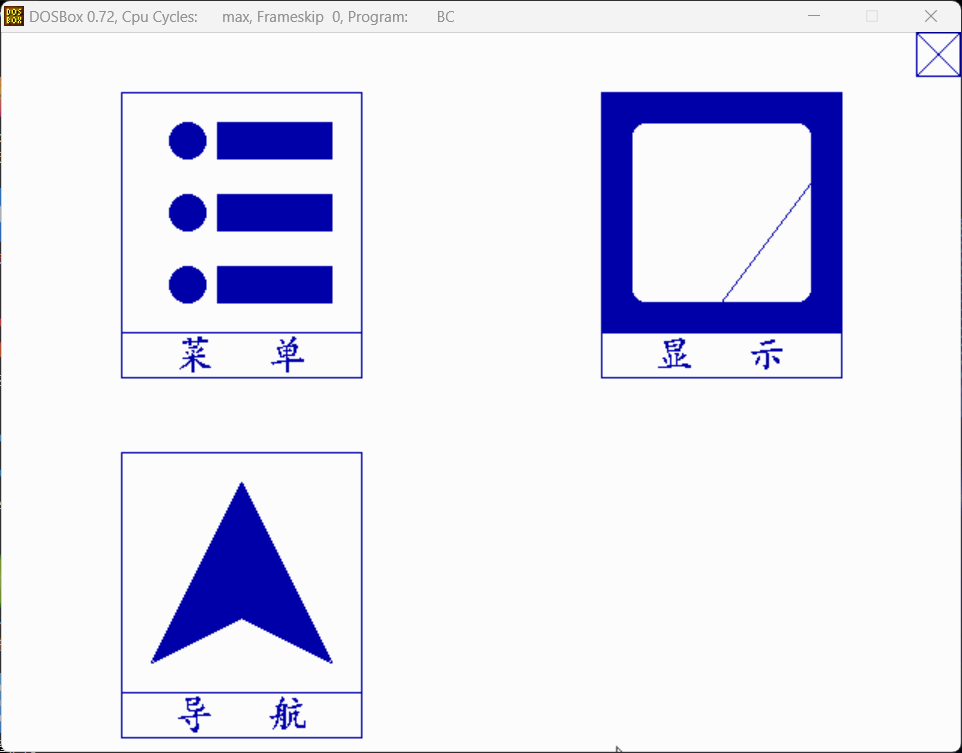


**华中科技大学**

**C语言课程设计报告**

**课程名称： 无人驾驶汽车中控仪表系统模拟**



**专业班级：**

**小组成员： Y**

**X**

**指导教师：\_\_周纯杰、桑农、汪国有\_\_**

**\_\_何顶新、周凯波、彭刚\_\_**

**\_\_左峥嵘、陈忠、高常鑫\_\_**

**报告日期： 2024年4月26日**

**目录**

[一、前言 3](#_Toc2522)

[1.编写背景： 3](#_Toc18023)

[2.编写目的： 3](#_Toc2487)

[3.参考资料： 4](#_Toc6161)

[4.参考软件： 4](#_Toc23188)

[5.编者的话： 4](#_Toc9996)

[二、任务概述 4](#_Toc12175)

[1.需求分析与目标功能： 4](#_Toc3702)

[2.编写规范： 5](#_Toc481)

[三、运行环境和配置 5](#_Toc19180)

[1.硬件接口 5](#_Toc19157)

[2.软件接口 6](#_Toc11051)

[3.控制 6](#_Toc22119)

[四、系统架构 6](#_Toc27302)

[1.基本框架: 6](#_Toc15359)

[五、系统设计 13](#_Toc28129)

[1. 主界面设计 13](#_Toc13796)

[2. 数据结构设计 14](#_Toc5779)

[3.文件操作 14](#_Toc5460)

[六、软件特色 14](#_Toc5590)

[1.界面交互设计 14](#_Toc16495)

[七、函数说明及函数原型 14](#_Toc5368)

[1.函数原型 14](#_Toc20559)

[2.函数功能 21](#_Toc4518)

[3.源代码 38](#_Toc22785)

[八、分工与时间安排 185](#_Toc28800)

[时间安排 185](#_Toc29252)

[分工安排 185](#_Toc8547)

**一、前言**

**1.编写背景：**

在2024年的中国，随着人工智能和自动驾驶技术的不断发展，无人驾驶汽车已经成为了一个备受关注的热门话题和前沿领域。国家要发展，民族要振兴，要建设的是一个全面崛起、多方位全面发展的新中国，计算机科学领域作为新时代广受欢迎的新兴领域，在我国的发展中尤其是不可忽视的重要组成部分。

当前，科技进步已经成为推进我国经济和社会发展、提高我国综合国力和国际竞争能力的关键因素之一。因此，自改革开放以来，我国一直把科学技术的研究放在靠前的位置。操作系统一直是我国计算机科学领域的短板之一，并且现在常见的Linux、Windows、Android等操作系统都是美国开发的。我国在计算机科学领域的发展既有机遇，也有挑战。

自动驾驶技术是当今世界科技领域的热点之一，也是中国政府重点支持和发展的领域。随着自动驾驶技术的不断成熟和普及，无人驾驶汽车的市场需求和潜力也在不断增长。中国作为全球最大的汽车市场之一，对于无人驾驶技术的应用和研究具有重要意义。随着科技的飞速发展，无人驾驶汽车作为未来智能交通的重要组成部分，其研发与应用正日益受到全球范围内的关注。无人驾驶汽车中控仪表系统作为车辆与用户之间交互的核心界面，不仅关乎用户体验，更是保障行车安全的关键环节因此，通过模拟无人驾驶汽车中控仪表系统的课程设计，可以深入了解这一前沿技术，为未来的发展做好准备。本次课程设计选择“无人驾驶汽车中控仪表系统模拟”作为实践主题，旨在通过编程实践，模拟中控仪表系统的基本功能，进一步加深对无人驾驶汽车技术的理解与应用。

早在2010年，谷歌就已经开始了无人驾驶汽车的研发，并于2012年获得许可证。而我国的百度等公司也在近年开始试运营无人驾驶汽车。虽然目前无人汽车尚未完全普及，但是当前在北京、广州、长沙、成都等地，已经有小规模的无人出租车或无人大巴在道路上运营，特斯拉、小鹏、蔚来等汽车公司也已推出无人驾驶汽车。随着计算机视觉、激光雷达、北斗导航系统的不断发展，未来无人驾驶汽车必将称为时代的潮流。从行业角度来看，无人驾驶汽车将会对交通运输、城市规划和个人出行习惯产生深远影响。在这个背景下，研发一款先进的无人驾驶汽车中控系统显得尤为重要。然而，传统的汽车中控系统已经无法满足未来无人驾驶汽车的需求。现代的智能汽车需要整合多种先进技术，包括但不限于传感器技术、人工智能算法、实时数据处理和大数据分析等。此外，无人驾驶汽车中控系统还需要考虑安全性、可靠性和用户体验等方面的挑战。因此，设计一款现代化的无人驾驶汽车中控系统有着巨大的市场潜力。

在当前社会背景下，无人驾驶汽车技术的发展已经逐渐从理论探索转向实际应用。各国纷纷加大研发投入，推动相关产业的发展。而中控仪表系统作为无人驾驶汽车的重要组成部分，其设计与实现对于提升车辆的智能化水平、增强用户体验具有重要意义。因此，本次课程设计不仅是对C语言编程能力的一次锻炼，更是对无人驾驶汽车技术的一次深入探索。

**2.编写目的：**

本报告是无人驾驶汽车中控仪表系统模拟编写组对该软件功能的分析和详解，以及对本编写组在编写该软件时调试和修改的经历的概括。

通过对“无人驾驶汽车中控系统”的需求者和使用者的调查和分析，对特斯拉等汽车中控界面的观摩和网络资料查询，我们编写出这一份软件需求分析和功能设计报告。

该项报告对于整个“无人驾驶汽车中控系统”进行了全面的用户需求和功能分析。包括可行性分析，需求分析，系统功能设计，代码实现，软件亮点和不足，集成测试等等。本报告明确了本软件系统架构设计，软件结构与数据结构设计，各模块之间的接口和调用，系统界面设计，系统功能设计（函数罗列），具体算法设计以及整个软件的源代码。

同时，该项报告也明确了两位开发者的设计与分工，增强了后期测试人员对于软件的调试和验收的可读性与可修改性。

**3.参考资料：**

1. 周纯杰，何顶新等. 程序设计与应用（用C/C++编程）. 北京：机械工业出版社. 2008年

2. 王士元. C高级实用程序设计. 北京：清华大学出版社. 1996年

3. 徐波. C专家编程. 北京：人民邮电出版社. 2002年

4. C语言高级编程技术. 北京：中国科学院希望高级电脑技术公司.

**4.参考软件：**

无人驾驶汽车中控仪表系统

无人驾驶电动汽车中控系统交互初设计

**5.编者的话：**

在编写过程中，我们遇到了许多挑战和困难，但通过团队协作和不懈努力，我们逐渐克服了这些难题，取得了进展。

这次课程设计不仅是对我们C语言编程能力的一次全面检验，更是对我们解决问题能力和团队协作精神的一次锻炼。通过实践操作，我们深刻体会到了编程的魅力和乐趣，也更加明白了理论知识与实际应用的紧密关系。

未来，我们将继续努力，不断完善和优化程序的功能和性能，争取在课程设计结束时呈现出一个完整、稳定、实用的无人驾驶汽车中控仪表系统模拟程序。同时，我们也希望能够将所学的知识和技能应用到更多的实际项目中。

**二、任务概述**

**1.需求分析与目标功能：**

在对系统需求进行全方位的分析，以及对无人驾驶汽车中控仪表系统等软件进行参考后，我们设置了如下功能：

菜单：提供配置选项，允许用户自定义某些功能。

显示驾驶信息：中控仪表系统通过显示器向驾驶员提供实时的驾驶信息，如车速、前后驱、导航信息、行车路线等。

模拟中控仪表显示车辆的实时数据，如车速、电池电量、里程等。模拟监测车辆各个系统的状态，如制动系统、电池等。根据车辆状态和用户操作，模拟显示警告信息和提示信息，如低电量、故障警告、事故警告等.

显示车辆周围环境的图像或实时视频，如前方障碍物等。模拟记录车辆数据并提供简单的数据分析功能，如行驶记录等。

按钮控制：提供按钮或选项来模拟控制车辆的功能，如启动、熄火、打开车窗、调节空调等。模拟车辆的诊断和故障排查功能，显示车辆的故障代码和建议的修复措施。

操作控制界面：提供模拟按钮和控制选项，允许用户模拟控制车辆的功能，如启动、停止、加速、刹车、转向等。

导航和地图功能：中控仪表系统通常集成了导航和地图功能，帮助驾驶员选择最佳路线。模拟显示导航路线以及附近的标志物、交通状况等。

**2.编写规范：**

命名规范：

变量命名：下划线命名法，即单词间用下划线隔开。其组成单词用英文表达其确定含义，或使用详细注释说明其主要功能。

函数命名：驼峰命名法。即除缩写和专有名词外，包括第一个单词在内的所有单词首字母大写，表达出函数或该文件所包含函数的主要功能。

注释规范:

文件注释：注释该文件所包含函数的主要功能。

函数注释：函数注释分为整个函数功能的注释（采用多行注释的形式），以及对部分可能令测试者难以理解的算法和流程而给出的相应注释（采用单行注释的形式）。

特殊注释：变量说明。

**三、运行环境和配置**

**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：VGA接口

系统运行内存：要求32MB以上

**2.软件接口**

开发软件工具：Borland C 3.1。

文字编辑工具：Visual Studio Code 1.49、Dev C++ 5.11

操作系统：DOS/WINDOWS XP

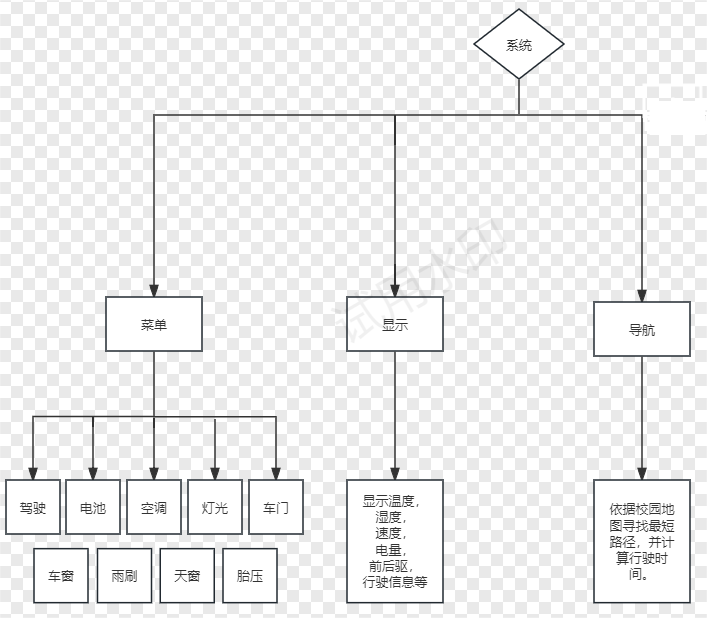
**3.控制**

该系统通过鼠标直接进行控制。鼠标操作中，左键单击可以完成不同的功能。操作完毕后点击右上角相应位置退出图标退出系统，点击左上角相应位置返回主界面。

**四、系统架构**

根据我们对于无人驾驶汽车中控仪表系统模拟用户的需求分析，以及对软件进行的参考，我们设置了如下功能：

**1.基本框架:**

****

根据对驾驶员的调研，我们设计了四个模块来满足其需求：菜单、显示、导航、娱乐。

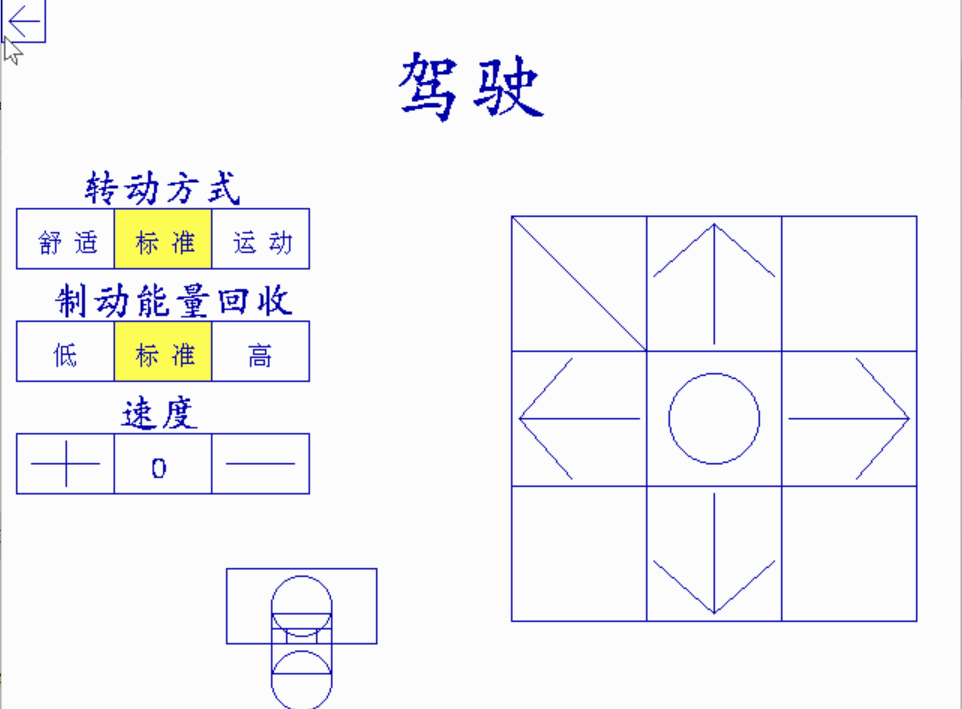
1. 菜单模块。该模块主要用于对全车各个模块进行全面的查看、分析、评估和设置，又分为以下九个子模块：驾驶、电池、空调、灯光、车门、车窗、雨刷、天窗、胎压。

点击右下角按钮可返回主界面。



* 1. 驾驶模块。该子模块主要用于实现对车的启动及控制功能，并控制车辆的速度，能量转换方式等。

可调节转动方式（舒适，标准，运动三档，一般为标准），制动能量回收（低，标准，高三档，一般为标准）。当右边启动按钮（中间圆形）激活时，可调节速度大小（0—99），并可点击前进，后退，左转，右转按钮；当刹车按钮激活时，速度归零；



* 1. 电池模块。该子模块主要用于实现对电池的调节及控制功能，如电池配置。

可实现对电池的两种充电模式（充至80%和充至100%），并显示两种充电模式优缺点。





* 1. 空调模块。该子模块主要用于实现对空调的开关及控制功能，如温度，湿度；

可预设温度（调节范围：16—30摄氏度）和湿度（20%—80%），之后开启空调，车内的温度湿度会发生变化，不断地接近空调的预设温度。



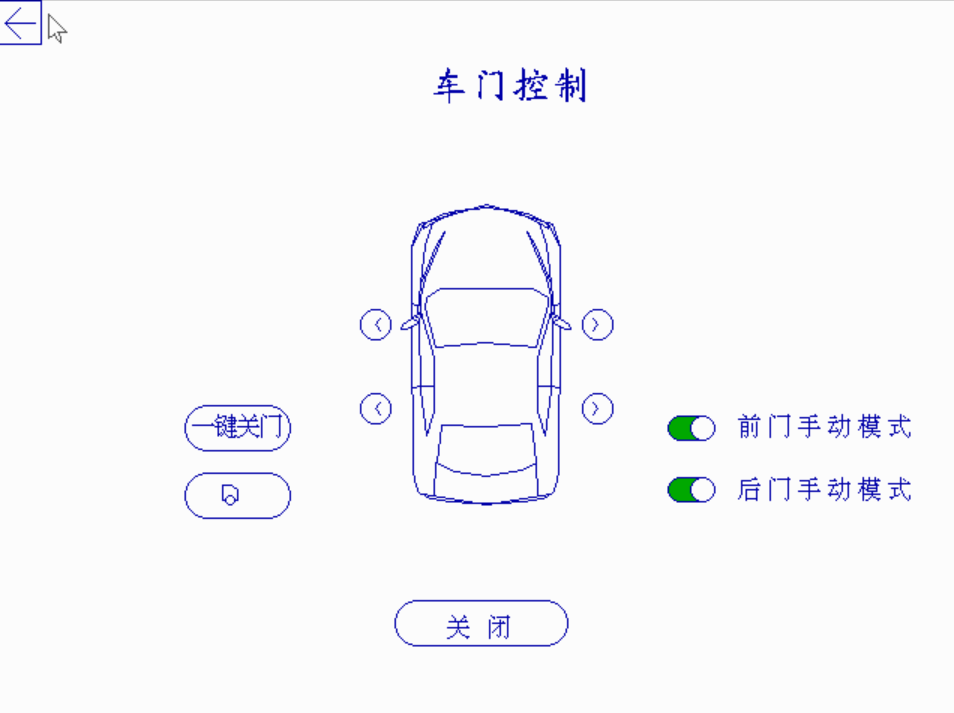
* 1. 灯光模块。该子模块主要用于实现对灯光的开关及调节功能，如外灯，内灯等。

外部灯光和内部灯光可设置四种模式（off,parking,auto,on），开启照明灯，雾灯，氛围灯等；



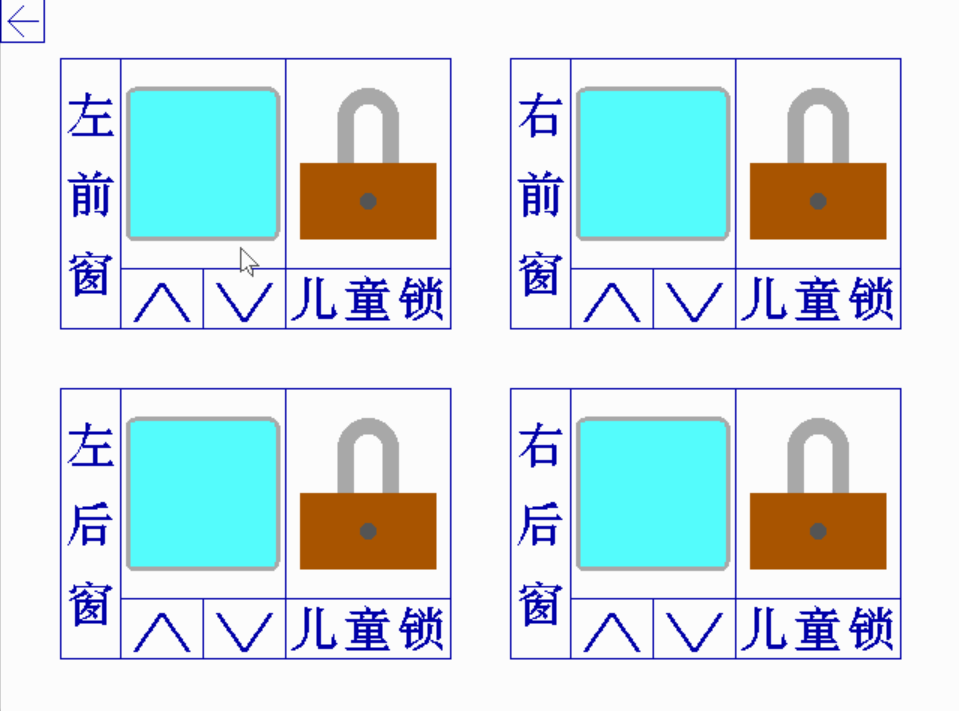
* 1. 车门模块。该子模块主要用于实现对车门的开关及锁定功能。

设置前后门的手动模式，自动模式（可以一键开门，关门）。

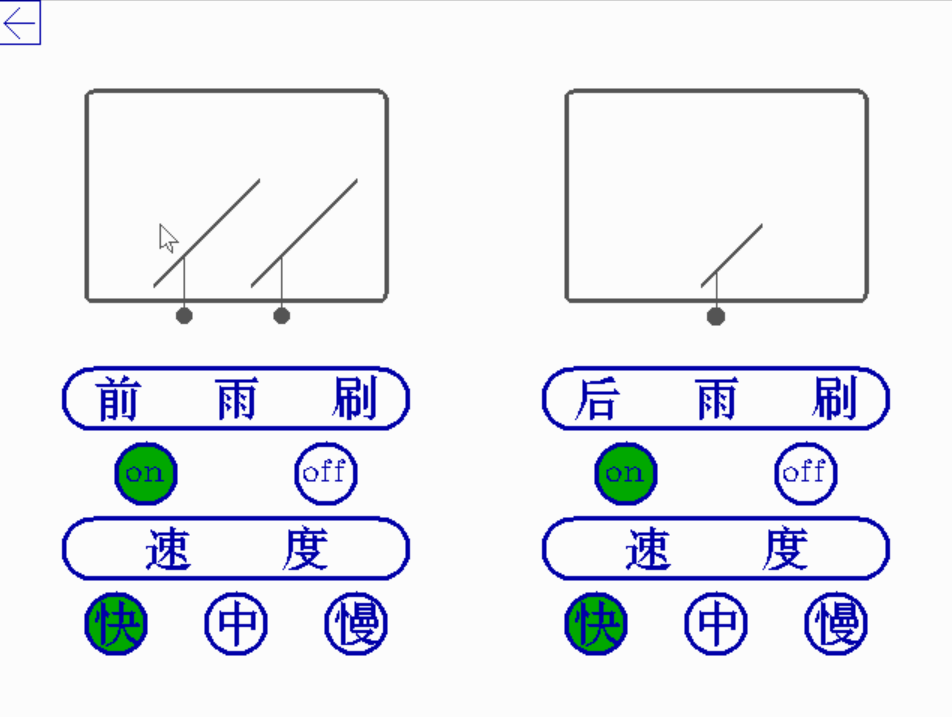


* 1. 车窗子模块。该子模块主要用于实现对车窗的开关及锁定功能。

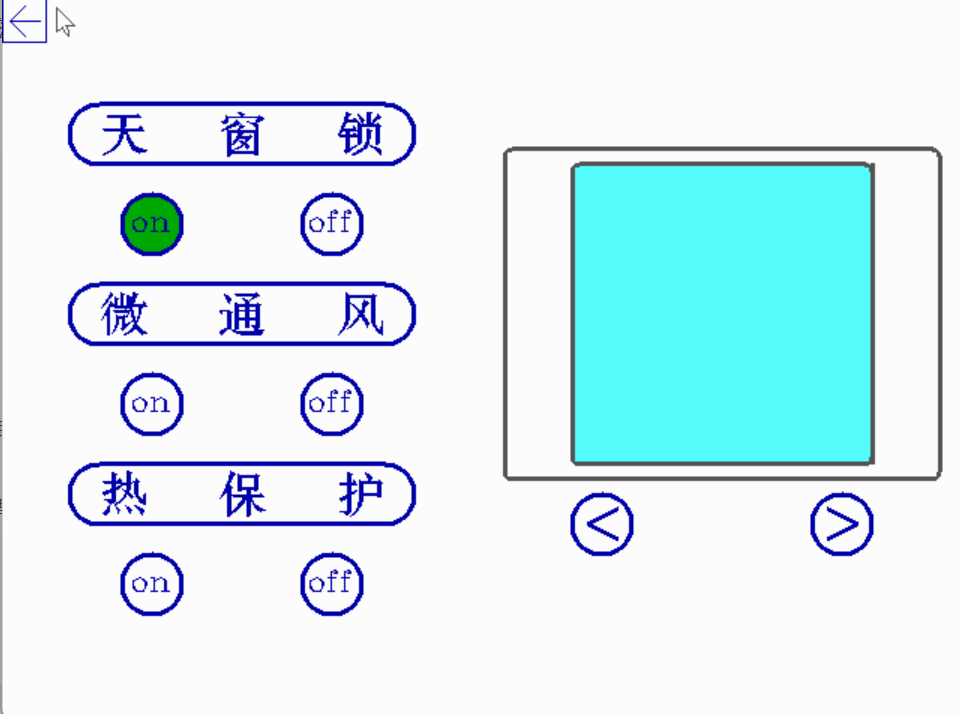
对左前窗，右前窗，左后窗，右后窗设置三个按钮（开启，关闭，儿童锁）；儿童锁设定后，相应车窗关闭，无法打开；未锁定时，可缓慢打开车窗，设定车窗打开程度。



* 1. 雨刷子模块。该子模块主要用于实现对前后雨刷的控制，如开关、速度调节。

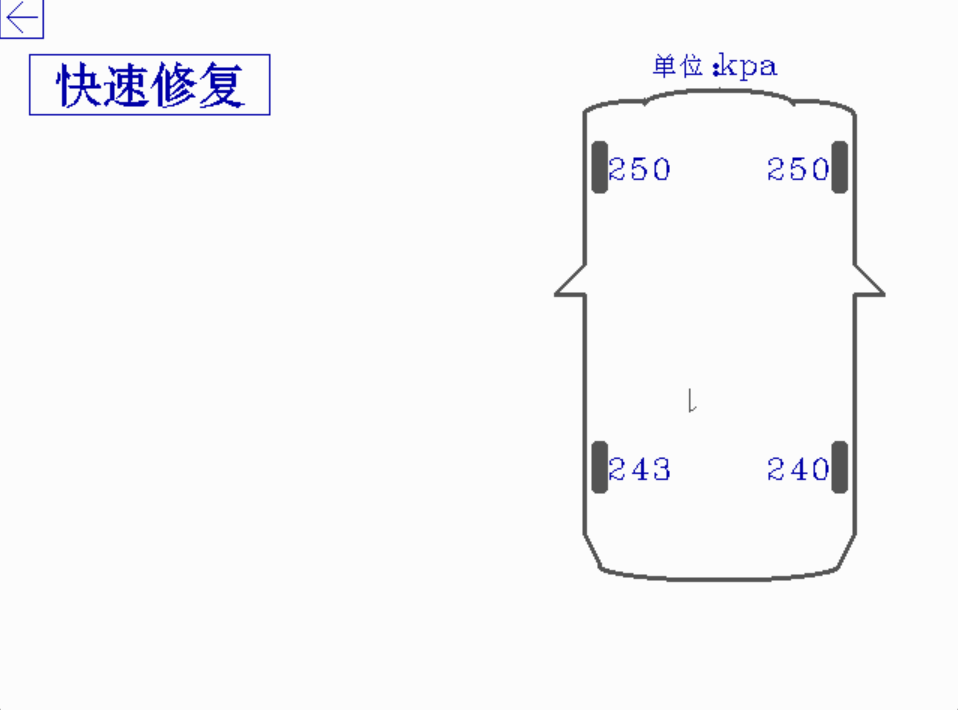


* 1. 天窗子模块。该子模块主要用于实现对天窗功能的控制，如天窗的锁定、开关，以及通风、热保护功能

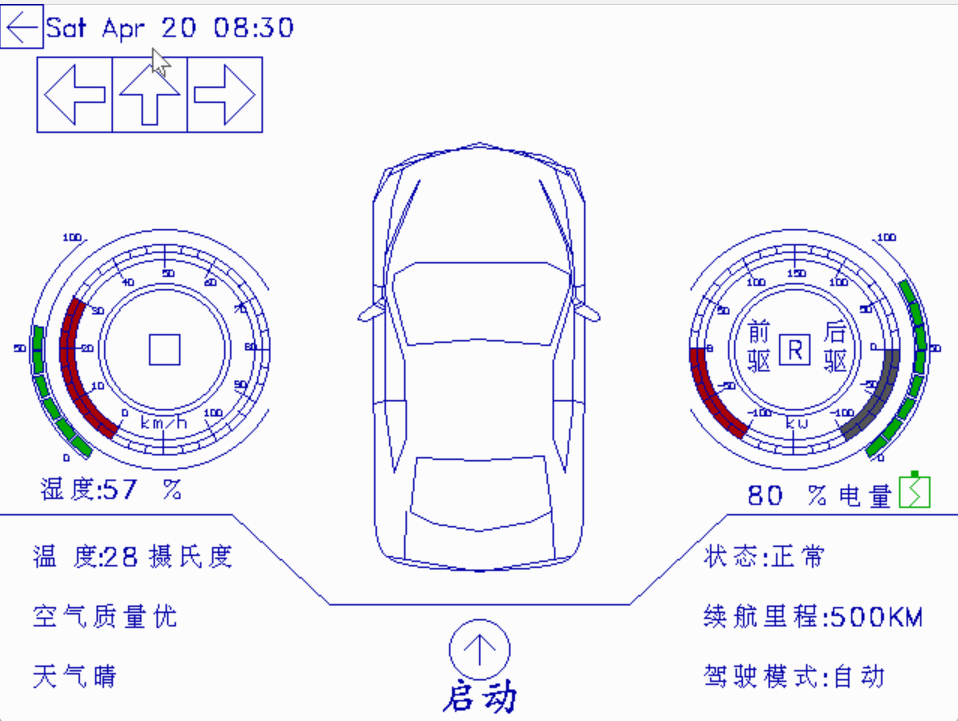


* 1. 胎压子模块。该子模块主要用于对轮胎的胎压进行监测，便于驾驶员分析轮胎情况。

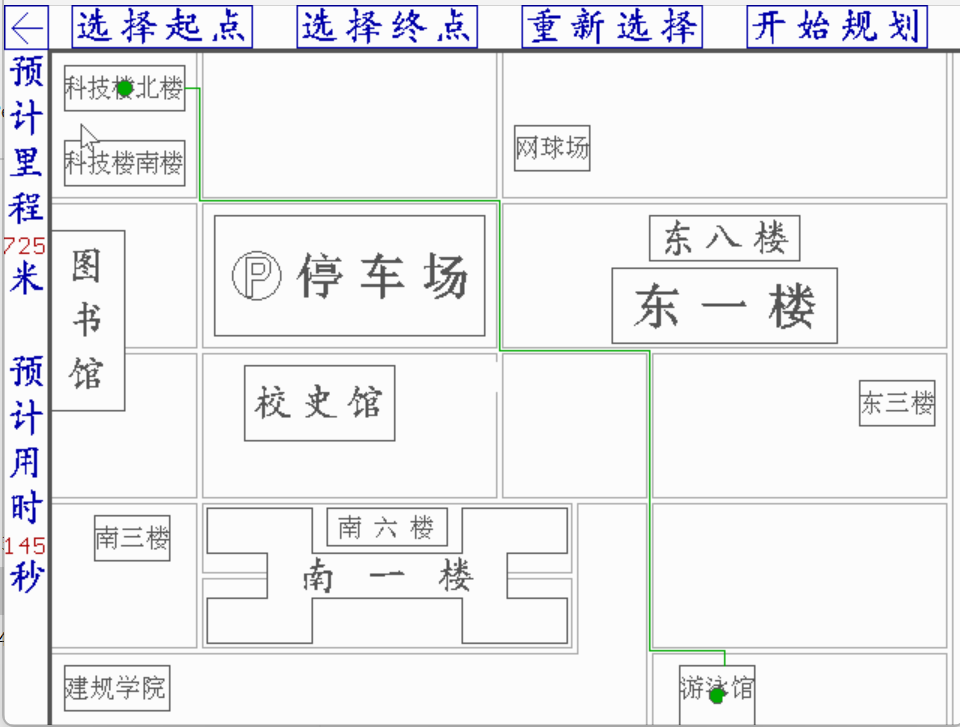
可显示胎压异常，并快速修复。



2）.显示模块。该模块集成了汽车各个模块关键参数，能够显示各个模块的重要信息，便于驾驶员及时获取当前汽车的运行状态，快捷地检查各个模块是否出现异常。



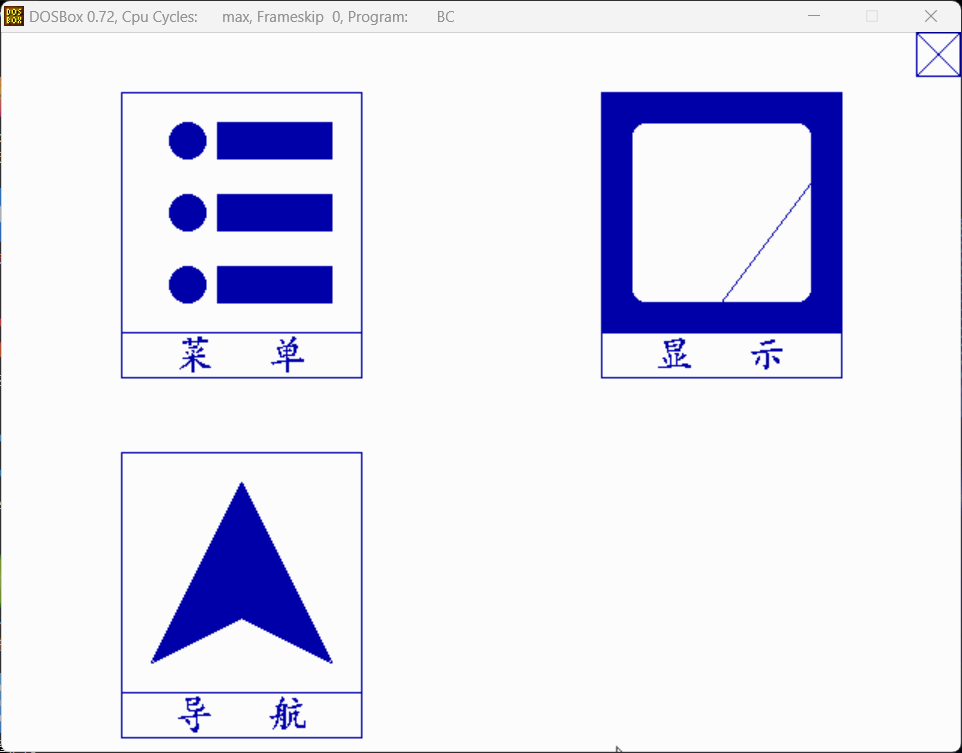
3）.导航模块。该模块主要实现无人驾驶汽车的导航功能。



4).娱乐模块。该模块主要实现无人驾驶汽车的娱乐功能。

**五、系统设计**

1. **主界面设计**



右上角可退出系统。

1. **数据结构设计**

PathNode:节点及指向下一个节点的指针

1. typedef struct PathNode {
2. int node;               *// 节点编号*
3. struct PathNode\* next;  *// 指向下一个节点的指针*
4. } PathNode;

