华中科技大学C语言课程设计报告

**极品飞车**

自动化校际交流1701

U201712087许可晗、U201715853曾文正

指导老师：

周纯杰、何顶新、周凯波、汪国有、彭刚等

上交日期：2018.11.17

目录

[1. 引言 3](#_Toc530221818)

[1.1 编写目的 3](#_Toc530221819)

[1.2 项目风险 3](#_Toc530221820)

[1.3 预期读者和阅读建议 3](#_Toc530221821)

[1.4 产品范围 3](#_Toc530221822)

[1.5 参考文献 3](#_Toc530221823)

[2. 综合描述 3](#_Toc530221824)

[2.1 产品的状况 3](#_Toc530221825)

[2.2 目标 4](#_Toc530221826)

[2.3 产品的功能 4](#_Toc530221827)

[2.4 用户类和特性 4](#_Toc530221828)

[2.5 运行环境 4](#_Toc530221829)

[硬件环境 4](#_Toc530221830)

[软件环境 4](#_Toc530221831)

[2.6 假设和约束(依赖) 5](#_Toc530221832)

[3. 外部接口需求 6](#_Toc530221833)

[3.1 用户界面 6](#_Toc530221834)

[3.2 用户操作说明书 20](#_Toc530221835)

[4. 系统功能需求 21](#_Toc530221836)

[4.1 功能需求描述 21](#_Toc530221837)

[4.2 系统设计 22](#_Toc530221838)

[5. 源代码 37](#_Toc530221839)

[6. 总结 216](#_Toc530221840)

# 

# 引言

## 编写目的

编写本程序的目的在于设计一款娱乐性质的赛车游戏，给人们的业余时间更多的选择，丰富人们的生活，体会竞速的乐趣。本程序包括程序系统的基本处理流程、程序系统的组织结构、模块划分和功能分配，为程序的详细设计提供基础。本程序所描述的需求信息将作为该项目最终验收的标准和依据。

本小组成员对于极品飞车游戏有着一定的兴趣，也为提高自身的软件编程能力，故选择了极品飞车这个课题作为C语言课程设计的设计项目。应用C语言完成对极品飞车基本功能的实现。

## 项目风险

本软件仅是为了我们更好地理解c语言而模拟的赛车游戏，无较大风险。

## 预期读者和阅读建议

该程序预期的读者为验收老师以及程序设计员。

## 产品范围

说明该软件产品及其开发目的的简短描述，包括利益和目标。把软件产品开发与企业目标，或者业务策略相联系。

描述产品范围时需注意，可以参考项目视图和范围文档，但是不能将其内容复制到这里。

## 参考文献

（1）王士元. C高级实用程序设计. 北京: 清华大学出版社. 1996

（2）周纯杰，刘正林等. 标准C语言程序及应用. 武汉: 华中科技大学出版社. 2005

（3）潭浩强. C语言程序设计. 北京: 清华大学出版社, 2000

# 综合描述

## 产品的状况

《极品飞车》（Need for Speed）是由美国艺电游戏公司出品研发的一款赛车

类游戏，于1994 年发行，为《极品飞车》的初代产品。，其续作有从1995 年至

2012 年已经出品了13 代共17 个作品。《极品赛车》的出现彻底改变了过去粗糙

的赛车游戏的格局，其他赛车游戏开发商为了与其抗衡，便不断提高游戏品质增

加游戏体验，也正是在这一时期， 赛车游戏有了长足的进步，毫不夸张的说，

《极品飞车》是赛车游戏史上的一座里程碑。《极品飞车》起初是在电脑端运行，

后来随着该游戏不断发展壮大，开发商把它移植到各种游戏平台，现在PS3、X360、

GBA、 PSP、 3DS, 手机各个游戏终端都能见到《极品飞车》，其火热程度可见一

斑。 游戏试图在爽快的赛车比赛和车辆的拟真上找到平衡点。这款游戏在世界

车迷心目中占有重要地位。

## 目标

本软件的开发意在让本小组成员以及其他用户更好地了解极品飞车游戏，提高本小组成员自身的软件编程能力。本软件的应用目标为极品飞车游戏，实现赛车游戏的基本功能。

本软件产品为独立软件，全部内容自含。

## 产品的功能

实现用户注册，登录，查询历史成绩，选赛道，选车，选择自动，手动挡，前进，转弯，超车，碰撞等功能。

## 用户类和特性

本软件的用户无需有太高的文化素质，操作人员只需掌握基本的计算机操作技能，以及说明书阅读能力，维护人员需要较好地掌握C语言程序编程与调试能力。本软件预期的使用频率不高，仅作为赛车游戏的模仿，程序设计员偶尔使用、调试，老师验收时使用。

## 运行环境

### 硬件环境

普通的计算机

### 软件环境

支持的操作系统包括Windows98/XP/NT/2000中文版及以上版本

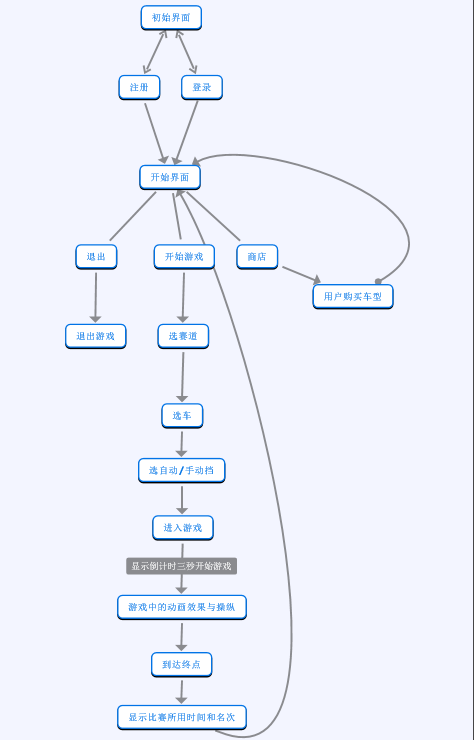
使用borlandc编译器

## 假设和约束(依赖)

本软件只是模仿赛车游戏，并不应用于市场，无开发经费，故也无经费限制。

# 外部接口需求

## 用户界面

1.总体流程







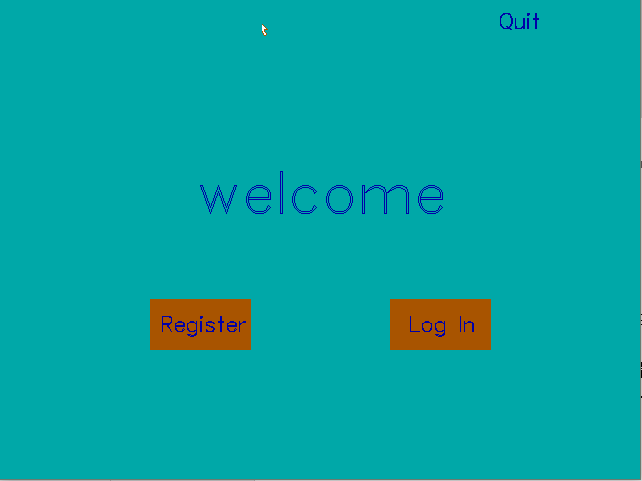






2.初始界面

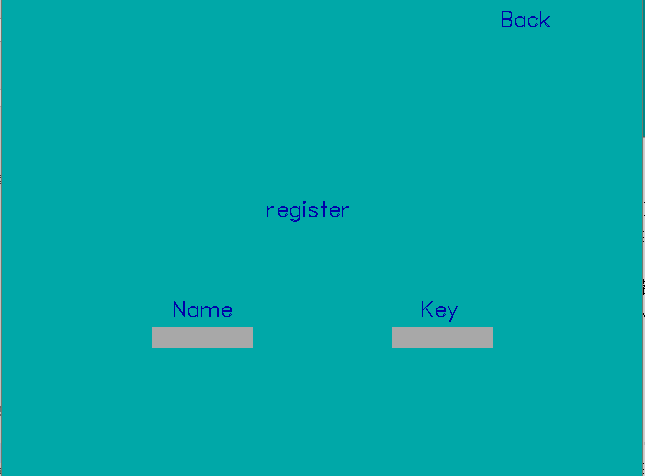
初始界面具有三个按键：退出（quit）、注册（register）、登录（log in）

在此界面中用户可以操作鼠标来实现特定功能。点击退出则退出游戏，点击注册则进入注册界面，点击登录则进入登录界面。

3.注册界面

注册界面具有三个模块：返回（back）、用户名（name）、密码（key）

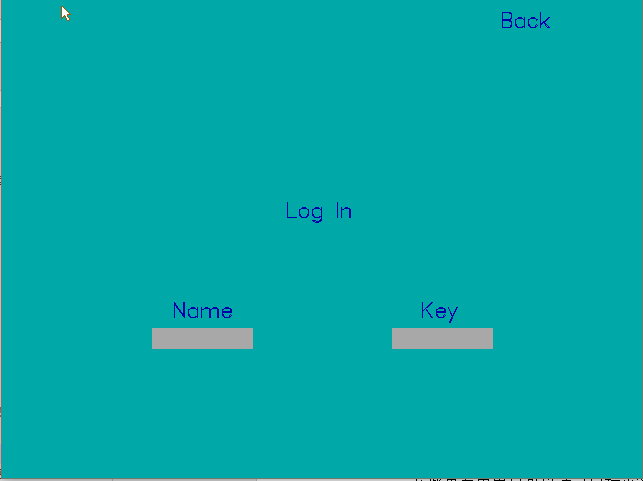
在此界面中用户可以通过鼠标和键盘来实现特定功能。点击返回则回到初始界面，点击“name”下方的方框则开始键入用户名。点击回车后，若用户名已被注册过，则程序会提示玩家并可重新输入用户名，若用户名没被注册过，则界面中会显示success，并提示玩家在“key”下方的方框中键入密码。玩家键入密码后点击回车后程序进入主菜单界面。



