

法汉字显示框

```
bar(IX,IY+WIDTH1+WIDTH2,IX+LENGTH,IY+WIDTH1+WIDTH2+WIDTH3,LIGHT_GRAY);
//文本输出框
    //bar(IX,IY-WIDTH4,IX+LENGTH,IY,MISTY_ROSE); //聊天框显示区域

    Init_qhwh(qhwh);
    flag=ShowChTab(txt_area,currentlu,16,qhwh,house,robot,usr,&mode);
    if(flag==0)
    {
        free(qhwh);
        free(quest);
        free(answer);
        free(incode);
        return 1;
    }
    if(flag==1)
        c_reply1(qhwh,&show_y);
    if(flag==2)
    {
        show_y=IY-245;
        continue ;
    }
    if(flag==3)
    {
        nocombo();
        free(qhwh);
        free(quest);
        free(answer);
        free(incode);
        clrmous(MouseX, MouseY);
        return 0;
    }
}
else if(mode==2)
{
    bar(IX,IY,IX+LENGTH,IY+WIDTH1,LIGHT_GRAY); //拼音显示框
    bar(IX,IY+WIDTH1,IX+LENGTH,IY+WIDTH1+WIDTH2,LIGHT_GRAY); //输入
```

法汉字显示框

```
bar(IX,IY+WIDTH1+WIDTH2,IX+LENGTH,IY+WIDTH1+WIDTH2+WIDTH3,LIGHT_GRAY);
```

```
//文本输出框
```

```
//bar(IX,IY-WIDTH4,IX+LENGTH,IY,MISTY_ROSE); //聊天框显示区域
```

```
if(f1==2)
```

```
{
```

```
    make_ques(quest,answer);
```

```
    show_reply(quest,&show_y);
```

```
    Init_qhwh(qhwh);
```

```
    flag=ShowChTab(txt_area,currentlu,16,qhwh,house,robot,usr,&mode);
```

```
    if(flag==0)
```

```
{
```

```
    free(qhwh);
```

```
    free(quest);
```

```
    free(answer);
```

```
    free(incode);
```

```
    return 1;
```

```
}
```

```
if(flag==1)
```

```
{
```

```
    get_str(qhwh,incode);
```

```
    show_incode(incode,&show_y);
```

```
    f1=check_answer(incode,answer);
```

```
    continue;
```

```
}
```

```
if(flag==2)
```

```
{
```

```
    show_y=IY-245;
```

```
    continue ;
```

```
}
```

```
if(flag==3)
```

```
{
```

```
    nocombo();
```

```
    free(qhwh);
```

```
    free(quest);
```

```
    free(answer);
```

```

        free(incode);
        clrmous(MouseX, MouseY);
        return 0;
    }
}
else if(f1==1)
{
    show_reply("恭喜你，答对啦！",&show_y);
    f1=2;
    Delaytime(1000);
    continue;
}
else
{
    show_reply("别灰心，再试一次！",&show_y);
    show_reply("也可输入“放弃”换一题",&show_y);
    Init_qhwh(qhwh);
    flag=ShowChTab(txt_area,currentlu,16,qhwh,house,robot,usr,&mode);
    if(flag==0)
    {
        free(qhwh);
        free(quest);
        free(answer);
        free(incode);
        return 1;
    }
    if(flag==1)
    {
        get_str(qhwh,incode);
        show_incode(incode,&show_y);
        if(check_answer(incode,"放弃"))
        {
            f1=2;
            continue;
        }
        f1=check_answer(incode,answer);
        continue;
    }
}

```

```

        if(flag==2) continue ;
        if(flag==3)
        {
            nocombo();
            free(qhwh);
            free(quest);
            free(answer);
            free(incode);
            clrmous(MouseX, MouseY);
            return 0;
        }
    }
}
if(mouse_press(LB+37,UB+410,LB+127,UB+450)==1) //进入闲聊模式
{
    nocombo();
    show_y=IY-245;
    if(mode!=1)
    {
        bar(IX,IY-WIDTH4,IX+LENGTH,IY,MISTY_ROSE); //聊天框显示区域
        mode=1;
    }
    else mode=0;
    clrmous(MouseX, MouseY);
    draw_react(usr,mode);
    draw_bactr(robot);
    write_statu(house,robot,3);
    fpde=fopen("debug\\debug.txt","a");
    fprintf(fpde,"1 %d\n",show_y);
    fclose(fpde);
    continue;
}
if(mouse_press(LB+147,UB+410,LB+237,UB+450)==1) //进入谜语模式
{
    nocombo();
    show_y=IY-245;
    if(mode!=2)
    {