**3.文件操作**

本工程使用了11个文件储存重要信息：

drive.text：用于储存驾驶信息

battery.text：用于储存电池电量

ac.text：用于储存空调预设温度湿度，环境温度湿度

light.text：用于储存灯光信息

door.text：用于储存车门信息

window.text：用于储存车窗信息

wiper.text：用于储存雨刷信息

sunroof.text：用于储存天窗信息

tire.text：用于储存胎压信息

display.text：用于储存行驶路线信息，车的各类参数

path.text：用于储存路径、距离信息

**六、软件特色**

**1.界面交互设计**

（1）本软件在设计时模仿无人驾驶汽车中控仪表系统，上手简单，操作容易。

（2）模拟充电，空调，车辆行进，可提示是否异常。

**七、函数说明及函数原型**

**1.函数原型**

**init.h**

1. #ifndef \_INIT\_H\_
2. #define \_INIT\_H\_
3. #include<stdio.h>
4. #include<graphics.h>
5. #include<stdlib.h>
6. #include<math.h>
7. #include<conio.h>
8. #include"hz.h"
9. #include"mouse.h"
10. #include"page.h"
11. void ClearGraph(int color);
12. void DrawCloseButton(void);
13. void DrawReturnButton(void);
14. void Draw(int color);
15. #endif

**menu.h**

1. #ifndef \_MENU\_H\_
2. #define \_MENU\_H\_
3. void OpenMeun(void);
4. void DrawDriveButton(void);
5. void DrawBatteryButton(void);
6. void DrawAirconditionButton(void);
7. void DrawLightButton(void);
8. void DrawDoorButton(void);
9. void DrawWindowButton(void);
10. void DrawWiperButton(void);
11. void DrawSunroofButton(void);
12. void DrawTireButton(void);
13. void DrawMenuBackButton(void);
14. int MenuMouse(void);
15. #endif

**page.h**

1. #ifndef \_PAGE\_H\_
2. #define \_PAGE\_H\_
3. void OpenMainPage(void);
4. void DrawMenuButton(void);
5. void DrawDisplayButton(void);
6. void DrawNavigationButton(void);
7. #endif

**drive.h**

1. #ifndef \_DRIVE\_H\_
2. #define \_DRIVE\_H\_
3. void OpenDrive(void);
4. void DrawDriveSButton(void);
5. void DrawDriveChangePart(int\* num);
6. void PutDriveHz(void);
7. int DriveMouse(int\* num);
8. #endif

**battery.h**

1. #ifndef \_BATTERY\_H\_
2. #define \_BATTERY\_H\_
3. void OpenBattery(void);
4. void DrawBatterySButton(int\* num);
5. void DrawBatteryChangePart(int\* num);
6. int BatteryMouse(int\* num);
7. void BatteryPlus(int\* num);
8. #endif

**ac.h**

1. #ifndef \_AIRCONDITION\_H\_
2. #define \_AIRCONDITION\_H\_
3. void OpenAircondition(void);
4. void DrawAirconditionSButton(int\* num);
5. void DrawAirconditionChangePart(int\* num);
6. int AirconditionMouse(int\* num);
7. void Aircondition(int\* num);
8. #endif

**light.h**

1. #ifndef \_LIGHT\_H\_
2. #define \_LIGHT\_H\_
3. void OpenLight(void);
4. void DrawLightSButton(void);
5. void DrawLightChangePart(int\* num);
6. void PutLightHz(void);
7. int LightMouse(int\* num);
8. #endif

**door.h**

1. #ifndef \_DOOR\_H\_
2. #define \_DOOR\_H\_
3. void OpenDoor(void);
4. void DrawDoorSButton(void);
5. void DrawDoorChangePart(int\* num);
6. void PutDoorHz(void);
7. int DoorMouse(int\* num);
8. #endif

**window.h**

1. #ifndef \_WINDOW\_H\_
2. #define \_WINDOW\_H\_
3. void OpenWindow(void);
4. void DrawOneWindow(int x, int y);
5. void DrawWindowChangePart(int\* num);
6. void PutWindowHz(void);
7. int WindowMouse(int\* num);
8. #endif

**wiper.h**

1. #ifndef \_WIPER\_H\_
2. #define \_WIPER\_H\_
3. void OpenWiper(void);
4. void DrawAheadWiper(void);
5. void DrawBehindWiper(void);
6. void DrawWiperChangePart(int\* num);
7. void PutWiperHz(void);
8. int WiperMouse(int\* num);
9. #endif

**sunroof.h**

1. #ifndef \_SUNROOF\_H\_
2. #define \_SUNROOF\_H\_
3. void OpenSunroof(void);
4. void DrawPartSunroofButton(void);
5. void DrawSunroof(void);
6. void DrawSunroofChangePart(int\* num);
7. void PutSunroofHz(void);
8. int SunroofMouse(int\* num);
9. #endif

**tire.h**

1. #ifndef \_TIRE\_H\_
2. #define \_TIRE\_H\_
3. void OpenTire(void);
4. int DrawTire(void);
5. void DrawOneTire(int, int);
6. void PutTireBar(int, int, int);
7. void DrawTireCar(void);
8. void DrawTireChangePart(int\* num);
9. int TireMouse(int\* num);
10. #endif

**display.h**

1. #ifndef \_DISPLAY\_H\_
2. #define \_DISPLAY\_H\_
3. void OpenDisplay(void);
4. void DrawDisplaySButton(int \*num);
5. void DrawCar(int x,int y,float k);
6. void DrawTime(int x,int y,int k);
7. int DisplayMouse(int\* num);
8. void DrawDailnumber();
9. void DrawDisplayChangePart(int \*num);
10. void Run(int \*num);
11. void DrawArrow();
12. void DrawBackground(int \*num);
13. void DisplayChange(int \*num,int battery);
14. #endif

**normal.h**

1. #ifndef \_NORMAL\_H\_
2. #define \_NORMAL\_H\_
3. int NormalRun(int\* bg);
4. void CaseRun(int\* bg, int\* place, int\* distance, int i, int\* now\_distance);
5. void RoutineRun(int\* bg, int\* pnow\_distance, int direction);
6. void FillArrow(int mode);
7. void PutMiles(int\* pnow\_distance);
8. void FragmentRun(int\* bg, int\*distance, int i, int direction, int\* pnow\_distance, int num, ...);
9. enum DIRECTION{LEFT, STRIGHT, RIGHT};
10. #endif

**map.h**

1. #ifndef \_MAP\_H\_
2. #define \_MAP\_H\_
3. void DrawMap(void);
4. void DrawRoad(void);
5. void DrawRoad(void);
6. void DrawParallelRoad(int x, int y, int d);
7. void DrawVerticalRoad(int x, int y, int d);
8. void DrawCrossing(int x, int y, int north, int south, int west, int east);
9. void DrawBuilding(void);
10. void DrawSouthBuildingOne(void);
11. void DrawParkingLot(void);
12. #endif

**navigat.h**

1. #ifndef \_NAVIGAT\_H\_
2. #define \_NAVIGAT\_H\_
3. void OpenNavigation(void);
4. void DrawNavigationTopButton(void);
5. int NavigationMouse(int\* mode);
6. int ChoosePlace(void);
7. #endif

**plan.h**

1. #ifndef \_PLAN\_H\_
2. #define \_PLAN\_H\_
3. typedef struct PathNode {
4. int node;               *// 节点编号*
5. struct PathNode\* next;  *// 指向下一个节点的指针*
6. } PathNode;
7. int PlanPath(int start, int end);
8. int Dijkstra(int start, int end, int distance[][34]);
9. int MinDistance(int\* start\_distance, int\* vertex);
10. void DrawRoute(void);
11. void ReturnCoordinates(int i, int\* place, int\* x, int\* y);
12. void DrawStartEnd(int num);
13. #endif

**hz.h**

1. #ifndef \_\_HZ\_H\_\_
2. #define \_\_HZ\_H\_\_
3. void puthz(int x, int y,char \*s,int flag,int part,int color);
4. #endif

**mouse.h**

1. #ifndef \_mouse\_h\_
2. #define \_mouse\_h\_
3. int mouse\_press(int x1, int y1, int x2, int y2);*//如果在框中点击，则返回1；在框中未点击，则返回2；不在框中则返回0*
4. void mouse(int,int);*//设计鼠标*
5. void mouseinit(void);*//初始化*
6. void mread(int \*,int \*,int\*);*//改坐标*
7. void save\_bk\_mou(int x,int y);*//存鼠标背背景*
8. void clrmous(int x,int y);*//清除鼠标*
9. void drawmous(int x,int y);*//画鼠标*
10. void newmouse(int \*nx,int \*ny,int \*nbuttons);   *//更新鼠标*
11. extern int MouseX;
12. extern int MouseY;
13. extern int MouseS;
14. extern int press;
15. extern union REGS regs;
16. #endif

**2.函数功能**

init.h

1.void ClearGraph(int color)

函数名:ClearGraph

函数描述:清空屏幕，用指定颜色填充，重新初始化鼠标

输入参数:int color

返回值:void

2.void DrawCloseButton(void)

函数名:DrawCloseButton

函数描述:绘制关闭按钮

输入参数:void

返回值:void

3.void DrawReturnButton(void)

函数名:DrawReturnButton

函数描述:绘制退出按钮

输入参数:void

返回值:void

4.void Draw(int color)

函数名:Draw

函数描述:绘制设置

输入参数:void

返回值:void

menu.h

5.void OpenMenu(void)

函数名:OpenMenu

函数描述:打开菜单页

输入参数:void

返回值:void

6.void DrawDriveButton(void)

函数名:DrawDriveButton

函数描述:绘制驾驶按钮

输入参数:void

返回值:void

7.void DrawBatteryButton(void)

函数名:DrawBatteryButton

函数描述:绘制电池按钮

输入参数:void

返回值:void

8.void DrawAirconditionButton(void)

函数名:DrawAirconditionButton

函数描述:绘制空调按钮

输入参数:void

返回值:void

9.void DrawLightButton(void)

函数名:DrawLightButton

函数描述:绘制灯光按钮

输入参数:void

返回值:void

10.void DrawDoorButton(void)

函数名:DrawDoorButton

函数描述:绘制车门按钮

输入参数:void

返回值:void

11.void DrawWindowButton(void)

函数名:DrawWindowButton

函数描述:绘制车窗按钮

输入参数:void

返回值:void

12.void DrawWiperButton(void)

函数名:DrawWiperButton

函数描述:绘制雨刷按钮

输入参数:void

返回值:void

13.void DrawSunroofButton(void)

函数名:DrawSunroofButton

函数描述:绘制天窗按钮

输入参数:void

返回值:void

14.void DrawTireButton(void)

函数名:DrawTireButton

函数描述:绘制胎压按钮

输入参数:void

返回值:void

15.void DrawMenuBackButton(void)

函数名:DrawMenuBackButton

函数描述:绘制菜单页返回按钮

输入参数:void

返回值:void

16.int MenuMouse(void)

函数名:MenuMouse

函数描述:菜单页鼠标

输入参数:void

返回值:int

page,h

17.void OpenMainPage(void)

函数名:OpenMainPage

函数描述:打开主页

输入参数:void

返回值:void

18.void DrawMenuButton(void)

函数名:DrawMenuButton

函数描述:绘制菜单按钮

输入参数:void

返回值:void

19.void DrawDisplayButton(void)

函数名:DrawDisplayButton

函数描述:绘制显示按钮

输入参数:void

返回值:void

20.void DrawNavigationButton(void)

函数名:DrawNavigationButton(void)

函数描述:绘制导航按钮

输入参数:void

返回值:void

21.void DrawEntertainmentButton(void)

函数名:DrawEntertainmentButton

函数描述:绘制娱乐按钮

输入参数:void

返回值:void

22.int MainPageMouse(void)

函数名:MainPageMouse

函数描述:主页鼠标

输入参数:void

返回值:int

23.void OpenDrive(void)

函数名:OpenDrive

函数描述:打开菜单中驾驶页

输入参数:void

返回值:void

drive.h

24.void DrawDriveSButton(void)

函数名:DrawDriveSButton

函数描述:绘制菜单中驾驶所有按钮

输入参数:void

返回值:void

25.void DrawDriveChangePart(int\* num)

函数名:DrawDriveChangePart

函数描述:绘制驾驶页面可变的部分

输入参数:int\* num

返回值:void

26.void PutDriveHz(void)

函数名:PutDriveHz

函数描述:显示驾驶页汉字

输入参数:void

返回值:void

27.int DriveMouse(int\* num)

函数名:DriveMouse

函数描述:驾驶页的鼠标功能

输入参数:int\*

返回值:int

battery.h

28.void OpenBattery(void)

函数名:OpenBattery

函数描述:打开菜单中电池页

输入参数:void

返回值:void

29.void DrawBatterySButton(int \*num)

函数名:DrawBatterySButton

函数描述:绘制菜单中电池所有按钮

输入参数:int

返回值:void

30.void DrawBatteryChangePart(int\* num)

函数名:DrawBatteryChangePart

函数描述:绘制电池页的可变部分

输入参数:int\* num

返回值:void

31.void BatteryPlus(int \*num)

函数名:BatteryPlus

函数描述:电池页的充电功能

输入参数:int\* num

返回值:void

32.int BatteryMouse(int\* num)

函数名:BatteryMouse

函数描述:电池页的鼠标功能

输入参数:int\* num

返回值:int

ac.h

33.void OpenAircondition(void)

函数名:OpenAircondition

函数描述:打开菜单中空调页

输入参数:void

返回值:void

34.void DrawAirconditionChangePart(int\* num)

函数名:DrawAirconditionChangePart

函数描述:绘制雨刷页的可变部分

输入参数:int\* num

返回值:void

35.void DrawAirconditionSButton(int \*num)

函数名:DrawAirconditionSButton

函数描述:绘制菜单中空调所有按钮

输入参数:void

返回值:void

36.int AirconditionMouse(int\* num)

函数名:AirconditionMouse

函数描述:空调页鼠标功能

输入参数:int\* num

返回值:int

37.void Aircondition(int\* num)

函数名:Aircondition

函数描述:空调页温度湿度的改变

输入参数:int\* num

返回值:void

light.h

38.void OpenLight(void)

函数名:OpenLight

函数描述:打开菜单中灯光页

输入参数:void

返回值:void

39.void DrawLightSButton(void)

函数名:DrawLightSButton

函数描述:绘制菜单中灯光所有按钮

输入参数:void

返回值:void

40.void DrawLightChangePart(int\* num)

函数名:DrawLightChangePart

函数描述:绘制灯光页的可变部分

输入参数:int\* num

返回值:void

41.void PutLightHz(void)

函数名:PutLightHz

函数描述:显示灯光页汉字

输入参数:void

返回值:void

42.int LightMouse(int\* num)

函数名:LightMouse

函数描述:灯光页的鼠标功能

输入参数:int\* num

返回值:int

door.h

43.void OpenDoor(void)

函数名:OpenDoor

函数描述:打开菜单中车门页

输入参数:void

返回值:void

44.void DrawDoorSButton(void)

函数名:DrawDoorSButton

函数描述:绘制菜单中车门所有按钮

输入参数:void

返回值:void

45.void DrawDoorChangePart(int\* num)

函数名:DrawDoorChangePart

函数描述:绘制车门页的可变部分

输入参数:int\* num

返回值:void

46.void PutDoorHz(void)

函数名:PutDoorHz

函数描述:显示车门页汉字

输入参数:void

返回值:void

47.int DoorMouse(int\* num)

函数名:DoorMouse

函数描述:车门页鼠标

输入参数:int\* num

返回值:int

window.h

48.void OpenWindow(void)

函数名:OpenWindow

函数描述:打开车窗页

输入参数:void

返回值:void

49.void DrawOneWindow(int x, int y)

函数名:DrawOneWindow

函数描述:绘制一个车窗

输入参数:int x, int y

返回值:void

50.void DrawWindowChangePart(int\* num)

函数名:DrawWindowChangePart

函数描述:绘制车窗页面可变的部分

输入参数:int\* num

返回值:void

51.void PutWindowHz(void)

函数名:PutWindowHz

函数描述:显示车窗页汉字

输入参数:void

返回值:void

52.int WindowMouse(int\* num)

函数名:WindowMouse

函数描述:车窗页的鼠标功能

输入参数:int\*

返回值:int

wiper.h

53.void OpenWiper(void)

函数名:OpenWiper

函数描述:打开雨刷页

输入参数:void

返回值:void

54.void DrawAheadWiper(void)

函数名:DrawAheadWiper

函数描述:绘制前雨刷

输入参数:void

返回值:void

55.void DrawBehindWiper(void)

函数名:DrawBehindWiper

函数描述:绘制后雨刷

输入参数:void

返回值:void

56.void DrawWiperChangePart(int\* num)

函数名:DrawWiperChangePart

函数描述:绘制雨刷页的可变部分

输入参数:int\* num

返回值:void

57.void PutWiperHz(void)

函数名:PutWiperHz

函数描述:显示雨刷页汉字

输入参数:void

返回值:void

58.int WiperMouse(int\* num)

函数名:WiperMouse

函数描述:雨刷页的鼠标功能

输入参数:int\*

返回值:int

sunroof.h

59.void OpenSunroof(void)

函数名:OpenSunroof

函数描述:打开天窗页

输入参数:void

返回值:void

60.void DrawPartSunroofButton(void)

函数名:DrawPartSunroofButton

函数描述:绘制天窗页部分按钮

输入参数:void

返回值:void

61.void DrawSunroof(void)

函数名:DrawSunroof

函数描述:绘制天窗

输入参数:void

返回值:void

62.void DrawSunroofChangePart(int\* num)

函数名:DrawSunroofChangePart

函数描述:绘制天窗页变化部分

输入参数:int"\*

返回值:void

63.void PutSunroofHz(void)

函数名:PutSunroofHz

函数描述:显示天窗页文本

输入参数:void

返回值:void

64.int SunroofMouse(int\* num)

函数名:SunroofMouse

函数描述:天窗页鼠标

输入参数:int\*

返回值:int

tire.h

65.void OpenTire(void)

函数名:OpenTire

函数描述:打开胎压页

输入参数:void

返回值:void

66.void DrawOneTire(int x, int y)

函数名:DrawOneTire

函数描述:绘制一个轮胎

输入参数:int x, int y

返回值:void

67.void PutTireBar(int x, int y, int bar)

函数名:PutTireBar

函数描述:显示胎压

输入参数:int x, int y

返回值:void

68.void DrawTireCar(void)

函数名:DrawTireCar

函数描述:绘制胎压页汽车

输入参数:void

返回值:void

69.void DrawTireChangePart(int\* num)

函数名:DrawTireChangePart

函数描述:绘制胎压页的可变部分

输入参数:int\* num

返回值:void

70.int TireMouse(int\* num)

函数名:TireMouse

函数描述:胎压页鼠标

输入参数:int\* num

返回值:int

display.h

71.void OpenDisplay(void)

函数名:OpenDisplay

函数描述:打开显示界面

输入参数:void

返回值:void

72.void DrawDisplaySButton(void)

函数名:DrawDisplaySButton

函数描述:绘制显示中所有按钮

输入参数:void

返回值:void

73.void DrawDailnumber(void)

函数名:DrawDailnumber

函数描述:绘制表盘等上的数据（如温度，湿度等）

输入参数:void

返回值:void

74.void DrawCar(int x,int y,float k)//x,y确定位置，k可确定大小

函数名:DrawCar

函数描述:绘制车

输入参数:int

返回值:void

75.void DrawArrow(void)

函数名:DrawArrow

函数描述:绘制箭头

输入参数:void

返回值:void

76.void DrawTime(int x,int y,int k)//x,y确定位置，k可确定大小

函数名:DrawTime

函数描述:显示时间

输入参数:int x,int y,int k

返回值:void

77.void DrawDisplayChangePart(int \*num)

函数名:DrawDisplayChangePart

函数描述:显示页可变部分

输入参数:int\* num

返回值:int

78.int DisplayMouse(int\* num)

函数名:DisplayMouse

函数描述:显示页鼠标

输入参数:int\* num

返回值:int

79.void Run(void)

函数名:Run

函数描述:运行

输入参数:void

返回值:void

80.void DrawBackground(int x,int y,float k)

函数名:DrawBackground

函数描述:画背景

输入参数:void

返回值:void

map.h

81.void DrawMap(void)

函数名:DrawMap

函数描述:绘制地图

输入参数:void

返回值:void

82.void DrawRoad(void)

函数名:DrawRoad

函数描述:绘制道路

输入参数:void

返回值:void

83.void DrawParallelRoad(int x, int y, int d)

函数名:DrawParallelRoad

函数描述:绘制横向道路

输入参数:int x, int y, int d

返回值:void

84.void DrawVerticalRoad(int x, int y, int d)

函数名:DrawVerticalRoad

函数描述:绘制纵向道路

输入参数:int x, int y, int d

返回值:void

85.void DrawCrossing(int x, int y, int north, int south, int west, int east)

函数名:DrawCrossing

函数描述:绘制十字路口

输入参数:int x, int y, int north, int south, int west, int east

返回值:void

86.void DrawBuilding(void)

函数名:DrawBuilding

函数描述:绘制建筑物

输入参数:void

返回值:void

87.void DrawSouthBuildingOne(void)

函数名:DrawSouthBuildingOne

函数描述:绘制南一楼

输入参数:void

返回值:void

88.void DrawParkingLot(void)

函数名:DrawParkingLot

函数描述:绘制

输入参数:void

返回值:void

navigation.h

89.void OpenNavigation(void)

函数名:OpenNavigation

函数描述:打开导航页

输入参数:void

返回值:void

90.void DrawNavigationTopButton(void)

函数名:DrawNavigationTopButton

函数描述:绘制导航页顶部按钮

输入参数:void

返回值:void

91.int NavigationMouse(int\* mode)

函数名:NavigationMouse

函数描述:导航页鼠标

输入参数:int\* mode

返回值:int

92.int ChoosePlace(void)

函数名:ChoosePlace

函数描述:选择地点

输入参数:void

返回值:int

plan.h

93.int PlanPath(int start, int end)

函数名:PlanPath

函数描述:规划路线

输入参数:int start, int end

返回值:int

94.int Dijkstra(int start, int end, int distance[][34])

函数名:Dijkstra

函数描述:实现迪杰斯特拉算法

输入参数:int start, int end, int distance[][34]

返回值:int

95.int MinDistance(int\* start\_distance, int\* vertex)

函数名:MinDistance

函数描述:返回尚未包含在最短路径树中的最小距离的顶点的索引

输入参数:int\* distance, int\* vertex

返回值:int

96.void DrawRoute(void)

函数名:DrawRoute

函数描述:绘制路线

输入参数:void

返回值:void

97.voidReturnCoordinates(int i, int\* place, int\* x, int\* y)

函数名:ReturnCoordinates

函数描述:返回路线的坐标

输入参数:int i, int\* place, int\* x, int\* y

返回值:void

98.void DrawStartEnd(int num)

函数名:DrawStartEnd

函数描述:绘制起点或终点处的路线

输入参数:int num

返回值:void

page.h

99.void DrawMenuButton(void)

函数名:DrawMenuButton

函数描述:绘制菜单按钮

输入参数:void

返回值:void

100.void DrawDisplayButton(void)

函数名:DrawDisplayButton

函数描述:绘制显示按钮

输入参数:void

返回值:void

101.void DrawNavigationButton(void)

函数名:DrawNavigationButton

函数描述:绘制导航按钮

输入参数:void

返回值:void

102.int MainPageMouse(void)

函数名:MainPageMouse

函数描述:主页鼠标功能

输入参数:void

返回值:void

hz.h

103. void puthz(int x, int y,char \*s,int flag,int part,int color)

函数名:puthz

函数描述:显示汉字

输入参数: int x, int y,char \*s,int flag,int part,int color

返回值:void

mouse.h

104.void mouseinit(void)

函数名:mouseinit

函数描述:初始化鼠标

输入参数:void

返回值:void

105.void mouse(int x, int y)

函数名:mouse

函数描述:画不同形态的鼠标

输入参数:int x, int y

返回值:void

106.void mread(int\* nx, int\* ny, int\* nbuttons)

函数名:mread

函数描述:获取最新的寄存器信息

输入参数: int\* nx, int\* ny, int\* nbuttons

返回值:void

107.void newmouse(int\* nx, int\* ny, int\* nbuttons)

函数名:newmouse

函数描述:鼠标状态发生变化则更新鼠标

输入参数: int\* nx, int\* ny, int\* nbuttons

返回值:void

108.void save\_bk\_mou(int nx, int ny)

函数名:save\_bk\_mou

函数描述:存鼠标背景

输入参数: int nx, int ny

返回值:void

109.void clrmous(int nx, int ny)

函数名:clrmous

函数描述:清除鼠标

输入参数: int nx, int ny

返回值:void

110.void drawmous(int nx, int ny)

函数名:drawmous

函数描述:绘制鼠标

输入参数: int nx, int ny

返回值:void

111.int mouse\_press(int x1, int y1, int x2, int y2)

函数名:mouse\_press

函数描述: 如果在框中点击，则返回1；在框中未点击，则返回2；不在框中则返回0

输入参数: int x1, int y1, int x2, int y2

返回值:void

main.c

112.int main(void)

函数名:main

函数描述:主函数

输入参数:int

返回值:0

**3.源代码**

**main.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. \*   版本号：1.0
4. \*   作者：X
5. \*   生成日期：2024-2-28
6. \*   说明：Main函数（主函数）
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
8. \*\*/
9. #include<graphics.h>
10. #include"init.h"
11. #include"page.h"
12. int main()
13. {
14. int gd = VGA, gm = VGAHI;
15. initgraph(&gd, &gm, "..\\BORLANDC\\BGI");
16. while(1)
17. {
18. ClearGraph(WHITE);
19. OpenMainPage();
20. }
21. }

**init.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
3. \*   版本号：1.0
4. \*   作者：X
5. \*   生成日期：2024—4—15
6. \*   说明：Init类函数包含初始化，退出，返回等基本函数
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
8. \*\*/
9. #include<graphics.h>
10. #include"init.h"
11. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
12. FUNCTION:ClearGraph
13. DESCRIBETION:清空屏幕，用指定颜色填充，重新初始化鼠标
14. INPUT:color
15. RETURN:void
16. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
17. void ClearGraph(int color)
18. {
19. mouseinit();
20. setfillstyle(1, color);
21. bar(0, 0, 640, 480);
22. }
23. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
24. FUNCTION:DrawCloseButton
25. DESCRIBETION:绘制关闭按钮
26. INPUT:void
27. RETURN:void
28. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
29. void DrawCloseButton()
30. {
31. setcolor(BLUE);
32. setlinestyle(0, 0, 1);
33. rectangle(610,0,639,29);
34. line(610,0,639,29);
35. line(610,29,639,0);
36. }
37. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
38. FUNCTION:DrawReturnButton
39. DESCRIBETION:绘制退出按钮
40. INPUT:void
41. RETURN:void
42. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
43. void DrawReturnButton()
44. {
45. setcolor(BLUE);
46. setlinestyle(0, 0, 1);
47. rectangle(0,0,29,29);
48. line(5,15,25,15);
49. line(5,15,15,5);
50. line(5,15,15,25);
51. }
52. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
53. FUNCTION:Draw
54. DESCRIBETION:绘制设置
55. INPUT:void
56. REUTRUN:void
57. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
58. void Draw(int color)
59. {
61. setbkcolor(WHITE);
62. setcolor(color);
63. setlinestyle(0,0,1);
64. setviewport(0,0,639,479,0);*//set*
65. }