4.登录界面

登录界面具有三个模块：返回（back）、用户名（name）、密码（key）

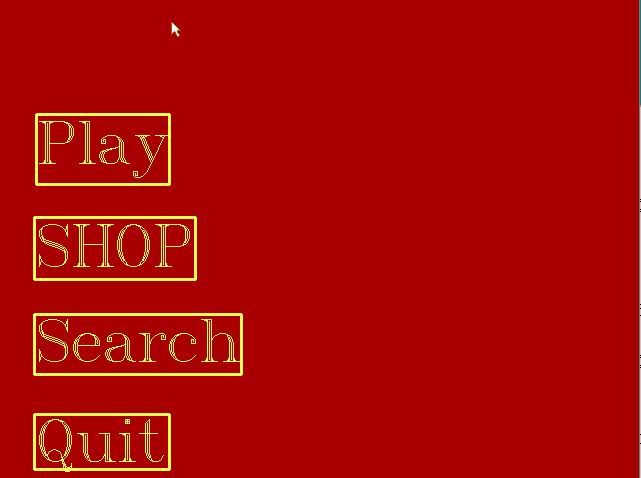
在此界面中用户可以通过鼠标和键盘来实现特定功能。点击返回则回到初始界面，点击“name”下方的方框则开始键入用户名。点击回车后，进入“key”下方的方框输入密码，点击回车后，若输入的用户名和密码与之前注册的不匹配则系统会提示“wrong key”，若输入的用户名不存在则界面中会显示“name doesn’t exist”。若输入的用户名和密码匹配则会进入主菜单界面。



5.主菜单界面

主菜单界面具有三个按键：开始（Play）、商店（shop）、退出（quit）

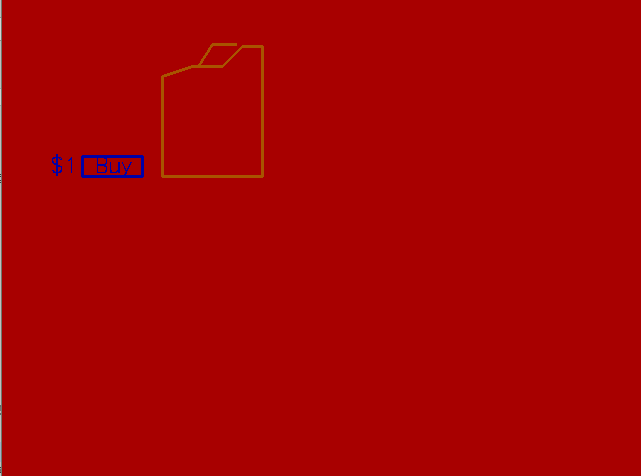
在此界面中用户可以通过鼠标来实现特定功能。点击开始则进入选择赛道界面，点击商店则进入商店界面，点击退出则游戏退出。



6.商店界面

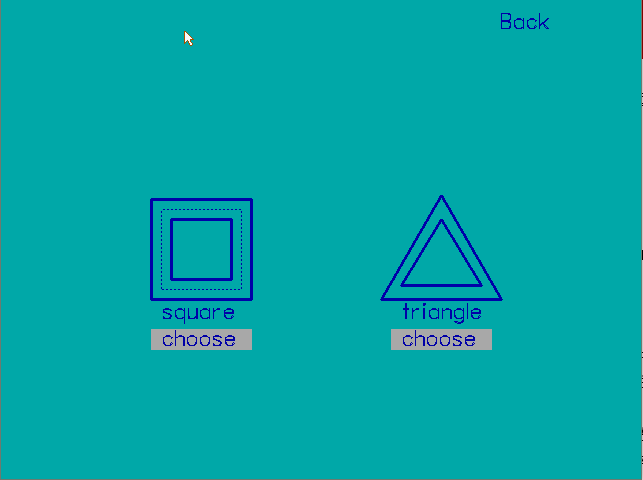
商店界面具有三个模块：显示金币、返回按键（back）、车辆和其相关信息（价格、车的样子、车的名称）

在此界面中用户可以通过鼠标来实现特定功能。用户可根据自己现有的金币数量购买相应车辆，购买后金币会自动减少，购买的车辆能在以后的比赛中使用，注：只能购买价格小于等于现有金币数的车型。用户点击购买即可完成购买，若不满足上述购买条件，则游戏会提示用户此车辆不可购买。若用户点击返回按键back，则游戏将返回到主菜单界面。



7.选择赛道界面

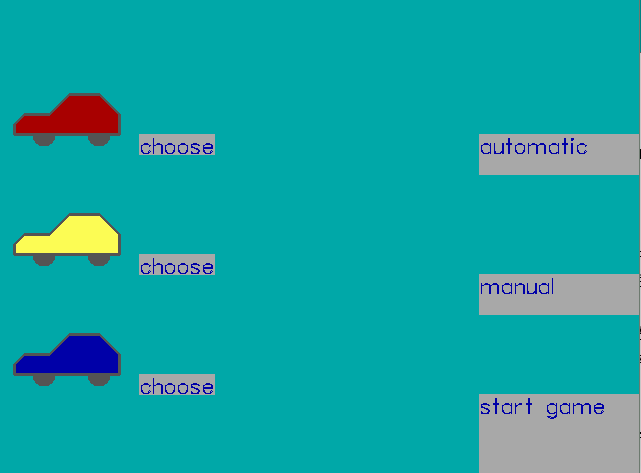
选择赛道界面具有两个模块：返回（back）、赛道即其相关信息（赛道的样子、赛道的名称）



8.选择车型、挡型界面

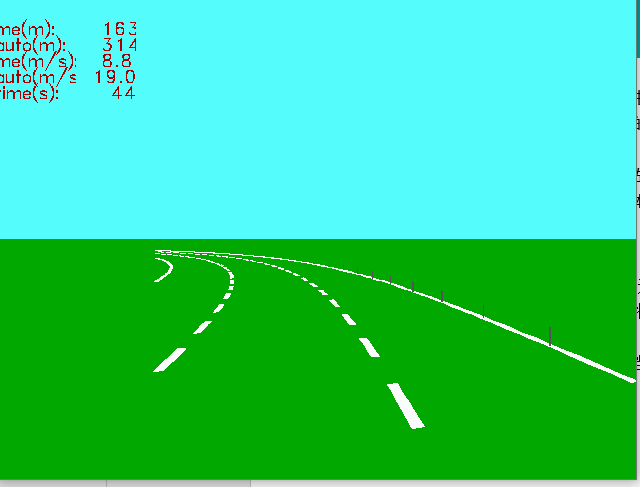
此界面有三个模块：返回（back）、可供选择的车型及其相关信息、选择挡型。

在此界面中用户可以通过鼠标来实现特定功能。用户可以选择自己已经拥有的车型来进行比赛，点击相应区域即可完成对车辆的选择。点击相应区域即可完成对挡型的选择。点击返回则回到选择赛道界面。



9.赛车行驶界面

该界面是程序中赛车行驶的界面，程序中的大部分功能在本界面下实现。它本身根据用途的不同分为3个板块。第一个是左上部分的赛车信息显示板块，用于显示敌方赛车的当前速度和行驶路程以及我方赛车的的当前速度，行驶路程和行驶时间；第二个是左下部分的档位显示板块。若是选择的手动挡模式，则在左下部分显示当前的档位。第三个是右半部分的赛车行驶动画板块，用于显示赛车比赛的具体情况。



游戏开始前选择：

1）跑道，有三角形和正方形两种选择，通过玩家的选择在游戏中生成相应的赛道。

2）选择行驶车型，不同车辆有不同的形状和不同的性能。有车辆不在一开始解锁，而是需要等玩家收集一定金币后购买解锁

3）选择交互行驶车是自动档还是手动挡。

## 用户操作说明书

1. 若用户未注册过，则在开始界面点击注册键。点击后将进入注册界面，点击下方输入框进行用户名、密码的输入。在输入用户名的时候，若改名字已经被注册过，则程序会提醒用户该用户名已经被注册过，并重新让用户键入用户名，直到成功为止。若用户想退出，则可以点击界面右上方返回键返回到初始界面，再在初始界面的右上角点击退出键方可退出游戏。
2. 若用户已经注册过，则可以在初始界面中选择登录而进入登录界面。点击进入登录界面，点击下方输入框进行用户名的输入，点击回车后进行密码的输入。点击回车后，若用户名和密码匹配，则程序会进入到主菜单界面。若用户名存在但密码与之不匹配或用户名不存在，则程序会显示用户名或密码错误，并可让用户重新输入用户名和密码。若用户想退出游戏，同样可以点击登录界面右上方的返回键返回初始界面，然后再点退出键退出。
3. 进入主菜单后。主菜有开始游戏、商店和退出功能。若用户点击开始游戏键，则会进入选择赛道界面，其中用户可以选择自己比赛用的赛道。选择完赛道后用户可点击选择车型，挡型，进入选择车型、并要选择自动挡还是手动挡的界面。选择赛道、车型等过程中的界面均可点击返回键来返回上一级界面。选择完毕后，游戏将会开始。
4. 赛道

画面上显示的只是局部的赛道，当用户选择好赛道并初始化赛道后，需要根据小车的位置刷新赛道在屏幕上能显示的部分。当快到终点时，要能显示出终点线，表示即将来到终点。

1. 赛车

与用户竞速的另一辆小车以默认的轨迹匀速行驶，当用户的车与另一辆车相隔一定的距离之内时可以把辆车同时显示，否则只显示用户的车。超车时，用户的车可以从另一辆车旁边超车。超过一段距离后不显示另一辆小车。按上键后小车启动，开始向前运动。按下键可以刹车。左右键可以使小车转弯。若选择的是手动挡，小车不能直接达到最大速度，需要用户手动换挡来提升速度上限。小车在运行过程中随时监测与赛道边缘或者另一小车的碰撞，若与边缘碰撞，则小车获得原速度镜面反射的速度，若与小车碰撞，则两车速度交换，另一辆小车会自动调整回原来的状态，而用户的小车需要手动调整。