```

```

        bar(IX,IY-WIDTH4,IX+LENGTH,IY,MISTY_ROSE); //聊天框显示区域
        mode=2;
    }
    else mode=0;
    clrmous(MouseX, MouseY);
    draw_react(usr,mode);
    draw_bactr(robot);
    write_statu(house,robot,3);
    fpde=fopen("debug\\debug.txt","a");
    fprintf(fpde,"2 %d\n",show_y);
    fclose(fpde);
    continue;
}
if(mouse_press(LB+57,UB+470,LB+217,UB+510)==1) //返回控制面板
{
    nocombo();
    free(qhwh);
    free(quest);
    free(answer);
    free(incode);
    clrmous(MouseX, MouseY);
    draw_react(usr,mode);
    fpde=fopen("debug\\debug.txt","a");
    fprintf(fpde,"0 %s\n",(*usr).user);
    fclose(fpde);
    return 0;
}
if(kbhit())
{
    Delaytime(50);
    value=getch();
    moveupdate(house,robot,value);
}
get_conbot(house,robot);
if(mouse_press(15,24,735,744)==1)
{
    nocombo();
    poscode=getposition(MouseX, MouseY);

```

```

        mto.x=poscode/18;
        mto.y=poscode%18;
        mp.x=(*robot).px;
        mp.y=(*robot).py;
        if(!Astartmove(mp,mto,robot,house))
        {
            draw_cantgo();
        }
    }
    if(mouse_press(LB+140,UB+10,LB+250,UB+40)==1)
    {
        free(qhwh);
        free(quest);
        free(answer);
        free(incode);
        return 1;
    }
}
}

```

```

void c_reply1(int *qhwh,int *show_y)
{
    char *reply=(char *)malloc(30*sizeof(char)); //之后标记到文件中的答句部分
    char *incode=(char *)malloc(30*sizeof(char)); //用户输入的内码序列
    char str1[30];
    char str2[30];

    if(reply==NULL)
    {
        puthz(0,0,"内存分配失败",16,16,'H',STRONG_RED);
        getch();
        exit(1);
    }
    if(incode==NULL)
    {
        puthz(0,0,"内存分配失败",16,16,'H',STRONG_RED);
        getch();
        exit(1);
    }
}

```



```

    }

    get_str(qhwh,incode); //将 qhwh 转换成文字存在 incode 里面
    strcpy(str1,incode);
    reply_match(incode,reply); //将用户的信息与文件里存储的信息进行对比并得到回
复
    strcpy(str2,reply);

    show_incode(str1,show_y); //将用户输入的信息打印到聊天框
    show_reply(str2,show_y); //将机器人回答的信息打印到聊天框

    free(reply), reply=NULL;
    free(incode), incode=NULL; //释放空间
}

int reply_match(char *str,char *reply)
{
    FILE *fp; //定义文件指针
    char *keyword=(char *)malloc(10*sizeof(char)); //关键词
    char *reply1=(char *)malloc(40*sizeof(char)); //答句
    int ti,cnt=0; //标记关键词出现的数量

    fp=fopen("chat\\match.txt","rt");
    if(fp==NULL)
    {
        puthz(0,0,"匹配文件打开错误",32,32,'H',STRONG_RED);
        exit(1);
    }
    rewind(fp);
    while(feof(fp)==0)
    {
        fscanf(fp,"%s",keyword);
        fscanf(fp,"%s",reply1);
        ti=count_str(str,keyword);
        if(ti>cnt)
        {
            strcpy(reply,reply1);
            cnt=ti;
        }
    }
}

```

```

    }
}
if(cnt==0)
    strcpy(reply,"抱歉，我不懂你在说什么");
fclose(fp);
free(keyword);
free(reply1);
return;
}