**page.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \*   版本号：1.0
4. \*   作者：X
5. \*   生成日期：2024—3—10
6. \*   说明：Page类函数实现打开页面功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include<graphics.h>
10. #include"display.h"
11. #include"hz.h"
12. #include"menu.h"
13. #include"mouse.h"
14. #include"navigat.h"
15. #include"page.h"
16. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
17. FUNCTION:OpenMainPage
18. DESCRIBETION:打开主页
19. INPUT:void
20. RETURN:void
21. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
22. void OpenMainPage()
23. {
24. while(1)
25. {
26. int main\_page\_ret = 1;
27. ClearGraph(WHITE);
28. DrawCloseButton();
29. DrawMenuButton();
30. DrawDisplayButton();
31. DrawNavigationButton();
32. main\_page\_ret = MainPageMouse();
33. if(!main\_page\_ret)
34. return ;
35. }
37. }
38. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
39. FUNCTION:DrawMenuButton
40. DESCRIBETION:绘制菜单按钮
41. INPUT:void
42. RETURN:void
43. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
44. void DrawMenuButton()
45. {
46. setcolor(BLUE);
47. setlinestyle(0, 0, 1);
48. setfillstyle(1, BLUE);
49. rectangle(80, 40, 240, 230);
50. line(80, 200, 240, 200);
51. puthz(117, 203, "菜单", 24, 62, BLUE);
52. fillellipse(124, 72, 12, 12);
53. fillellipse(124, 120, 12, 12);
54. fillellipse(124, 168, 12, 12);
55. bar(144, 60, 220, 84);
56. bar(144, 108, 220, 132);
57. bar(144, 156, 220, 180);
58. }
59. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
60. FUNCTION:DrawDisplayButton
61. DESCRIBETION:绘制显示按钮
62. INPUT:void
63. RETURN:void
64. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
65. void DrawDisplayButton()
66. {
67. setcolor(BLUE);
68. setlinestyle(0, 0, 1);
69. setfillstyle(1, BLUE);
70. rectangle(400, 40, 560, 230);
71. line(400, 200, 560, 200);
72. puthz(437, 203, "显示", 24, 62, BLUE);
73. line(430, 60, 530, 60);
74. line(430, 180, 530, 180);
75. line(420, 70, 420, 170);
76. line(540, 70, 540, 170);
77. ellipse(430, 70, 90, 180, 10, 10);
78. ellipse(530, 70, 0, 90, 10, 10);
79. ellipse(430, 170, 180, 270, 10, 10);
80. ellipse(530, 170, 270, 360, 10, 10);
81. line(480, 180, 540, 100);
82. floodfill(410, 50, BLUE);
83. }
84. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
85. FUNCTION:DrawNavigationButton()
86. DESCRIBETION:绘制导航按钮
87. INPUT:void
88. RETURN:void
89. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
90. void DrawNavigationButton()
91. {
92. int point[10] = {160, 300, 100, 420, 160, 390, 220, 420, 160, 300};
93. setcolor(BLUE);
94. setlinestyle(0, 0, 1);
95. setfillstyle(1, BLUE);
96. rectangle(80, 280, 240, 470);
97. line(80, 440, 240, 440);
98. puthz(117, 443, "导航", 24, 62, BLUE);
99. drawpoly(5, point);
100. floodfill(160, 360, BLUE);
101. }
102. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
103. FUNCTION:MainPageMouse
104. DESCRIBETION:主页鼠标
105. INPUT:void
106. RETURN:int
107. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
108. int MainPageMouse()
109. {
110. int page[5] = {0, 0, 0, 0, 0};
111. while(1)
112. {
113. *//绘制鼠标和设置按钮*
114. newmouse(&MouseX, &MouseY, &press);
115. page[1] = mouse\_press(80, 40, 240, 230);
116. page[2] = mouse\_press(400, 40, 560, 230);
117. page[3] = mouse\_press(80, 280, 240, 470);
118. page[0] = mouse\_press(610, 0, 639, 29);
119. *//判断选项*
120. if(page[1] == 1)
121. {
122. ClearGraph(WHITE);
123. OpenMenu();
124. break;
125. }
126. else if(page[2] == 1)
127. {
128. ClearGraph(WHITE);
129. OpenDisplay();
130. break;
131. }
132. else if(page[3] == 1)
133. {
134. ClearGraph(WHITE);
135. OpenNavigation();
136. break;
137. }
138. else if(page[0] == 1)
139. {
140. closegraph();
141. return 0;
142. }
143. }
144. return 1;
145. }

**menu.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \*   版本号：1.0
4. \*   作者：X
5. \*   生成日期：2024—3—19
6. \*   说明：Menu类函数实现菜单功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include"ac.h"
10. #include"battery.h"
11. #include"door.h"
12. #include"drive.h"
13. #include"init.h"
14. #include"light.h"
15. #include"menu.h"
16. #include"page.h"
17. #include"window.h"
18. #include"wiper.h"
19. #include"sunroof.h"
20. #include"tire.h"
21. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
22. FUNCTION:OpenMenu
23. DESCRIBETION:打开菜单页
24. INPUT:void
25. RETURN:void
26. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
27. void OpenMenu()
28. {
29. while(1)
30. {
31. int menu\_ret = 1;
32. *//绘制界面*
33. ClearGraph(WHITE);
34. DrawDriveButton();
35. DrawBatteryButton();
36. DrawAirconditionButton();
37. DrawLightButton();
38. DrawDoorButton();
39. DrawWindowButton();
40. DrawWiperButton();
41. DrawSunroofButton();
42. DrawTireButton();
43. DrawMenuBackButton();
44. menu\_ret = MenuMouse();
45. if(!menu\_ret)
46. return ;
47. }
48. }
49. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
50. FUNCTION:DrawDriveButton
51. DESCRIBETION:绘制驾驶按钮
52. INPUT:void
53. RETURN:void
54. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
55. void DrawDriveButton()
56. {
57. setcolor(BLUE);
58. setlinestyle(0, 0, 1);
59. rectangle(14, 20, 114, 220);
60. line(14, 180, 114, 180);
61. puthz(31, 188, "驾驶", 24, 42, BLUE);
62. setcolor(DARKGRAY);
63. setfillstyle(1, DARKGRAY);
64. circle(64, 100, 45);
65. circle(64, 100, 35);
66. circle(64, 98, 5);
67. ellipse(64, 100, 30, 150, 15, 15);
68. line(30, 92, 60, 92);
69. line(68, 92, 98, 92);
70. line(30, 104, 50, 104);
71. line(78, 104, 98, 104);
72. line(60, 114, 60, 135);
73. line(68, 114, 68, 135);
74. ellipse(50, 114, 0, 90, 10, 10);
75. ellipse(78, 114, 90, 180, 10, 10);
76. floodfill(64, 60, DARKGRAY);
77. floodfill(64, 90, DARKGRAY);
78. }
79. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
80. FUNCTION:DrawBatteryButton
81. DESCRIBETION:绘制电池按钮
82. INPUT:void
83. RETURN:void
84. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
85. void DrawBatteryButton()
86. {
87. int point[12] = {162, 30, 187, 30, 187, 25, 197, 25, 197, 30, 222, 30};
88. setcolor(BLUE);
89. setlinestyle(0, 0, 1);
90. rectangle(142, 20, 242, 220);
91. line(142, 180, 242, 180);
92. puthz(159, 188, "电池", 24, 42, BLUE);
93. setcolor(DARKGRAY);
94. setfillstyle(1, DARKGRAY);
95. line(147, 45, 147, 160);
96. line(157, 45, 157, 160);
97. line(162, 165, 222, 165);
98. line(162, 175, 222, 175);
99. line(227, 45, 227, 160);
100. line(237, 45, 237, 160);
101. line(162, 40, 222, 40);
102. drawpoly(6, point);
103. ellipse(222, 45, 0, 90, 5, 5);
104. ellipse(222, 45, 0, 90, 15, 15);
105. ellipse(162, 45, 90, 180, 5, 5);
106. ellipse(162, 45, 90, 180, 15, 15);
107. ellipse(162, 160, 180, 270, 5, 5);
108. ellipse(162, 160, 180, 270, 15, 15);
109. ellipse(222, 160, 270, 0, 5, 5);
110. ellipse(222, 160, 270, 0, 15, 15);
111. floodfill(192, 30, DARKGRAY);
112. setfillstyle(1, GREEN);
113. floodfill(192, 100, DARKGRAY);
114. }
115. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
116. FUNCTION:DrawAirconditionButton
117. DESCRIBETION:绘制空调按钮
118. INPUT:void
119. RETURN:void
120. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
121. void DrawAirconditionButton()
122. {
123. setcolor(BLUE);
124. setlinestyle(0, 0, 1);
125. setfillstyle(1, BLUE);
126. rectangle(270, 20, 370, 220);
127. line(270, 180, 370, 180);
128. puthz(287, 188, "空调", 24, 42, BLUE);
129. setcolor(LIGHTBLUE);
130. line(320, 75, 310, 65);
131. line(320, 75, 330, 65);
132. line(320, 125, 310, 135);
133. line(320, 125, 330, 135);
134. line(298, 112, 294, 126);
135. line(298, 112, 284, 108);
136. line(298, 87, 294, 73);
137. line(298, 87, 284, 91);
138. line(342, 87, 356, 91);
139. line(342, 87, 346, 73);
140. line(342, 112, 346, 127);
141. line(342, 112, 356, 109);
142. setlinestyle(0, 0, 3);
143. line(320, 60, 320, 140);
144. line(285, 80, 355, 120);
145. line(285, 120, 355, 80);
146. }
147. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
148. FUNCTION:DrawLightButton
149. DESCRIBETION:绘制灯光按钮
150. INPUT:void
151. RETURN:void
152. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
153. void DrawLightButton()
154. {
155. setcolor(BLUE);
156. setlinestyle(0, 0, 1);
157. rectangle(398, 20, 498, 220);
158. line(398, 180, 498, 180);
159. puthz(415, 188, "灯光", 24, 42, BLUE);
160. setcolor(DARKGRAY);
161. setfillstyle(1, DARKGRAY);
163. ellipse(448, 80, -60, 240, 45, 45);
164. line(425, 118, 425, 138);
165. line(471, 118, 471, 138);
166. line(425, 118, 471, 118);
167. ellipse(448, 138, 180, 360, 23, 23);
168. floodfill(448, 138, DARKGRAY);
169. setfillstyle(1, YELLOW);
170. floodfill(448, 80, DARKGRAY);
171. }
172. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
173. FUNCTION:DrawDoorButton
174. DESCRIBETION:绘制车门按钮
175. INPUT:void
176. RETURN:void
177. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
178. void DrawDoorButton()
179. {
180. int point\_out[12] = {571, 30, 531, 100, 531, 170, 621, 170, 621, 30, 571, 30};
181. int point\_in[10] = {571, 35, 536, 100, 616, 100, 616, 35, 571, 35};
182. setcolor(BLUE);
183. setlinestyle(0, 0, 1);
184. rectangle(526, 20, 626, 220);
185. line(526, 180, 626, 180);
186. puthz(543, 188, "车门", 24, 42, BLUE);
187. setcolor(DARKGRAY);
188. setfillstyle(1, DARKGRAY);
189. drawpoly(6, point\_out);
190. drawpoly(5, point\_in);
191. floodfill(576, 101, DARKGRAY);
192. }
193. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
194. FUNCTION:DrawWindowButton
195. DESCRIBETION:绘制车窗按钮
196. INPUT:void
197. RETURN:void
198. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
199. void DrawWindowButton()
200. {
201. int point\_out[12] = {59, 270, 19, 340, 19, 410, 109, 410, 109, 270, 59, 270};
202. int point\_in[10] = {59, 275, 24, 340, 104, 340, 104, 275, 59, 275};
203. setcolor(BLUE);
204. setlinestyle(0, 0, 1);
205. rectangle(14, 260, 114, 460);
206. line(14, 420, 114, 420);
207. puthz(31, 428, "车窗", 24, 42, BLUE);
208. setcolor(DARKGRAY);
209. drawpoly(6, point\_out);
210. drawpoly(5, point\_in);
211. setfillstyle(1, LIGHTCYAN);
212. floodfill(64, 339, DARKGRAY);
213. }
214. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
215. FUNCTION:DrawWiperButton
216. DESCRIBETION:绘制雨刷按钮
217. INPUT:void
218. RETURN:void
219. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
220. void DrawWiperButton()
221. {
222. setcolor(BLUE);
223. setlinestyle(0, 0, 1);
224. rectangle(142, 260, 242, 460);
225. line(142, 420, 242, 420);
226. puthz(159, 428, "雨刷", 24, 42, BLUE);
227. setcolor(DARKGRAY);
228. line(177, 365, 177, 340);
229. line(207, 365, 207, 340);
230. setlinestyle(0, 0, 3);
231. circle(207, 365, 2);
232. circle(177, 365, 2);
233. line(152, 300, 232, 300);
234. line(237, 305, 237, 355);
235. line(152, 360, 232, 360);
236. line(147, 305, 147, 355);
237. ellipse(232, 305, 0, 90, 5, 5);
238. ellipse(152, 305, 90, 180, 5, 5);
239. ellipse(152, 355, 180, 270, 5, 5);
240. ellipse(232, 355, 270, 360, 5, 5);
241. line(162, 355, 197, 320);
242. line(192, 355, 227, 320);
243. }
244. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
245. FUNCTION:DrawSunroofButton
246. DESCRIBETION:绘制天窗按钮
247. INPUT:void
248. RETURN:void
249. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
250. void DrawSunroofButton()
251. {
252. setcolor(BLUE);
253. setlinestyle(0, 0, 1);
254. rectangle(270, 260, 370, 460);
255. line(270, 420, 370, 420);
256. puthz(287, 428, "天窗", 24, 42, BLUE);
257. setcolor(DARKGRAY);
258. setlinestyle(0, 0, 3);
259. setfillstyle(1, LIGHTCYAN);
260. line(280, 270, 360, 270);
261. line(365, 275, 365, 405);
262. line(280, 410, 360, 410);
263. line(275, 275, 275, 405);
264. ellipse(360, 275, 0, 90, 5, 5);
265. ellipse(280, 275, 90, 180, 5, 5);
266. ellipse(280, 405, 180, 270, 5, 5);
267. ellipse(360, 405, 270, 360, 5, 5);
268. line(285, 300, 355, 300);
269. line(360, 305, 360, 375);
270. line(285, 380, 355, 380);
271. line(280, 305, 280, 375);
272. ellipse(355, 305, 0, 90, 5, 5);
273. ellipse(285, 305, 90, 180, 5, 5);
274. ellipse(285, 375, 180, 270, 5, 5);
275. ellipse(355, 375, 270, 360, 5, 5);
276. floodfill(320, 340, DARKGRAY);
277. }
278. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
279. FUNCTION:DrawTireButton
280. DESCRIBETION:绘制胎压按钮
281. INPUT:void
282. RETURN:void
283. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
284. void DrawTireButton()
285. {
286. setcolor(BLUE);
287. setlinestyle(0, 0, 1);
288. rectangle(398, 260, 498, 460);
289. line(398, 420, 498, 420);
290. puthz(415, 428, "胎压", 24, 42, BLUE);
291. setcolor(DARKGRAY);
292. setfillstyle(1, DARKGRAY);
293. circle(448, 340, 5);
294. circle(448, 340, 10);
295. circle(448, 340, 20);
296. circle(448, 340, 25);
297. circle(448, 340, 30);
298. circle(448, 340, 45);
299. floodfill(448, 347, DARKGRAY);
300. floodfill(448, 362, DARKGRAY);
301. floodfill(448, 380, DARKGRAY);
302. setlinestyle(0, 0, 3);
303. line(448, 330, 448, 320);
304. line(448, 350, 448, 360);
305. line(438, 340, 428, 340);
306. line(458, 340, 468, 340);
307. line(433, 355, 441, 347);
308. line(463, 325, 455, 333);
309. line(433, 325, 441, 333);
310. line(463, 355, 455, 347);
311. }
312. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
313. FUNCTION:DrawMenuBackButton
314. DESCRIBETION:绘制菜单页返回按钮
315. INPUT:void
316. RETURN:void
317. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
318. void DrawMenuBackButton()
319. {
320. int point\_up[14] = {596, 315, 541, 315, 541, 310, 531, 323, 541, 335, 541, 330, 596, 330};
321. int point\_down[8] = {596, 350, 531, 350, 531, 365, 596, 365};
322. setcolor(BLUE);
323. setlinestyle(0, 0, 1);
324. setfillstyle(1, BLUE);
325. rectangle(526, 260, 626, 460);
326. line(526, 420, 626, 420);
327. puthz(543, 428, "返回", 24, 42, BLUE);
329. setcolor(DARKGRAY);
330. setfillstyle(1, DARKGRAY);
331. drawpoly(7, point\_up);
332. drawpoly(4, point\_down);
333. ellipse(596, 340, -90, 90, 10, 10);
334. ellipse(596, 340, -90, 90, 25, 25);
335. floodfill(541, 318, DARKGRAY);
336. }
337. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
338. FUNCTION:MenuMouse
339. DESCRIBETION:菜单页鼠标
340. INPUT:void
341. RETURN:int
342. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
343. int MenuMouse()
344. {
345. int choice[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
346. while(1)
347. {
348. *//绘制鼠标和设置按钮*
349. newmouse(&MouseX, &MouseY, &press);
350. choice[1] = mouse\_press(14, 20, 114, 220);
351. choice[2] = mouse\_press(142, 20, 242, 220);
352. choice[3] = mouse\_press(270, 20, 370, 220);
353. choice[4] = mouse\_press(398, 20, 498, 220);
354. choice[5] = mouse\_press(526, 20, 626, 220);
355. choice[6] = mouse\_press(14, 260, 114, 460);
356. choice[7] = mouse\_press(142, 260, 242, 460);
357. choice[8] = mouse\_press(270, 260, 370, 460);
358. choice[9] = mouse\_press(398, 260, 498, 460);
359. choice[0] = mouse\_press(526, 260, 626, 460);
360. *//判断选项*
361. if(choice[1] == 1)
362. {
363. ClearGraph(WHITE);
364. OpenDrive();
365. break;
366. }
367. else if(choice[2] == 1)
368. {
369. ClearGraph(WHITE);
370. OpenBattery();
371. break;
372. }
373. else if(choice[3] == 1)
374. {
375. ClearGraph(WHITE);
376. OpenAircondition();
377. break;
378. }
379. else if(choice[4] == 1)
380. {
381. ClearGraph(WHITE);
382. OpenLight();
383. break;
384. }
385. else if(choice[5] == 1)
386. {
387. ClearGraph(WHITE);
388. OpenDoor();
389. break;
390. }
391. else if(choice[6] == 1)
392. {
393. ClearGraph(WHITE);
394. OpenWindow();
395. break;
396. }
397. else if(choice[7] == 1)
398. {
399. ClearGraph(WHITE);
400. OpenWiper();
401. break;
402. }
403. else if(choice[8] == 1)
404. {
405. ClearGraph(WHITE);
406. OpenSunroof();
407. break;
408. }
409. else if(choice[9] == 1)
410. {
411. ClearGraph(WHITE);
412. OpenTire();
413. break;
414. }
415. else if(choice[0] == 1)
416. {
417. return 0;
418. }
419. }
420. return 1;
421. }

**drive.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \* 版本号：1.0
4. \* 作者：Y
5. \* 生成日期：2024—3—10
6. \* 说明：Drive类函数实现驾驶功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include"drive.h"
10. #include"init.h"
11. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
12. FUNCTION:OpenDrive
13. DESCRIBETION:打开菜单中驾驶页
14. INPUT:void
15. REUTRUN:void
16. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
17. void OpenDrive()
18. {
19. while(1)
20. {
21. int drive\_ret = 1, num[5]={0}, i=0;
22. FILE\* drive\_text;
23. setfillstyle(1, WHITE);
24. bar(0, 0, 640, 480);
25. drive\_text=fopen("..\\CODE\\text\\drive.txt","rt");
26. for(i = 0; i < 5; i++)
27. {
28. fscanf(drive\_text, "%d " ,&num[i]);
29. }
30. fclose(drive\_text);
31. DrawReturnButton();
32. DrawDriveSButton();*//绘制*
33. DrawDriveChangePart(num);
34. PutDriveHz();
35. drive\_ret = DriveMouse(num);
37. drive\_text = fopen("..\\CODE\\text\\drive.txt", "wt");
38. for(i = 0; i < 5; i++)
39. {
40. fprintf(drive\_text, "%d ", num[i]);
41. }
42. fclose(drive\_text);
43. if(!drive\_ret)
44. return ;
45. }
46. }
47. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
48. FUNCTION:DrawDriveSButton
49. DESCRIBETION:绘制菜单中驾驶所有按钮
50. INPUT:void
51. REUTRUN:void
52. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
53. void DrawDriveSButton()
54. {
55. rectangle(0+10,140,195+10,180);
56. rectangle(0+10,215,195+10,255);
57. rectangle(0+10,290,195+10,330);
58. line(65+10,140,65+10,180);
59. line(130+10,140,130+10,180);
60. line(65+10,215,65+10,255);
61. line(130+10,215,130+10,255);
62. line(65+10,290,65+10,330);
63. line(130+10,290,130+10,330);*//方框*
64. line(33+10,295,33+10,325);
65. line(10+10,310,55+10,310);*//+*
66. line(140+10,310,185+10,310);*//-*
67. rectangle(340,155-10,610,425-10);
68. rectangle(430,155-10,520,425-10);
69. rectangle(340,245-10,610,335-10);
70. circle(475,290-10,40-10);
71. line(340,155-10,430,245-10);
72. line(475,240-10,475,160-10);
73. line(475,160-10,435,195-10);
74. line(475,160-10,515,195-10);
75. line(475,420-10,475,340-10);
76. line(475,420-10,435,385-10);
77. line(475,420-10,515,385-10);
78. line(345,290-10,425,290-10);
79. line(345,290-10,380,250-10);
80. line(345,290-10,380,330-10);*//car set*
81. line(525,290-10,605,290-10);
82. line(605,290-10,570,250-10);
83. line(605,290-10,570,330-10);
84. rectangle(150,380,150+100,380+50);
85. circle(150+50,380+25,20);
86. circle(150+50,380+75,20);
87. rectangle(150+30,380+30,150+70,380+70);
88. rectangle(150+30,380+40,150+60,380+50);
89. rectangle(150+40,380+40,150+70,380+50);*//绘制车*
90. }
91. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
92. FUNCTION:DrawDriveChangePart
93. DESCRIBETION:绘制驾驶页面可变的部分
94. INPUT:int\* num
95. RETURN:void
96. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
97. void DrawDriveChangePart(int\* num)
98. {
99. char \*p;
100. if(num[0]==1)
101. {
102. setfillstyle(1,YELLOW);
103. floodfill(1+10,141,BLUE);
104. }
105. if(num[0]==2)
106. {
107. setfillstyle(1,YELLOW);
108. floodfill(66+10,141,BLUE);
109. }
110. if(num[0]==3)
111. {
112. setfillstyle(1,YELLOW);
113. floodfill(136+10,141,BLUE);
114. }
115. if(num[1]==1)
116. {
117. setfillstyle(1,YELLOW);
118. floodfill(1+10,216,BLUE);
119. }
120. if(num[1]==2)
121. {
122. setfillstyle(1,YELLOW);
123. floodfill(66+10,216,BLUE);
124. }
125. if(num[1]==3)
126. {
127. setfillstyle(1,YELLOW);
128. floodfill(136+10,216,BLUE);
129. }
130. if(num[3]==1)
131. {
132. setfillstyle(1,YELLOW);
133. floodfill(421+10,156-10,BLUE);
134. }
135. if(num[3]==2)
136. {
137. setfillstyle(1,YELLOW);
138. floodfill(421+10,336-10,BLUE);
139. }
140. if(num[3]==3)
141. {
142. setfillstyle(1,YELLOW);
143. floodfill(331+10,246-10,BLUE);
144. }
145. if(num[3]==4)
146. {
147. setfillstyle(1,YELLOW);
148. floodfill(511+10,246-10,BLUE);
149. }
150. if(num[4]==1)
151. {
152. setfillstyle(1,YELLOW);
153. floodfill(421+10,246-10,BLUE);
154. }
155. if(num[4]==2)
156. {
157. setfillstyle(1,YELLOW);
158. floodfill(331+10,156-10,BLUE);
159. }
161. p=(int \*) malloc(11\*sizeof(int));
162. itoa(num[2],p,10);
163. settextstyle(3,0,1);
164. outtextxy(95+6,300,p);
165. free(p);
166. }
167. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
168. FUNCTION:PutDriveHz
169. DESCRIBETION:显示驾驶页汉字
170. INPUT:void
171. RETURN:void
172. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
173. void PutDriveHz()
174. {
175. puthz(250+10,35,"驾驶",48,54,BLUE);
176. puthz(45+10,115,"转动方式",24,27,BLUE);
177. puthz(25+10,190,"制动能量回收",24,27,BLUE);
178. puthz(70+10,265,"速度",24,27,BLUE);*//set*
179. puthz(15+10,155,"舒适",16,24,BLUE);
180. puthz(80+10,155,"标准",16,24,BLUE);
181. puthz(145+10,155,"运动",16,24,BLUE);
182. puthz(25+10,230,"低",16,24,BLUE);
183. puthz(80+10,230,"标准",16,24,BLUE);
184. puthz(155+10,230,"高",16,24,BLUE);*//档位*
185. }
186. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
187. FUNCTION:DriveMouse
188. DESCRIBETION:驾驶页的鼠标功能
189. INPUT:int\*
190. RETURN:int
191. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
192. int DriveMouse(int\* num)
193. {
194. int choice[15]={0};
195. while(1)
196. {
197. *//绘制鼠标和设置按钮*
198. newmouse(&MouseX,&MouseY,&press);
199. choice[0]=mouse\_press(0,0,29,29);
201. choice[1]=mouse\_press(0+10,140,65+10,180);
202. choice[2]=mouse\_press(65+10,140,130+10,180);
203. choice[3]=mouse\_press(130+10,140,195+10,180);
205. choice[4]=mouse\_press(0+10,215,65+10,255);
206. choice[5]=mouse\_press(65+10,215,130+10,255);
207. choice[6]=mouse\_press(130+10,215,195+10,255);
209. choice[7]=mouse\_press(0+10,290,65+10,330);
210. choice[8]=mouse\_press(130+10,290,195+10,330);
212. choice[9]=mouse\_press(420+10,155-10,510+10,245-10);
213. choice[10]=mouse\_press(420+10,335-10,510+10,425-10);
214. choice[11]=mouse\_press(330+10,245-10,420+10,335-10);
215. choice[12]=mouse\_press(510+10,245-10,600+10,335-10);
216. choice[13]=mouse\_press(420+10,245-10,510+10,335-10);
217. choice[14]=mouse\_press(330+10,155-10,420+10,245-10);
218. *//判断选项*
219. if(choice[0] == 1)  *//返回*
220. {
221. ClearGraph(WHITE);
222. return 0;
223. }
224. else if(choice[1] == 1) *//舒适*
225. {
226. num[0]=1;
227. break;
228. }
229. else if(choice[2] == 1) *//标准*
230. {
231. num[0]=2;
232. break;
233. }
234. else if(choice[3] == 1) *//运动*
235. {
236. num[0]=3;
237. break;
238. }
239. else if(choice[4] == 1) *//低*
240. {
241. num[1]=1;
242. break;
243. }
244. else if(choice[5] == 1) *//标准*
245. {
246. num[1]=2;
247. break;
248. }
249. else if(choice[6] == 1) *//高*
250. {
251. num[1]=3;
252. break;
253. }
254. else if(choice[7] == 1) *//速度增加*
255. {
256. if(num[2]<99&&num[4]==1)
257. num[2]++;
258. break;
259. }
260. else if(choice[8] == 1) *//速度减少*
261. {
262. if(num[2]>0&&num[4]==1)
263. num[2]--;
264. break;
265. }
266. else if(choice[9] == 1) *//前进*
267. {
268. num[4]=1;
269. num[3]=1;
270. break;
271. }
272. else if(choice[10] == 1) *//后退*
273. {
274. num[4]=1;
275. num[3]=2;
276. break;
277. }
278. else if(choice[11] == 1) *//左转*
279. {
280. num[4]=1;
281. num[3]=3;
282. break;
283. }
284. else if(choice[12] == 1) *//右转*
285. {
286. num[4]=1;
287. num[3]=4;
288. break;
289. }
290. else if(choice[13] == 1) *//启动*
291. {
292. num[4]=1;
293. break;
294. }
295. else if(choice[14] == 1) *//刹车*
296. {
297. num[4]=2;
298. num[3]=0;
299. num[2]=0;
300. break;
301. }
302. }
303. delay(100);
304. return 1;
305. }