1. 商店

用户可在商店中购买自己喜爱的车型，根据自己拥有的金币量，若金币量不足则会提示购买失败，点击购买键即可购买相关车型。

# 系统功能需求

## 功能需求描述

1. 注册

玩家需在游戏内建立自己的用户。通过游戏中的注册功能建立自己的账号。

注：同一用户名不能被注册两次。

2.登录

在玩家已经建立过自己的账号后，下次进入游戏时，可登录自己之前注册过的账号继续游戏。进入游戏前必须登录或注册，登录时要求玩家输入的用户名、密码和之前注册的用户名、密码匹配，否则视为无效。

3.信息保存

每个账号中的信息都会永久储存，当玩家下一次进入游戏时，原信息仍存在并可读取。

4.信息更新

玩家在游戏中进行的一系列和自身账号信息变更有关的操作都会实时完成，更改后的账号相关信息会替代原信息。

5.商城

在商城中，玩家可通过自己拥有的金币购买不同的赛车，赛车只有购买后才能使用。

6.金币

在玩家的账户中有金币，金币可用来在商城中购买物品。金币可通过比赛获得。

7.界面可操作性

玩家可通过界面的提示选择自己想实现的功能，界面的设计需符合基本需求。

8.赛道选择

玩家可选择不同的赛道进行游戏

9.赛车选择

玩家可选择不同的赛车进行游戏，前提是该赛车已被解锁（玩家已经在商城购买过此赛车）。

10.手动、自动挡选择

玩家可选择手动挡车型或自动挡车型来进行游戏，手动挡车型需要手动换挡，自动挡车型则不需要。

11.键盘控制：

一直按左键左转，一直按右键右转，松开左，右键则停止左右的转向。一直按上键油门踩到底，点击空格键换档（只有两档，循环变化）。松开上键赛车减速直到停止。

12.游戏规则：

键盘控制小车行驶一圈结束，最先到达终点为第一名。

13.碰撞处理：

与道路边线碰撞，新位置在碰撞点，按光线反射改变方向。两车相撞，交换速度。

14.信息显示：

两车行驶路程，速度，比赛经历时间，键盘控制小车的档位，跑完一圈各自所用时间。

## 系统设计

1.主函数模块

该模块为整个程序的入口

int main()

{

int driver = VGA;

int mode = VGAHI;

user \*head=NULL;

int mark=1;

initgraph(&driver, &mode, "C:\\BORLANDC\\bgi");

if ((head = (user \*)malloc(sizeof(user))) == NULL)

{

printf("\nmeineicun");

return 0;

}

CreateUserList(head);

while(1)

{

if(mark==1)

mark=jiemian();

if(mark==2)

mark=loadzhuce();

if(mark==3)

mark=denglu();

if(mark==4)

mark=zhucaidan();

if(mark==5)

mark=shop();

if(mark==6)

{

mark=saidao();

mark=cartype();

Game\_Init();

Game\_Main();

delay(dly);

mark=4;

}

}

if(mark==0)

exit(1);

}

return 1;

}

主函数是整个程序的入口，反映出了各个模块之间的关系和程序整体的运行流程。

2．数据类型模块

该模块用来定义在程序中会用到的自定义数据类型

文件：jiegouti.h

内部的数据类型：

用户信息的结构体：

typedef struct account

{

char name[11];

char key[11];

char coin[4];

char car[10];

struct account \*next;

}user;

name指用户的用户名

key指用户的密码

coin指用户现有金币数量

car指用户可支配的车的类型

1. 鼠标模块

该模块实现鼠标的相关操作

文件：mouse.c mouse.h

内部具有的具体功能：

鼠标后台，为实现鼠标的各种功能做准备。

void mouseInit(int \*mx,int \*my, int \*mbutt)

该函数为初始化鼠标所需要的各种参数，为把鼠标驱动程 序装入内存做准备。

将鼠标驱动程序装入内存

int init(int xmi, int xma, int ymi, int yma)

该函数初始化鼠标，将鼠标驱动程序装入内存。使用int86中断并选择相应通道和参数，完成鼠标初始化。其中，regs是一个联合体，它用来模拟寄存器，通过对寄存器不同位输入不同入口参数完成鼠标初始化。

在鼠标位置上画出鼠标

void cursor(int x, int y)

该函数在鼠标位置上画出鼠标

获取鼠标未覆盖前图像

void getMousebk(int x, int y)

该函数存储即将要被因画鼠标而被覆盖掉的原图像

抹掉鼠标图案，显示被覆盖掉的原图像

void mousehide(int x, int y)

读鼠标状态

int read(int \*mx, int \*my, int \*mbutt)

该函数读取鼠标所在的位置，以及按键情况。利用int86中断对应的通道和参数实现。

刷新鼠标

void newxy(int \*mx, int \*my, int \*mbutt)

4.用户信息操作模块

该模块实现有关用户信息的相关操作

文件：yonghu.c yonghu.h

内部具有的具体功能：

将用户的信息从.txt文件中读入内存中

void createuserlist（user\*head）

此函数用来将用户的信息从.txt文件中读入程序启动后建立的链表中。在名为user.txt的文件中存放这每一个已注册账号的信息，存储格式为：用户名！密码@金币数#车型$用户名！依此类推。通过循环与fgetc（）函数将文件中的字符一位一位地读取，通过选择语句来接收需要的字符串段。最终将全部有用信息存储。链表中的每一个节点存放一个账户的信息，节点的数据类型为user。将head指针传入本函数，以head为头指针建立链表，每当用fgetc()读到“$”,说明已读完一个节点，此时建立下一个节点，即让current指针指向current->next，再利用一个p指针来读出结构体中的每一个成员。如，当读到“$”时，让p指向current->name,然后每读一个字符让p++。当fgetc()读到“！”时让\*p=‘\0’,再开始读密码，即让p指向current->key,依此类推。

在用户选择注册账号时，寻找是否已经有和和用户输入名相同的用户名存在

int searchname( user \*head,char \*p )

此函数在用户选择注册账号时，寻找是否已经有和和用户输入名相同的用户名存在。函数会对比存有用户信息的链表中是否存在与传进来的用户名相同的用户名（通过循环与选择语句）。若存在则返回0，不存在则返回1.

在用户登录时，判断用户输入的用户名和密码是否存在且匹配

int judge(user \*head,char \*p1,char \*p2)

此函数会判断对比存有用户信息的链表，判断用户在登录时输入的用户名、密码是否匹配。若匹配，则返回1，否则返回0.

在用户注册时，需要在链表中建立新的节点以存放新注册的账号信息。

void newuser(user \*head, char \*p1, char \*p2)

此函数会将用户键入的用户名、密码存入链表中和user.txt中。循环指向链表节点的指针，直到最后一个节点（判断条件为current->next=NULL）。然后开辟一个新的节点，在将信息写入user.txt时，在读用户名、密码和车型时可直接用fputc（）将字符串直接写入文件中并在后面加一个标志符，用户名和密码使用用户注册时输入的用户名和密码，而金币和车型则先给一个初始值，语句为fprintf(fp,"4 #A$");对于金币信息，游戏中设定的最大金币数为999，为了便于在文件中进行数据的读取，所以把金币数量设为字符串形式，这意味着金币这个字符串要留出3个空位，初始化为用户拥有4个金币，故输入文件时为4加上两个空格，限制好金币的位数，就便于以后对金币数量的修改操作，只要控制指针，再将原信息覆盖掉，并补上剩余空格数即可。对于车型信息，通过A、B、C、D区分用户拥有车型的情况，故车型信息在文件中只占一位，在修改时将原信息覆盖掉后不用再补上剩余空格数。

在进行对用户金币、用户车型信息的处理时，涉及到将数字（整形变量）转化成字符串

void tozifu(int number, char \*p)

此函数将整形变量转化成字符串\*p

在进行对用户金币、用户车型信息的处理时，涉及到将字符串转化成数字（整形变量）

int toshuzi(char \*p)

此函数将字符串\*p转化为整形变量，返回此整形变量的数值

当用户在商店购买车辆时，涉及到用户金币的减少，需要将金币数量更新、在链表和txt文件中的对应位置将信息更新。

void changecoin(user \*head,char \*p1,int num)

此函数会利用当前登录的用户（利用当前登录的用户名），在链表中找到对应的账号全部信息，然后修改链表中的金币信息（根据传入参数num），并同时将txt文件中的相应信息更新。在文件中，通过循环取每一个用户名，当取出的用户名和传入函数的当前玩家的用户名一样时，再继续用中间字符变量c=fgetc（）读字符，当读到”$”时，说明文件指针已经指到这一用户信息的末尾位置，由于车型和金币信息所占的位数已知，故可以利用fseek（）函数移动文件指针，使文件指针移到金币信息开始的位置，再利用fputc将新信息把原信息覆盖（金币数量字符串+剩余空格数）。

剩余空格数通过3-strlen（coin）来确定。注意，传入的金币数量为整形，而这里我们存入的时字符串，所以需要先利用将整形变为字符串的函数将金币信息化为字符串。

当用户在商店中购买车辆时，涉及到用户可用车型的变更，需要将车型信息更新，在链表和txt文件中的对应位置将信息更新。

void changecar(user \*head,char \*p1,char \*type)

此函数会利用当前登录的用户（利用当前登录的用户名），在链表中找到对应的账号全部信息，然后修改链表中的车型信息（根据传入参数type），并同时将txt文件中的相应信息更新。这里的算法思路和修改金币数量时的大致相似，有一点不同是这里在用fputc（）将原信息覆盖后不用再补上剩余空格数，因为车型信息始终只占一位。

在商店、选择车型进行比赛时，需要提取用户当前金币、车型信息

void tiqu(user \*head,char \*p0,char \*p1,char \*p2)