```

```

int count_str(char *str,char *tar)
{
    int num=0,ne[40]={0},l1,l2;
    int i,j;
    l1=strlen(str);
    l2=strlen(tar);
    str--;
    tar--;
    j=0,ne[0]=-1;
    for(i=2;i<=l2;i++)
    {
        while(j&&tar[i]!=tar[j+1]) j=ne[j];
        if(tar[j+1]==tar[i]) j++;
        ne[i]=j;
    }
    j=0;
    for(i=1;i<=l1;i++)
    {
        while(j>0&&tar[j+1]!=str[i]) j=ne[j];
        if(tar[j+1]==str[i]) j++;
        if(j==l2) num++;
    }
    return num;
}

```

```

int make_ques(char *question,char *answer)
{
    FILE *game;

```

```

int tm; //用来确定最终选择的题号
int i=0; //标记暂时选择的题号

srand((unsigned)(time(NULL)));
tm=rand()%100;
game=fopen("chat\\zimi.txt","r");
if(game==NULL)
{
    puthz(0,0,"文件打开错误",32,32,'H',STRONG_RED);
    getch();
    exit(1);
}
rewind(game); //使指针指向文件流的开头
while(feof(game)==0) //当文件读取完毕或标记题号与选中题号相符时
{
    fscanf(game,"%s",question);
    fscanf(game,"%s",answer);
    if(i==tm) break;
    i++;
}
fclose(game);
return tm;
}

int check_answer(char *incode,char *answer)
{
    if(strcmp(incode,answer)==0) return 1;
    return 0;
}

void show_incode(char *incode,int *y)
{
    int len=strlen(incode), line=1;
    int IY1=IY-5, WIDTH1=WIDTH4-10;
    unsigned int back[260][90];
    line=len/24;
    if(len%24!=0) line++;
    if(*y>=IY-5)

```

```

{
    clrmous(MouseX, MouseY);
    get_image(IX,IY1-WIDTHI+line*20,IX+LENGTH,IY1-WIDTHI*2/3+line*20,*back);
    put_image(IX,IY1-WIDTHI,IX+LENGTH,IY1-WIDTHI*2/3,*back);
    get_image(IX,IY1-WIDTHI*2/3+line*20,IX+LENGTH,IY1-
WIDTHI/3+line*20,*back);
    put_image(IX,IY1-WIDTHI*2/3,IX+LENGTH,IY1-WIDTHI/3,*back);
    get_image(IX,IY1-WIDTHI/3+line*20,IX+LENGTH,IY1,*back);
    put_image(IX,IY1-WIDTHI/3,IX+LENGTH,IY1-line*20,*back);
    bar(IX,IY1-line*20,IX+LENGTH,IY1,MISTY_ROSE);

    puthz1(IX,*y-line*20,"用户： ");
    puthz1(IX+48,*y-line*20,incode);
}
else
{
    puthz1(IX,*y,"用户： ");
    puthz1(IX+48,*y,incode);
    *y+=(line*20);
}
}

void show_reply(char *reply,int *y)
{
    int len=strlen(reply), line=1;
    int IY1=IY-5, WIDTHI=WIDTH4-10;
    unsigned int back[260][90];
    line=len/24;
    if(len%24!=0) line++;
    if(*y>=IY-5)
    {
        get_image(IX,IY1-WIDTHI+line*20,IX+LENGTH,IY1-WIDTHI*2/3+line*20,*back);
        put_image(IX,IY1-WIDTHI,IX+LENGTH,IY1-WIDTHI*2/3,*back);
        get_image(IX,IY1-WIDTHI*2/3+line*20,IX+LENGTH,IY1-
WIDTHI/3+line*20,*back);
        put_image(IX,IY1-WIDTHI*2/3,IX+LENGTH,IY1-WIDTHI/3,*back);
        get_image(IX,IY1-WIDTHI/3+line*20,IX+LENGTH,IY1,*back);
        put_image(IX,IY1-WIDTHI/3,IX+LENGTH,IY1-line*20,*back);
    }
}

```

```

        bar(IX,IY1-line*20,IX+LENGTH,IY1,MISTY_ROSE);

        puthz1(IX,*y-line*20,"小科: ");
        puthz1(IX+48,*y-line*20,reply);
    }
    else
    {
        puthz1(IX,*y,"小科: ");
        puthz1(IX+48,*y,reply);
        *y+=(line*20);
    }
}
}