**battery.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \* 版本号：1.0
4. \* 作者：Y
5. \* 生成日期：2024—3—17
6. \* 说明：Battery类函数实现电池功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include"init.h"
10. #include"battery.h"
11. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
12. FUNCTION:OpenBattery
13. DESCRIBETION:打开菜单中电池页
14. INPUT:void
15. REUTRUN:void
16. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
17. void OpenBattery()
18. {
19. while(1)
20. {
21. int battery\_ret = 1, num[4], i = 0;
22. FILE\* battery\_text;
23. setfillstyle(1, WHITE);
24. bar(0, 0, 640, 480);
25. battery\_text = fopen("..\\CODE\\text\\battery.txt", "rt");
26. for(i = 0; i < 4; i++)
27. {
28. fscanf(battery\_text, "%d ", &num[i]);
29. }
30. fclose(battery\_text);
31. DrawBatterySButton(num);
32. DrawBatteryChangePart(num);
33. puthz(305-40,50,"电池充电",24,27,BLUE);
34. battery\_ret = BatteryMouse(num);
36. battery\_text = fopen("..\\CODE\\text\\battery.txt", "wt");
37. for(i = 0; i < 4; i++)
38. {
39. fprintf(battery\_text, "%d ", num[i]);
40. }
41. fclose(battery\_text);
42. if(!battery\_ret)
43. return ;
44. }
45. }
46. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
47. FUNCTION:DrawBatterySButton
48. DESCRIBETION:绘制菜单中电池所有按钮
49. INPUT:int
50. REUTRUN:void
51. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
52. void DrawBatterySButton(int \*num)
53. {
54. char p[5];
55. int i;
56. Draw(BLUE);
58. rectangle(280-40,20,430-40,100);
59. circle(65,115,10);
60. circle(65,115,6);
61. circle(420,115,10);
62. circle(420,115,6);
64. puthz(85,105,"电池充满模式",24,27,BLUE);
65. puthz(440,105,"电池养护模式",24,27,BLUE);*//set*
67. line(60,150,615,150);
68. line(60,385,615,385);
69. line(45,165,45,370);
70. line(630,165,630,370);
71. ellipse(60,165,90,180,15,15);
72. ellipse(60,370,180,270,15,15);
73. ellipse(615,165,0,90,15,15);
74. ellipse(615,370,-90,0,15,15);
75. rectangle(170,170,605,360);
76. line(305,170,305,360);
77. line(170,230,605,230);
78. line(170,280,605,280);
79. puthz(190,185,"电池容量",24,27,BLUE);
80. puthz(330,185,"可充至",24,27,BLUE);
81. puthz(190,240,"电池寿命",24,27,BLUE);
82. puthz(190,305,"推荐对象",24,27,BLUE);
83. DrawReturnButton();
84. setcolor(DARKGRAY);
85. setfillstyle(1, DARKGRAY);
86. line(105,190,120,190);
87. line(105,198,120,198);
88. ellipse(105,194,90,270,5,4);
89. ellipse(120,194,-90,90,5,4);
90. floodfill(105,194,DARKGRAY);
91. line(95,200,130,200);
92. line(95,340,130,340);
93. line(85,210,85,330);
94. line(140,210,140,330);
95. ellipse(95,210,90,180,10,10);
96. ellipse(95,330,180,270,10,10);
97. ellipse(130,330,-90,0,10,10);
98. ellipse(130,210,0,90,10,10);
99. line(97,211,128,211);
100. line(97,329,128,329);
101. line(95,213,95,327);
102. line(130,213,130,327);
103. ellipse(97,213,90,180,2,2);
104. ellipse(97,327,180,270,2,2);
105. ellipse(128,327,-90,0,2,2);
106. ellipse(128,213,0,90,2,2);
107. floodfill(95,210,DARKGRAY);
108. setcolor(GREEN);
109. setfillstyle(1,GREEN);
111. itoa(num[0],p,10);
112. settextstyle(3,0,3);
113. outtextxy(80,345,p);
114. outtextxy(125,345,"%");
116. for(i=0;i< num[0]/10;i++)
117. {
118. rectangle(97,216+11\*(9-i),128,225+11\*(9-i));
119. floodfill(98,217+11\*(9-i),GREEN);
120. }
121. }
122. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
123. FUNCTION:DrawBatteryChangePart
124. DESCRIBETION:绘制电池页的可变部分
125. INPUT:int\* num
126. RETURN:void
127. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
128. void DrawBatteryChangePart(int\* num)
129. {
130. int i;
131. if(num[1] == 1)
132. {
133. setfillstyle(1, GREEN);
134. floodfill(65,115,BLUE);
135. settextstyle(3,0,3);
136. outtextxy(415,180,"100%");
137. puthz(330,240,"较长",24,27,BLUE);
138. puthz(320,290,"不会对电池造成永久性",16,27,BLUE);
139. puthz(320,312,"损害，长期满充可能会",16,27,BLUE);
140. puthz(320,334,"降低电池性能。",16,27,BLUE);
141. }
142. if(num[1] == 2)
143. {
144. setfillstyle(1,GREEN);
145. floodfill(420,115,BLUE);
146. settextstyle(3,0,3);
147. outtextxy(415,180,"80%");
148. puthz(330,240,"更长",24,27,BLUE);
149. puthz(320,290,"经常使用外接电源供电，",24,27,BLUE);
150. puthz(320,320,"续航能力要求并不太高",24,27,BLUE);*//表格*
151. }
152. }
153. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
154. FUNCTION:BatteryPlus
155. DESCRIBETION:电池页的充电功能
156. INPUT:int\* num
157. REUTRUN:void
158. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
159. void BatteryPlus(int \*num)
160. {
161. int i,t,mode;
162. char p[5],q[5];
164. if(num[1]==1)
165. {
166. mode=100;
167. }
168. if(num[1]==2)
169. {
170. mode=80;
171. }
173. while(num[2])
174. {
175. t=(mode-num[0])/10.0;
176. if(t<=0)
177. {
178. t=0;
179. }
180. num[3]=t;
181. setcolor(WHITE);
182. setfillstyle(1, WHITE);
183. bar(80,345,125,380);
184. bar(150,395,180,500);
185. itoa(num[3],p,10);
186. itoa(num[0],q,10);
188. setcolor(GREEN);
189. settextstyle(3,0,3);
190. setfillstyle(1, GREEN);
191. floodfill(320,35,BLUE);
192. puthz(40,400,"还需充电",24,27,BLUE);
193. puthz(180,400,"小时",24,27,BLUE);
194. outtextxy(80,345,q);
195. outtextxy(150,395,p);
197. delay(100);
199. if(num[2] && num[0]<mode)
200. {
201. num[0]++;
203. for(i=0;i< num[0]/10;i++)
204. {
205. setcolor(GREEN);
206. setfillstyle(1, GREEN);
207. rectangle(97,216+11\*(9-i),128,225+11\*(9-i));
208. floodfill(98,217+11\*(9-i),GREEN);
209. }
210. if(num[0]>=mode)
211. {
212. num[2]=0;
213. }
214. }
215. else
216. {
217. num[2]=0;
218. }
219. sleep(1);
220. }
221. }
222. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
223. FUNCTION:BatteryMouse
224. DESCRIBETION:电池页的鼠标功能
225. INPUT:int\* num
226. REUTRUN:int
227. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
228. int BatteryMouse(int\* num)
229. {
230. int choice[10]={0};
231. while(1)
232. {
233. *//绘制鼠标和设置按钮*
234. newmouse(&MouseX, &MouseY, &press);
236. choice[1]= mouse\_press(55,105,75,125);
237. choice[2]= mouse\_press(410,105,430,125);
238. choice[3]= mouse\_press(300,40,380,80);
239. choice[0]= mouse\_press(0,0,29,29);
240. *//判断选项*
241. if(choice[0] == 1)
242. {
243. ClearGraph(WHITE);
244. return 0;
245. }
246. else if(choice[1] == 1)
247. {
248. num[1]=1;
249. break;
250. }
251. else if(choice[2] == 1)
252. {
253. num[1]=2;
254. break;
255. }
256. else if(choice[3] == 1)
257. {
258. num[2]^=1;
259. BatteryPlus(num);
260. break;
261. }
262. }
263. delay(100);
264. return 1;
265. }

**light.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \*   版本号：1.0
4. \*   作者：Y
5. \*   生成日期：2024—3—5
6. \*   说明：Light类函数实现灯光功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include"init.h"
10. #include"light.h"
11. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
12. FUNCTION:OpenLight
13. DESCRIBETION:打开菜单中灯光页
14. INPUT:void
15. REUTRUN:void
16. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
17. void OpenLight()
18. {
19. while(1)
20. {
21. int light\_ret = 1, i=0, num[10]={0};
22. FILE \*light\_text;
24. setfillstyle(1, WHITE);
25. bar(0, 0, 640, 480);
26. light\_text = fopen("..\\CODE\\text\\light.txt", "rt");
27. for(i = 0; i < 6; i++)
28. {
29. fscanf(light\_text, "%d ", &num[i]);
30. }
31. fclose(light\_text);
33. DrawLightSButton();
34. DrawLightChangePart(num);
35. PutLightHz();
36. light\_ret = LightMouse(num);
38. light\_text = fopen("..\\CODE\\text\\light.txt", "wt");
39. for(i = 0; i < 6; i++)
40. {
41. fprintf(light\_text, "%d ", num[i]);
42. }
43. fclose(light\_text);
44. if(!light\_ret)
45. return ;
46. }
47. }
48. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
49. FUNCTION:DrawLightSButton
50. DESCRIBETION:绘制菜单中灯光所有按钮
51. INPUT:void
52. REUTRUN:void
53. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
54. void DrawLightSButton()
55. {
56. Draw(BLUE);
57. line(130,85,480,85);
58. line(130,115,480,115);
59. ellipse(130,100,90,270,30,15);
60. ellipse(480,100,-90,90,30,15);
61. line(140,165,210,165);
62. line(140,195,210,195);
63. ellipse(140,180,90,270,30,15);
64. ellipse(210,180,-90,90,30,15);
65. line(420,165,490,165);
66. line(420,195,490,195);
67. ellipse(420,180,90,270,30,15);
68. ellipse(490,180,-90,90,30,15);*//out*
69. line(130,320,480,320);
70. line(130,350,480,350);
71. ellipse(130,335,90,270,30,15);
72. ellipse(480,335,-90,90,30,15);
73. line(130,400,200,400);
74. line(130,430,200,430);
75. ellipse(130,415,90,270,30,15);
76. ellipse(200,415,-90,90,30,15);
77. line(420,405,480,405);
78. line(420,435,480,435);
79. ellipse(420,420,90,270,30,15);
80. ellipse(480,420,-90,90,30,15);*//in*
82. line(217,85,217,115);
83. line(304,85,304,115);
84. line(391,85,391,115);
85. line(217,320,217,350);
86. line(304,320,304,350);
87. line(391,320,391,350);
89. DrawReturnButton();
90. }
91. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
92. FUNCTION:DrawLightChangePart
93. DESCRIBETION:绘制灯光页的可变部分
94. INPUT:int\* num
95. RETURN:void
96. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
97. void DrawLightChangePart(int\* num)
98. {
99. if(num[0] == 1)
100. {
101. setfillstyle(1, GREEN);
102. floodfill(130, 100, BLUE);
103. }
104. if(num[0] == 2)
105. {
106. setfillstyle(1, GREEN);
107. floodfill(221, 100, BLUE);
108. }
109. if(num[0] == 3)
110. {
111. setfillstyle(1, GREEN);
112. floodfill(311, 100, BLUE);
113. }
114. if(num[0] == 4)
115. {
116. setfillstyle(1, GREEN);
117. floodfill(401, 100, BLUE);
118. }
119. if(num[1] == 1)
120. {
121. setfillstyle(1, GREEN);
122. floodfill(130, 330, BLUE);
123. }
124. if(num[1] == 2)
125. {
126. setfillstyle(1, GREEN);
127. floodfill(220, 330, BLUE);
128. }
129. if(num[1] == 3)
130. {
131. setfillstyle(1, GREEN);
132. floodfill(310, 330, BLUE);
133. }
134. if(num[1] == 4)
135. {
136. setfillstyle(1, GREEN);
137. floodfill(400,330, BLUE);
138. }
139. if(num[2] == 1)
140. {
141. setfillstyle(1, GREEN);
142. floodfill(140, 180, BLUE);
143. }
144. if(num[3] == 1)
145. {
146. setfillstyle(1, GREEN);
147. floodfill(420, 180, BLUE);
148. }
149. if(num[4] == 1)
150. {
151. setfillstyle(1, GREEN);
152. floodfill(130, 415, BLUE);
153. }
154. if(num[5] == 1)
155. {
156. setfillstyle(1, GREEN);
157. floodfill(420, 420, BLUE);
158. }
159. }
160. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
161. FUNCTION:PutLightHz
162. DESCRIBETION:显示灯光页汉字
163. INPUT:void
164. RETURN:void
165. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
166. void PutLightHz()
167. {
168. puthz(120,30,"外部灯光",32,54,BLUE);
169. puthz(140,175,"照明灯",16,27,BLUE);
170. puthz(430,175,"雾灯",16,27,BLUE);
171. puthz(110,260,"内部灯光",32,54,BLUE);
172. puthz(130,405,"照明灯",16,27,BLUE);
173. puthz(420,415,"氛围灯",16,27,BLUE);
174. settextstyle(3,0,1);
175. outtextxy(130,85,"off        parking     auto         on");
176. outtextxy(130,320,"off        parking     auto         on");
177. }
178. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
179. FUNCTION:LightMouse
180. DESCRIBETION:灯光页的鼠标功能
181. INPUT:int\* num
182. REUTRUN:int
183. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
184. int LightMouse(int\* num)
185. {
186. int choice[13]={0};
187. while(1)
188. {
189. *//绘制鼠标和设置按钮*
190. newmouse(&MouseX, &MouseY, &press);
192. choice[1]= mouse\_press(130,85,217,115);
193. choice[2]= mouse\_press(217,85,304,115);
194. choice[3]= mouse\_press(304,85,391,115);
195. choice[4]= mouse\_press(391,85,480,115);*//out  auto....*
196. choice[5]= mouse\_press(130,320,217,350);
197. choice[6]= mouse\_press(217,320,304,350);
198. choice[7]= mouse\_press(304,320,391,350);
199. choice[8]= mouse\_press(391,320,480,350);*//in  auto....*
201. choice[9]= mouse\_press(140, 165, 210, 195);
202. choice[10]= mouse\_press(420, 165, 490, 195);
203. choice[11]= mouse\_press(130,400,200,430);
204. choice[12]= mouse\_press(420,405,480,435);
205. choice[0]= mouse\_press(0,0,29,29);
207. *//判断选项*
208. if(choice[0]==1)
209. {
210. *//返回*
211. ClearGraph(WHITE);
212. return 0;
213. }
214. else if(choice[1]==1)
215. {
216. *//off*
217. num[0]=1;
218. break;
219. }
220. else if(choice[2]==1)
221. {
222. *//parking*
223. num[0]=2;
224. break;
225. }
226. else if(choice[3]==1)
227. {
228. *//auto*
229. num[0]=3;
230. break;
231. }
232. else if(choice[4]==1)
233. {
234. *//on*
235. num[0]=4;
236. break;
237. }
238. else if(choice[5]==1)
239. {
240. *//off*
241. num[1]=1;
242. break;
243. }
244. else if(choice[6]==1)
245. {
246. *//parking*
247. num[1]=2;
248. break;
249. }
250. else if(choice[7]==1)
251. {
252. *//auto*
253. num[1]=3;
254. break;
255. }
256. else if(choice[8]==1)
257. {
258. *//on*
259. num[1]=4;
260. break;
261. }
262. else if(choice[9]==1)
263. {
264. *//out 照明灯*
265. num[2]^=1;
266. break;
267. }
268. else if(choice[10]==1)
269. {
270. *//out 雾灯*
271. num[3]^=1;
272. break;
273. }
274. else if(choice[11]==1)
275. {
276. *//in 照明灯*
277. num[4]^=1;
278. break;
279. }
280. else if(choice[12]==1)
281. {
282. *//in  氛围灯*
283. num[5]^=1;
284. break;
285. }
286. }
287. delay(100);
288. return 1;
289. }

**ac.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \* 版本号：1.0
4. \* 作者：Y
5. \* 生成日期：2024—3—18
6. \* 说明：Ac类函数实现空调功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include"ac.h"
10. #include"init.h"
11. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
12. FUNCTION:OpenAircondition
13. DESCRIBETION:打开菜单中空调页
14. INPUT:void
15. REUTRUN:void
16. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
17. void OpenAircondition()
18. {
19. while(1)
20. {
21. int aircondition\_ret = 1, i = 0;
22. int num[8];
23. FILE\* ac\_text;
24. setfillstyle(1, WHITE);
25. bar(0, 0, 640, 480);
26. ac\_text = fopen("..\\CODE\\text\\ac.txt", "rt");
27. for(i = 0; i < 8; i++)
28. {
29. fscanf(ac\_text, "%d ", &num[i]);
30. }
31. fclose(ac\_text);
32. DrawAirconditionSButton(num);
33. DrawAirconditionChangePart(num);
34. aircondition\_ret = AirconditionMouse(num);
35. ac\_text = fopen("..\\CODE\\text\\ac.txt", "wt");
36. for(i = 0; i < 8; i++)
37. {
38. fprintf(ac\_text, "%d ", num[i]);
39. }
40. fclose(ac\_text);
41. if(!aircondition\_ret)
42. return ;
43. }
44. }
45. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
46. FUNCTION:DrawAirconditionChangePart
47. DESCRIBETION:绘制雨刷页的可变部分
48. INPUT:int\* num
49. RETURN:void
50. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
51. void DrawAirconditionChangePart(int\* num)
52. {
53. if(num[0] == 1)
54. {
55. setfillstyle(1, GREEN);
56. floodfill(145,360, BLUE);
57. }
58. if(num[0] == 2)
59. {
60. setfillstyle(1,GREEN);
61. floodfill(472,360,BLUE);
62. }
63. if(num[1] == 1)
64. {
65. setfillstyle(1, GREEN);
66. floodfill(171,161,BLUE);
67. }
68. if(num[2] == 1)
69. {
70. setfillstyle(1, GREEN);
71. floodfill(471,161,BLUE);
72. }
73. }
74. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
75. FUNCTION:DrawAirconditionSButton
76. DESCRIBETION:绘制菜单中空调所有按钮
77. INPUT:void
78. REUTRUN:void
79. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
80. void DrawAirconditionSButton(int \*num)
81. {
82. float pi=3.1415926535,a,b;
83. char \*p;
84. int i,k;
85. a=-pi/2;
86. b=pi/100;
87. Draw(BLUE);
88. DrawReturnButton();
89. circle(170,200-40,20);
90. line(160,200-40,180,200-40);
91. circle(320,200-40,60);
92. circle(320,200-40,75);
93. circle(320,200-40,90);
94. circle(470,200-40,20);
95. line(460,200-40,480,200-40);
96. line(470,190-40,470,210-40);
97. setfillstyle(1,GREEN);
98. floodfill(320,200-40,BLUE);
99. circle(245-100,365-30,30);
100. puthz(220-100,400-30,"温度",24,27,BLUE);
101. ellipse(245-100,370-30,-230,50,10,10);
102. ellipse(245-100,370-30,-230,50,6,6);
103. line(250-100,368-30,250-100,363-30);
104. line(240-100,368-30,240-100,363-30);
105. line(250-100,363-30,240-100,363-30);
106. line(251-100,366-30,251-100,356-30);
107. line(239-100,366-30,239-100,356-30);
108. ellipse(245-100,356-30,0,180,6,6);
109. setfillstyle(1,DARKGRAY);
110. floodfill(245-100,370-30,BLUE);*//温度按钮*
111. circle(372+100,365-30,30);
112. ellipse(372+100,370-30,-180,0,8,8);
113. ellipse(372+100,370-30-4,-40,210,8,20);
114. ellipse(372+100,370-30,180,260,4,4);
115. puthz(352+100,400-30,"湿度",24,27,BLUE);*//湿度按钮*
117. if(num[0]==2)
118. {
119. k=(num[4]-20)/60.0\*200;
120. p=(int \*) malloc(11\*sizeof(int));
121. itoa(num[4],p,10);
123. settextstyle(3,0,5);
124. outtextxy(285,170-40,p);
125. settextstyle(3,0,3);
126. outtextxy(345,170-40,"%");
127. puthz(284,230-40,"预设湿度",16,20,BLUE);*//湿度显示*
128. free(p);
129. }
130. else
131. {
133. k=(num[3]-16)/14.0\*200;
134. p=(int \*) malloc(11\*sizeof(int));
135. itoa(num[3],p,10);
137. settextstyle(3,0,5);
138. outtextxy(285,170-40,p);
139. puthz(284,230-40,"预设温度",16,20,BLUE);
140. free(p);
142. }
144. for(i=0;i<k;i++)
145. {
146. line(320+75\*cos(a-b\*i),200-40-75\*sin(a-b\*i),320+90\*cos(a-b\*i),200-40-90\*sin(a-b\*i));
147. }
149. settextstyle(3,0,3);
150. circle(300,335,30);
151. ellipse(300,335,-180,0,15,15);
152. line(300,320,300,340);
153. puthz(275,370,"开关",24,27,BLUE);
154. puthz(50,400,"当前温度",24,27,BLUE);
156. p=(int \*) malloc(11\*sizeof(int));
157. itoa(num[5],p,10);
158. outtextxy(155,395,":");
159. outtextxy(165,395,p);
160. free(p);
162. puthz(205,400,"摄氏度",24,27,BLUE);
163. puthz(350,400,"当前湿度",24,27,BLUE);
165. p=(int \*) malloc(11\*sizeof(int));
166. itoa(num[6],p,10);
167. outtextxy(460,395,":");
168. outtextxy(470,395,p);
169. outtextxy(520,395,"%");
170. free(p);
172. }
173. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
174. FUNCTION:AirconditionMouse
175. DESCRIBETION:空调页鼠标功能
176. INPUT:int\* num
177. REUTRUN:int
178. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
179. int AirconditionMouse(int\* num)
180. {
181. int choice[10]={0};
183. while(1)
184. {
185. *//绘制鼠标和设置按钮*
186. newmouse(&MouseX, &MouseY, &press);
187. choice[1]= mouse\_press(150,140,190,180);
188. choice[2]= mouse\_press(450,140,490,180);
189. choice[3]= mouse\_press(115,305,175,365);
190. choice[4]= mouse\_press(442,305,502,365);
191. choice[5]= mouse\_press(270,305,330,365);
193. choice[0]= mouse\_press(0,0,29,29);
194. *//判断选项*
195. if(choice[0] == 1)
196. {
197. ClearGraph(WHITE);
198. return 0;
199. }
200. else if(choice[1] == 1)
201. {
202. num[1]^=1;
203. if(num[0]==1)
204. {
205. if(num[3]>16)
206. {
207. num[3]--;
208. }
209. }
210. if(num[0]==2)
211. {
212. if(num[4]>20)
213. {
214. num[4]--;
215. }
216. }
217. break;
218. }
219. else if(choice[2] == 1)
220. {
221. num[2]^=1;
222. if(num[0]==1)
223. {
224. if(num[3]<30)
225. {
226. num[3]++;
227. }
228. }
229. if(num[0]==2)
230. {
231. if(num[4]<80)
232. {
233. num[4]++;
234. }
235. }
236. break;
237. }
238. else if(choice[3] == 1)
239. {
240. num[0]=1;
241. break;
242. }
243. else if(choice[4] == 1)
244. {
245. num[0]=2;
246. break;
247. }
248. else if(choice[5] == 1)
249. {
250. num[7]^=1;
251. Aircondition(num);
252. break;
253. }
254. }
255. delay(100);
256. return 1;
257. }
258. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
259. FUNCTION:Aircondition
260. DESCRIBETION:空调页温度湿度的改变
261. INPUT:int\* num
262. REUTRUN:void
263. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
264. void Aircondition(int\* num)
265. {
266. char \*p;
267. int i;
269. while(num[7])
270. {
271. setfillstyle(1, GREEN);
272. floodfill(301,335,BLUE);
273. setfillstyle(1, WHITE);
274. Draw(BLUE);
275. if(num[3]>num[5])
276. {
277. for(i=0;i<=num[3]-num[5];i++)
278. {
279. bar(165,395,200,420);
280. num[5]++;
281. p=(int \*) malloc(11\*sizeof(int));
282. itoa(num[5],p,10);
283. outtextxy(155,395,":");
284. outtextxy(165,395,p);
285. free(p);
287. delay(1000);
288. }
289. }
290. else if(num[3]<num[5])
291. {
292. for(i=0;i<=num[5]-num[3];i++)
293. {
294. bar(165,395,200,420);
295. num[5]--;
297. p=(int \*) malloc(11\*sizeof(int));
298. itoa(num[5],p,10);
299. outtextxy(155,395,":");
300. outtextxy(165,395,p);
301. free(p);
303. delay(1000);
304. }
305. }
307. if(num[4]>num[6])
308. {
309. for(i=0;i<=num[4]-num[6];i++)
310. {
311. bar(470,395,520,420);
312. num[6]++;
314. p=(int \*) malloc(11\*sizeof(int));
315. itoa(num[6],p,10);
316. outtextxy(460,395,":");
317. outtextxy(470,395,p);
318. free(p);
320. outtextxy(520,395,"%");
321. delay(1000);
322. }
323. }
324. else if(num[4]<num[6])
325. {
326. for(i=0;i<=num[6]-num[4];i++)
327. {
328. bar(470,395,520,420);
329. num[6]--;
331. p=(int \*) malloc(11\*sizeof(int));
332. itoa(num[6],p,10);
333. outtextxy(460,395,":");
334. outtextxy(470,395,p);
335. free(p);
337. outtextxy(520,395,"%");
338. delay(1000);
339. }
340. }
342. if(num[3]==num[5] && num[4]==num[6])
343. {
344. num[7]=0;
345. }
346. }
347. }

**door.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \* 版本号：1.0
4. \* 作者：Y
5. \* 生成日期：2024—3—15
6. \* 说明：Door类函数实现车门功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include"init.h"
10. #include"display.h"
11. #include"door.h"
12. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
13. FUNCTION:OpenDoor
14. DESCRIBETION:打开菜单中车门页
15. INPUT:void
16. REUTRUN:void
17. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
18. void OpenDoor()
19. {
20. while(1)
21. {
22. int door\_ret = 1, num[9], i = 0;
23. FILE\* door\_text;
24. setfillstyle(1, WHITE);
25. bar(0, 0, 640, 480);
26. door\_text = fopen("..\\CODE\\text\\door.txt", "rt");
27. for(i = 0; i < 9; i++)
28. {
29. fscanf(door\_text, "%d ", &num[i]);
30. }
31. fclose(door\_text);
32. DrawDoorSButton();
33. DrawDoorChangePart(num);
34. PutDoorHz();
35. door\_ret = DoorMouse(num);
36. door\_text = fopen("..\\CODE\\text\\door.txt", "wt");
37. for(i = 0; i < 9; i++)
38. {
39. fprintf(door\_text, "%d ", num[i]);
40. }
41. fclose(door\_text);
42. if(!door\_ret)
43. return ;
44. }
45. }
46. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
47. FUNCTION:DrawDoorSButton
48. DESCRIBETION:绘制菜单中车门所有按钮
49. INPUT:void
50. REUTRUN:void
51. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
52. void DrawDoorSButton()
53. {
54. *//int v1[]={276,179,263,192,275,188},v2[]={379,180,390,195,379,189};*
55. Draw(BLUE);
57. DrawReturnButton();
59. puthz(290,45,"车门控制",24,27,BLUE);
60. line(140,270,180,270);
61. line(140,300,180,300);
62. ellipse(140,285,90,270,15,15);
63. ellipse(180,285,-90,90,15,15);
65. line(140,315,180,315);
66. line(140,345,180,345);
67. ellipse(140,330,90,270,15,15);
68. ellipse(180,330,-90,90,15,15);
69. line(150,323,156,323);
70. line(156,323,160,326);
71. line(160,326,160,333);
72. line(150,323,150,333);
73. line(150,333,152,333);
74. line(160,333,158,333);
75. circle(155,333,3);
76. floodfill(151,324,BLUE);
78. line(280,400,365,400);
79. line(280,430,365,430);
80. ellipse(280,415,90,270,15,15);
81. ellipse(365,415,-90,90,15,15);*//关闭按钮*
83. line(455,277,470,277);
84. line(455,293,470,293);
85. line(455,318,470,318);
86. line(455,334,470,334);
87. ellipse(455,285,90,270,8,8);
88. ellipse(455,326,90,270,8,8);
89. circle(470,285,8);
90. circle(470,326,8);
91. puthz(494,276,"前门手动模式",16,20,BLUE);
92. puthz(494,318,"后门手动模式",16,20,BLUE);
93. circle(400,216,10);
94. line(400,216,400-3,216-4);
95. line(400,216,400-3,216+4);
96. circle(400,272,10);
97. line(400,272,400-3,272-4);
98. line(400,272,400-3,272+4);
99. circle(252,216,10);
100. line(252,216,252+3,216-4);
101. line(252,216,252+3,216+4);
102. circle(252,272,10);
103. line(252,272,252+3,272-4);
104. line(252,272,252+3,272+4);*//set*
106. DrawCar(326,240,0.35);
107. }
108. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
109. FUNCTION:DrawDoorChangePart
110. DESCRIBETION:绘制车门页的可变部分
111. INPUT:int\* num
112. RETURN:void
113. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
114. void DrawDoorChangePart(int\* num)
115. {
116. int i;
117. if(num[0] == 1)
118. {
119. setfillstyle(1,DARKGRAY);
120. floodfill(160,285,BLUE);
121. num[1]=0;
122. num[2]=0;
123. num[5]=0;
124. num[6]=0;
125. num[7]=0;
126. num[8]=0;
127. }
128. if(num[1] == 1 )
129. {
130. setfillstyle(1, GREEN);
131. floodfill(140,316,BLUE);
132. num[0]=0;
133. num[2]=0;
134. num[5]=1;
135. num[6]=1;
136. num[7]=1;
137. num[8]=1;
138. }
139. if(num[2] == 1)
140. {
141. setfillstyle(1, GREEN);
142. floodfill(280,415,BLUE);
143. num[0]=0;
144. num[1]=0;
145. num[3]=0;
146. num[4]=0;
147. }
148. if(num[3] == 1)
149. {
150. setfillstyle(1, GREEN);
151. floodfill(455,285,BLUE);
152. num[0]=0;
153. num[1]=0;
154. }
155. if(num[4] == 1)
156. {
157. setfillstyle(1,GREEN);
158. floodfill(455,326,BLUE);
159. num[0]=0;
160. num[1]=0;
161. }
162. if(num[5] == 1)
163. {
164. setfillstyle(1, GREEN);
165. floodfill(401,217,BLUE);
166. }
167. if(num[6] == 1)
168. {
169. setfillstyle(1, GREEN);
170. floodfill(401,273,BLUE);
171. }
172. if(num[7] == 1)
173. {
174. setfillstyle(1, GREEN);
175. floodfill(252,217,BLUE);
176. }
177. if(num[8] == 1)
178. {
179. setfillstyle(1, GREEN);
180. floodfill(252,273,BLUE);
181. }
182. }
183. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
184. FUNCTION:PutDoorHz
185. DESCRIBETION:显示车门页汉字
186. INPUT:void
187. RETURN:void
188. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
189. void PutDoorHz()
190. {
191. puthz(300,410,"关闭",16,27,BLUE);
192. puthz(130,276,"一键关门",16,15,BLUE);
193. }
194. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
195. FUNCTION:DoorMouse
196. DESCRIBETION:车门页鼠标
197. INPUT:int\* num
198. REUTRUN:int
199. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
200. int DoorMouse(int\* num)
201. {
202. int choice[10]={0};
203. while(1)
204. {
205. *//绘制鼠标和设置按钮*
206. newmouse(&MouseX, &MouseY, &press);
207. choice[1]=mouse\_press(140,270,180,300);
208. choice[2]=mouse\_press(140,315,180,345);*//*
209. choice[3]=mouse\_press(265,400,380,430);*//*
210. choice[4]=mouse\_press(455,277,470,293);
211. choice[5]=mouse\_press(455,318,470,334);*//*
212. choice[6]=mouse\_press(390,206,410,226);
213. choice[7]=mouse\_press(390,262,410,282);
214. choice[8]=mouse\_press(242,206,262,226);
215. choice[9]=mouse\_press(242,262,262,282);
216. choice[0]=mouse\_press(0,0,29,29);
217. *//判断选项*
218. if(choice[0] == 1)
219. {
220. ClearGraph(WHITE);
221. return 0;
222. }
223. else if(choice[1] == 1)
224. {
225. num[0]^=1;
226. break;
227. }
228. else if(choice[2] == 1)
229. {
230. num[1]^=1;
231. break;
232. }
233. else if(choice[3] == 1)
234. {
235. num[2]^=1;
236. }
237. else if(choice[4] == 1)
238. {
239. num[3]^=1;
240. break;
241. }
242. else if(choice[5] == 1)
243. {
244. num[4]^=1;
245. break;
246. }
247. else if(choice[6] == 1)
248. {
249. if(num[3])
250. {
251. num[5]^=1;
252. }
253. break;
254. }
255. else if(choice[7] == 1)
256. {
257. if(num[4])
258. {
259. num[6]^=1;
260. }
261. break;
262. }
263. else if(choice[8] == 1)
264. {
265. if(num[3])
266. {
267. num[7]^=1;
268. }
269. break;
270. }
271. else if(choice[9] == 1)
272. {
273. if(num[4])
274. {
275. num[8]^=1;
276. }
277. break;
278. }
279. }
280. delay(100);
281. return 1;
282. }