此函数会利用当前登录的用户（利用当前登录的用户名），在链表中找到对应的账号全部信息，然后提取出金币和车型信息。

5．初始界面模块

该模块用于显示初始界面，并可从此调入注册、登录界面或退出游戏。

文件：jiemian.c jiemian.h

内部具有的具体功能：

显示初始界面

int jiemian1(user \*head)

该函数用户显示初始界面并实现用户在初始界面的相关操作

6.注册界面模块

该模块用来实现和注册有关的人机交互

文件: zhuce.c zhuce.h

内部具有的具体功能：

显示注册界面

int loadzhuce(user \*head)

此函数用来显示注册界面，并实现人机交互

此函数将调用输入用户名、输入密码函数

void inputname(char \*, void \*);

void inputkey(char \*);

这两函数属于输入模块，将在后面详细介绍。

7.登录界面模块

该模块用来实现和登录有关的人机交互

文件: denglu.c denglu.h

内部具有的具体功能：

显示登录界面

int loaddenglu(user \*)

此函数用来显示登录界面，并实现和登录有关的人机交互

此函数将调用输入用户名、输入密码函数

void inputname(char \*, void \*);

void inputkey(char \*);

这两函数属于输入模块，将在后面详细介绍。

8.输入模块

此模块用户实现有关输入的相关操作（如在注册登录时）

文件：shuru.c shuru.h

内部具有的具体功能：

输入用户名

void inputname(char \*, void \*);

此函数用于将用户输入的用户名实时显示在指定区域，并以字符串形式形成用户输入的用户名（整体）。用中间字符变量c=getkey（）读用户输入的每一个字符，每读一次，就存入用户名字符串中，并在屏幕上利用outtextxy显示出用户刚刚输入的字符，用户输入完毕后会按回车，这时在用户名末尾加字符串结束标志’\0’，此时用户名提取完毕。

输入密码

void inputkey(char \*);

此函数用于将用户输入的密码实时显示在指定区域，并以字符串形式形成用户输入的密码（整体）。这里算法和输入用户名时的类似。

9.键盘模块

此模块用于实现相关键盘操作

文件：keyboard.c keyboard.h

内部具有的具体功能：

读取用户按下的按键的键值

int getkey();

此函数读取用户按下的键值，将其转化为ASCII码返回。

10．主菜单模块

此模块用于显示主菜单界面并完成相应的人机交互

文件: caidan.c caidan.h

内部具有的具体功能：

显示主菜单界面

int zhucaidan(user \*, char\*);

此函数用于显示主菜单界面并完成相应的人机交互

11.商店模块

此模块用于加载商店并完成相应的人机交互

文件： shop.c shop.h

内部具有的具体功能：

商店后台，根据用户的不同情况选择不同的商店界面加载

void shop(user \*head,char \*p0);

此函数根据用户拥有车型的情况，选择不同的商店界面加载

int loadshop1(user \*head,char \*p0,char \*p1,char \*p2);

若用户只有一辆车，加载此界面

int loadshop2(user \*head,char \*p0,char \*p1,char \*p2);

若用户有1、2号车，加载此界面

int loadshop3(user \*head,char \*p0,char \*p1,char \*p2);

若用户有1、3号车，加载此界面

int loadshop4(user \*head,char \*p0,char \*p1,char \*p2);

若用户有1、2、3号车，加载此界面

在这些函数中，会嵌套修改用户金币数量和拥有车型的函数，这意味着同时需要嵌套将字符串转化为整形变量函数和将整形变量函数转化为字符串的函数。用户在界面函数中完成购买后，用户的信息也将会被更改。

12.选赛道模块

此模块用于加载选赛道界面并完成相应的人机交互

文件： saidao.c saidao.h

内部具有的具体功能：

显示选赛道界面

void loadsaidao（user \***head，char**\*p0）；

此函数用来显示选赛道界面并完成相应的人机交互

13.选车型，挡型界面

该模块用来实现和选车、选档型有关的人机交互

文件：choose.c choose.h

内部具有的具体功能：

界面的后台，根据用户的不同情况选择不同的选车界面加载

void cartype（user \*head，char \*p0）;

此函数根据用户拥有车型的情况，选择不同的选车界面进行加载

根据后台函数判断用户情况所得到的不同结果，调用一下不同的函数。这些函数生成的界面中玩家可以选择比赛用车和挡型。

void loadshop1();

若用户只有1号车

void loadshop2();

若用户有1、2号车，加载此界面

void loadshop3();

若用户有1、3号车，加载此界面

void loadshop4();

若用户有1、2、3号车，加载此界面

运行完界面后，主函数首先调用Game\_Init()初始化函数设定显示器图形模式、赛道数据、小车数据等，绘制第1帧赛道和赛车的透视图，显示状态信息。

然后在一个循环里调用游Game\_Main()主体函数，读取按键命令，改变赛车的各参数，计算赛车新位置和新方向，清除之前的画面，相机与键盘控制小车相对位置不变，绘制新位置的赛道和赛车的透视图，显示状态信息。经过void delay(clock\_t td)延时函数使画面定格约55ms。如此循环往复形成赛车行驶的动画效果。按ESQ键或键盘控制的小车跑完一圈，结束循环。

程序退出前，调用Game\_Shutdown()退出图形模式。

14.赛车模块

Game\_Main()主体函数调用了多个模块：

1）键盘读取模块；

2）自动行驶小车行驶模块；

3）键盘控制小车行驶模块；

4）两车相碰检测模块；

5）相机位置设置模块；

6）透视图绘制模块；

7）信息显示模块；

8）比赛结束判断模块。

15.键盘读取模块

该模块主要用于读取当前用户对键盘的操作，控制小车油门有无、转弯以及比赛提前结束。

文件：key.c key.h

内部具有的功能

读玩家按键

int getkey()

该函数获取键盘命令，执行踩油门、转弯、换档、结束程序动作。

void interrupt newint9(...)

该函数代替原键盘中断服务程序，可以直接获取键盘按下、松开情况，将方向键、ESC键的按下状态存储在变量keystate里，作为踩油门、转弯、提前结束比赛的命令。另外将当前空格键按下状态存储在变量oldSPACE里，将空格键的单击状态存储在变量trig里，作为换档的命令。从而实现多按键同时操作。

16.自动行驶小车行驶模块

本模块主要用于自动行驶小车行驶。

文件：autocar.c autocar.h

本模块中的主函数：

void autoruncar(ROADAREA ra[])

首先，获得小车一帧后行驶的速度和路程，之后，要找到小车所在的赛道区段，最后，通过小车当前所在赛道区段与行驶路程计算出小车在一帧后的新位置。在这个过程之中，默认小使用自动挡，沿固定线路行驶。

主要调用5个函数：

void get\_automatic\_vlen(double \*vcur,double vmax,double throttle,double \*len)

int whereroadarea(ROADAREA ra[], MYPOINT p1,double offset)

double runline(ROADAREA ra[],double xingshi,int frompos)

double runarc(ROADAREA ra[],double xingshi,int frompos)

double jicheng(MYPOINT p1,MYPOINT p2,double \*sum)

5个函数的功能如下：

void get\_automatic\_vlen(double \*vcur,double vmax,double throttle,double \*len)函数用于计算自动档时，一步路程和下一步小车速度。其中包括档位的判断，计算速度和计算路程。

首先计算加速度。若踩油门，因碰撞 速度高于最大速度，则给定减加速度；否则根据速度判断是否换档，若速度高于当前档位上限，则自动换档，通过赛车当前的速度、档位和油门计算风阻、摩擦阻力和牵引力，从而计算出赛车的加速度。

再计算一帧时间间隔后的速度和行驶的路程。若没有踩油门，速度小于0则调整为0；若踩油门，如果由低于最大速度加速到超过最大速度，则调整为最大速度。

int whereroadarea(ROADAREA ra[], MYPOINT p1,double offset)函数用于读取此时赛车在赛道的哪一段。

函数通过循环比较当前位置与赛道每段的关系，其中函数int isAtLineRegion(LINE line1, MYPOINT p,double hw)用于判断赛车是否在直线区域内，函数int isAtArcRegion(ARC arc1, MYPOINT p,double hw)用于判断赛车是否在圆弧段区域内，从而得出赛车在哪个区域内。

double runline(ROADAREA ra[],double xingshi,int frompos)和double runarc(ROADAREA ra[],double xingshi,int frompos)这两个函数分别计算小车在直道和弯道上行驶一段路程后的新位置和方向。

double jicheng(MYPOINT p1,MYPOINT p2,double \*sum)计算小车映射到跑道中心线上的路程。

17.键盘控制小车行驶模块

本模块主要用于键盘控制小车行驶。

文件：car.c car.h

本模块中的主函数：

void drivecar()

主要调用4个函数：

void get\_automatic\_vlen(&v,vdrivemax,throttledrive,&len)

void get\_stick\_vlen(&v,throttledrive,&len)

void twowheelstraight(double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong, LINE \*line2)

void twowheelstraight(double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong, LINE \*line2)

三级函数：

void onewheelstraight(double dr, double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong, LINE \*line2)

onewheelturn(double dr, double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong,ARC \*arc2)

void drivecar()函数判断赛车是自动挡还是手动挡，根据选择的模式计算一步的路程和一帧过后的速度和方向，考虑与赛道边界碰撞，计算小车的新位置和方向。若碰撞，小车只能运行到恰好与赛道边界接触的位置，新方向按光线反射原理计算。

void get\_automatic\_vlen(&v,vdrivemax,throttledrive,&len)功能前面已经描述。

计算手动档时，一步路程和下一步小车速度

void get\_stick\_vlen(&v,throttledrive,&len) 函数用于计算手动档时，一步路程和下一步小车速度。不同的档位对应不同的最大速度，对应不同的变速箱变速比。同时，要判断油门的有无。油门的有无通过“↑”键是否按下来判断。有油门时，因碰撞速度高于最大速度，则给定减加速度；否则赛车加速，不超过当前档位最大速度。无油门时赛车减速，速度最小为0。