```

7 时间与工作安排

7.1 时间安排

时间	任务
第一周	复习开学考试
第二周	分析需求与功能，完成报告； 搭建工程，配置环境，完成注册登陆界面
第三周	开始绘制核心界面，并构建核心程序框架
第四周	开始实现基本功能，并编写算法
第五周	完善功能，并开始优化算法
第六周	基本完成所有功能，优化算法，并完善交互界面内容
第七周	完成交互界面，继续优化算法
第八周	优化核心框架，细化绘图，并优化一些细节
第九周	调试程序，整理报告，准备验收

7.2 分工情况

函数	文件	行数
孟繁鹏	board.c	300
	chat.c	366
	comfort.c	95
	drawbas.c	144
	drawrob.c	128

	Interf.c	185
	mainwin..c	240
	move.c	332
	mysys.c	35
	plan.c	128
	合计	1953
黄瀚文	drawmap.c	300
	electr.c	366
	login.c	95
	main.c	144
	register.c	128
	rubbish.c	185
	合计	1107

8 总结与感想

8.1 程序亮点

- 1.. 本程序在路径规划中使用了 Astar 算法,节省了时空成本,同时手动实现了优先队列,增加了队员对于优先队列这一数据结构的理解。
2. 本程序的交互功能在前人的基础上实现了较大的突破,尝试使用一些 NLP 知识,利用 KMP 算法统计词频来预测关键词并给出答案。
3. 开发者手动实现了大多数鼠标,汉字库,输入法等基层函数,并进行了一定的优化以适配自己的使用,虽然消耗了比较多的时间成本并且影响了最终呈现效果,但是加深了对底层编程内容的理解。

8.2 程序不足

1. 功能相对较为单一,没有加入过多的创新功能。
2. 由于时间关系,绘图较为简单,图像内容并不算丰富美观。

8.3 组员感想

孟繁鹏:

C 语言课程设计,自动化的第一个程序设计,也是大家讨论比较多,比较看重的一个课程设计,很快就过去了。

由于种种原因,我和队友真正开始启动项目是很晚的。不过我也早在项目启动之前做了不少尝试和准备,自学了算法和数据结构,也研读了包括 C Primer Plus 和深入理解

计算机系统书籍，来提升自己的有关能力，而最后我也成功的应用了一些效果不错的算法。早期我们的进展也很慢，直到中期验收，我们拿得出的也只有几个画了一般的界面和卡成 ppt 的动画，环境的恶劣和难以驾驭程度我们早就有所耳闻，但是第一次感受到的时候还是感到十分的吃力和苦恼。

后来我们转变了策略，一切以高效流畅为准，这也导致我们在很长一段时间一直用的是一个“毛坯房”。包括后来我们尝试了几个算法，也都小心翼翼地不断优化，才能使得他在环境的限制下尽可能好的运行。

我和队友其实在课设的节奏上并不太相似，我是属于一个比较喜欢催促和赶进度的人，有的时候就会不断催促队友赶进度之类，但是队友也有手头的事情，所有也很感谢队友的理解和包容，之前很多时候也是担心不能顺利完成进度，但是当实现最后一个功能时候，我发现有时候似乎操之过急也未必是一个好的策略。总而言之，这次的合作让我和队友开心顺利的完成了自己的任务。而且我们很注重平时的沟通交流，当一个人在忙其他事情时候另一个人往往会主动承担更多任务和责任，给彼此留下了更多自由和空间。

我觉得我们的编程习惯还算良好，大家都会去尝试封装一些东西，这大大提高了我们的工作效率，有时候看看对方的注释或者接口，就能直接顺利地调用对方实现的一些功能。此外，我们最开始就商量好了一些基本的小原则，比如，留出一些显眼的接口，边写边合并代码，这也直接让我们最后的临验收测试变得很轻松愉快，当别人苦恼于合并两个人的功能和代码的时候，我们可以很轻松地拿出一份两个人合力开发的完整的程序。