**window.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \*   版本号：1.0
4. \*   作者：X
5. \*   生成日期：2024—3—8
6. \*   说明：Window类函数实现车窗功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include"init.h"
10. #include"window.h"
11. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
12. FUNCTION:OpenWindow
13. DESCRIBETION:打开车窗页
14. INPUT:void
15. RETURN:void
16. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
17. void OpenWindow()
18. {
19. while(1)
20. {
21. int i = 0, num[8], window\_ret = 1;
22. FILE\* window\_text;
23. window\_text = fopen("..\\CODE\\text\\window.txt", "rt");
24. for(i = 0; i < 8; i++)
25. {
26. fscanf(window\_text, "%d ", &num[i]);
27. }
28. fclose(window\_text);
29. setfillstyle(1, WHITE);
30. bar(0, 0, 640, 480);
31. DrawReturnButton();
32. DrawOneWindow(0, 0);    *//Left Ahead*
33. DrawOneWindow(300, 0);  *//Right Ahead*
34. DrawOneWindow(0, 220);  *//Left Behind*
35. DrawOneWindow(300, 220);*//Right Behind*
36. DrawWindowChangePart(num);
37. PutWindowHz();
38. window\_ret = WindowMouse(num);
39. window\_text = fopen("..\\CODE\\text\\window.txt", "wt");
40. for(i = 0; i < 8; i++)
41. {
42. fprintf(window\_text, "%d ", num[i]);
43. }
44. fclose(window\_text);
45. if(!window\_ret) *//If zero, then return*
46. return ;
47. }
48. }
49. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
50. FUNCTION:DrawOneWindow
51. DESCRIBETION:绘制一个车窗
52. INPUT:int x, int y
53. RETURN:void
54. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
55. void DrawOneWindow(int x, int y)
56. {
57. *//绘制方框*
58. setcolor(BLUE);
59. setlinestyle(0, 0, 1);
60. rectangle(40+x, 40+y, 300+x, 220+y);
61. line(80+x, 40+y, 80+x, 220+y);
62. line(190+x, 40+y, 190+x, 220+y);
63. line(80+x, 180+y, 300+x, 180+y);
64. line(135+x, 180+y, 135+x, 220+y);
65. *//绘制上下按钮*
66. setlinestyle(0, 0, 3);
67. line(90+x, 215+y, 107+x, 190+y);
68. line(108+x, 190+y, 125+x, 215+y);
69. line(145+x, 190+y, 162+x, 215+y);
70. line(163+x, 215+y, 180+x, 190+y);
71. *//绘制车窗 (85, 60) (185, 160)*
72. setcolor(LIGHTGRAY);
73. ellipse(90+x, 65+y, 90, 180, 5, 5);
74. ellipse(90+x, 155+y, 180, 270, 5, 5);
75. ellipse(180+x, 155+y, 270, 360, 5, 5);
76. ellipse(180+x, 65+y, 0, 90, 5, 5);
77. line(90+x, 60+y, 180+x, 60+y);
78. line(90+x, 160+y, 180+x, 160+y);
79. line(85+x, 65+y, 85+x, 155+y);
80. line(185+x, 65+y, 185+x, 155+y);
81. *//绘制儿童锁*
82. puthz(193+x, 184+y, "儿童锁", 32, 36, BLUE);
83. setcolor(BROWN);
84. setlinestyle(0, 0, 1);
85. setfillstyle(1, BROWN);
86. rectangle(200+x, 110+y, 290+x, 160+y);
87. floodfill(245+x, 135+y, BROWN);
88. setcolor(DARKGRAY);
89. setfillstyle(1, DARKGRAY);
90. circle(245+x, 135+y, 5);
91. floodfill(245+x, 135+y, DARKGRAY);
92. setcolor(LIGHTGRAY);
93. setfillstyle(1, LIGHTGRAY);
94. line(225+x, 80+y, 225+x, 109+y);
95. line(225+x, 109+y, 235+x, 109+y);
96. line(235+x, 109+y, 235+x, 80+y);
97. line(265+x, 80+y, 265+x, 109+y);
98. line(265+x, 109+y, 255+x, 109+y);
99. line(255+x, 109+y, 255+x, 80+y);
100. ellipse(245+x, 80+y, 0, 180, 10, 10);
101. ellipse(245+x, 80+y, 0, 180, 20, 20);
102. floodfill(245+x, 65+y, LIGHTGRAY);
103. }
104. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
105. FUNCTION:DrawWindowChangePart
106. DESCRIBETION:绘制车窗页面可变的部分
107. INPUT:int\* num
108. RETURN:void
109. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
110. void DrawWindowChangePart(int\* num)
111. {
112. if(num[4])
113. {
114. setfillstyle(1, LIGHTCYAN);
115. floodfill(135, 110, LIGHTGRAY);
116. }
117. else
118. {
119. if(num[0] && num[0]!=10)
120. {
121. setcolor(LIGHTGRAY);
122. setlinestyle(0, 0, 1);
123. line(85, 160-10\*num[0], 185, 160-10\*num[0]);
124. setfillstyle(1, LIGHTCYAN);
125. floodfill(135, 155, LIGHTGRAY);
126. }
127. else if(num[0]==10)
128. {
129. setfillstyle(1, LIGHTCYAN);
130. floodfill(135, 110, LIGHTGRAY);
131. }
132. setfillstyle(1, WHITE);
133. bar(225, 80, 235, 109);
134. }
135. if(num[5])
136. {
137. setfillstyle(1, LIGHTCYAN);
138. floodfill(135 + 300, 110, LIGHTGRAY);
139. }
140. else
141. {
142. if(num[1] && num[1]!=10)
143. {
144. setcolor(LIGHTGRAY);
145. setlinestyle(0, 0, 1);
146. line(85+300, 160-10\*num[1], 185+300, 160-10\*num[1]);
147. setfillstyle(1, LIGHTCYAN);
148. floodfill(135+300, 155, LIGHTGRAY);
149. }
150. if(num[1]==10)
151. {
152. setfillstyle(1, LIGHTCYAN);
153. floodfill(135 + 300, 110, LIGHTGRAY);
154. }
155. setfillstyle(1, WHITE);
156. bar(225+300, 80, 235+300, 109);
157. }
158. if(num[6])
159. {
160. setfillstyle(1, LIGHTCYAN);
161. floodfill(135, 110 + 220, LIGHTGRAY);
162. }
163. else
164. {
165. if(num[2] && num[2]!=10)
166. {
167. setcolor(LIGHTGRAY);
168. setlinestyle(0, 0, 1);
169. line(85, 160-10\*num[2]+220, 185, 160-10\*num[2]+220);
170. setfillstyle(1, LIGHTCYAN);
171. floodfill(135, 155+220, LIGHTGRAY);
172. }
173. else if(num[2]==10)
174. {
175. setfillstyle(1, LIGHTCYAN);
176. floodfill(135, 110 + 220, LIGHTGRAY);
177. }
178. setfillstyle(1, WHITE);
179. bar(225, 80+220, 235, 109+220);
180. }
181. if(num[7])
182. {
183. setfillstyle(1, LIGHTCYAN);
184. floodfill(135 + 300, 110 + 220, LIGHTGRAY);
185. }
186. else
187. {
188. if(num[3] && num[3]!=10)
189. {
190. setcolor(LIGHTGRAY);
191. setlinestyle(0, 0, 1);
192. line(85+300, 160-10\*num[3]+220, 185+300, 160-10\*num[3]+220);
193. setfillstyle(1, LIGHTCYAN);
194. floodfill(135+300, 155+220, LIGHTGRAY);
195. }
196. else if(num[3]==10)
197. {
198. setfillstyle(1, LIGHTCYAN);
199. floodfill(135 + 300, 110 + 220, LIGHTGRAY);
200. }
201. setfillstyle(1, WHITE);
202. bar(225+300, 80+220, 235+300, 109+220);
203. }
204. }
205. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
206. FUNCTION:PutWindowHz
207. DESCRIBETION:显示车窗页汉字
208. INPUT:void
209. RETURN:void
210. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
211. void PutWindowHz()
212. {
213. *//左前窗*
214. puthz(44, 61, "左", 32, 0, BLUE);
215. puthz(44, 114, "前", 32, 0, BLUE);
216. puthz(44, 167, "窗", 32, 0, BLUE);
217. *//右前窗*
218. puthz(44 + 300, 61, "右", 32, 0, BLUE);
219. puthz(44 + 300, 114, "前", 32, 0, BLUE);
220. puthz(44 + 300, 167, "窗", 32, 0, BLUE);
221. *//左后窗*
222. puthz(44, 61 + 220, "左", 32, 0, BLUE);
223. puthz(44, 114 + 220, "后", 32, 0, BLUE);
224. puthz(44, 167 + 220, "窗", 32, 0, BLUE);
225. *//右后窗*
226. puthz(44 + 300, 61 + 220, "右", 32, 0, BLUE);
227. puthz(44 + 300, 114 + 220, "后", 32, 0, BLUE);
228. puthz(44 + 300, 167 + 220, "窗", 32, 0, BLUE);
229. }
230. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
231. FUNCTION:WindowMouse
232. DESCRIBETION:车窗页的鼠标功能
233. INPUT:int\*
234. RETURN:int
235. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
236. int WindowMouse(int\* num)
237. {
238. int choice[13] = {0};
239. while(1)
240. {
241. *//绘制鼠标和设置按钮*
242. newmouse(&MouseX,&MouseY,&press);
243. choice[0]=mouse\_press(0,0,29,29);
244. *//左前窗*
245. choice[1]=mouse\_press(80,180,135,220);   *//上*
246. choice[2]=mouse\_press(135,180,190,220);   *//下*
247. choice[3]=mouse\_press(190,180,300,220);  *//锁*
248. *//右前窗*
249. choice[4]=mouse\_press(80+300, 180, 135+300, 220);   *//上*
250. choice[5]=mouse\_press(135+300, 180, 190+300, 220);   *//下*
251. choice[6]=mouse\_press(190+300, 180, 300+300, 220);  *//锁*
252. *//左后窗*
253. choice[7]=mouse\_press(80, 180+220, 135, 220+220);   *//上*
254. choice[8]=mouse\_press(135, 180+220, 190, 220+220);   *//下*
255. choice[9]=mouse\_press(190, 180+220, 300, 220+220);  *//锁*
256. *//右后窗*
257. choice[10]=mouse\_press(80+300, 180+220, 135+300, 220+220);   *//上*
258. choice[11]=mouse\_press(135+300, 180+220, 190+300, 220+220);   *//下*
259. choice[12]=mouse\_press(190+300, 180+220, 300+300, 220+220);  *//锁*
260. *//判断选项*
261. if(choice[0]==1)
262. {
263. ClearGraph(WHITE);
264. return 0;
265. }
266. else if(choice[1]==1)
267. {
268. if(num[0]<10)
269. num[0]+=1;
270. break;
271. }
272. else if(choice[2]==1)
273. {
274. if(num[0]>0)
275. num[0]-=1;
276. break;
277. }
278. else if(choice[3]==1)
279. {
280. num[4]^=1;
281. num[0]=10;
282. break;
283. }
284. else if(choice[4]==1)
285. {
286. if(num[1]<10)
287. num[1]+=1;
288. break;
289. }
290. else if(choice[5]==1)
291. {
292. if(num[1]>0)
293. num[1]-=1;
294. break;
295. }
296. else if(choice[6]==1)
297. {
298. num[5]^=1;
299. num[1]=10;
300. break;
301. }
302. else if(choice[7]==1)
303. {
304. if(num[2]<10)
305. num[2]+=1;
306. break;
307. }
308. else if(choice[8]==1)
309. {
310. if(num[2]>0)
311. num[2]-=1;
312. break;
313. }
314. else if(choice[9]==1)
315. {
316. num[6]^=1;
317. num[2]=10;
318. break;
319. }
320. else if(choice[10]==1)
321. {
322. if(num[3]<10)
323. num[3]+=1;
324. break;
325. }
326. else if(choice[11]==1)
327. {
328. if(num[3]>0)
329. num[3]-=1;
330. break;
331. }
332. else if(choice[12]==1)
333. {
334. num[7]^=1;
335. num[3]=10;
336. break;
337. }
338. }
339. return 1;
340. }

**wiper.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \*   版本号：1.0
4. \*   作者：X
5. \*   生成日期：2024—3—9
6. \*   说明：Wiper类函数实现雨刷功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include"init.h"
10. #include"wiper.h"
11. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
12. FUNCTION:OpenWiper
13. DESCRIBETION:打开雨刷页
14. INPUT:void
15. REUTRUN:void
16. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
17. void OpenWiper()
18. {
19. while(1)
20. {
21. int i = 0, num[4] = {0}, wiper\_ret = 1;
22. FILE\* wiper\_text;
23. setfillstyle(1, WHITE);
24. bar(0, 0, 640, 480);
25. wiper\_text = fopen("..\\CODE\\text\\wiper.txt", "rt");
26. for(i = 0; i < 4; i++)
27. {
28. fscanf(wiper\_text, "%d ", &num[i]);
30. }
31. fclose(wiper\_text);
32. DrawReturnButton();
33. DrawAheadWiper();
34. DrawBehindWiper();
35. DrawWiperChangePart(num);
36. PutWiperHz();
37. wiper\_ret = WiperMouse(num);
38. wiper\_text = fopen("..\\CODE\\text\\wiper.txt", "wt");
39. for(i = 0; i < 4; i++)
40. {
41. fprintf(wiper\_text, "%d ", num[i]);
42. }
43. fclose(wiper\_text);
44. if(!wiper\_ret)
45. return ;
46. }
47. }
48. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
49. FUNCTION:DrawAheadWiper
50. DESCRIBETION:绘制前雨刷
51. INPUT:void
52. RETURN:void
53. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
54. void DrawAheadWiper()
55. {
56. *//绘制前雨刷*
57. setlinestyle(0, 0, 3);
58. line(65, 245, 255, 245);
59. line(65, 285, 255, 285);
60. ellipse(65, 265, 90, 270, 20, 20);
61. ellipse(255, 265, 270, 450, 20, 20);
62. puthz(65, 249, "前雨刷", 32, 47+32, BLUE);
63. settextstyle(8, 0, 2);
64. circle(100, 315, 20);   *//on*
65. circle(220, 315, 20);   *//off*
66. line(65, 345, 255, 345);
67. line(65, 385, 255, 385);
68. ellipse(65, 365, 90, 270, 20, 20);
69. ellipse(255, 365, 270, 450, 20, 20);
70. puthz(98, 349, "速度", 32, 60+32, BLUE);
71. circle(80, 415, 20);    *//快*
72. circle(160, 415, 20);   *//中*
73. circle(240, 415, 20);   *//慢*
74. *//绘制雨刷图形*
75. setcolor(DARKGRAY);
76. line(65, 60, 255, 60);
77. line(60, 65, 60, 195);
78. line(65, 200, 255, 200);
79. line(260, 65, 260, 195);
80. ellipse(255, 65, 0, 90, 5, 5);
81. ellipse(65, 65, 90, 180, 5, 5);
82. ellipse(65, 195, 180, 270, 5, 5);
83. ellipse(255, 195, 270, 360, 5, 5);
84. setlinestyle(0, 0, 1);
85. setfillstyle(1, DARKGRAY);
86. circle(125, 210, 5);
87. circle(190, 210, 5);
88. floodfill(125, 210, DARKGRAY);
89. floodfill(190, 210, DARKGRAY);
90. line(125, 210, 125, 170);
91. line(190, 210, 190, 170);
92. setlinestyle(0, 0, 3);
93. line(105, 190, 175, 120);
94. line(170, 190, 240, 120);
95. }
96. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
97. FUNCTION:DrawBehindWiper
98. DESCRIBETION:绘制后雨刷
99. INPUT:void
100. RETURN:void
101. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
102. void DrawBehindWiper()
103. {
104. *//绘制后雨刷*
105. setcolor(BLUE);
106. setlinestyle(0, 0, 1);
107. setlinestyle(0, 0, 3);
108. line(65+320, 245, 255+320, 245);
109. line(65+320, 285, 255+320, 285);
110. ellipse(65+320, 265, 90, 270, 20, 20);
111. ellipse(255+320, 265, 270, 450, 20, 20);
112. puthz(65+320, 249, "后雨刷", 32, 47+32, BLUE);
113. settextstyle(8, 0, 2);
114. circle(100+320, 315, 20);   *//on*
115. circle(220+320, 315, 20);   *//off*
116. line(65+320, 345, 255+320, 345);
117. line(65+320, 385, 255+320, 385);
118. ellipse(65+320, 365, 90, 270, 20, 20);
119. ellipse(255+320, 365, 270, 450, 20, 20);
120. puthz(98+320, 349, "速度", 32, 60+32, BLUE);
121. circle(80+320, 415, 20);    *//快*
122. circle(160+320, 415, 20);   *//中*
123. circle(240+320, 415, 20);   *//慢*
124. *//绘制雨刷图形*
125. setcolor(DARKGRAY);
126. line(65+320, 60, 255+320, 60);
127. line(60+320, 65, 60+320, 195);
128. line(65+320, 200, 255+320, 200);
129. line(260+320, 65, 260+320, 195);
130. ellipse(255+320, 65, 0, 90, 5, 5);
131. ellipse(65+320, 65, 90, 180, 5, 5);
132. ellipse(65+320, 195, 180, 270, 5, 5);
133. ellipse(255+320, 195, 270, 360, 5, 5);
134. setfillstyle(1, DARKGRAY);
135. circle(480, 210, 5);
136. floodfill(480, 210, DARKGRAY);
137. setlinestyle(0, 0, 1);
138. line(480, 210, 480, 180);
139. setlinestyle(0, 0, 3);
140. line(470, 190, 510, 150);
141. }
142. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
143. FUNCTION:DrawWiperChangePart
144. DESCRIBETION:绘制雨刷页的可变部分
145. INPUT:int\* num
146. RETURN:void
147. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
148. void DrawWiperChangePart(int\* num)
149. {
150. if(num[0] == 1)
151. {
152. setfillstyle(1, GREEN);
153. floodfill(100, 315, BLUE);
154. }
155. if(num[0] == 2)
156. {
157. setfillstyle(1, GREEN);
158. floodfill(220, 315, BLUE);
159. }
160. if(num[1] == 1)
161. {
162. setfillstyle(1, GREEN);
163. floodfill(80, 415, BLUE);
164. }
165. if(num[1] == 2)
166. {
167. setfillstyle(1, GREEN);
168. floodfill(160, 415, BLUE);
169. }
170. if(num[1] == 3)
171. {
172. setfillstyle(1, GREEN);
173. floodfill(240, 415, BLUE);
174. }
175. if(num[2] == 1)
176. {
177. setfillstyle(1, GREEN);
178. floodfill(100+320, 315, BLUE);
179. }
180. if(num[2] == 2)
181. {
182. setfillstyle(1, GREEN);
183. floodfill(220+320, 315, BLUE);
184. }
185. if(num[3] == 1)
186. {
187. setfillstyle(1, GREEN);
188. floodfill(80+320, 415, BLUE);
189. }
190. if(num[3] == 2)
191. {
192. setfillstyle(1, GREEN);
193. floodfill(160+320, 415, BLUE);
194. }
195. if(num[3] == 3)
196. {
197. setfillstyle(1, GREEN);
198. floodfill(240+320, 415, BLUE);
199. }
200. }
201. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
202. FUNCTION:PutWiperHz
203. DESCRIBETION:显示雨刷页汉字
204. INPUT:void
205. RETURN:void
206. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
207. void PutWiperHz()
208. {
209. setcolor(BLUE);
210. outtextxy(87, 297, "on");
211. outtextxy(205, 297, "off");
212. puthz(64, 399, "快", 32, 0, BLUE);
213. puthz(144, 399, "中", 32, 0, BLUE);
214. puthz(224, 399, "慢", 32, 0, BLUE);
215. outtextxy(87+320, 297, "on");
216. outtextxy(205+320, 297, "off");
217. puthz(64+320, 399, "快", 32, 0, BLUE);
218. puthz(144+320, 399, "中", 32, 0, BLUE);
219. puthz(224+320, 399, "慢", 32, 0, BLUE);
220. }
221. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
222. FUNCTION:WiperMouse
223. DESCRIBETION:雨刷页的鼠标功能
224. INPUT:int\*
225. RETURN:int
226. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
227. int WiperMouse(int\* num)
228. {
229. int choice[11] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
230. while(1)
231. {
232. *//绘制鼠标和设置按钮*
234. newmouse(&MouseX,&MouseY,&press);
235. choice[0]=mouse\_press(0,0,29,29);
236. *//前雨刷*
237. choice[1]=mouse\_press(80, 295, 120, 335);  *//on*
238. choice[2]=mouse\_press(200, 295, 240, 335);  *//off*
239. choice[3]=mouse\_press(60, 395, 100, 435);  *//快*
240. choice[4]=mouse\_press(140, 395, 180, 435);  *//中*
241. choice[5]=mouse\_press(220, 395, 260, 435);  *//慢*
242. *//后雨刷*
243. choice[6]=mouse\_press(80+320, 295, 120+320, 335);  *//on*
244. choice[7]=mouse\_press(200+320, 295, 240+320, 335);  *//off*
245. choice[8]=mouse\_press(60+320, 395, 100+320, 435);  *//快*
246. choice[9]=mouse\_press(140+320, 395, 180+320, 435);  *//中*
247. choice[10]=mouse\_press(220+320, 395, 260+320, 435);  *//慢*
248. *//判断选项*
249. if(choice[0]==1)
250. {
251. ClearGraph(WHITE);
252. return 0;
253. }
254. else if(choice[1]==1)
255. {
256. num[0] = 1;
257. break;
258. }
259. else if(choice[2]==1)
260. {
261. num[0] = 2;
262. break;
263. }
264. else if(choice[3]==1)
265. {
266. num[1] = 1;
267. break;
268. }
269. else if(choice[4]==1)
270. {
271. num[1] = 2;
272. break;
273. }
274. else if(choice[5]==1)
275. {
276. num[1] = 3;
277. break;
278. }
279. else if(choice[6]==1)
280. {
281. num[2] = 1;
282. break;
283. }
284. else if(choice[7]==1)
285. {
286. num[2] = 2;
287. break;
288. }
289. else if(choice[8]==1)
290. {
291. num[3] = 1;
292. break;
293. }
294. else if(choice[9]==1)
295. {
296. num[3] = 2;
297. break;
298. }
299. else if(choice[10]==1)
300. {
301. num[3] = 3;
302. break;
303. }
304. }
305. return 1;
306. }

**sunroof.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \*   版本号：1.0
4. \*   作者：X
5. \*   生成日期：2024—3—6
6. \*   说明：Sunroof类函数实现天窗功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include"init.h"
10. #include"sunroof.h"
11. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
12. FUNCTION:OpenSunroof
13. DESCRIBETION:打开天窗页
14. INPUT:void
15. RETURN:void
16. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
17. void OpenSunroof()
18. {
19. while(1)
20. {
21. int sunroof\_ret = 1, i = 0, num[4];
22. FILE\* sunroof\_text;
23. setfillstyle(1, WHITE);
24. bar(0, 0, 640, 480);
25. sunroof\_text = fopen("..\\CODE\\text\\sunroof.txt", "rt");
26. for(i = 0; i < 5; i++)
27. {
28. fscanf(sunroof\_text, "%d ", &num[i]);
29. }
30. fclose(sunroof\_text);
31. DrawReturnButton();
32. DrawPartSunroofButton();
33. DrawSunroof();
34. DrawSunroofChangePart(num);
35. PutSunroofHz();
36. sunroof\_ret = SunroofMouse(num);
37. sunroof\_text = fopen("..\\CODE\\text\\sunroof.txt", "wt");
38. for(i = 0; i < 4; i++)
39. {
40. fprintf(sunroof\_text, "%d ", num[i]);
41. }
42. fclose(sunroof\_text);
43. if(!sunroof\_ret)
44. return ;
45. }
46. }
47. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
48. FUNCTION:DrawPartSunroofButton
49. DESCRIBETION:绘制天窗页部分按钮
50. INPUT:void
51. RETURN:void
52. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
53. void DrawPartSunroofButton()
54. {
55. setlinestyle(0, 0, 3);
56. line(65, 70, 255, 70);
57. line(65, 110, 255, 110);
58. ellipse(65, 90, 90, 270, 20, 20);
59. ellipse(255, 90, 270, 450, 20, 20);
60. puthz(65, 74, "天窗锁", 32, 47+32, BLUE);
61. circle(100, 150, 20);   *//on*
62. circle(220, 150, 20);   *//off*
63. setlinestyle(0, 0, 3);
64. line(65, 70+120, 255, 70+120);
65. line(65, 110+120, 255, 110+120);
66. ellipse(65, 90+120, 90, 270, 20, 20);
67. ellipse(255, 90+120, 270, 450, 20, 20);
68. puthz(65, 74+120, "微通风", 32, 47+32, BLUE);
69. circle(100, 150+120, 20);   *//on*
70. circle(220, 150+120, 20);   *//off*
71. setlinestyle(0, 0, 3);
72. line(65, 70+240, 255, 70+240);
73. line(65, 110+240, 255, 110+240);
74. ellipse(65, 90+240, 90, 270, 20, 20);
75. ellipse(255, 90+240, 270, 450, 20, 20);
76. puthz(65, 74+240, "热保护", 32, 47+32, BLUE);
77. settextstyle(8, 0, 2);
78. circle(100, 150+240, 20);   *//on*
79. circle(220, 150+240, 20);   *//off*
80. }
81. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
82. FUNCTION:DrawSunroof
83. DESCRIBETION:绘制天窗
84. INPUT:void
85. RETURN:void
86. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
87. void DrawSunroof()
88. {
89. setcolor(DARKGRAY);
90. *//天窗外框*
91. ellipse(620, 105, 0, 90, 5, 5);
92. ellipse(340, 105, 90, 180, 5, 5);
93. ellipse(340, 315, 180, 270, 5, 5);
94. ellipse(620, 315, 270, 360, 5, 5);
95. line(620, 100, 340, 100);
96. line(335, 105, 335, 315);
97. line(340, 320, 620, 320);
98. line(625, 315, 625, 105);
99. *//天窗内框*
100. ellipse(575, 115, 0, 90, 5, 5);
101. ellipse(385, 115, 90, 180, 5, 5);
102. ellipse(385, 305, 180, 270, 5, 5);
103. ellipse(575, 305, 270, 360, 5, 5);
104. line(575, 110, 385, 110);
105. line(380, 115, 380, 305);
106. line(385, 310, 575, 310);
107. line(580, 310, 580, 110);
109. *//天窗按钮*
110. setcolor(BLUE);
111. circle(400, 350, 20);
112. line(410, 340, 390, 350);
113. line(410, 360, 390, 350);
114. circle(560, 350, 20);
115. line(550, 340, 570, 350);
116. line(550, 360, 570, 350);
117. }
118. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
119. FUNCTION:DrawSunroofChangePart
120. DESCRIBETION:绘制天窗页变化部分
121. INPUT:int"\*
122. RETURN:void
123. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
124. void DrawSunroofChangePart(int\* num)
125. {
126. if(num[0] == 1)
127. {
128. setfillstyle(1, GREEN);
129. floodfill(100, 150, BLUE);
130. setfillstyle(1, LIGHTCYAN);
131. floodfill(480, 210, DARKGRAY);
132. }
133. else if(num[0] == 2)
134. {
135. setfillstyle(1, GREEN);
136. floodfill(220, 150, BLUE);
137. if(num[1] == 1)
138. {
139. floodfill(100, 150+120, BLUE);
140. }
141. else if(num[1] == 2)
142. {
143. floodfill(220, 150+120, BLUE);
144. }
145. if(num[2] == 1)
146. {
147. floodfill(100, 150+240, BLUE);
148. }
149. else if(num[2] == 2)
150. {
151. floodfill(220, 150+240, BLUE);
152. }
153. if(num[3])
154. {
155. setlinestyle(0, 0, 3);
156. setcolor(DARKGRAY);
157. line(380+22\*num[3], 110, 380+22\*num[3], 310);
158. setfillstyle(1, LIGHTCYAN);
159. floodfill(390, 210, DARKGRAY);
160. setcolor(LIGHTCYAN);
161. line(380+22\*num[3], 112, 380+22\*num[3], 308);
162. }
163. }
164. }
165. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
166. FUNCTION:PutSunroofHz
167. DESCRIBETION:显示天窗页文本
168. INPUT:void
169. RETURN:void
170. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
171. void PutSunroofHz()
172. {
173. settextstyle(8, 0, 2);
174. setcolor(BLUE);
175. outtextxy(87, 132, "on");
176. outtextxy(205, 132, "off");
177. outtextxy(87, 132+120, "on");
178. outtextxy(205, 132+120, "off");
179. outtextxy(87, 132+240, "on");
180. outtextxy(205, 132+240, "off");
181. }
182. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
183. FUNCTION:SunroofMouse
184. DESCRIBETION:天窗页鼠标
185. INPUT:int\*
186. RETURN:int
187. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
188. int SunroofMouse(int\* num)
189. {
190. int choice[9] = {0, 0, 0, 0, 0, 0, 0, 0, 0};
191. while(1)
192. {
193. *//绘制鼠标和设置按钮*
194. newmouse(&MouseX,&MouseY,&press);
195. choice[0]=mouse\_press(0,0,29,29);
196. *//天窗锁*
197. choice[1]=mouse\_press(80, 130, 120, 170);  *//on*
198. choice[2]=mouse\_press(200, 130, 240, 170);  *//off*
199. *//微通风*
200. choice[3]=mouse\_press(80, 130+120, 120, 170+120);  *//on*
201. choice[4]=mouse\_press(200, 130+120, 240, 170+120);  *//off*
202. *//热保护*
203. choice[5]=mouse\_press(80, 130+240, 120, 170+240);  *//on*
204. choice[6]=mouse\_press(200, 130+240, 240, 170+240);  *//off*
205. *//天窗*
206. choice[7]=mouse\_press(380, 330, 420, 370);  *//左*
207. choice[8]=mouse\_press(540, 330, 580, 370);  *//右*
208. *//判断选项*
209. if(choice[0]==1)
210. {
211. ClearGraph(WHITE);
212. return 0;
213. }
214. else if(choice[1]==1)
215. {
216. num[0] = 1;
217. num[3] = 9;
218. break;
219. }
220. else if(choice[2]==1)
221. {
222. num[0] = 2;
223. break;
224. }
225. else if(choice[3]==1)
226. {
227. num[1] = 1;
228. break;
229. }
230. else if(choice[4]==1)
231. {
232. num[1] = 2;
233. break;
234. }
235. else if(choice[5]==1)
236. {
237. num[2] = 1;
238. break;
239. }
240. else if(choice[6]==1)
241. {
242. num[2] = 2;
243. break;
244. }
245. else if(choice[7]==1)
246. {
247. if(num[3] > 0)
248. {
249. num[3]-=1;
250. }
251. break;
252. }
253. else if(choice[8]==1)
254. {
255. if(num[3] < 9)
256. {
257. num[3]+=1;
258. }
259. break;
260. }
261. }
262. return 1;
263. }

**tire.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \*   版本号：1.0
4. \*   作者：X
5. \*   生成日期：2024—3—7
6. \*   说明：Tire类函数实现胎压功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include<time.h>
10. #include"init.h"
11. #include"tire.h"
12. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
13. FUNCTION:OpenTire
14. DESCRIBETION:打开胎压页
15. INPUT:void
16. RETURN:void
17. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
18. void OpenTire()
19. {
20. while(1)
21. {
22. int tire\_ret = 1, i = 0, num[5]={0};
23. FILE\* tire\_text;
24. setfillstyle(1, WHITE);
25. bar(0, 0, 640, 480);
26. tire\_text = fopen("..\\CODE\\text\\tire.txt", "rt");
27. for(i = 0; i < 4; i++)
28. {
29. fscanf(tire\_text, "%d ", &num[i]);
30. if(num[i]<230 || num[i]>280)
31. {
32. num[4] = 1;
33. }
34. }
35. fclose(tire\_text);
36. DrawReturnButton();
37. DrawTireCar();
38. PutTireBar(407, 100, num[0]);
39. PutTireBar(407, 300, num[1]);
40. PutTireBar(513, 100, num[2]);
41. PutTireBar(513, 300, num[3]);
42. DrawTireChangePart(num);
43. tire\_ret = TireMouse(num);
44. tire\_text = fopen("..\\CODE\\text\\tire.txt", "wt");
45. for(i = 0; i < 5; i++)
46. {
47. fprintf(tire\_text, "%d ", num[i]);
48. }
49. fclose(tire\_text);
50. if (!tire\_ret)
51. return ;
52. }
53. }
54. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
55. FUNCTION:DrawOneTire
56. DESCRIBETION:绘制一个轮胎
57. INPUT:int x, int y
58. RETURN:void
59. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
60. void DrawOneTire(int x, int y)
61. {
62. setcolor(DARKGRAY);
63. setfillstyle(1, DARKGRAY);
64. setlinestyle(0, 0, 1);
65. line(0+x, 0+y, 0+x, 30+y);
66. line(10+x, 0+y, 10+x, 30+y);
67. ellipse(5+x, 0+y, 0, 180, 5, 2);
68. ellipse(5+x, 30+y, 180, 360, 5, 2);
69. floodfill(5+x, 15+y, DARKGRAY);
70. }
71. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
72. FUNCTION:PutTireBar
73. DESCRIBETION:显示胎压
74. INPUT:int x, int y
75. RETURN:void
76. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
77. void PutTireBar(int x, int y, int bar)
78. {
79. char pressure[4];
80. itoa(bar, pressure, 10);
81. if(bar>=230 && bar<=280)
82. {
83. setcolor(BLUE);
84. }
85. else if(bar<230 || bar>280)
86. {
87. setcolor(RED);
88. }
89. settextstyle(8, 0, 2);
90. outtextxy(x, y, pressure);
91. }
92. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
93. FUNCTION:DrawTireCar
94. DESCRIBETION:绘制胎压页汽车
95. INPUT:void
96. RETURN:void
97. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
98. void DrawTireCar()
99. {
100. setcolor(DARKGRAY);
101. setlinestyle(0, 0, 3);
102. moveto(390, 80);
103. lineto(390, 180);
104. lineto(370, 200);
105. lineto(390, 200);
106. lineto(390, 360);
107. lineto(400, 380);
108. moveto(560, 380);
109. lineto(570, 360);
110. lineto(570, 200);
111. lineto(590, 200);
112. lineto(570, 180);
113. lineto(570, 80);
114. ellipse(430, 80, 90, 180, 40, 10);
115. ellipse(530, 80, 0, 90, 40, 10);
116. ellipse(480, 73, 0, 180, 50, 10);
117. ellipse(480, 380, 180, 360, 80, 10);
118. DrawOneTire(395, 100);
119. DrawOneTire(395, 300);
120. DrawOneTire(555, 100);
121. DrawOneTire(555, 300);
122. puthz(436, 40, "单位：", 16, 18, BLUE);
123. setcolor(BLUE);
124. settextstyle(8, 0, 2);
125. outtextxy(480, 30, "kpa");
126. puthz(36, 44, "快速修复", 32, 32, BLUE);
127. rectangle(20, 40, 180, 80);
128. }
129. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
130. FUNCTION:DrawTireChangePart
131. DESCRIBETION:绘制胎压页的可变部分
132. INPUT:int\* num
133. RETURN:void
134. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
135. void DrawTireChangePart(int\* num)
136. {
137. if(num[4] == 1)
138. {
139. if(num[0]<230 || num[0]>280)
140. {
141. puthz(20, 200, "左前轮胎压异常", 32, 32, RED);
142. }
143. if(num[1]<230 || num[1]>280)
144. {
145. puthz(20, 240, "左后轮胎压异常", 32, 32, RED);
146. }
147. if(num[2]<230 || num[2]>280)
148. {
149. puthz(20, 280, "右前轮胎压异常", 32, 32, RED);
150. }
151. if(num[3]<230 || num[3]>280)
152. {
153. puthz(20, 320, "右后轮胎压异常", 32, 32, RED);
154. }
155. }
156. }
157. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
158. FUNCTION:TireMouse
159. DESCRIBETION:胎压页鼠标
160. INPUT:int\* num
161. RETURN:int
162. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
163. int TireMouse(int\* num)
164. {
165. int choice[2] = {0}, i;
166. while(1)
167. {
168. *//绘制鼠标和设置按钮*
169. newmouse(&MouseX,&MouseY,&press);
170. choice[0]=mouse\_press(0,0,29,29);
171. choice[1]=mouse\_press(20, 40, 180, 80);
172. *//判断选项*
173. if(choice[0]==1)
174. {
175. ClearGraph(WHITE);
176. return 0;
177. }
178. else if(choice[1]==1)
179. {
180. for(i = 0; i < 4; i++)
181. {
182. if(num[i]<230 || num[i]>280)
183. {
184. srand(time(NULL));
185. num[i] = rand()%50+230;
186. delay(1000);
187. }
188. }
189. break;
190. }
191. }
192. return 1;
193. }