直行时考虑与道路边界碰撞的新位置和方向。

void twowheelstraight(double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong, LINE \*line2)函数用于直行时考虑与道路边界碰撞的新位置和方向。由于游戏中赛车不能看作一个点，因此，要分别计算左轮和右轮可以行使的最大路程，按较小路程确定赛车的新位置和方向。一个轮子可以行使的最大路程通过void onewheelstraight(double dr, double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong, LINE \*line2)计算，考虑了与道路内外边界是否碰撞。

转弯时考虑与道路边界碰撞的新位置和方向

void twowheelturn(double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong, ARC \*arc2) 函数用于转弯时考虑与道路边界碰撞的新位置和方向，计算原理与直行时相似，调用了void onewheelturn(double dr, double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong,ARC \*arc2)。

double jicheng(MYPOINT p1,MYPOINT p2,double \*sum)计算小车映射到跑道中心线上的路程。

18.两车相碰检测模块

该模块可以检测两个赛车之间的碰撞，对赛车碰撞后的状态进行相应的调整。

文件：collide.c collide.h

本模块中的主函数：

proccarcollide()。

调用的函数：

int carcollide()

proccarcollide()函数检测碰撞并使赛车做出反应，分为检测碰撞和发生碰撞后的反应。首先检测两车是否碰撞，若碰撞则按照质量相同的完全弹性碰撞处理，使速度交换。

int carcollide()函数检测碰撞，检查一个赛车的4个顶点是否在以另一个赛车四个顶点为顶点的矩形内，若在，则表示两车发生碰撞。

19.相机位置设置模块

本模块根据键盘控制赛车当前的位置和方向计算相机的位置和方向。

文件：camera.c camera.h

本模块中的函数为void setcamera(MYPOINT p1,double ang)。

20.透视图绘制模块

该模块用于在屏幕上画出赛道、终点线、赛车。

文件：toushi.c toushi.h

自定义结构类型：

typedef struct \_MYPOINT//坐标系中的点

{

double x;//横坐标

double y;//纵坐标

}MYPOINT, \*PMYPOINT;创建坐标系中的点的结构，是之后画图的基础，其中横纵坐标用double类型时可使点的计算更精确。

typedef struct \_POINT//屏幕坐标系中的点

{

int x; //横坐标

int y; //纵坐标

}POINT, \*PPOINT;创建横纵坐标均为整型的点的结构，是因为使用头文件graphics.h中的画图函数时函数中的参数都为整数。

typedef struct \_ARC//圆弧

{

double x;//圆心横坐标

double y;//圆心纵坐标

double r;//圆弧半径

double angstart;//起始角

double dang;//圆弧扫过的圆心角

}ARC,\*PARC;游戏中的弯道由圆弧构成，圆弧结构中的数据成员包括圆心的横纵坐标，圆弧半径以及圆弧的起始角和扫过的圆心角，创建圆弧结构便于画弯道。

typedef struct \_LINE//线段

{

MYPOINT p1;//线段的一个端点

MYPOINT p2;//线段的另一端点

}LINE,\*PLINE;

typedef struct \_ROADAREA//赛道区域

{

char type[5];//各段赛道类型

void \*pcurve;//指向具体赛道的指针

}ROADAREA;绘制赛道的过程中将赛道中直道和弯道拆分为多个片段，分别独立计算直道和弯道的情况，若有需要则进行组合，便于处理。其中用char型数组存储片段类型，确定赛道为直道或弯道，void类型指针用于指向具体的直线段、圆弧段。使用时，根据片段类型，为直线段时指针强制转换为结构指针PLINE，为圆弧段时指针强制转换为结构指针PARC。

主要几何计算函数：

void x2xp(MYPOINT pntsight, double alpha0, MYPOINT pnt, MYPOINT \*pnt\_p)函数用于将全局坐标系转换为局部坐标系。

void xp2x(MYPOINT pntsight, double alpha0, MYPOINT pnt\_p, MYPOINT \*pnt)函数用于将局部坐标系转换为全局坐标系，与x2xp函数功能相反。

void line\_intersection(MYPOINT p1, MYPOINT p2, MYPOINT p3, MYPOINT p4, int \*count, MYPOINT pout[])求线段与线段的交点。

void arcline\_intersection(MYPOINT p1, MYPOINT p2, ARC arc1, MYPOINT intersection[], int \*n)求线段与圆弧的交点。

透视基础函数：

void Pnt\_Toushi(MYPOINT pnt, MYPOINT \*pnt\_t)用透视原理将空间三维点转换为屏幕上的二维点。

本模块中的主函数：

void toushitu()

调用的函数：

void draw\_solidloop(int nseg,ROADAREA ra[])

void draw\_dashloop(int nseg, ROADAREA ra[])

void drawautocar()

void drawdrivecar()

void drawendline()

三级函数：

int seeline(MYPOINT p1, MYPOINT p2, LINE \*line1)

int judgein(MYPOINT p1)

void draw\_line(LINE line1)

int seearc(ARC arc2, ARC arc3[])

void draw\_solidarc(ARC arc1)

void draw\_dashline(LINE linearea,LINE line)

void draw\_dasharc(ARC arcarea,ARC arc)

void drawcarline(LINE line1,double h);

void drawcarupline(LINE line1,double h);

void drawcarsideline();

void draw\_solidloop(int nseg,ROADAREA ra[])函数用于绘制赛道实线，分为两个模块，一个用于绘制直实线，一个用于绘制圆弧实线。

对赛道直线段则调用int seeline(MYPOINT p1, MYPOINT p2, LINE \*line1)函数获取直线在相机视野内的部分。函数中首先用int judgein(MYPOINT p1)函数判断直线段的端点是否在视野内，通过函数void x2xp(MYPOINT pntsight, double alpha0, MYPOINT pnt, MYPOINT \*pnt\_p)和void xp2x(MYPOINT pntsight, double alpha0, MYPOINT pnt\_p, MYPOINT \*pnt)的坐标转换之后比较点与视野边界的横纵坐标和角度便可判断。再利用void line\_intersection(MYPOINT p1, MYPOINT p2, MYPOINT p3, MYPOINT p4, int \*count, MYPOINT pout[])函数分别求直线段和视野的四个边界的交点。综合这两步最终整合出视野内的线段。最后，用void draw\_line(LINE line1)画出视野内的实直线赛道。

对赛道圆弧线段则调用int seearc(ARC arc2, ARC arc3[])函数获取弯道在相机视野内的部分。首先用int judgein(MYPOINT p1)函数判断圆弧的端点是否在视野内。再利用void arcline\_intersection(MYPOINT p1, MYPOINT p2, ARC arc1, MYPOINT intersection[], int \*n)函数分别求转弯段赛道和视野的四个边界的交点。综合这两步最终整合出视野内的圆弧。最后用void draw\_solidarc(ARC arc1)画出视野内的实线圆弧赛道。

void draw\_dashloop(int nseg, ROADAREA ra[])函数用于绘制赛道虚线，分为两个模块，一个用于绘制直虚线，一个用于绘制圆弧虚线。。首先与绘制赛道实线相同，获取相机视野内的直线、圆弧段。之后用void draw\_dashline(LINE linearea,LINE line)来绘制直虚线，用void draw\_dasharc(ARC arcarea,ARC arc)绘制圆弧虚线。void draw\_dashline(LINE linearea,LINE line)函数首先画出两端不完整虚线部分，再画出中间完整的虚线部分。用int wheredashline(linearea,line.p1,&n1)函数检测端点是实心部分还是透明部分，若是实心部分，则结合车的位置和视野位置画出不完整的实心部分。之后用void draw\_line(LINE line1) 函数画中间每一段完整的直虚线。void draw\_dasharc(ARC arcarea,ARC arc) 函数首先画出两端不完整虚线部分，再画出中间完整的虚线部分。用int wheredasharc(arcarea,ang1,&n1) 函数检测端点是实心部分还是透明部分，若是实心部分，则结合车的位置和视野位置画出不完整的实心部分。之后用void draw\_solidarc(arc1)函数画中间每一段完整的圆弧虚线。

void drawautocar()画自动行驶赛车，调用void drawcarupline(LINE line1,double h)画顶面轮廓线，调用void drawcarline(LINE line1,double h)画后面轮廓线，调用 void drawcarsideline()画侧面轮廓线。隐藏线不画。

void drawdrivecar()画键盘控制赛车。

void drawendline()画终点线。

21.信息显示模块

本模块用于显示双方赛车行驶时的信息。

文件：info.c info.h

本模块中的函数：

void displayinf()

将两车行驶路程、速度、比赛经历时间、键盘控制小车的档位、跑完一圈各自所用时间显示在屏幕上。

22. 比赛结束判断模块

判断游戏结束，游戏结束后切换界面。

文件：end.c end.h

本模块中的函数：

void judgetoendline()。

键盘控制小车行驶一圈结束，最先到达终点为第一名。

# 源代码

1. main.c

#include<stdio.h>

#include<stdlib.h>

#include <graphics.h>

#include <conio.h>

#include<dos.h>

#include<math.h>

#include<time.h>

#include "mouse1.h"

#include "zhuce2.h"

#include "denglu1.h"

#include "jiemian2.h"

#include "yonghu.h"

#include "jiegouti.h"

#include "shop.h"

#include "xuanche.h"

#include "saidao.h"

#define MYMACRO

#include "mytype.h"

#include "common.h"

#include "record.h"

#include "chaxun.h"

#include "result.h"

int main()