我认为课程设计这种实践性的课程存在是非常有价值的。现在太多同学只会死刷题，我见过许多加权不低的同学，在 C 课设面前一头雾水，完全不知该从何下手，说直白了，出了大学校园，考试的分数就显得无足轻重，更多的是需要我们能学到的知识应用起来，比如利用学过的 C 语言知识，实现一个有一定功能的小程序。在课设的过程中，对文件的处理，对逻辑思维的锻炼，对指针的灵活运用，对堆栈存储区的思考，完全不是像上个学期应付考试就能收获到的。而且这样的经历也让我明白，课业只是生活的一部分，因为有些东西是课堂上不会讲的，是只能通过自学来完成的，如果只是像大一的生活那样把所有精力放在学校已安排好的课程上，那反而是一种对时间的浪费。对于知识的实践应用的探索，往往比死抠知识本身更重要。

黄瀚文：

经历了 C 课设的每个人，内心一定都是五味杂陈的，有着颇多的感慨。

首先，C 语言课程设计，对于刚刚接触 C 语言不久的我们是一个极大的提高和锻炼的机会，在短时间内将所学甚至未学的知识，都通过实践深化了理解，也学会了如何合理运用。从大一下开始接触 C 语言，虽然有上机，但是其对于 C 语言这个东西怎样运用到实际中，怎样去实现一些现实中的需要的功能，只掌握了皮毛的我们对于这样得力的工具一头雾水，无法合理地运用。从未接触过编程语言的我在一学期的学习之后其实仍对编程充满了疑惑。而紧着，在大一下结束的时候，我终于迎来了 C 课设。C 课设是我第一次面对一个如此巨大的整个工程的设计，一开始面对它的时候内心其实充满

了忐忑，也充满了疑惑，确实不知道怎么下手，也不知道到底需要做些什么。我觉得 C 课设带给我的第一个巨大的收获就是我的队友，以及协作的能力。C 课设是两个人的事情，而整体又是一个大的框架，这就对协作和分工能力有较高的一个要求。我和我的队友在这方面做的比较好，我们在选题的时候就选择了一个两人都感兴趣并能接受的题目，并且在后期的合作当中结合各自的优点缺点进行了明确的分工，很好地结合了各自的优势一起完成了我们的课设，也收获了一份真挚的友谊。而在知识上的收获，毋庸置疑也是巨大的。通过这次的课程设计，我不仅对课本上的知识有了更深的理解，也学到了很多很多课本内没有的知识。一开始的我面对着课设一头雾水，但渐渐的在看了很多遍课本以及网络上的各种教学以后，我也慢慢开始步入正轨，开始掌握一些最简单的操作。当第一次成功跑出图形界面的时候那个激动高兴的心情是在平时的学习过程中都体会不到的，看着自己画出来的界面心情别提有多开心，然后就充满了干劲，很快把登录界面全部画完了。整个做课设的过程都是在这种不断看不懂写不下去和好不容易搞出来很开心充满干劲的心情中来回切换，并且在中期验收后因为心累经历了短暂的空白，导致了后期的紧张兮兮疯狂赶进度。前期的停滞虽然看似挺浪费时间的，但其实在不断翻看课本和代码的过程中我对于那些知识的理解更加深刻了，也有了更多的体会，知道了如何去运用，学会了如何去思考 and 解决。这个过程异常艰辛，但是在最后看到自己完成的课设的时候，还是充满了感慨的，确实来之不易，付出了很多，也收获了很多，心血凝成的课设，希望能够以最好的一面展现出来。

9 参考资料

1. Stephen Prata 等，C Primer Plus 中文版（第五版），北京，人民邮电出版社，2005-2；
2. Cormen,T.H.等著，殷建平译，算法导论（第三版），北京，机械工业出版社，2013-1；
3. Rosen K、罗森、袁崇义等，离散数学及其应用，北京，机械工业出版社，2011.；
4. 刘汝佳，算法竞赛入门经典，北京，清华大学出版社，2009-11；
5. 王士元，C 高级实用程序设计.，北京：清华大学出版社.，1996；
6. 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。

其他参考资料和网站

1. github: <https://github.com/>
2. csdn 论坛: <https://www.csdn.net/>
3. stack overflow 论坛: <https://stackoverflow.com/>