**display.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \* 版本号：1.0
4. \* 作者：Y
5. \* 生成日期：2024—4—20
6. \* 说明：Display类函数实现显示功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include"init.h"
10. #include<time.h>
11. #include<math.h>
12. #include"display.h"
13. #include<stdlib.h>
14. #include<time.h>
15. #include"normal.h"
16. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
17. FUNCTION:OpenDisplay
18. DESCRIBETION:打开显示界面
19. INPUT:void
20. RETURN:void
21. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
22. void OpenDisplay()
23. {
24. while(1)
25. {
26. int display\_ret=-1,num[10]={0},i;
27. FILE \*display\_text,\*battery\_text;
29. setfillstyle(1, WHITE);
30. bar(0, 0, 640, 480);
32. display\_text=fopen("..\\CODE\\text\\display.txt","rt");
33. for(i=0;i<10;i++)
34. {
35. fscanf(display\_text,"%d ",&num[i]);
36. }
37. fclose(display\_text);
39. DrawDisplaySButton(num);
40. DrawDisplayChangePart(num);
41. display\_ret =DisplayMouse(num);
43. display\_text=fopen("..\\CODE\\text\\display.txt","wt");
44. for(i=0;i<10;i++)
45. {
46. fprintf(display\_text,"%d ",num[i]);
47. }
48. fclose(display\_text);
49. battery\_text = fopen("..\\CODE\\text\\battery.txt", "wt");
50. fprintf(battery\_text, "%d %d %d %d ",num[2],1,0,0);
51. fclose(battery\_text);
52. if(!display\_ret)
53. return ;
54. }
55. }
56. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
57. FUNCTION:DrawDisplaySButton
58. DESCRIBETION:绘制显示中所有按钮
59. INPUT:void
60. RETURN:void
61. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
62. void DrawDisplaySButton(int \*num)
63. {
64. int i,j,t;
65. char \*p;
66. float pi=3.1415926535,k;
67. k=pi/180;
68. num[1]=0;
70. Draw(BLUE);
71. DrawReturnButton();
72. DrawCar(320,240,0.2);
74. DrawArrow();
76. for(i=-1;i<2;i+=2)
77. {
78. circle(320+210\*i,230,40);
79. circle(320+210\*i,230,44);
80. circle(320+210\*i,230,60);
81. circle(320+210\*i,230,65);
82. circle(320+210\*i,230,70);
83. ellipse(320+210\*i,230,-60-90\*i,60+90\*i,80,80);
84. ellipse(320+210\*i,230,35-90\*i,145-90\*i,90,90);
85. rectangle(310+210\*i,220,330+210\*i,240);
86. for(j=0;j<36;j++)
87. {
88. t=j/3;
89. line(320+210\*i+65\*cos(10\*k\*j),230-65\*sin(10\*k\*j),320+210\*i+70\*cos(10\*k\*j),230-70\*sin(10\*k\*j));
90. line(320+210\*i+55\*cos(30\*k\*t),230-55\*sin(30\*k\*t),320+210\*i+65\*cos(30\*k\*t),230-65\*sin(30\*k\*t));
91. }
93. p=(int \*) malloc(10\*sizeof(int));
94. for(j=0;j<10;j++)
95. {
96. t=(j+1)/5;
97. itoa(t\*50,p,10);
98. settextstyle(2,0,3);
99. outtextxy(315+210\*i+95\*cos((-90+40\*i+50\*t\*i)\*k),225-96\*sin((-90+40\*i+50\*t\*i)\*k),p);
100. }
101. free(p);
102. p=(int \*) malloc(11\*sizeof(int));
103. for(j=0;j<11;j++)
104. {
105. itoa(j\*10,p,10);
106. settextstyle(2,0,3);
107. outtextxy(110+55\*cos((-120-30\*j)\*k),225-50\*sin((-120-30\*j)\*k),p);
108. }
109. free(p);
110. p=(int \*) malloc(12\*sizeof(int));
111. for(j=0;j<6;j++)
112. {
113. itoa(-100+50\*j,p,10);
114. outtextxy(527+55\*cos((-120-30\*j)\*k),225-50\*sin((-120-30\*j)\*k),p);
115. itoa(150-50\*j,p,10);
116. outtextxy(527+55\*cos((-270-30\*j)\*k),225-50\*sin((-270-30\*j)\*k),p);
117. }
118. free(p);
119. }
120. settextstyle(2,0,5);
121. outtextxy(95,270,"km/h");
122. outtextxy(525,270,"kw");
123. puthz(28,315,"湿度",16,20,BLUE);
124. puthz(499,210,"前",16,20,BLUE);
125. puthz(499,230,"驱",16,20,BLUE);
126. puthz(550,210,"后",16,20,BLUE);
127. puthz(550,230,"驱",16,20,BLUE);
128. settextstyle(3,0,1);
129. outtextxy(527,217,"R");
130. puthz(560,320,"电量",16,20,BLUE);
131. setcolor(GREEN);
132. setfillstyle(1, GREEN);
133. rectangle(600,315,620,335);
134. rectangle(608,311,612,315);
135. floodfill(609,312,GREEN);
136. line(613,316,607,322);
137. line(607,322,613,328);
138. line(613,328,607,334);*//电池*
139. DrawDailnumber(num);*//表盘*
141. setcolor(BLUE);
142. line(0,340,155,340);
143. line(155,340,220,400);
144. line(420,400,485,340);
145. line(485,340,640,340);
146. line(220,400,420,400);*//分界*
147. circle(320,430,20);
148. line(320,440,320,420);
149. line(320,420,310,430);
150. line(320,420,330,430);
151. puthz(295,450,"启动",24,27,BLUE);*//启动*
153. puthz(23,360,"温度",16,27,BLUE);
154. outtextxy(67,355,":");
155. puthz(100,360,"摄氏度",16,20,BLUE);
156. puthz(23,400,"空气质量优",16,20,BLUE);
157. puthz(23,440,"天气晴",16,20,BLUE);
158. puthz(470,360,"状态",16,20,BLUE);
159. outtextxy(510,355,":");
160. puthz(515,360,"正常",16,20,BLUE);*/////////*
161. puthz(470,400,"续航里程",16,20,BLUE);
162. outtextxy(550,395,":");
163. outtextxy(555,395,"500KM");
164. puthz(470,440,"驾驶模式",16,20,BLUE);
165. outtextxy(550,435,":");
166. puthz(555,440,"自动",16,20,BLUE);
167. line(250,40,250,400);
168. line(270,40,270,400);
169. line(380,40,380,400);
170. line(400,40,400,400);
171. }
172. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
173. FUNCTION:DrawDailnumber
174. DESCRIBETION:绘制表盘等上的数据（如温度，湿度等）
175. INPUT:void
176. RETURN:void
177. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
178. void DrawDailnumber(int \*num)
179. {
180. int i,j,n,a,b,temperature,humidity,battery,front,rear,m[5]={1};
181. char \*p;
182. FILE \*ac\_text,\*battery\_text;
183. float pi=3.1415926535,k;
184. k=pi/180;
185. front=rear=0;
186. ac\_text=fopen("..\\CODE\\text\\ac.txt","rt");
187. fscanf(ac\_text,"%d %d %d %d %d %d %d ",&i,&j,&n,&a,&b,&temperature,&humidity);
188. fclose(ac\_text);
189. battery\_text=fopen("..\\CODE\\text\\battery.txt","rt");
190. fscanf(battery\_text,"%d %d %d %d ",&battery,&m[0],&m[1],&m[2]);
191. fclose(battery\_text);
192. num[2]=battery;
193. DisplayChange(num,num[2]);
194. Draw(BLUE);
195. for(j=0;j<humidity/10.0;j++)
196. {
197. line(110+82\*cos((-125-11\*j)\*k),230-82\*sin((-125-11\*j)\*k),110+88\*cos((-125-11\*j)\*k),230-88\*sin((-125-11\*j)\*k));
198. line(110+82\*cos((-135-11\*j)\*k),230-82\*sin((-135-11\*j)\*k),110+88\*cos((-135-11\*j)\*k),230-88\*sin((-135-11\*j)\*k));
199. line(110+82\*cos((-125-11\*j)\*k),230-82\*sin((-125-11\*j)\*k),110+82\*cos((-135-11\*j)\*k),230-82\*sin((-135-11\*j)\*k));
200. line(110+88\*cos((-125-11\*j)\*k),230-88\*sin((-125-11\*j)\*k),110+88\*cos((-135-11\*j)\*k),230-88\*sin((-135-11\*j)\*k));
201. setfillstyle(1, GREEN);
202. floodfill(110+85\*cos((-130-11\*j)\*k),230-85\*sin((-130-11\*j)\*k), BLUE);
203. }
204. p=(int \*) malloc(sizeof(int));
205. itoa(humidity,p,10);
206. settextstyle(3,0,1);
207. outtextxy(66,310,":");
208. outtextxy(70,310,p);
209. outtextxy(110,310,"%");*//湿度*
210. free(p);
211. p=(int \*) malloc(sizeof(int));
212. itoa(temperature,p,10);
213. outtextxy(70,355,p);*//温度*
214. free(p);
215. for(j=0;j<(front+100)\*0.06;j++)
216. {
217. setfillstyle(1, RED);
218. floodfill(530+67\*cos((-125-10\*j)\*k),230-67\*sin((-125-10\*j)\*k),BLUE);
219. floodfill(530+62\*cos((-125-10\*j)\*k),230-62\*sin((-125-10\*j)\*k),BLUE);
220. }*//前驱*
221. for(j=0;j<(rear+100)\*0.06;j++)
222. {
223. setfillstyle(1, DARKGRAY);
224. floodfill(530+67\*cos((-55+10\*j)\*k),230-67\*sin((-55+10\*j)\*k),BLUE);
225. floodfill(530+62\*cos((-55+10\*j)\*k),230-62\*sin((-55+10\*j)\*k),BLUE);
226. }*//后驱*
227. }
228. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
229. FUNCTION:DrawCar
230. DESCRIBETION:绘制车
231. INPUT:int
232. RETURN:void
233. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
234. void DrawCar(int x,int y,float k)*//x,y确定位置，k可确定大小*
235. {
236. int i;
237. line((325-230)\*k+x,(275-300)\*k+y,(350-230)\*k+x,(196-300)\*k+y);
238. line((350-230)\*k+x,(181-300)\*k+y,(350-230)\*k+x,(196-300)\*k+y);
239. line((350-230)\*k+x,(181-300)\*k+y,(321-230)\*k+x,(164-300)\*k+y);
240. line((321-230)\*k+x,(164-300)\*k+y,(144-230)\*k+x,(165-300)\*k+y);
241. line((144-230)\*k+x,(165-300)\*k+y,(116-230)\*k+x,(180-300)\*k+y);
242. line((116-230)\*k+x,(180-300)\*k+y,(115-230)\*k+x,(201-300)\*k+y);
243. line((135-230)\*k+x,(275-300)\*k+y,(115-230)\*k+x,(201-300)\*k+y);
244. line((229-230)\*k+x,(266-300)\*k+y,(135-230)\*k+x,(275-300)\*k+y);
245. line((229-230)\*k+x,(266-300)\*k+y,(325-230)\*k+x,(275-300)\*k+y);*//前窗*
246. line((147-230)\*k+x,(424-300)\*k+y,(230-230)\*k+x,(428-300)\*k+y);
247. line((230-230)\*k+x,(428-300)\*k+y,(316-230)\*k+x,(422-300)\*k+y);
248. line((316-230)\*k+x,(422-300)\*k+y,(326-230)\*k+x,(496-300)\*k+y);
249. line((326-230)\*k+x,(496-300)\*k+y,(296-230)\*k+x,(511-300)\*k+y);
250. line((296-230)\*k+x,(511-300)\*k+y,(230-230)\*k+x,(522-300)\*k+y);
251. line((230-230)\*k+x,(522-300)\*k+y,(171-230)\*k+x,(512-300)\*k+y);
252. line((171-230)\*k+x,(512-300)\*k+y,(138-230)\*k+x,(497-300)\*k+y);
253. line((138-230)\*k+x,(497-300)\*k+y,(147-230)\*k+x,(424-300)\*k+y);*//后窗*
254. for(i=-1;i<2;i+=2)
255. {
256. line((150-230)\*i\*k+x,(55-300)\*k+y,(114-230)\*i\*k+x,(143-300)\*k+y);
257. line((114-230)\*i\*k+x,(143-300)\*k+y,(107-230)\*i\*k+x,(180-300)\*k+y);
258. line((107-230)\*i\*k+x,(180-300)\*k+y,(105-230)\*i\*k+x,(207-300)\*k+y);
259. line((105-230)\*i\*k+x,(207-300)\*k+y,(131-230)\*i\*k+x,(296-300)\*k+y);
260. line((131-230)\*i\*k+x,(296-300)\*k+y,(132-230)\*i\*k+x,(398-300)\*k+y);
261. line((132-230)\*i\*k+x,(398-300)\*k+y,(116-230)\*i\*k+x,(444-300)\*k+y);
262. line((116-230)\*i\*k+x,(444-300)\*k+y,(105-230)\*i\*k+x,(360-300)\*k+y);
263. line((105-230)\*i\*k+x,(360-300)\*k+y,(104-230)\*i\*k+x,(227-300)\*k+y);
264. line((104-230)\*i\*k+x,(227-300)\*k+y,(105-230)\*i\*k+x,(214-300)\*k+y);
265. line((105-230)\*i\*k+x,(214-300)\*k+y,(101-230)\*i\*k+x,(211-300)\*k+y);
266. line((101-230)\*i\*k+x,(211-300)\*k+y,(105-230)\*i\*k+x,(156-300)\*k+y);
267. line((105-230)\*i\*k+x,(156-300)\*k+y,(134-230)\*i\*k+x,(84-300)\*k+y);
268. line((134-230)\*i\*k+x,(84-300)\*k+y,(150-230)\*i\*k+x,(55-300)\*k+y);
269. line((115-230)\*i\*k+x,(82-300)\*k+y,(105-230)\*i\*k+x,(156-300)\*k+y);
270. line((115-230)\*i\*k+x,(82-300)\*k+y,(129-230)\*i\*k+x,(39-300)\*k+y);
271. line((129-230)\*i\*k+x,(39-300)\*k+y,(174-230)\*i\*k+x,(16-300)\*k+y);
272. line((174-230)\*i\*k+x,(16-300)\*k+y,(230-230)\*i\*k+x,(11-300)\*k+y);
273. line((174-230)\*i\*k+x,(16-300)\*k+y,(142-230)\*i\*k+x,(23-300)\*k+y);
274. line((142-230)\*i\*k+x,(23-300)\*k+y,(115-230)\*i\*k+x,(39-300)\*k+y);
275. line((115-230)\*i\*k+x,(39-300)\*k+y,(111-230)\*i\*k+x,(49-300)\*k+y);
276. line((111-230)\*i\*k+x,(49-300)\*k+y,(129-230)\*i\*k+x,(38-300)\*k+y);
277. line((112-230)\*i\*k+x,(42-300)\*k+y,(89-230)\*i\*k+x,(84-300)\*k+y);
278. line((89-230)\*i\*k+x,(84-300)\*k+y,(104-230)\*i\*k+x,(40-300)\*k+y);
279. line((104-230)\*i\*k+x,(40-300)\*k+y,(104-230)\*i\*k+x,(40-300)\*k+y);
280. line((104-230)\*i\*k+x,(40-300)\*k+y,(230-230)\*i\*k+x,(4-300)\*k+y);
281. line((89-230)\*i\*k+x,(84-300)\*k+y,(89-230)\*i\*k+x,(221-300)\*k+y);
282. line((89-230)\*i\*k+x,(221-300)\*k+y,(102-230)\*i\*k+x,(211-300)\*k+y);
283. line((94-230)\*i\*k+x,(217-300)\*k+y,(103-230)\*i\*k+x,(222-300)\*k+y);
284. line((91-230)\*i\*k+x,(221-300)\*k+y,(103-230)\*i\*k+x,(227-300)\*k+y);
285. line((89-230)\*i\*k+x,(221-300)\*k+y,(72-230)\*i\*k+x,(232-300)\*k+y);
286. line((72-230)\*i\*k+x,(232-300)\*k+y,(69-230)\*i\*k+x,(240-300)\*k+y);
287. line((69-230)\*i\*k+x,(240-300)\*k+y,(78-230)\*i\*k+x,(242-300)\*k+y);
288. line((78-230)\*i\*k+x,(242-300)\*k+y,(104-230)\*i\*k+x,(231-300)\*k+y);
289. line((89-230)\*i\*k+x,(236-300)\*k+y,(91-230)\*i\*k+x,(495-300)\*k+y);
290. ellipse((109-230)\*i\*k+x,(495-300)\*k+y,-90-45\*(i+1),0-45\*(i+1),18\*k,62\*k);
291. line((109-230)\*i\*k+x,(557-300)\*k+y,(230-230)\*i\*k+x,(576-300)\*k+y);
292. ellipse((230-230)\*i\*k+x,(557-300)\*k+y,-90-45\*(i+1),0-45\*(i+1),121\*k,19\*k);
293. line((133-230)\*i\*k+x,(497-300)\*k+y,(130-230)\*i\*k+x,(559-300)\*k+y);
294. line((104-230)\*i\*k+x,(349-300)\*k+y,(132-230)\*i\*k+x,(349-300)\*k+y);
295. line((90-230)\*i\*k+x,(352-300)\*k+y,(104-230)\*i\*k+x,(349-300)\*k+y);
296. line((89-230)\*i\*k+x,(194-300)\*k+y,(101-230)\*i\*k+x,(197-300)\*k+y);*//左右主体*
297. }
298. }
299. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
300. FUNCTION:DrawArrow
301. DESCRIBETION:绘制箭头
302. INPUT:void
303. RETURN:void
304. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
305. void DrawArrow()
306. {
307. setcolor(BLUE);
308. setlinestyle(0, 0, 1);
309. rectangle(25, 35, 175, 85);
310. rectangle(75, 35, 125, 85);
311. moveto(30, 60);
312. lineto(50, 40);
313. lineto(50, 55);
314. lineto(70, 55);
315. lineto(70, 65);
316. lineto(50, 65);
317. lineto(50, 80);
318. lineto(30, 60);
319. moveto(100, 40);
320. lineto(120, 60);
321. lineto(105, 60);
322. lineto(105, 80);
323. lineto(95, 80);
324. lineto(95, 60);
325. lineto(80, 60);
326. lineto(100, 40);
327. moveto(170, 60);
328. lineto(150, 40);
329. lineto(150, 55);
330. lineto(130, 55);
331. lineto(130, 65);
332. lineto(150, 65);
333. lineto(150, 80);
334. lineto(170, 60);
335. }
336. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
337. FUNCTION:DrawTime
338. DESCRIBETION:显示时间
339. INPUT:int
340. now[0]=info->tm\_sec;
341. RETURN:void
342. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
343. void DrawTime(int x,int y,int k)*//x,y确定位置，k可确定大小*
344. {
345. time\_t curtime;
347. setfillstyle(1, WHITE);
348. time(&curtime);
349. bar(x,y,x+270\*k,y+20\*k);
350. settextstyle(3,0,k);
351. setcolor(BLUE);
352. outtextxy(x,y,ctime(&curtime));
353. bar(x+170\*k,y,x+270\*k,y+20);
354. delay(100);
355. }
356. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
357. FUNCTION:DrawDisplayChangePart
358. DESCRIBETION:显示页可变部分
359. INPUT:int\* num
360. RETURN:int
361. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
362. void DrawDisplayChangePart(int \*num)
363. {
364. if(num[0]==1)
365. {
366. if(num[2]>20 && num[2]<=100)
367. {
368. Run(num);
369. num[0]=0;
370. }
371. else if(num[2]<=20)
372. {
373. puthz(410,40,"电量不足",16,27,BLUE);
374. num[0]=0;
375. }
376. else
377. {
378. puthz(410,40,"电量异常",16,27,BLUE);
379. Run(num);
380. num[0]=0;
381. }
382. }
383. }
384. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
385. FUNCTION:DisplayMouse
386. DESCRIBETION:显示页鼠标
387. INPUT:int\* num
388. RETURN:int
389. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
390. int DisplayMouse(int\* num)
391. {
392. int choice[10]={0};
394. time\_t current\_time;
395. DrawTime(32,2,1);
396. while(1)
397. {
398. *//绘制鼠标和设置按钮*
399. newmouse(&MouseX, &MouseY, &press);
401. choice[1]=mouse\_press(300,410,340,450);
402. choice[0]=mouse\_press(0,0,29,29);
404. current\_time=time(NULL);
406. if(current\_time%60==0)
407. {
408. DrawTime(32,2,1);*//显示时间*
409. }
411. *//判断选项*
412. if(choice[0] == 1)
413. {
414. num[0]=0;
415. ClearGraph(WHITE);
416. return 0;
417. }
418. else if(choice[1] == 1)
419. {
420. num[0]^=1;
421. break;
422. }
424. }
425. delay(100);
426. return 1;
427. }
428. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
429. FUNCTION:Run
430. DESCRIBETION:运行,车辆行进
431. INPUT:int \*num
432. RETURN:void
433. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
434. void Run(int \*num)
435. {
436. int i,j,k,n[5]={1},battery;
437. FILE \*battery\_text,\*display\_text;
438. NormalRun(num);
439. line(220,400,420,400);*//分界*
440. circle(320,430,20);
441. line(320,440,320,420);
442. line(320,420,310,430);
443. line(320,420,330,430);
444. puthz(295,450,"启动",24,27,BLUE);*//启动*
445. num[3]=num[4]=num[5]=0;
446. num[0]=num[1]=0;
447. }
448. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
449. FUNCTION:DrawBackground
450. DESCRIBETION:绘制背景，设置障碍物
451. INPUT:int \*num
452. RETURN:void
453. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
454. void DrawBackground(int \*num)
455. {
456. int i=0,j,k,n[5]={1},m,s=5;
457. FILE \*battery\_text;
459. i=num[3];
460. j=num[4];
461. k=num[5];
462. Draw(BLUE);
463. setfillstyle(1,WHITE);
464. line(250,40,250,400);
465. line(270,40,270,400);
466. line(380,40,380,400);
467. line(400,40,400,400);
469. srand((unsigned) time(NULL));
470. m=rand()%20;
471. while(1)
472. {
473. DisplayChange(num,num[2]);
474. while(1)
475. {
476. Draw(BLUE);
477. setfillstyle(1,WHITE);
478. delay(30);
479. bar(300,0,340,479);
480. DrawCar(320,240,0.2);
481. for(k=0;k<200;k++)
482. {
483. setfillstyle(1,WHITE);
484. rectangle(310,-100+s\*j-110\*k,330,0+s\*j-110\*k);
485. }
486. srand((unsigned) time(NULL));
487. if(m<30)
488. {
489. n[i]=rand()%12000;*//不可避开障碍物*
490. setfillstyle(1,RED);
491. fillellipse(320,-fabs(n[i])+s\*j,20,20);
492. }
493. setfillstyle(1,WHITE);
494. bar(230,0,400,40);
495. bar(230,400,400,479);
496. bar(280,182,360,292);
497. DrawCar(320,240,0.2);
498. if(j%200==0)
499. {
500. num[2]--;
501. }
502. num[1]=s\*6;
503. DisplayChange(num,num[2]);
504. if(m<30)
505. {
506. if(-fabs(n[i])+s\*j+20>42 && -fabs(n[i])+s\*j-20<160)
507. {
508. s=0;
509. num[1]=s\*6;
510. goto end;
511. }
512. else if(-fabs(n[i])+s\*j+20>160 && -fabs(n[i])+s\*j-20<295)
513. {
514. bar(300,-fabs(n[i])+s\*j-20,340,-fabs(n[i])+s\*j+20);
515. }
516. else ;
517. }
518. j++;
519. if(j%5==0)
520. {
521. break;
522. }
523. }
524. DisplayChange(num,num[2]);
526. end:
527. if(!s)
528. {
529. DisplayChange(num,num[2]);
530. rectangle(310,410,370,450);
531. puthz(320,420,"清除",16,27,BLUE);
532. puthz(500,10,"发现障碍物",16,27,RED);
533. do
534. {
535. newmouse(&MouseX, &MouseY, &press);
536. }while(!(mouse\_press(310,410,370,450)==1));
537. setfillstyle(1,WHITE);
538. bar(500, 10, 640, 30);
539. bar(300,-fabs(n[i])+s\*j-20,340,-fabs(n[i])+s\*j+20);
540. s=5;
541. }
542. else
543. {
544. num[3]=i;num[4]=j;num[5]=k;
545. break;
546. }
547. }
548. }
549. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
550. FUNCTION:DisplayChange
551. DESCRIBETION:改变速度和电量
552. INPUT:int \*num,int battery
553. RETURN:void
554. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
555. void DisplayChange(int \*num,int battery)
556. {
557. FILE \*battery\_text;
558. int speed,a,n[5]={1};
559. float pi=3.1415926535,k;
560. char \*p;
561. k=pi/180;
562. speed=num[1];
563. if(battery>100 || battery<0)
564. {
565. battery=0;
566. }
568. Draw(BLUE);
569. for(a=0;a<11;a++)
570. {
571. line(530+80\*cos((-55+11\*a)\*k),230-82\*sin((-55+11\*a)\*k),530+90\*cos((-55+11\*a)\*k),230-90\*sin((-55+11\*a)\*k));
572. }
573. for(a=0;a<battery/10.0;a++)
574. {
575. setfillstyle(1, GREEN);
576. floodfill(530+85\*cos((-50+11\*a)\*k),230-85\*sin((-50+11\*a)\*k), BLUE);
577. }
578. for(a=battery/10.0;a<10;a++)
579. {
580. setfillstyle(1, WHITE);
581. floodfill(530+85\*cos((-50+11\*a)\*k),230-85\*sin((-50+11\*a)\*k), BLUE);
582. }
584. Draw(BLUE);
585. p=(int \*) malloc(sizeof(int));
586. itoa(num[2],p,10);
587. settextstyle(3,0,1);
588. setfillstyle(1,WHITE);
589. bar(500,314,560,343);
590. outtextxy(500,314,p);
591. outtextxy(540,314,"%");*//电量*
592. free(p);
594. for(a=0;a<speed\*0.3;a++)
595. {
596. setfillstyle(1, RED);
597. floodfill(110+67\*cos((-125-10\*a)\*k),230-67\*sin((-125-10\*a)\*k),BLUE);
598. floodfill(110+62\*cos((-125-10\*a)\*k),230-62\*sin((-125-10\*a)\*k),BLUE);
599. }
600. for(a=speed\*0.3;a<10;a++)
601. {
602. setfillstyle(1, WHITE);
603. floodfill(110+67\*cos((-125-10\*a)\*k),230-67\*sin((-125-10\*a)\*k),BLUE);
604. floodfill(110+62\*cos((-125-10\*a)\*k),230-62\*sin((-125-10\*a)\*k),BLUE);
605. }*//速度*
606. }