{

int driver = VGA;

int mode = VGAHI;

user \*head=NULL;

record \*head1=NULL;

char p0[15];

int saidao;

int mark=1;

int \*p1,\*p2,\*p3;

int \*p5;

int a=1,b=1,c=1;

int d=1;

int rank;

int x1,x2,x3; //x1判断比赛是否完成，x2记录比赛结束后玩家所用时间，x3记录电脑车用时

x1=1; //0代表完成比赛， 1代表未完成

x2=1;

x3=2;

p1=&a; //对应赛道

p2=&b; //对应车型

p3=&c; //对应挡型

p5=&d; //对应燃油购买情况

initgraph(&driver, &mode, "..\\BORLANDC\\bgi");

if ((head = (user \*)malloc(sizeof(user))) == NULL)

{

printf("\nOverflow");

return 0;

}

userlist(head);

if ((head1 = (record \*)malloc(sizeof(record))) == NULL)

{

printf("\nOverflow");

return 0;

}

recordlist(head1);

// jiemian1(head);

// newrecord(head1,"123",1,1,99,1);

while(1)

{

if(mark==1)

mark=jiemian1();

if(mark==2)

mark=loadzhuce(head,p0);

if(mark==3)

mark=loaddenglu(head,p0);

if(mark==4)

mark=zhucaidan(head,p0);

if(mark==5)

mark=shop(head,p0,p5);

if(mark==6)

{

mark=sai(head,p0,p1);

}

if(mark==7)

{

mark=cartype(head,p0,p2,p3);

}

if(mark==9)

{

mark=loadchaxun(head1,p0);

}

if (mark==8)

{

jiazai();

setbkcolor(BLACK);

cleardevice();

settextstyle(3,0,1);

setlinestyle(0,0,1);

if(\*p1==1)

segs=4;

else if(\*p1==2)

segs=3;

if(\*p2==1)

{

if(\*p5==0)

{

vdrivemax=30;

}

else if(\*p5==1)

{

vdrivemax=20;

}

mecolor=RED;

}

else if(\*p2==2)

{

if(\*p5==0)

{

vdrivemax=50;

}

else if (\*p5==1)

{

vdrivemax=40;

}

mecolor=YELLOW;

}

else if(\*p2==3)

{

if(\*p5==0)

{

vdrivemax =70;

}

else if (\*p5==1)

{

vdrivemax =60;

}

mecolor=BLUE;

}

if(\*p3==1)

geartype=AUTO;

else if(\*p3==2)

geartype=STICK;

carRace(&x1);

x2=timeme;

x3=timeauto;

if(!x1)

{

rank=compare(x2,x3);

result(head,p0,rank);

newrecord(head1,p0,\*p1,\*p2,x2,rank);

}

\*p5=1;

mark=4;

}

if(mark==0)

{

delay(100); //防止下次进入时出bug

exit(0);

}

}

return 1;

}

----------------------------------------------------------------------------------------------------------------------------

saidao.c

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<graphics.h>

#include<conio.h>

#include<dos.h>

#include<bios.h>

#include "mouse1.h"

#include "jiegouti.h"

#include "yonghu.h"

#include "saidao.h"

int sai(user \*head,char \*p0,int \*p1)

{

int buttons, mx, my;

int a1[]={380,300,500,300,440,196,380,300};

int a2[]={390,292,490,292,440,203,390,292};

int a3[]={400,286,480,286,440,220,400,286};

cleardevice();

setcolor(BLUE);

setbkcolor(CYAN);

//setlinestyle(2,0,3);

setfillstyle(1,LIGHTGRAY);

settextstyle(SANS\_SERIF\_FONT ,HORIZ\_DIR,3);

setlinestyle(1,0,1);

rectangle(160,210,240,290);

setlinestyle(0,0,3);

rectangle(150,200,250,300);

rectangle(170,220,230,280);

setlinestyle(0,0,3);

drawpoly(4,a1);

drawpoly(4,a3);

bar(150,330,250,350);

bar(390,330,490,350);

outtextxy(162,322,"choose");

outtextxy(402,322,"choose");

outtextxy(162,295,"square");

outtextxy(402,295,"triangle");

outtextxy(500,5,"Back");

mouseInit(&mx, &my,&buttons);

while (1)

{

/\*if(kbhit()!=0)

{

temp=getkey();

}\*/

newxy(&mx, &my, &buttons);

if(mx>=150&&mx<=250&&my>=330&&my<=350&&buttons)

{

\*p1=1;

delay(100);

return 7;

}

if(mx>=390&&mx<=490&&my>=330&&my<=350&&buttons)

{

\*p1=2;

delay(100);

return 7;

}

if(mx>=495&&mx<=545&&my>=12&&my<=33&&buttons)

{

return 4;

}

}

}

----------------------------------------------------------------------------------------------------------------------------

cartype.c

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<graphics.h>

#include<conio.h>

#include<dos.h>

#include<bios.h>

#include "mouse1.h"

#include "jiegouti.h"

#include "yonghu.h"

#include "xuanche.h"

int cartype(user \*head,char \*p0,int \*p2,int \*p3)

{

char \*type[3];

char \*num[3];

// void \*buffer;

int buttons, mx, my;

int flag1=1,flag2=0,flag3=0;

//int size;

setcolor(CYAN);

outtextxy(402,295,"triangle");

cleardevice();

setcolor(BLUE);

setbkcolor(CYAN);

setfillstyle(1,LIGHTGRAY);

mouseInit(&mx, &my,&buttons);

//size=imagesize(0,120,220,420);

//buffer=malloc(size);

//getimage(120,120,220,420,buffer);

tiqu(head,p0,num,type);

settextstyle(SANS\_SERIF\_FONT ,HORIZ\_DIR,3);

draw(flag1,flag2,flag3);

while(1)

{

newxy(&mx, &my, &buttons);

if(mx>=140&&mx<=215&&my>=140&&my<=160&&buttons)

{

//putimage(120,120,buffer,COPY\_PUT);

draw(1,flag2,flag3);

rectangle(10,80,125,180);

\*p2=1;

flag3=1;

}

if(mx>=140&&mx<=215&&my>=260&&my<=280&&buttons)

{

if(strcmp(type,"B")==0||strcmp(type,"D")==0)

{

//putimage(120,120,buffer,COPY\_PUT);

draw(1,flag2,flag3);

rectangle(10,200,125,300);

\*p2=2;

flag3=2;

}

else

{

flag3=0;

draw(1,flag2,flag3);

outtextxy(180,200,"you don't have this car");

}

}

if(mx>=140&&mx<=215&&my>=380&&my<=400&&buttons)

{

if(strcmp(type,"C")==0||strcmp(type,"D")==0)

{

//putimage(120,120,buffer,COPY\_PUT);

draw(1,flag2,flag3);

rectangle(10,320,125,410);

\*p2=3;

flag3=3;

}

else

{

flag3=0;

draw(1,flag2,flag3);

outtextxy(180,200,"you don't have this car");

}

}

if(mx>=480&&mx<=640&&my>=140&&my<=180&&buttons)

{

draw(2,flag2,flag3);

rectangle(470,120,640,200);

\*p3=1;

flag2=1;

}

if(mx>=480&&mx<=640&&my>=280&&my<=320&&buttons)

{

draw(2,flag2,flag3);

rectangle(470,260,640,340);

\*p3=2;

flag2=2;

}

if(mx>=480&&mx<=640&&my>=400&&my<=480&&buttons)

{

if(flag2==0||flag3==0)

{

outtextxy(180,200,"you need to choose the type");

}

else return 8;

}

}

}

/////////////////////////////////////////////////////////////////////

//draw函数说明： //

// 取值 0 1 2 3 //

//参数：flag1 无 选车时flag1=1 选档时flag1=2 无 //

// flag2 未选挡 已选自动挡 已选手动挡 无 //

// flag3 未选车 已选1车 已选2车 已选3车 //

// //

// //

/////////////////////////////////////////////////////////////////////

void draw(int flag1, int flag2,int flag3)

{

if(flag1==1)

{

if(flag2==0)

{

drawback();

}

else if(flag2==1)

{

drawback();

rectangle(470,120,640,200);

}

else

{

drawback();

rectangle(470,260,640,340);

}

}

else if(flag1=2)

{

if(flag3==0)

{

drawback();

}

else if(flag3==1)

{

drawback();

rectangle(10,80,125,180);

}

else if(flag3==2)

{

drawback();

rectangle(10,200,125,300);

}

else

{

drawback();

rectangle(10,320,125,410);

}

}

}

////////////////////////////////////////

//函数drawback：画出基础界面（包括车）

////////////////////////////////////

void drawback()

{

int a1[]={15,140,15,130,25,120,50,120,70,100,100,100,120,120,120,140,15,140};

int a2[]={15,260,15,250,25,240,50,240,70,220,100,220,120,240,120,260,15,260};

int a3[]={15,380,15,370,25,360,50,360,70,340,100,340,120,360,120,380,15,380};

cleardevice();

//////////////////////

//画背景框和文字信息

setcolor(BLUE);

setbkcolor(CYAN);

setfillstyle(1,LIGHTGRAY);

settextstyle(SANS\_SERIF\_FONT ,HORIZ\_DIR,3);

bar(140,140,215,160);

bar(140,260,215,280);

bar(140,380,215,400);

bar(480,140,640,180);

bar(480,280,640,320);

bar(480,400,640,480);

outtextxy(142,135,"choose");

outtextxy(142,255,"choose");

outtextxy(142,375,"choose");

outtextxy(482,135,"automatic");

outtextxy(482,275,"manual");

outtextxy(482,395,"start game");

//////////////////////////////////////////

//画红车

setcolor(DARKGRAY);

drawpoly(9,a1);

arc(45,140,180,360,10);

arc(100,140,180,360,10);

setfillstyle(1,RED);

floodfill(40,130,DARKGRAY);

setfillstyle(1,DARKGRAY);

floodfill(45,146,DARKGRAY);

floodfill(100,146,DARKGRAY);

/////////////////////////////////////////

//画黄车

setcolor(DARKGRAY);

drawpoly(9,a2);

arc(45,260,180,360,10);

arc(100,260,180,360,10);

setfillstyle(1,YELLOW);

floodfill(40,250,DARKGRAY);

setfillstyle(1,DARKGRAY);

floodfill(45,266,DARKGRAY);

floodfill(100,266,DARKGRAY);

////////////////////////////////////

//画蓝车

setcolor(DARKGRAY);

drawpoly(9,a3);

arc(45,380,180,360,10);

arc(100,380,180,360,10);

setfillstyle(1,BLUE);

floodfill(40,370,DARKGRAY);

setfillstyle(1,DARKGRAY);

floodfill(45,386,DARKGRAY);

floodfill(100,386,DARKGRAY);

setcolor(BLUE);

}

--------------------------------------------------------------------------------------------------------------

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<graphics.h>

#include<conio.h>

#include<dos.h>

#include<bios.h>

#include "mouse1.h"

#include "jiegouti.h"

#include "yonghu.h"

#include "xuanche.h"

int cartype(user \*head,char \*p0,int \*p2,int \*p3)

{

char \*type[3];

char \*num[3];

// void \*buffer;

int buttons, mx, my;

int flag1=1,flag2=0,flag3=0;

//int size;

setcolor(CYAN);

outtextxy(402,295,"triangle");

cleardevice();

setcolor(BLUE);

setbkcolor(CYAN);

setfillstyle(1,LIGHTGRAY);

mouseInit(&mx, &my,&buttons);

//size=imagesize(0,120,220,420);

//buffer=malloc(size);

//getimage(120,120,220,420,buffer);

tiqu(head,p0,num,type);

settextstyle(SANS\_SERIF\_FONT ,HORIZ\_DIR,3);

draw(flag1,flag2,flag3);

while(1)

{

newxy(&mx, &my, &buttons);

if(mx>=140&&mx<=215&&my>=140&&my<=160&&buttons)

{

//putimage(120,120,buffer,COPY\_PUT);

draw(1,flag2,flag3);

rectangle(10,80,125,180);

\*p2=1;

flag3=1;

}

if(mx>=140&&mx<=215&&my>=260&&my<=280&&buttons)

{

if(strcmp(type,"B")==0||strcmp(type,"D")==0)

{

//putimage(120,120,buffer,COPY\_PUT);

draw(1,flag2,flag3);

rectangle(10,200,125,300);

\*p2=2;

flag3=2;

}

else

{

flag3=0;

draw(1,flag2,flag3);

outtextxy(180,200,"you don't have this car");

}

}

if(mx>=140&&mx<=215&&my>=380&&my<=400&&buttons)

{

if(strcmp(type,"C")==0||strcmp(type,"D")==0)

{

//putimage(120,120,buffer,COPY\_PUT);

draw(1,flag2,flag3);

rectangle(10,320,125,410);

\*p2=3;

flag3=3;

}

else

{

flag3=0;

draw(1,flag2,flag3);

outtextxy(180,200,"you don't have this car");

}

}

if(mx>=480&&mx<=640&&my>=140&&my<=180&&buttons)

{

draw(2,flag2,flag3);

rectangle(470,120,640,200);

\*p3=1;

flag2=1;

}

if(mx>=480&&mx<=640&&my>=280&&my<=320&&buttons)

{

draw(2,flag2,flag3);

rectangle(470,260,640,340);

\*p3=2;

flag2=2;

}

if(mx>=480&&mx<=640&&my>=400&&my<=480&&buttons)

{

if(flag2==0||flag3==0)

{

outtextxy(180,200,"you need to choose the type");

}

else return 8;

}

}

}

/////////////////////////////////////////////////////////////////////

//draw函数说明： //

// 取值 0 1 2 3 //

//参数：flag1 无 选车时flag1=1 选档时flag1=2 无 //

// flag2 未选挡 已选自动挡 已选手动挡 无 //

// flag3 未选车 已选1车 已选2车 已选3车 //

// //

// //

/////////////////////////////////////////////////////////////////////

void draw(int flag1, int flag2,int flag3)

{

if(flag1==1)

{

if(flag2==0)

{

drawback();

}

else if(flag2==1)

{

drawback();

rectangle(470,120,640,200);

}

else

{

drawback();

rectangle(470,260,640,340);

}

}

else if(flag1=2)

{

if(flag3==0)

{

drawback();

}

else if(flag3==1)

{

drawback();

rectangle(10,80,125,180);

}

else if(flag3==2)

{

drawback();

rectangle(10,200,125,300);

}

else

{

drawback();

rectangle(10,320,125,410);

}

}

}

////////////////////////////////////////

//函数drawback：画出基础界面（包括车）

////////////////////////////////////

void drawback()

{

int a1[]={15,140,15,130,25,120,50,120,70,100,100,100,120,120,120,140,15,140};

int a2[]={15,260,15,250,25,240,50,240,70,220,100,220,120,240,120,260,15,260};

int a3[]={15,380,15,370,25,360,50,360,70,340,100,340,120,360,120,380,15,380};

cleardevice();

//////////////////////

//画背景框和文字信息

setcolor(BLUE);

setbkcolor(CYAN);

setfillstyle(1,LIGHTGRAY);

settextstyle(SANS\_SERIF\_FONT ,HORIZ\_DIR,3);

bar(140,140,215,160);

bar(140,260,215,280);

bar(140,380,215,400);

bar(480,140,640,180);

bar(480,280,640,320);

bar(480,400,640,480);

outtextxy(142,135,"choose");

outtextxy(142,255,"choose");

outtextxy(142,375,"choose");

outtextxy(482,135,"automatic");

outtextxy(482,275,"manual");

outtextxy(482,395,"start game");

//////////////////////////////////////////

//画红车

setcolor(DARKGRAY);

drawpoly(9,a1);

arc(45,140,180,360,10);

arc(100,140,180,360,10);

setfillstyle(1,RED);

floodfill(40,130,DARKGRAY);

setfillstyle(1,DARKGRAY);

floodfill(45,146,DARKGRAY);

floodfill(100,146,DARKGRAY);

/////////////////////////////////////////

//画黄车

setcolor(DARKGRAY);

drawpoly(9,a2);

arc(45,260,180,360,10);

arc(100,260,180,360,10);

setfillstyle(1,YELLOW);

floodfill(40,250,DARKGRAY);

setfillstyle(1,DARKGRAY);

floodfill(45,266,DARKGRAY);

floodfill(100,266,DARKGRAY);

////////////////////////////////////

//画蓝车

setcolor(DARKGRAY);

drawpoly(9,a3);

arc(45,380,180,360,10);

arc(100,380,180,360,10);

setfillstyle(1,BLUE);

floodfill(40,370,DARKGRAY);

setfillstyle(1,DARKGRAY);

floodfill(45,386,DARKGRAY);

floodfill(100,386,DARKGRAY);

setcolor(BLUE);

}

caidan1.c

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<graphics.h>

#include<conio.h>

#include<math.h>

#include<dos.h>

#include<bios.h>

#include "mouse1.h"

#include "jiegouti.h"

#include "shop.h"

int zhucaidan(user \*head, char \*p0)

{

int size;

int buttons, mx, my;

setcolor(YELLOW);

setbkcolor(RED);

settextstyle(1,0,3); //必要的初始化

setlinestyle(0,0,3);

cleardevice();

settextstyle(1,HORIZ\_DIR,7);

outtextxy(40,102,"Play");

outtextxy(40,205,"SHOP");

outtextxy(40,400,"Quit");

outtextxy(40,300,"Search");

mouseInit(&mx, &my,&buttons);

rectangle(37,115,170,185);

rectangle(35,218,196,280);

rectangle(35,415,170,470);

rectangle(35,315,242,375);

while (1)

{

newxy(&mx, &my, &buttons);

if(mx>=37&&mx<=170&&my>=115&&my<=185&&buttons)

{

//进入游戏

return 6;

}

if(mx>=35&&mx<=196&&my>=218&&my<=280&&buttons)

{

//返回main，进入商店

return 5;

}

if(mx>=35&&mx<=170&&my>=415&&my<=470&&buttons)

{

// 退出游戏

delay(100);

cleardevice();

exit(0);

}

if(mx>=35&&mx<=242&&my>=150&&my<=375&&buttons)

{

// 返回main，进入查询

return 9;

}

}

}

shop.c

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<bios.h>

#include<graphics.h>

#include "jiegouti.h"

#include "yonghu.h"

#include "caidan1.h"

#include "shop.h"

#include "mouse1.h"

//////////////////////////////////

//int shop函数说明：

//该函数提取用户金币、车型等信息，根据信息加载不同界面

//用户完成购买后会回到此函数，是一个循环，继续根据修改后的数据再次加载商店

/////////////////////////////////

int shop(user \*head,char \*p0, int \*p)

{

char \*type[3];

char \*num[3];

int mark;

tiqu(head,p0,num,type);

settextstyle(SANS\_SERIF\_FONT ,HORIZ\_DIR,3);

if(strcmp(type,"A")==0&&(\*p==0))

{

mark=loadshop1(head,p0,num,type);

}

else if(strcmp(type,"B")==0&&(\*p==0))

{

mark=loadshop2(head,p0,num,type);

}

else if(strcmp(type,"C")==0&&(\*p==0))

{

mark=loadshop3(head,p0,num,type);

}

else if(strcmp(type,"D")==0&&(\*p==0))

{

mark=loadshop4(head,p0,num,type);

}

else if(strcmp(type,"A")==0&&(\*p==1))

{

mark=loadshop5(head,p0,num,type,p);

}

else if(strcmp(type,"B")==0&&(\*p==1))

{

mark=loadshop6(head,p0,num,type,p);

}

else if(strcmp(type,"C")==0&&(\*p==1))

{

mark=loadshop7(head,p0,num,type,p);

}

else if(strcmp(type,"D")==0&&(\*p==1))