**normal.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \*版本号：1.0
4. \*作者：X
5. \*生成日期：2024—4—21
6. \*说明：Normal类函数表示正常情况下运行
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include<stdarg.h>
10. #include"init.h"
11. #include"normal.h"
12. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
13. FUNCTION:NormalRun
14. DESCRIBETION:正常情况下运行
15. INPUT:int\*
16. REUTRUN:int
17. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
18. int NormalRun(int\* bg)
19. {
20. int distance[40] = {0}, place[40] = {0};
21. int i, num = -1, sum\_distance = 0, now\_distance;
22. char text[10];
23. FILE\* path\_text;
24. path\_text = fopen("..\\CODE\\text\\path.txt", "rt");
25. do
26. {
27. num++;
28. fscanf(path\_text, "%d %d ", &place[num], &distance[num]);
29. sum\_distance+=distance[num];
30. } while (distance[num]!=1000);
31. fclose(path\_text);
32. place[num+1] = 1000;
33. sum\_distance-=1000;
34. now\_distance=sum\_distance;
35. setfillstyle(1, WHITE);
36. bar(0, 106, 200, 140);
37. puthz(25, 108, "剩余里程", 16, 16, BLUE);
38. puthz(25, 124, "剩余时间", 16, 16, BLUE);
39. puthz(113, 108, "米", 16, 16, BLUE);
40. puthz(113, 124, "秒", 16, 16, BLUE);
41. setcolor(RED);
42. settextstyle(2, 0, 5);
43. itoa(sum\_distance, text, 10);
44. outtextxy(89, 108, text);
45. itoa(sum\_distance/5, text, 10);
46. outtextxy(89, 124, text);
48. for(i = num; i > -1; i--)
49. {
50. CaseRun(bg, place, distance, i, &now\_distance);
51. }
52. return 1;
53. }
54. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
55. FUNCTION:CaseRun
56. DESCRIBETION:根据不同的情况选择在不同的道路上运行
57. INPUT:int\* place, int\* distance, int i, int\* now\_distance
58. REUTRUN:void
59. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
60. void CaseRun(int\* bg, int\* place, int\* distance, int i, int\* pnow\_distance)
61. {
62. int j;
63. if(i == 0)
64. {
65. FillArrow(4);
66. setfillstyle(1, WHITE);
67. bar(0, 90, 200, 140);
68. puthz(25, 90, "到达目的地附近", 16, 16, BLUE);
69. return;
70. }
71. if(place[i]==0){
72. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过华中路");}
73. else if(place[i]==1){
74. if(place[i-1]==0){
75. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过华中路");}
76. else if(place[i-1]==14){
77. if(place[i-2]==15){
78. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过华中路");}
79. else if(place[i-2]==19){
80. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过华中路", "左转，进入红棉路");}}}
81. else if(place[i]==2){
82. if(place[i-1]==15){
83. if(place[i-2]==7){
84. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过南三路");}
85. else if(place[i-2]==14){
86. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过南三路", "左转，进入华中路");}
87. else if(place[i-2]==16){
88. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过南三路", "右转，进入华中路");}}}
89. else if(place[i]==3){
90. if(place[i-2]==6){
91. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过紫荆路");}
92. else if(place[i-2]==15){
93. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过紫荆路", "左转，进入华中路");}
94. else if(place[i-2]==17){
95. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过紫荆路", "右转，进入华中路");}}
96. else if(place[i]==4){
97. if(place[i-2]==17){
98. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过无名路", "左转，进入华中路");}
99. else if(place[i-2]==26){
100. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过无名路");}}
101. else if(place[i]==5){
102. if(place[i-1]==17){
103. if(place[i-2]==16){
104. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行后右转，进入华中路", "右转，进入华中路");}
105. else if(place[i-2]==18){
106. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行后左转，进入华中路", "左转，进入华中路");}}
107. if(place[i-1]==25){
108. if(place[i-2]==24){
109. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行后左转，进入无名路", "左转，进入无名路");}
110. else if(place[i-2]==26){
111. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行后右转，进入无名路", "右转，进入无名路");}}}
112. else if(place[i]==6){
113. if(place[i-1]==16){
114. if(place[i-2]==3){
115. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过紫荆路");}
116. else if(place[i-2]==15){
117. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过紫荆路", "右转，进入华中路");}
118. else if(place[i-2]==17){
119. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过紫荆路", "左转，进入华中路");}}
120. else if(place[i-1]==23){
121. if(place[i-2]==22){
122. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过紫荆路", "左转，进入东一路");}
123. else if(place[i-2]==24){
124. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过紫荆路");}}}
125. else if(place[i]==7){
126. if(place[i-1]==15){
127. if(place[i-2]==2){
128. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过南三路");}
129. else if(place[i-2]==14){
130. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过南三路", "右转，进入华中路");}
131. else if(place[i-2]==16){
132. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过南三路", "左转，进入华中路");}}
133. else if(place[i-1]==20){
134. if(place[i-2]==8){
135. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过南三路", "左转，进入停车场");}
136. else if(place[i-2]==22){
137. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过南三路");}}}
138. else if(place[i]==8){
139. if(place[i-1]==19){
140. if(place[i-2]==14){
141. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "", "左转，进入红棉路");}
142. else if(place[i-2]==21){
143. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "", "右转，进入红棉路");}}
144. else if(place[i-1]==20){
145. if(place[i-2]==7){
146. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "", "右转，进入南三路");}
147. else if(place[i-2]==22){
148. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "", "左转，进入南三路");}}}
149. else if(place[i]==9){
150. if(place[i-2]==10){
151. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过东一路", "左转，进入红棉路");}
152. else if(place[i-2]==19){
153. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过东一路", "右转，进入红棉路");}
154. else if(place[i-2]==22){
155. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过东一路");}}
156. else if(place[i]==10){
157. if(place[i-1]==21){
158. if(place[i-2]==9){
159. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过红棉路", "右转，进入东一路");}
160. else if(place[i-2]==19){
161. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过红棉路");}
162. else if(place[i-2]==22){
163. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过红棉路", "左转，进入东一路");}}
164. if(place[i-1]==30){
165. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过红棉路", "右转，进入醉晚路");}}
166. else if(place[i]==11){
167. if(place[i-1]==27){
168. if(place[i-2]==22){
169. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过南三路");}
170. else if(place[i-2]==28){
171. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过南三路", "左转，进入无名路");}}
172. else if(place[i-1]==31){
173. if(place[i-2]==12){
174. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过南三路", "右转，进入醉晚路");}
175. else if(place[i-2]==30){
176. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过南三路", "左转，进入醉晚路");}}}
177. else if(place[i]==12){
178. if(place[i-1]==31){
179. if(place[i-2]==11){
180. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过醉晚路", "左转，进入南三路");}
181. else if(place[i-2]==30){
182. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过醉晚路");}}
183. else if(place[i-1]==32){
184. if(place[i-2]==28){
185. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过醉晚路", "右转，进入紫荆路");}
186. else if(place[i-2]==33){
187. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过醉晚路");}}}
188. else if(place[i]==13){
189. if(place[i-1]==29){
190. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过无名路", "右转，进入无名路");}
191. else if(place[i-1]==33){
192. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过无名路", "左转，进入醉晚场");}}
193. else if(place[i]==14){
194. if(place[i-1]==1){
195. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过华中路");}
196. else if(place[i-1]==15){
197. if(place[i-2]==2){
198. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过华中路", "右转，进入南三路");}
199. else if(place[i-2]==7){
200. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过华中路", "左转，进入南三路");}
201. else if(place[i-2]==16){
202. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过华中路");}}
203. else if(place[i-1]==19){
204. if(place[i-2]==8){
205. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过红棉路", "右转，进入停车场");}
206. else if(place[i-2]==21){
207. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过红棉路");}}}
208. else if(place[i]==15){
209. if(place[i-1]==2 || place[i-1]==7){
210. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过南三路");}
211. else if(place[i-1]==14){
212. if(place[i-2]==1){
213. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过华中路");}
214. else if(place[i-2]==19){
215. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过华中路", "右转，进入红棉路");}}
216. else if(place[i-1]==16){
217. if(place[i-2]==3){
218. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过华中路", "右转，进入紫荆路");}
219. else if(place[i-2]==6){
220. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过华中路", "左转，进入紫荆路");}
221. else if(place[i-2]==17){
222. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过华中路");}}}
223. else if(place[i]==16){
224. if(place[i-1]==3 || place[i-1]==6){
225. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过紫荆路");}
226. else if(place[i-1]==15){
227. if(place[i-2]==2){
228. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过华中路", "左转，进入南三路");}
229. else if(place[i-2]==7){
230. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过华中路", "右转，进入南三路");}
231. else if(place[i-2]==14){
232. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过华中路");}}
233. else if(place[i-1]==17){
234. if(place[i-2]==5){
235. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过华中路", "左转，到达南一楼");}
236. else if(place[i-2]==18){
237. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过华中路");}}}
238. else if(place[i]==17){
239. if(place[i-1]==5){
240. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，到达南一楼");}
241. else if(place[i-1]==16){
242. if(place[i-2]==3){
243. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过华中路", "左转，进入紫荆路");}
244. else if(place[i-2]==6){
245. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过华中路", "右转，进入紫荆路");}
246. else if(place[i-2]==15){
247. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过华中路");}}
248. else if(place[i-1]==18){
249. if(place[i-2]==4){
250. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过华中路", "右转，到达建规学院");}
251. else if(place[i-2]==26){
252. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过华中路", "左转，进入无名路");}}}
253. else if(place[i]==18){
254. if(place[i-1]==4){
255. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，到达建规学院");}
256. else if(place[i-1]==17){
257. if(place[i-2]==5){
258. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过华中路", "右转，到达南一楼");}
259. else if(place[i-2]==16){
260. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过华中路");}}
261. else if(place[i-1]==26){
262. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过无名路", "左转，进入无名路");}}
263. else if(place[i]==19){
264. if(place[i-1]==8){
265. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，到达停车场");}
266. else if(place[i-1]==14){
267. if(place[i-2]==1){
268. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过红棉路", "右转，进入华中路");}
269. else if(place[i-2]==15){
270. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过红棉路", "左转，进入华中路");}}
271. else if(place[i-1]==21){
272. if(place[i-2]==9){
273. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过红棉路", "左转，进入东一路");}
274. else if(place[i-2]==10){
275. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过红棉路");}
276. else if(place[i-2]==22){
277. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过红棉路", "右转，进入东一路");}}}
278. else if(place[i]==20){
279. if(place[i-1]==7){
280. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过南三路");}
281. else if(place[i-1]==8){
282. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，到达停车场");}
283. else if(place[i-1]==22){
284. if(place[i-2]==21){
285. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过南三路", "左转，进入东一路");}
286. else if(place[i-2]==23){
287. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过南三路", "右转，进入华中路");}
288. else if(place[i-2]==27){
289. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过南三路");}}}
290. else if(place[i]==21){
291. if(place[i-1]==9){
292. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过东一路");}
293. else if(place[i-1]==10){
294. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过红棉路");}
295. else if(place[i-1]==19){
296. if(place[i-2]==8){
297. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过红棉路", "左转，进入停车场");}
298. else if(place[i-2]==14){
299. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过红棉路");}}
300. else if(place[i-1]==22){
301. if(place[i-2]==20){
302. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过东一路", "右转，进入南三路");}
303. else if(place[i-2]==23){
304. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过东一路");}
305. else if(place[i-2]==27){
306. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过东一路");}}}
307. else if(place[i]==22){
308. if(place[i-1]==20){
309. if(place[i-2]==8){
310. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过南三路", "右转，进入停车场");}
311. else if(place[i-2]==7){
312. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过南三路");}}
313. else if(place[i-1]==21){
314. if(place[i-2]==9){
315. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过东一路");}
316. else if(place[i-2]==10){
317. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过东一路", "右转，进入红棉路");}
318. else if(place[i-2]==19){
319. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过东一路", "左转，进入红棉路");}}
320. else if(place[i-1]==23){
321. if(place[i-2]==6){
322. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过东一路", "右转，进入紫荆路");}
323. else if(place[i-2]==24){
324. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过东一路", "左转，进入紫荆路");}}
325. else if(place[i-1]==27){
326. if(place[i-2]==11){
327. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过南三路");}
328. else if(place[i-2]==28){
329. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过东一路", "右转，进入无名路");}}}
330. else if(place[i]==23){
331. if(place[i-1]==6){
332. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过紫荆路");}
333. else if(place[i-1]==22){
334. if(place[i-2]==20){
335. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过东一路", "左转，进入南三路");}
336. else if(place[i-2]==21){
337. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过东一路");}
338. else if(place[i-2]==27){
339. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过东一路", "右转，进入南三路");}}
340. else if(place[i-1]==24){
341. if(place[i-2]==25){
342. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过紫荆路", "右转，进入无名路");}
343. else if(place[i-2]==28){
344. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过紫荆路");}}}
345. else if(place[i]==24){
346. if(place[i-1]==23){
347. if(place[i-2]==6){FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过紫荆路");}
348. else if(place[i-2]==22){
349. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过紫荆路", "右转，进入东一路");}}
350. else if(place[i-1]==25){
351. if(place[i-2]==5){
352. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过无名路", "右转，到达南一楼");}
353. else if(place[i-2]==26){
354. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过无名路");}}
355. else if(place[i-1]==28){
356. if(place[i-2]==27){
357. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过紫荆路", "左转，进入无名路");}
358. else if(place[i-2]==29){
359. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过紫荆路", "右转，进入无名路");}
360. else if(place[i-2]==32){
361. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过紫荆路");}}}
362. else if(place[i]==25){
363. if(place[i-1]==5){
364. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，到达南一楼");}
365. else if(place[i-1]==24){
366. if(place[i-2]==23){
367. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过无名路", "左转，进入紫荆路");}
368. else if(place[i-2]==28){
369. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过无名路", "右转，进入紫荆路");}}
370. else if(place[i-1]==26){
371. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过无名路", "右转，进入无名路");}}
372. else if(place[i]==26){
373. if(place[i-1]==18){
374. if(place[i-2]==4){
375. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过无名路");}
376. else if(place[i-2]==17){
377. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过无名路", "右转，进入华中路");}}
378. else if(place[i-1]==25){
379. if(place[i-2]==5){
380. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过无名路", "左转，到达南一楼");}
381. else if(place[i-2]==24){
382. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过无名路");}}}
383. else if(place[i]==27){
384. if(place[i-1]==11){
385. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过南三路");}
386. else if(place[i-1]==22){
387. if(place[i-2]==20){
388. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过南三路");}
389. else if(place[i-2]==21){
390. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过南三路", "右转，进入东一路");}
391. else if(place[i-2]==23){
392. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过南三路", "左转，进入东一路");}}
393. else if(place[i-1]==28){
394. if(place[i-2]==24){
395. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过无名路", "右转，进入紫荆路");}
396. else if(place[i-2]==29){
397. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过无名路");}
398. else if(place[i-2]==32){
399. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过无名路", "左转，进入紫荆路");}}}
400. else if(place[i]==28){
401. if(place[i-1]==24){
402. if(place[i-2]==23){
403. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过紫荆路");}
404. else if(place[i-2]==25){
405. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过紫荆路", "左转，进入无名路");}}
406. else if(place[i-1]==27){
407. if(place[i-2]==11){
408. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过无名路", "右转，进入南三路");}
409. else if(place[i-2]==22){
410. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过无名路", "左转，进入南三路");}}
411. else if(place[i-1]==29){
412. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过无名路", "左转，进入无名路");}
413. else if(place[i-1]==32){
414. if(place[i-2]==31){
415. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过紫荆路", "左转，进入醉晚路");}
416. else if(place[i-2]==33){
417. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过紫荆路", "右转，进入醉晚路");}}}
418. else if(place[i]==29){
419. if(place[i-1]==13){
420. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过无名路");}
421. else if(place[i-1]==24){
422. if(place[i-2]==25){
423. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过紫荆路", "右转，进入无名路");}
424. else if(place[i-2]==28){
425. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过紫荆路");}}
426. else if(place[i-1]==28){
427. if(place[i-2]==24){
428. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过无名路", "左转，进入紫荆路");}
429. else if(place[i-2]==27){
430. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过无名路");}
431. else if(place[i-2]==32){
432. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过无名路", "右转，进入紫荆路");}}}
433. else if(place[i]==30){
434. if(place[i-1]==10){
435. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过红棉路");}
436. else if(place[i-1]==31){
437. if(place[i-2]==11){
438. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过醉晚路", "右转，进入南三路");}
439. else if(place[i-2]==12){
440. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过醉晚路");}}}
441. else if(place[i]==31){
442. if(place[i-1]==11){
443. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过南三路");}
444. else if(place[i-1]==12){
445. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过醉晚路");}
446. else if(place[i-1]==30){
447. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过醉晚路", "左转，进入红棉路");}}
448. else if(place[i]==32){
449. if(place[i-1]==12){
450. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过醉晚路");}
451. else if(place[i-1]==28){
452. if(place[i-2]==24){
453. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过紫荆路");}
454. else if(place[i-2]==27){
455. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过紫荆路", "右转，进入无名路");}
456. else if(place[i-2]==29){
457. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过紫荆路", "左转，进入无名路");}}
458. else if(place[i-1]==33){
459. FragmentRun(bg, distance, i, RIGHT, pnow\_distance, 2, "直行，通过醉晚路", "右转，进入无名路");}}
460. else if(place[i]==33){
461. if(place[i-1]==13){
462. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过无名路");}
463. else if(place[i-1]==32){
464. if(place[i-2]==12){
465. FragmentRun(bg, distance, i, STRIGHT, pnow\_distance, 1, "直行，通过醉晚路");}
466. else if(place[i-2]==28){
467. FragmentRun(bg, distance, i, LEFT, pnow\_distance, 2, "直行，通过醉晚路", "左转，进入紫荆路");}}}
468. }
469. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
470. FUNCTION:FragmentRun
471. DESCRIBETION:运行一段道路
472. INPUT:int\*distance, int i, int direction, int\* pnow\_distance, int num, ...
473. REUTRUN:void
474. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
475. void FragmentRun(int\* bg, int\*distance, int i, int direction, int\* pnow\_distance, int num, ...)
476. {
477. int j;
478. char hanzi[2][100];
479. va\_list args;
480. va\_start(args, num);
481. strcpy(hanzi[0], va\_arg(args, const char\*));
482. if(num==2)
483. strcpy(hanzi[1], va\_arg(args, const char\*));
484. setfillstyle(1, WHITE);
485. bar(0, 90, 200, 106);
486. puthz(25, 90, hanzi[0], 16, 16, BLUE);
487. for(j = 0; j < distance[i-1]-10; j++)
488. RoutineRun(bg,pnow\_distance, STRIGHT);
489. if(direction == STRIGHT)
490. {
491. for(j = distance[i-1]-10; j < distance[i-1]; j++)
492. RoutineRun(bg,pnow\_distance, direction);
493. }
494. else if(direction == LEFT)
495. {
496. setfillstyle(1, WHITE);
497. bar(0, 90, 200, 106);
498. puthz(25, 90, hanzi[1], 16, 16, BLUE);
499. for(j = distance[i-1]-10; j < distance[i-1]; j++)
500. RoutineRun(bg,pnow\_distance, direction);
501. }
502. else if(direction == RIGHT)
503. {
504. setfillstyle(1, WHITE);
505. bar(0, 90, 200, 106);
506. puthz(25, 90, hanzi[1], 16, 16, BLUE);
507. for(j = distance[i-1]-10; j < distance[i-1]; j++)
508. RoutineRun(bg,pnow\_distance, direction);
509. }
510. }
511. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
512. FUNCTION:RoutineRun
513. DESCRIBETION:直行
514. INPUT:int\* pnow\_distance, int direction
515. REUTRUN:
516. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
517. void RoutineRun(int\* bg, int\* pnow\_distance, int direction)
518. {
519. FillArrow(direction);
520. \*pnow\_distance-=1;
521. PutMiles(pnow\_distance);
522. DrawBackground(bg);
523. }
524. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
525. FUNCTION:FillArrow
526. DESCRIBETION:填充箭头
527. INPUT:mode
528. REUTRUN:void
529. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
530. void FillArrow(int mode)
531. {
532. setfillstyle(1, WHITE);
533. floodfill(50, 60, BLUE);
534. floodfill(100, 60, BLUE);
535. floodfill(150, 60, BLUE);
536. setfillstyle(1, YELLOW);
537. switch (mode)
538. {
539. case LEFT:
540. floodfill(50, 60, BLUE);
541. break;
542. case STRIGHT:
543. floodfill(100, 60, BLUE);
544. break;
545. case RIGHT:
546. floodfill(150, 60, BLUE);
547. break;
548. default:
549. break;
550. }
551. }
552. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
553. FUNCTION:PutMiles
554. DESCRIBETION:显示里程等信息
555. INPUT:int\* pnow\_distance
556. REUTRUN:void
557. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
558. void PutMiles(int\* pnow\_distance)
559. {
560. char text[10];
561. setfillstyle(1, WHITE);
562. bar(89, 108, 113, 140);
563. setcolor(RED);
564. settextstyle(2, 0, 5);
565. itoa(\*pnow\_distance, text, 10);
566. outtextxy(89, 108, text);
567. itoa((\*pnow\_distance)/5, text, 10);
568. outtextxy(89, 124, text);
569. }

**navigation.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \*   版本号：1.0
4. \*   作者：X
5. \*   生成日期：2024—3—28
6. \*   说明：Navigation类函数实现导航功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include"init.h"
10. #include"map.h"
11. #include"navigat.h"
12. #include"plan.h"
13. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
14. FUNCTION:OpenNavigation
15. DESCRIBETION:打开导航页
16. INPUT:void
17. RETURN:void
18. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
19. void OpenNavigation()
20. {
21. int mode[2] = {1, 1};
22. while(1)
23. {
24. int navigation\_ret = 1;
25. ClearGraph(WHITE);
26. DrawReturnButton();
27. DrawMap();
28. DrawNavigationTopButton();
29. navigation\_ret = NavigationMouse(mode);
30. if(!navigation\_ret)
31. return ;
32. }
33. }
34. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
35. FUNCTION:DrawNavigationTopButton
36. DESCRIBETION:绘制导航页顶部按钮
37. INPUT:void
38. RETURN:void
39. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
40. void DrawNavigationTopButton()
41. {
42. setlinestyle(0, 0, 1);
43. setcolor(BLUE);
44. rectangle(45, 0, 165, 28);
45. puthz(48, 2, "选择起点", 24, 30, BLUE);
46. rectangle(195, 0, 315, 28);
47. puthz(198, 2, "选择终点", 24, 30, BLUE);
48. rectangle(345, 0, 465, 28);
49. puthz(348, 2, "重新选择", 24, 30, BLUE);
50. rectangle(495, 0, 615, 28);
51. puthz(498, 2, "开始规划", 24, 30, BLUE);
52. }
53. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
54. FUNCTION:NavigationMouse
55. DESCRIBETION:导航页鼠标
56. INPUT:int\* mode
57. RETURN:int
58. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
59. int NavigationMouse(int\* mode)
60. {
61. int choice[5] = {0}, start = -1, end = -1, distance = 0;
62. char dist[10];
63. while(1)
64. {
65. newmouse(&MouseX,&MouseY,&press);
66. choice[0]=mouse\_press(0,0,29,29);
67. choice[1]=mouse\_press(45, 0, 165, 28);
68. choice[2]=mouse\_press(195, 0, 315, 28);
69. choice[3]=mouse\_press(345, 0, 465, 28);
70. choice[4]=mouse\_press(495, 0, 615, 28);
71. if(choice[0] == 1)
72. {
73. return 0;
74. }
75. if(choice[1] == 1)
76. {
77. if(mode[0])
78. {
79. start = ChoosePlace();
80. mode[0] = 0;
81. }
82. }
83. if(choice[2] == 1)
84. {
85. if(mode[1])
86. {
87. end = ChoosePlace();
88. mode[1] = 0;
89. }
90. }
91. if(choice[3] == 1)
92. {
93. mode[0] = 1;
94. mode[1] = 1;
95. break;
96. }
97. if(choice[4] == 1 && start != -1 && end != -1)
98. {
99. distance = PlanPath(start, end);
100. DrawRoute();
101. setcolor(RED);
102. settextstyle(2, 0, 6);
103. itoa(distance, dist, 10);
104. puthz(3, 33, "预", 24, 0, BLUE);
105. puthz(3, 63, "计", 24, 0, BLUE);
106. puthz(3, 93, "里", 24, 0, BLUE);
107. puthz(3, 123, "程", 24, 0, BLUE);
108. outtextxy(0, 150, dist);
109. puthz(3, 170, "米", 24, 0, BLUE);
110. itoa(distance/5, dist, 10);
111. puthz(3, 233, "预", 24, 0, BLUE);
112. puthz(3, 263, "计", 24, 0, BLUE);
113. puthz(3, 293, "用", 24, 0, BLUE);
114. puthz(3, 323, "时", 24, 0, BLUE);
115. outtextxy(0, 350, dist);
116. puthz(3, 370, "秒", 24, 0, BLUE);
117. start = -1, end = -1;
118. }
119. }
120. return 1;
121. }
122. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
123. FUNCTION:ChoosePlace
124. DESCRIBETION:选择地点
125. INPUT:void
126. RETURN:int
127. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
128. int ChoosePlace()
129. {
130. int place[14] = {0}, i = 0;
131. while(1)
132. {
133. newmouse(&MouseX,&MouseY,&press);
134. place[ 0]=mouse\_press(40,40,120,70);        *//科技楼北楼*
135. place[ 1]=mouse\_press(40,90,120,120);       *//科技楼南楼*
136. place[ 2]=mouse\_press(30, 150, 80, 270);    *//图书馆*
137. place[ 3]=mouse\_press(60, 340, 110, 370);   *//南三楼*
138. place[ 4]=mouse\_press(40, 440, 110, 470);   *//建规学院*
139. place[ 5]=mouse\_press(135, 335, 205, 365)+mouse\_press(135, 395, 205, 425)+\
140. mouse\_press(305, 335, 375, 365)+mouse\_press(305, 395, 375, 425)+mouse\_press(175, 365, 335, 395);
141. place[ 6]=mouse\_press(215, 335, 295, 360);  *//南六楼*
142. place[ 7]=mouse\_press(160, 240, 260, 290);  *//校史馆*
143. place[ 8]=mouse\_press(140, 140, 320, 220);  *//停车场*
144. place[ 9]=mouse\_press(340, 80, 390, 110);   *//网球场*
145. place[10]=mouse\_press(430, 140, 530, 170);  *//东八楼*
146. place[11]=mouse\_press(405, 175, 555, 225);  *//东一楼*
147. place[12]=mouse\_press(570, 250, 620, 280);  *//东三楼*
148. place[13]=mouse\_press(450, 440, 500, 480);  *//游泳馆*
150. setfillstyle(1, GREEN);
151. if(place[0]==1)
152. {
153. fillellipse(80, 55, 5, 5);
154. }
155. else if(place[1]==1)
156. {
157. fillellipse(80, 105, 5, 5);
158. }
159. else if(place[2]==1)
160. {
161. fillellipse(55, 210, 5, 5);
162. }
163. else if(place[3]==1)
164. {
165. fillellipse(85, 355, 5, 5);
166. }
167. else if(place[4]==1)
168. {
169. fillellipse(75, 455, 5, 5);
170. }
171. else if(place[5]==1)
172. {
173. fillellipse(255, 380, 5, 5);
174. }
175. else if(place[6]==1)
176. {
177. fillellipse(255, 348, 5, 5);
178. }
179. else if(place[7]==1)
180. {
181. fillellipse(210, 265, 5, 5);
182. }
183. else if(place[8]==1)
184. {
185. fillellipse(230, 180, 5, 5);
186. }
187. else if(place[9]==1)
188. {
189. fillellipse(365, 95, 5, 5);
190. }
191. else if(place[10]==1)
192. {
193. fillellipse(480, 155, 5, 5);
194. }
195. else if(place[11]==1)
196. {
197. fillellipse(480, 200, 5, 5);
198. }
199. else if(place[12]==1)
200. {
201. fillellipse(595, 265, 5, 5);
202. }
203. else if(place[13]==1)
204. {
205. fillellipse(475, 460, 5, 5);
206. }
207. for(i = 0; i < 14; i++)
208. {
209. if(place[i] == 1)
210. return i;
211. }
212. }
213. }

**map.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \*   版本号：1.0
4. \*   作者：
5. \*   生成日期：2024—4—10
6. \*   说明：Map类函数绘制地图
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include<graphics.h>
10. #include<hz.h>
11. #include"map.h"
12. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
13. FUNCTION:DrawMap
14. DESCRIBETION:绘制地图
15. INPUT:void
16. RETURN:void
17. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
18. void DrawMap()
19. {
20. DrawRoad();
21. DrawBuilding();
22. setlinestyle(0, 0, 3);
23. setcolor(DARKGRAY);
24. line(30, 30, 30, 479);
25. line(30, 30, 639, 30);
26. }
27. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
28. FUNCTION:DrawRoad
29. DESCRIBETION:绘制道路
30. INPUT:void
31. RETURN:void
32. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
33. void DrawRoad()
34. {
35. setlinestyle(0, 0, 1);
36. setcolor(LIGHTGRAY);
37. DrawVerticalRoad(130, 30, 400);     *//华中路 x=130, 30<y<430*
38. DrawVerticalRoad(230, 130, 10);       *//x=230, 130<y<140*
39. DrawVerticalRoad(230, 220, 10);       *//x=230, 220<y<230*
40. DrawVerticalRoad(330, 30, 300);     *//东一路 x=330, 30<y<330*
41. DrawVerticalRoad(630, 30, 450);     *//醉晚路 x=630, 30<y<480*
42. DrawVerticalRoad(380, 330, 102);    *// x=380, 330<y<430*
43. DrawVerticalRoad(430, 230, 250);    *// x=430, 230<y<480*
44. DrawParallelRoad(30, 130, 600);     *//红棉路 y=130, 30<x<630*
45. DrawParallelRoad(80, 230, 550);     *//南三路 y=230, 80<x<630*
46. DrawParallelRoad(30, 330, 600);     *//紫荆路 y=330, 30<x<630*
47. DrawParallelRoad(130, 380, 45);     *//y=380, 130<x<175*
48. DrawParallelRoad(335, 380, 45);     *//y=380, 335<x<380*
49. DrawParallelRoad(30, 430, 352);     *// y=430, 30<x<380*
50. DrawParallelRoad(430, 430, 200);    *// y=430, 430<x<630*
51. setlinestyle(0, 0, 1);
52. setcolor(WHITE);
54. DrawCrossing(130, 130, 1, 1, 1, 1); *//华中路 红棉路 14*
55. DrawCrossing(130, 230, 1, 1, 1, 1); *//华中路 南三路 15*
56. DrawCrossing(130, 330, 1, 1, 1, 1); *//华中路 紫荆路 16*
57. DrawCrossing(130, 380, 1, 1, 0, 1); *//华中路 y=380  17*
58. DrawCrossing(130, 430, 1, 0, 1, 1); *//华中路 y=430  18*
59. DrawCrossing(230, 130, 0, 1, 1, 1); *//              19*
60. DrawCrossing(230, 230, 1, 0, 1, 1); *//              20*
61. DrawCrossing(330, 130, 1, 1, 1, 1); *//东一路 红棉路 21*
62. DrawCrossing(330, 230, 1, 1, 1, 1); *//东一路 南三路 22*
63. DrawCrossing(330, 330, 1, 0, 1, 1); *//东一路 紫荆路 23*
65. DrawCrossing(380, 330, 0, 1, 1, 1); *//x=380 紫荆路  24*
66. DrawCrossing(380, 380, 1, 1, 1, 0); *//x=380 y=380   25*
67. DrawCrossing(380, 430, 1, 0, 1, 0); *//x=380 y=430   26*
68. DrawCrossing(430, 230, 0, 1, 1, 1); *//x=430 y=230   27*
69. DrawCrossing(430, 330, 1, 1, 1, 1); *//x=430 y=330   28*
70. DrawCrossing(430, 430, 1, 1, 0, 1); *//x=430 y=430   29*
71. DrawCrossing(630, 130, 1, 1, 1, 0); *//醉晚路 红棉路 30*
72. DrawCrossing(630, 230, 1, 1, 1, 0); *//醉晚路 南三路 31*
73. DrawCrossing(630, 330, 1, 1, 1, 0); *//醉晚路 紫荆路 32*
74. DrawCrossing(630, 430, 1, 1, 1, 0); *//醉晚路 y=430  33*
75. }
76. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
77. FUNCTION:DrawParallelRoad
78. DESCRIBETION:绘制横向道路
79. INPUT:int x, int y, int d
80. RETURN:void
81. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
82. void DrawParallelRoad(int x, int y, int d)
83. {
84. line(x, y-2, x+d, y-2);
85. line(x, y+2, x+d, y+2);
86. }
87. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
88. FUNCTION:DrawVerticalRoad
89. DESCRIBETION:绘制纵向道路
90. INPUT:int x, int y, int d
91. RETURN:void
92. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
93. void DrawVerticalRoad(int x, int y, int d)
94. {
95. line(x-2, y, x-2, y+d);
96. line(x+2, y, x+2, y+d);
97. }
98. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
99. FUNCTION:DrawCrossing
100. DESCRIBETION:绘制纵向十字路口
101. INPUT:int x, int y, int north, int south, int west, int east
102. RETURN:void
103. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
104. void DrawCrossing(int x, int y, int north, int south, int west, int east)
105. {
106. if(north)
107. line(x-1, y-2, x+1, y-2);
108. if(south)
109. line(x-1, y+2, x+1, y+2);
110. if(west)
111. line(x-2, y-1, x-2, y+1);
112. if(east)
113. line(x+2, y-1, x+2, y+1);
114. }
115. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
116. FUNCTION:DrawBuilding
117. DESCRIBETION:绘制建筑物
118. INPUT:void
119. RETURN:void
120. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
121. void DrawBuilding()
122. {
123. setlinestyle(0, 0, 1);
124. setcolor(DARKGRAY);
125. rectangle(40, 40, 120, 70);     *//科技楼北楼    0*
126. puthz(40, 47, "科技楼北楼", 16, 16, DARKGRAY);
127. rectangle(40, 90, 120, 120);    *//科技楼南楼    1*
128. puthz(40, 97, "科技楼南楼", 16, 16, DARKGRAY);
129. rectangle(30, 150, 80, 270);    *//图书馆        2*
130. puthz(43, 162, "图", 24, 0, DARKGRAY);
131. puthz(43, 198, "书", 24, 0, DARKGRAY);
132. puthz(43, 234, "馆", 24, 0, DARKGRAY);
133. rectangle(60, 340, 110, 370);   *//南三楼        3*
134. puthz(61, 347, "南三楼", 16, 17, DARKGRAY);
135. rectangle(40, 440, 110, 470);   *//建规学院      4*
136. puthz(41, 447, "建规学院", 16, 17, DARKGRAY);
137. DrawSouthBuildingOne();         *//南一楼        5*
138. rectangle(215, 335, 295, 360);  *//南六楼        6*
139. puthz(223, 340, "南六楼", 16, 24, DARKGRAY);
140. rectangle(160, 240, 260, 290);  *//校史馆        7*
141. puthz(167, 253, "校史馆", 24, 31, DARKGRAY);
142. DrawParkingLot();               *//停车场        8*
143. rectangle(340, 80, 390, 110);   *//网球场        9*
144. puthz(341, 87, "网球场", 16, 17, DARKGRAY);
145. rectangle(430, 140, 530, 170);  *//东八楼        10*
146. puthz(437, 143, "东八楼", 24, 31, DARKGRAY);
147. rectangle(405, 175, 555, 225);  *//东一楼        11*
148. puthz(419, 184, "东一楼", 32, 45, DARKGRAY);
149. rectangle(570, 250, 620, 280);  *//东三楼        12*
150. puthz(571, 257, "东三楼", 16, 17, DARKGRAY);
151. rectangle(450, 440, 500, 480);  *//游泳馆        13*
152. puthz(451, 447, "游泳馆", 16, 17, DARKGRAY);
153. }
154. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
155. FUNCTION:DrawSouthBuildingOne
156. DESCRIBETION:绘制南一楼
157. INPUT:void
158. RETURN:void
159. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
160. void DrawSouthBuildingOne()
161. {
162. moveto(135, 335);
163. lineto(205, 335);
164. lineto(205, 365);
165. lineto(305, 365);
166. lineto(305, 335);
167. lineto(375, 335);
168. lineto(375, 365);
169. lineto(335, 365);
170. lineto(335, 395);
171. lineto(375, 395);
172. lineto(375, 425);
173. lineto(305, 425);
174. lineto(305, 395);
175. lineto(205, 395);
176. lineto(205, 425);
177. lineto(135, 425);
178. lineto(135, 395);
179. lineto(175, 395);
180. lineto(175, 365);
181. lineto(135, 365);
182. lineto(135, 335);               *//南一楼*
183. puthz(197, 368, "南一楼", 24, 46, DARKGRAY);
184. }
185. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
186. FUNCTION:DrawParkingLot
187. DESCRIBETION:绘制
188. INPUT:void
189. RETURN:void
190. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
191. void DrawParkingLot()
192. {
193. rectangle(140, 140, 320, 220);  *//停车场*
194. circle(168, 180, 16);
195. settextstyle(10, 0, 2);
196. outtextxy(161, 154, "P");
197. puthz(194, 164, "停车场", 32, 42, DARKGRAY);
198. }

**plan.c**

1. */\*\**
2. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
3. \*   版本号：1.0
4. \*   作者：X
5. \*   生成日期：2024—3—30
6. \*   说明：Plan类函数实现路径规划功能
7. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*?
8. \*\*/
9. #include<graphics.h>
10. #include<stdio.h>
11. #include"plan.h"
12. #define MAX 9999
13. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
14. FUNCTION:PlanPath
15. DESCRIBETION:规划路线
16. INPUT:int start, int end
17. RETURN:int
18. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
19. int PlanPath(int start, int end)
20. {
21. int distance[34][34] =
22. {
23. *//0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33*
24. {  0, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *// 0*
25. { 50,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 25,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *// 1*
26. {MAX,MAX,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *// 2*
27. {MAX,MAX,MAX,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *// 3*
28. {MAX,MAX,MAX,MAX,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *// 4*
29. {MAX,MAX,MAX,MAX,MAX,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *// 5*
30. {MAX,MAX,MAX,MAX,MAX,MAX,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,125,MAX,MAX,MAX,MAX,MAX,MAX, 75,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *// 6*
31. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 75,MAX,MAX,MAX,MAX, 25,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *// 7*
32. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 10, 10,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *// 8*
33. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 25,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *// 9*
34. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,150,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,150,MAX,MAX,MAX},  *//10*
35. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,MAX,MAX,MAX,150,MAX,MAX},  *//11*
36. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 25, 75,MAX},  *//12*
37. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,MAX,MAX,MAX,150},  *//13*
38. {MAX, 25,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,  0,100,MAX,MAX,MAX,100,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *//14*
39. {MAX,MAX, 50,MAX,MAX,MAX,MAX, 75,MAX,MAX,MAX,MAX,MAX,MAX,100,  0,100,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *//15*
40. {MAX,MAX,MAX, 50,MAX,MAX,125,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,100,  0, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *//16*
41. {MAX,MAX,MAX,MAX,MAX, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,  0, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *//17*
42. {MAX,MAX,MAX,MAX, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX,250,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *//18*
43. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 10,MAX,MAX,MAX,MAX,MAX,100,MAX,MAX,MAX,MAX,  0,MAX,100,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *//19*
44. {MAX,MAX,MAX,MAX,MAX,MAX,MAX, 25, 10,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,  0,MAX,100,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *//20*
45. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 25,150,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,100,MAX,  0,100,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *//21*
46. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,100,100,  0,100,MAX,MAX,MAX,100,MAX,MAX,MAX,MAX,MAX,MAX},  *//22*
47. {MAX,MAX,MAX,MAX,MAX,MAX, 75,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,100,  0, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *//23*
48. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,  0, 50,MAX,MAX, 50,MAX,MAX,MAX,MAX,MAX},  *//24*
49. {MAX,MAX,MAX,MAX,MAX, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,  0, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *//25*
50. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,250,MAX,MAX,MAX,MAX,MAX,MAX, 50,  0,MAX,MAX,MAX,MAX,MAX,MAX,MAX},  *//26*
51. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,100,MAX,MAX,MAX,MAX,  0,100,MAX,MAX,MAX,MAX,MAX},  *//27*
52. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,MAX,MAX,100,  0,100,MAX,MAX,200,MAX},  *//28*
53. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 50,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,100,  0,MAX,MAX,MAX,MAX},  *//29*
54. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,150,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,  0,100,MAX,MAX},  *//30*
55. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,150, 25,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,100,  0,MAX,MAX},  *//31*
56. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX, 75,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,200,MAX,MAX,MAX,  0,100},  *//32*
57. {MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,150,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,MAX,100,  0},  *//33*
58. };
59. return Dijkstra(start, end, distance);
60. }
61. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
62. FUNCTION:Dijkstra
63. DESCRIBETION:实现迪杰斯特拉算法
64. INPUT:int start, int end, int distance[][34]
65. RETURN:int
66. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
67. int Dijkstra(int start, int end, int distance[][34])
68. {
69. int start\_distance[34], vertex[34] = {0};
70. int i, u, v;
71. FILE\* path\_text;
72. PathNode path\_node[34], \*temp = &path\_node[end];
73. for(i = 0; i < 34; i++)
74. {
75. start\_distance[i] = MAX;
76. path\_node[i].node = i;
77. path\_node[i].next = NULL;
78. }
79. start\_distance[start] = 0;
80. while(!vertex[end])
81. {
82. u = MinDistance(start\_distance, vertex);
83. vertex[u] = 1;
84. for(v = 0; v < 34; v++)
85. {
86. if((u!=5&&u!=8&&start!=5&&start!=8)||start==5||start==8)
87. {
88. if(!vertex[v] && distance[u][v]!=MAX && start\_distance[u] + distance[u][v] < start\_distance[v])
89. {
90. start\_distance[v] = start\_distance[u] + distance[u][v];
91. path\_node[v].next = &path\_node[u];
92. }
93. }
94. }
95. }
97. path\_text = fopen("..\\CODE\\text\\path.txt", "wt");
98. do
99. {
100. fprintf(path\_text, "%d %d ", (\*temp).node, distance[(\*temp).node][(\*((\*temp).next)).node]);
101. temp = (\*temp).next;
102. }while((\*temp).next!=NULL);
103. fprintf(path\_text, "%d %d ", start, 1000);
104. fclose(path\_text);
105. return start\_distance[end];
106. }
107. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
108. FUNCTION:MinDistance
109. DESCRIBETION:返回尚未包含在最短路径树中的最小距离的顶点的索引
110. INPUT:int\* distance, int\* vertex
111. RETURN:int
112. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
113. int MinDistance(int\* start\_distance, int\* vertex)
114. {
115. int min = MAX, min\_index, v;
116. for (v = 0; v < 34; v++)
117. if (vertex[v] == 0 && start\_distance[v] <= min)
118. min = start\_distance[v], min\_index = v;
119. return min\_index;
120. }
121. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
122. FUNCTION:DrawRoute
123. DESCRIBETION:绘制路线
124. INPUT:void
125. RETURN:void
126. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
127. void DrawRoute()
128. {
129. int place[35] = {0}, i = -1;
130. int x\_1, x\_2, y\_1, y\_2;
131. FILE\* path\_text;
132. path\_text = fopen("..\\CODE\\text\\path.txt", "rt");
133. do
134. {
135. i++;
136. fscanf(path\_text, "%d %d ", &place[i], &place[i+1]);
137. } while (place[i+1] != 1000);
138. fclose(path\_text);
139. i = 0;
140. setcolor(GREEN);
141. DrawStartEnd(place[i]);     *//end*
142. while(place[i+1] != 1000)
143. {
144. ReturnCoordinates(i, place, &x\_1, &y\_1);
145. ReturnCoordinates(i+1, place, &x\_2, &y\_2);
146. line(x\_1, y\_1, x\_2, y\_2);
147. i++;
148. }
149. DrawStartEnd(place[i]);     *//start*
150. return ;
151. }
152. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
153. FUNCTION:ReturnCoordinates
154. DESCRIBETION:返回路线的坐标
155. INPUT:int i, int\* place, int\* x, int\* y
156. RETURN:void
157. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
158. void ReturnCoordinates(int i, int\* place, int\* x, int\* y)
159. {
160. switch (place[i])
161. {
162. case 0:
163. \*x = 130, \*y = 55;
164. break;
165. case 1:
166. \*x = 130, \*y = 105;
167. break;
168. case 2:
169. \*x = 80, \*y = 230;
170. break;
171. case 3:
172. \*x = 80, \*y = 330;
173. break;
174. case 4:
175. \*x = 80, \*y = 430;
176. break;
177. case 5:
178. if(place[i+1] == 1000)
179. {
180. if(place[i-1] == 17)
181. \*x = 175, \*y = 380;
182. else if(place[i-1] == 25)
183. \*x = 335, \*y = 380;
184. }
185. else
186. {
187. if(place[i+1] == 17)
188. \*x = 175, \*y = 380;
189. else if(place[i+1] == 25)
190. \*x = 335, \*y = 380;
191. }
192. break;
193. case 6:
194. \*x = 255, \*y = 330;
195. break;
196. case 7:
197. \*x = 205, \*y = 230;
198. break;
199. case 8:
200. if(place[i+1] == 1000)
201. {
202. if(place[i-1] == 19)
203. \*x = 230, \*y = 140;
204. else if(place[i-1] == 20)
205. \*x = 230, \*y = 220;
206. }
207. else
208. {
209. if(place[i+1] == 19)
210. \*x = 230, \*y = 140;
211. else if(place[i+1] == 20)
212. \*x = 230, \*y = 220;
213. }
214. break;
215. case 9:
216. \*x = 330, \*y = 105;
217. break;
218. case 10:
219. \*x = 480, \*y = 130;
220. break;
221. case 11:
222. \*x = 480, \*y = 230;
223. break;
224. case 12:
225. \*x = 630, \*y = 255;
226. break;
227. case 13:
228. \*x = 480, \*y = 430;
229. break;
230. case 14:
231. \*x = 130, \*y = 130;
232. break;
233. case 15:
234. \*x = 130, \*y = 230;
235. break;
236. case 16:
237. \*x = 130, \*y = 330;
238. break;
239. case 17:
240. \*x = 130, \*y = 380;
241. break;
242. case 18:
243. \*x = 130, \*y = 430;
244. break;
245. case 19:
246. \*x = 230, \*y = 130;
247. break;
248. case 20:
249. \*x = 230, \*y = 230;
250. break;
251. case 21:
252. \*x = 330, \*y = 130;
253. break;
254. case 22:
255. \*x = 330, \*y = 230;
256. break;
257. case 23:
258. \*x = 330, \*y = 330;
259. break;
260. case 24:
261. \*x = 380, \*y = 330;
262. break;
263. case 25:
264. \*x = 380, \*y = 380;
265. break;
266. case 26:
267. \*x = 380, \*y = 430;
268. break;
269. case 27:
270. \*x = 430, \*y = 230;
271. break;
272. case 28:
273. \*x = 430, \*y = 330;
274. break;
275. case 29:
276. \*x = 430, \*y = 430;
277. break;
278. case 30:
279. \*x = 630, \*y = 130;
280. break;
281. case 31:
282. \*x = 630, \*y = 230;
283. break;
284. case 32:
285. \*x = 630, \*y = 330;
286. break;
287. case 33:
288. \*x = 630, \*y = 430;
289. break;
290. case 34:
291. \*x = 630, \*y = 530;
292. break;
293. }
294. return ;
295. }
296. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
297. FUNCTION:DrawStartEnd
298. DESCRIBETION:绘制起点或终点处的路线
299. INPUT:int num
300. RETURN:void
301. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
302. void DrawStartEnd(int num)
303. {
304. setcolor(GREEN);
305. switch (num)
306. {
307. case 0:
308. line(130, 55, 120, 55);
309. break;
310. case 1:
311. line(130, 105, 120, 105);
312. break;
313. case 2:
314. break;
315. case 3:
316. line(80, 330, 80, 340);
317. break;
318. case 4:
319. line(80, 430, 80, 440);
320. break;
321. case 5:
322. break;
323. case 6:
324. line(255, 330, 255, 335);
325. break;
326. case 7:
327. line(205, 230, 205, 240);
328. break;
329. case 8:
330. break;
331. case 9:
332. line(330, 105, 340, 105);
333. break;
334. case 10:
335. line(480, 130, 480, 140);
336. break;
337. case 11:
338. line(480, 230, 480, 225);
339. break;
340. case 12:
341. line(630, 255, 620, 255);
342. break;
343. case 13:
344. line(480, 430, 480, 440);
345. break;
346. }
347. }

**hz.c**

1. #include <graphics.h>
2. #include<conio.h>
3. #include<stdio.h>
4. #include<stdlib.h>
5. #include<string.h>
6. #include"hz.h"
7. void puthz(int x, int y,char \*s,int flag,int part,int color)
8. {
9. FILE \*hzk\_p=NULL;                                       *//定义汉字库文件指针*
10. unsigned char quma,weima;                 *//定义汉字的区码和位码*
11. unsigned long offset;                          *//定义汉字在字库中的偏移量*
12. unsigned char mask[] = {0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01};  *//功能数组，用于显示汉字点阵中的亮点*
13. int i,j,pos;
14. switch(flag)    *//不同的flag对应不同的汉字库，实现了汉字的大小可根据需要改变*
15. {
16. case 16 :
17. {
18. char mat[32];   *//16\*16的汉字需要32个字节的数组来存储*
19. int y0=y;
20. int x0=x;
21. hzk\_p = fopen("HZK\\HZ16","rb");            *//使用相对路径*
22. if(hzk\_p==NULL)
23. {
24. settextjustify(LEFT\_TEXT,TOP\_TEXT);          *//左部对齐，顶部对齐*
25. settextstyle(GOTHIC\_FONT,HORIZ\_DIR,1);     *//黑体笔划输出，水平输出，24\*24点阵*
26. outtextxy(10,10,"Can't open hzk16 file!Press any key to quit...");
27. getch();
28. exit(1);
29. }
30. while(\*s!=NULL)
31. {
32. while (x<640-flag && (\*s!=NULL))
33. {
34. y=y0;
35. quma=s[0]-0xa0;                      *//求出区码*
36. weima=s[1]-0xa0;                     *//求出位码*
37. offset=(94\*(quma-1)+(weima-1))\*32L;   *//求出要显示的汉字在字库文件中的偏移*
38. fseek(hzk\_p,offset,SEEK\_SET);         *//重定位文件指针*
39. fread (mat,32,1,hzk\_p);            *//读出该汉字的具体点阵数据,1为要读入的项数*
40. for(i=0;i<16;i++)
41. {
42. pos=2\*i;       *//16\*16矩阵中有每一行有两外字节*
43. for(j=0;j<16;j++)    *//一行一行地扫描，将位上为了1的点显示出来*
44. {
45. if((mask[j%8]&mat[pos+j/8])!=NULL)   *//j%8只能在0—8之间循环，j/8在0，1之间循环*
46. {
47. putpixel(x+j,y,color);
48. }
49. }
50. y++;
51. }
52. */\*====================================================*
53. 以上是一个汉字显示完
54. ====================================================\*/
55. x+=part;        *//给x 一个偏移量part*
56. s+=2;           *//汉字里存放的是内码，2个字节，所以要加2*
57. }
58. x=x0;y0+=flag+10; *//一行汉字显示完后,重新从左侧开始输出汉字，给y一个偏移量*
59. }
60. break;
61. }
62. case 24 :
63. {
64. char mat[72];   *//24\*24矩阵要72个字节来存储*
65. int y0=y;
66. int x0=x;
67. hzk\_p = fopen("HZK\\Hzk24k","rb");
68. if (hzk\_p==NULL)
69. {
70. settextjustify(LEFT\_TEXT,TOP\_TEXT);          *//左部对齐，顶部对齐*
71. settextstyle(GOTHIC\_FONT,HORIZ\_DIR,3);     *//黑体笔划输出，水平输出，24\*24点阵*
72. outtextxy(10,10,"Can't open hzk24 file!Press any key to quit...");
73. getch();
74. exit(1);
75. }
76. while(\*s!=NULL)
77. {
78. while(x<640-flag && (\*s!=NULL))
79. {
80. y=y0;
81. quma=s[0]-0xa0;                      *//求出区码*
82. weima=s[1]-0xa0;                     *//求出位码*
83. offset=(94\*(quma-1)+(weima-1))\*72L;
84. fseek(hzk\_p,offset,SEEK\_SET);
85. fread (mat,72,1,hzk\_p);
86. for (i=0;i<24;i++)
87. {
88. pos=3\*i;   *//矩阵中每一行有三个字节*
89. for (j=0;j<24;j++)   *// 每一行有24位*
90. {
91. if ((mask[j%8]&mat[pos+j/8])!=NULL)
92. putpixel(x+j,y,color);
93. }
94. y++;
95. }
96. x+=part;
97. s+=2;
98. }
99. x=x0;y0+=flag+10;
100. }
101. break;
102. }
103. case 32 :
104. {
105. char mat[128];   *//32\*32的汉字需要128个字节的数组来存储*
106. int y0=y;
107. int x0=x;
108. hzk\_p = fopen("HZK\\HZK32S","rb");
109. if(hzk\_p==NULL)
110. {
111. settextjustify(LEFT\_TEXT,TOP\_TEXT);          *//左部对齐，顶部对齐*
112. settextstyle(GOTHIC\_FONT,HORIZ\_DIR,3);     *//黑体笔划输出，水平输出，24\*24点阵*
113. outtextxy(10,10,"Can't open hzk32 file!Press any key to quit...");
114. getch();
115. exit(1);
116. }
117. while(\*s!=NULL)
118. {
119. while (x<640-flag && (\*s!=NULL))
120. {
121. y=y0;
122. quma=s[0]-0xa0;                      *//求出区码*
123. weima=s[1]-0xa0;                     *//求出位码*
124. offset=(94\*(quma-1)+(weima-1))\*128L;
125. fseek(hzk\_p,offset,SEEK\_SET);
126. fread (mat,128,1,hzk\_p);
127. for(i=0;i<32;i++)
128. {
129. pos=4\*i;       *//32\*32矩阵中有每一行有两外字节*
130. for(j=0;j<32;j++)
131. {
132. if((mask[j%8]&mat[pos+j/8])!=NULL)
133. {
134. putpixel(x+j,y,color);
135. }
136. }
137. y++;
138. }
139. *//以上是一个汉字显示完*
140. x+=part;    *//给x 一个偏移量part*
141. s+=2;          *//汉字里存放的是内码，2个字节，所以要加2*
142. }
143. x=x0;y0+=flag+10;   *//一行汉字显示完后，给y一个偏移量*
144. }
145. break;
146. }
147. case 48:
148. {
149. char mat[288];   *//48\*48的汉字需要288个字节的数组来存储*
150. int y0=y;
151. int x0=x;
152. hzk\_p = fopen("HZK\\Hzk48k","rb");
153. if(hzk\_p==NULL)
154. {
155. settextjustify(LEFT\_TEXT,TOP\_TEXT);          *//左部对齐，顶部对齐*
156. settextstyle(GOTHIC\_FONT,HORIZ\_DIR,3);     *//黑体笔划输出，水平输出，24\*24点阵*
157. outtextxy(10,10,"Can't open hzk48 file!Press any key to quit...");
158. getch();
159. exit(1);
160. }
161. while(\*s!=NULL)
162. {
163. while (x<640-flag && (\*s!=NULL))
164. {
165. y=y0;
166. quma=s[0]-0xa0;                      *//求出区码*
167. weima=s[1]-0xa0;                     *//求出位码*
168. offset=(94\*(quma-1)+(weima-1))\*288L;   *//求出要显示的汉字在字库文件中的偏移*
169. fseek(hzk\_p,offset,SEEK\_SET);         *//重定位文件指针*
170. fread (mat,288,1,hzk\_p);            *//读出该汉字的具体点阵数据,1为要读入的项数*
171. for(i=0;i<48;i++)
172. {
173. pos=6\*i;
174. for(j=0;j<48;j++)    *//一行一行地扫描，将位上为了1的点显示出来*
175. {
176. if((mask[j%8]&mat[pos+j/8])!=NULL)   *//j%8只能在0—8之间循环，j/8在0，1之间循环*
177. {
178. putpixel(x+j,y,color);
179. }
180. }
181. y++;
182. }
183. *//以上是一个汉字显示完*
184. x+=part;    *//给x 一个偏移量part*
185. s+=2;          *//汉字里存放的是内码，2个字节，所以要加2*
186. }
187. x=x0;y0+=flag+10;   *//一行汉字显示完后，给y一个偏移量*
188. }
189. break;
190. }
191. default:
192. break;
193. }
194. fclose(hzk\_p);
195. }

**mouse.c**

1. #include<conio.h>
2. #include<graphics.h>
3. #include<dos.h>
4. #include<stdio.h>
5. #include<stdlib.h>
6. #include "mouse.h"
7. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
8. MOUSE.c
9. UPDATER: dengshuumin
10. FUNCTION: mouse action
11. ABSTRACT:
12. A.mread
13. B.newmouse
14. VERSION: 3.0
15. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
16. int MouseX;
17. int MouseY;
18. int MouseS;
19. int press;
20. void \*buffer;
21. union REGS regs;
22. int flag=0;
23. void mouseinit()*//初始化*
24. {
25. int retcode;
26. int xmin,xmax,ymin,ymax,x\_max=625,y\_max=480;
27. int size;
28. xmin=2;
29. xmax=x\_max-1;
30. ymin=8;
31. ymax=y\_max-2;
32. regs.x.ax=0;
33. int86(51,&regs,&regs);
34. retcode=regs.x.ax;
35. if(retcode==0)
36. {
37. printf("Mouse or Mouse Driver Obsent,Please Install!");
38. delay(5000);
39. }
40. else
41. {
42. regs.x.ax=7;
43. regs.x.cx=xmin;
44. regs.x.dx=xmax;
45. int86(51,&regs,&regs);
46. regs.x.ax=8;
47. regs.x.cx=ymin;
48. regs.x.dx=ymax;
49. int86(51,&regs,&regs);
50. }
51. MouseS = 0;
52. MouseX=320,MouseY=240;
53. save\_bk\_mou(320,240);
54. mouse(MouseX,MouseY);
55. flag=1;
56. }
57. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
58. FUNCTION: mouse
59. DESCRIPTION: 画不同形态的鼠标
60. INPUT: x,y
61. RETURN: 无
62. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
63. void mouse(int x,int y)
64. {
66. switch(MouseS)
67. {
68. case 1:                                  *//手势鼠标*
69. {
70. setcolor(WHITE);
71. setlinestyle(0,0,1);
72. line(x-1,y+9,x-1,y+8);
73. line(x,y+7,x,y+11);
74. line(x+1,y+6,x+1,y+13);
75. line(x+2,y+8,x+2,y+14);
76. line(x+3,y-1,x+3,y+15);
77. arc(x+4,y-1,0,180,1);
78. line(x+4,y-2,x+4,y+15);
79. line(x+5,y-1,x+5,y+16);
80. arc(x+6,y+3,0,180,1);
81. line(x+6,y+2,x+6,y+16);
82. line(x+7,y+3,x+7,y+17);
83. arc(x+8,y+5,0,180,1);
84. line(x+8,y+4,x+8,y+17);
85. line(x+9,y+5,x+9,y+16);
86. arc(x+10,y+7,0,180,1);
87. line(x+10,y+6,x+10,y+16);
88. line(x+11,y+7,x+11,y+13);
89. setcolor(DARKGRAY);
90. line(x-1,y+9,x-1,y+8);
91. line(x-1,y+8,x+1,y+6);
92. line(x+1,y+6,x+3,y+10);
93. line(x+3,y+10,x+3,y-1);
94. arc(x+4,y-1,0,180,1);
95. line(x+5,y-1,x+5,y+5);
96. arc(x+6,y+3,0,180,1);
97. line(x+7,y+3,x+7,y+7);
98. arc(x+8,y+5,0,180,1);
99. line(x+9,y+5,x+9,y+9);
100. arc(x+10,y+7,0,180,1);
101. line(x+11,y+7,x+11,y+13);
102. arc(x+7,y+13,-90,0,4);
103. line(x+7,y+17,x+3,y+15);
104. line(x+3,y+15,x+1,y+13);
105. line(x+1,y+13,x-1,y+9);
106. }
107. break;
108. case 2:                        *//光标*
109. {
110. setcolor(DARKGRAY);
111. setlinestyle(0,0,1);
112. line(x+1,y-1,x+9,y-1);
113. line(x+1,y+15,x+9,y+15);
114. line(x+5,y-1,x+5,y+15);
115. }
116. break;
117. case 3:                        *//十字*
118. {
119. setcolor(WHITE);
120. setlinestyle(0,0,1);
121. line(x-1,y+7,x+11,y+7);
122. line(x+5,y-1,x+5,y+15);
123. }
124. break;
125. default:              *//默认鼠标*
126. {
127. setlinestyle(0,0,1);
128. setcolor(WHITE);
129. line(x,y,x,y+13);
130. line(x+1,y+1,x+1,y+12);
131. line(x+2,y+2,x+2,y+11);
132. line(x+3,y+3,x+3,y+10);
133. line(x+4,y+4,x+4,y+12);
134. line(x+5,y+5,x+5,y+9);
135. line(x+5,y+11,x+5,y+14);
136. line(x+6,y+6,x+6,y+9);
137. line(x+6,y+13,x+6,y+15);
138. line(x+7,y+7,x+7,y+9);
139. line(x+8,y+8,x+8,y+9);
140. line(x+9,y+9,x+9,y+9);
141. setcolor(DARKGRAY);
142. line(x-1,y-1,x-1,y+14);
143. line(x-1,y+14,x+3,y+11);
144. line(x+3,y+11,x+3,y+12);
145. line(x+3,y+12,x+4,y+13);
146. line(x+4,y+13,x+4,y+14);
147. line(x+4,y+14,x+7,y+17);
148. line(x+7,y+17,x+7,y+13);
149. line(x+7,y+13,x+6,y+12);
150. line(x+6,y+12,x+6,y+11);
151. line(x+6,y+11,x+5,y+10);
152. line(x+5,y+10,x+11,y+10);
153. line(x+11,y+10,x-1,y-2);
154. }
155. break;
156. }
157. }
158. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
159. FUNCTION: mread
160. DESCRIPTION: 获取新的寄存器信息
161. INPUT: nx,ny,nbuttons
162. RETURN: 无
163. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
164. void mread(int \*nx,int \*ny,int \*nbuttons)
165. {
166. regs.x.ax=3;
167. int86(51,&regs,&regs);
168. \*nx = regs.x.cx;
169. \*ny = regs.x.dx;
170. \*nbuttons = regs.x.bx;
171. }
172. */\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**
173. FUNCTION: newmouse
174. DESCRIPTION: 鼠标状态发生变化则更新鼠标
175. INPUT: nx,ny,nbuttons
176. RETURN: 无
177. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
178. void newmouse(int \*nx,int \*ny,int \*nbuttons)
179. {
180. int xn,yn,buttonsn;
181. int x0=\*nx,y0=\*ny,buttons0=\*nbuttons;
182. mread(&xn,&yn,&buttonsn);
183. \*nx = xn;
184. \*ny = yn;
185. \*nbuttons = buttonsn;
186. if(buttons0 == \*nbuttons)
187. \*nbuttons = 0;    *//使得能连续按键*
188. if(xn == x0 && yn == y0 && buttonsn == buttons0)
189. return;            *//鼠标状态不变则直接返回S*
190. clrmous(x0,y0);        *//说明鼠标状态发生了改变*
191. save\_bk\_mou(\*nx,\*ny);
192. drawmous(\*nx,\*ny);
193. }
194. void save\_bk\_mou(int nx,int ny)*//存鼠标背景*
195. {
196. int size;
197. size=imagesize(nx-1,ny-2,nx+11,ny+17);
198. buffer=malloc(size);
199. if(buffer!=NULL)
200. getimage(nx-1,ny-2,nx+11,ny+17,buffer);
201. else
202. printf("Error");
203. }
204. void clrmous(int nx,int ny)*//清除鼠标*
205. {
206. if(flag==1)
207. {
208. setwritemode(XOR\_PUT);
209. mouse(nx,ny);
210. putimage(nx-1,ny-2,buffer,COPY\_PUT);
211. free(buffer);
212. flag=0;
213. setwritemode(COPY\_PUT);
214. }
215. }
216. void drawmous(int nx,int ny)
217. {
218. if(flag==0)
219. {
220. setwritemode(COPY\_PUT);
221. mouse(nx,ny);
222. flag=1;
223. }
224. }
225. *//如果在框中点击，则返回1；在框中未点击，则返回2；不在框中则返回0*
226. int mouse\_press(int x1, int y1, int x2, int y2)
227. {
228. *//在框中点击，则返回1*
229. if(MouseX > x1
230. &&MouseX < x2
231. &&MouseY > y1
232. &&MouseY < y2
233. &&press == 1)
234. {
235. return 1;
236. }
238. *//在框中未点击，则返回2*
239. else if(MouseX > x1
240. &&MouseX < x2
241. &&MouseY > y1
242. &&MouseY < y2
243. &&press == 0)
244. {
245. return 2;
246. }
248. *//在框中点击右键，则返回3*
249. else if(MouseX > x1
250. &&MouseX < x2
251. &&MouseY > y1
252. &&MouseY < y2
253. &&press == 2)
254. {
255. return 3;
256. }
258. else
259. {
260. return 0;
261. }
262. }

**八、分工与时间安排**

## 时间安排

第一周：需求分析，构思程序框架，熟悉画图操作及汉字库、鼠标库的使用，编写主函数。

第二周：绘制主页，设计菜单页框架，编写车窗页、雨刷页、天窗页、胎压页、灯光页。

第三周：完善菜单页，编写驾驶页、电池页、车门页。

第四周：编写空调页，菜单页定稿，将两人编写的菜单页的所有子页面进行整合。

第五周：设计地图和导航页面的草图，学习迪杰斯特拉算法。

第六周：编写导航页、路径规划算法。

第七周：完成地图的绘制，构思显示页面框架。

第八周：编写显示页面，绘制汽车运行时的动画，设计运行时遇到障碍物的报警功能。

第九周：调试程序，迭代出稳定版本，准备验收。

## 分工安排

Y：负责drive,battery,aircondition,light,door,display模块的编写以及中期、末期报告的撰写工作等。总代码行数2208，有效代码行数1724。

X：负责main,menu,page,window,wiper,sunroof,tire,init,map,navigation,plan模块的编写工作，中期、末期报告的撰写工作以及对整个工程架构的维护工作等。总代码行数3293，有效代码行数2526。