{

mark=loadshop8(head,p0,num,type,p);

}

return mark;

}

/////////////////////////////////////////////////

//loadshop1函数说明：当玩家未购买过车且购买了燃油时加载的界面及一些后台操作

//参数：p0该玩家用户名 p1为该玩家持有的金币数，但此时为字符串形式

// p2为该玩家持有的车

//

/////////////////////////////////////////////////

int loadshop1(user \*head,char \*p0,char \*p1,char \*p2)

{

int buttons, mx, my;

int n;

int mb=3,mc=4;

cleardevice();

setviewport(0,0,640,480,1);

drawcar2();

drawcar3();

setfillstyle(1,BROWN);

bar(0,0,640,40);

settextstyle(1,0,3);

outtextxy(500,5,"Back");

outtextxy(55,5,"coin");

outtextxy(120,5,p1);

mouseInit(&mx, &my,&buttons);

n=toshuzi(p1);

while (1)

{

newxy(&mx, &my, &buttons);

if(mx>=150&&mx<=250&&my>=350&&my<=370&&buttons)

{

if(n>=mb)

{

changecar(head,p0,"B");

changecoin(head,p0,n-mb);

//outtextxy(192,125,"now you can use the yellow car");

//outtextxy(192,150,"press any key to continue");

msgbox(1);

// getch();

return 5;

}

else

{

//outtextxy(192,125,"your coin is not enough");

//getch();

msgbox(0);

return 5;

}

}

if(mx>=390&&mx<=490&&my>=350&&my<=370&&buttons)

{

if(n>=mc)

{

changecar(head,p0,"C");

changecoin(head,p0,n-mc);

//outtextxy(192,125,"now you can use the blue car");

//outtextxy(192,150,"press any key to continue");

msgbox(1);

// getch();

return 5;

}

else

{

// outtextxy(192,125,"your coin is not enough");

//outtextxy(192,150,"press any key to continue");

//getch();

msgbox(0);

return 5;

}

}

if(mx>=495&&mx<=545&&my>=12&&my<=33&&buttons)

{

//返回主菜单

return 4;

}

}

}

/////////////////////////////////////////////////

//loadshop2函数说明：当玩家未购买过黄车且购买了燃油时加载的界面及一些后台操作

//参数：p0该玩家用户名 p1为该玩家持有的金币数，但此时为字符串形式

// p2为该玩家持有的车

//

/////////////////////////////////////////////////

int loadshop2(user \*head,char \*p0,char \*p1,char \*p2)

{

int buttons, mx, my;

int n;

int mc=4;

cleardevice();

setviewport(0,0,640,480,1);

drawcar3();

setfillstyle(1,BROWN);

bar(0,0,640,40);

settextstyle(1,0,3);

outtextxy(500,5,"Back");

outtextxy(55,5,"coin");

outtextxy(120,5,p1);

mouseInit(&mx, &my,&buttons);

n=toshuzi(p1);

while (1)

{

newxy(&mx, &my, &buttons);

if(mx>=390&&mx<=490&&my>=350&&my<=370&&buttons)

{

if(n>=mc)

{

changecar(head,p0,"D");

changecoin(head,p0,n-mc);

// outtextxy(192,125,"now you can use the blue car");

//outtextxy(192,150,"press any key to continue");

//getch();

msgbox(1);

return 4;

}

else

{

//outtextxy(192,125,"your coin is not enough");

//outtextxy(192,150,"press any key to continue");

//getch();

msgbox(0);

return 4;

}

}

if(mx>=495&&mx<=545&&my>=12&&my<=33&&buttons)

{

return 4;

}

}

}

/////////////////////////////////////////////////

//loadshop3函数说明：当玩家未购买过蓝车且购买了燃油时加载的界面及一些后台操作

//参数：p0该玩家用户名 p1为该玩家持有的金币数，但此时为字符串形式

// p2为该玩家持有的车

//

/////////////////////////////////////////////////

int loadshop3(user \*head,char \*p0,char \*p1,char \*p2)

{

int buttons, mx, my;

int n;

int mb=3;

cleardevice();

setviewport(0,0,640,480,1);

drawcar2();

setfillstyle(1,BROWN);

bar(0,0,640,40);

settextstyle(1,0,3);

outtextxy(500,5,"Back");

outtextxy(55,5,"coin");

outtextxy(120,5,p1);

mouseInit(&mx, &my,&buttons);

n=toshuzi(p1);

while (1)

{

newxy(&mx, &my, &buttons);

if(mx>=150&&mx<=250&&my>=350&&my<=370&&buttons)

{

if(n>=mb)

{

changecar(head,p0,"D");

changecoin(head,p0,n-mb);

//outtextxy(192,125,"now you can use the yellow car");

//outtextxy(192,150,"press any key to continue");

//getch();

msgbox(1);

return 4;

}

else

{

//outtextxy(192,125,"your coin is not enough");

//outtextxy(192,150,"press any key to continue");

//getch();

msgbox(0);

return 4;

}

}

if(mx>=495&&mx<=545&&my>=12&&my<=33&&buttons)

{

return 4;

}

}

}

/////////////////////////////////////////////////

//loadshop4函数说明：当玩家未购买过黄车和蓝车且购买了燃油时加载的界面及一些后台操作

//参数：p0该玩家用户名 p1为该玩家持有的金币数，但此时为字符串形式

// p2为该玩家持有的车

//

/////////////////////////////////////////////////

int loadshop4(user \*head,char \*p0,char \*p1,char \*p2)

{

int buttons, mx, my;

mouseInit(&mx, &my,&buttons);

cleardevice();

setfillstyle(1,LIGHTGRAY);

bar(170,180,470,250);

rectangle(290,215,340,250);

outtextxy(300,215,"OK");

outtextxy(190,180,"you have bought all things");

while(1)

{

newxy(&mx, &my, &buttons);

if(mx>=290&&mx<=340&&my>=215&&my<=250&&buttons)

{

return 4;

}

}

}

/////////////////////////////////////////////////

//loadshop5函数说明：当玩家未购买过车和燃油时加载的界面及一些后台操作

//参数：p0该玩家用户名 p1为该玩家持有的金币数，但此时为字符串形式

// p2为该玩家持有的车

// p实际代表从主函数传过来的一个标准，\*p=0则玩家此次游戏已经购买了燃油，\*p=1则还未购买

/////////////////////////////////////////////////

int loadshop5(user \*head,char \*p0,char \*p1,char \*p2,int \*p)

{

int buttons, mx, my;

int n;

int mb=3,mc=4;

int mg=1;

cleardevice();

setviewport(0,0,640,480,1);

drawcar2();

drawcar3();

drawgas();

setfillstyle(1,BROWN);

bar(0,0,640,40);

settextstyle(1,0,3);

outtextxy(500,5,"Back");

outtextxy(55,5,"coin");

outtextxy(120,5,p1);

mouseInit(&mx, &my,&buttons);

n=toshuzi(p1);

while (1)

{

newxy(&mx, &my, &buttons);

if(mx>=150&&mx<=250&&my>=350&&my<=370&&buttons)

{

if(n>=mb)

{

changecar(head,p0,"B");

changecoin(head,p0,n-mb);

//outtextxy(192,125,"now you can use the yellow car");

//outtextxy(192,150,"press any key to continue");

msgbox(1);

// getch();

return 5;

}

else

{

//outtextxy(192,125,"your coin is not enough");

//getch();

msgbox(0);

return 5;

}

}

if(mx>=390&&mx<=490&&my>=350&&my<=370&&buttons)

{

if(n>=mc)

{

changecar(head,p0,"C");

changecoin(head,p0,n-mc);

//outtextxy(192,125,"now you can use the blue car");

//outtextxy(192,150,"press any key to continue");

msgbox(1);

// getch();

return 5;

}

else

{

// outtextxy(192,125,"your coin is not enough");

//outtextxy(192,150,"press any key to continue");

//getch();

msgbox(0);

return 5;

}

}

if(mx>=80&&mx<=160&&my>=140&&my<=180&&buttons)

{

if(n>=mg)

{

changecoin(head,p0,n-mg);

\*p=0;

msgbox(1);

return 5;

}

else

{

msgbox(0);

return 5;

}

}

if(mx>=495&&mx<=545&&my>=12&&my<=33&&buttons)

{

//返回主菜单

return 4;

}

}

}

/////////////////////////////////////////////////

//loadshop6函数说明：当玩家只购买了黄车时加载的界面及一些后台操作

//参数：p0该玩家用户名 p1为该玩家持有的金币数，但此时为字符串形式

// p2为该玩家持有的车

// p实际代表从主函数传过来的一个标准，\*p=0则玩家此次游戏已经购买了燃油，\*p=1则还未购买

/////////////////////////////////////////////////

int loadshop6(user \*head,char \*p0,char \*p1,char \*p2,int \*p)

{

int buttons, mx, my;

int n;

int mc=4;

int mg=1;

cleardevice();

setviewport(0,0,640,480,1);

drawcar3();

drawgas();

setfillstyle(1,BROWN);

bar(0,0,640,40);

settextstyle(1,0,3);

outtextxy(500,5,"Back");

outtextxy(55,5,"coin");

outtextxy(120,5,p1);

mouseInit(&mx, &my,&buttons);

n=toshuzi(p1);

while (1)

{

newxy(&mx, &my, &buttons);

if(mx>=390&&mx<=490&&my>=350&&my<=370&&buttons)

{

if(n>=mc)

{

changecar(head,p0,"D");

changecoin(head,p0,n-mc);

// outtextxy(192,125,"now you can use the blue car");

//outtextxy(192,150,"press any key to continue");

//getch();

msgbox(1);

return 5;

}

else

{

//outtextxy(192,125,"your coin is not enough");

//outtextxy(192,150,"press any key to continue");

//getch();

msgbox(0);

return 5;

}

}

if(mx>=80&&mx<=160&&my>=140&&my<=180&&buttons)

{

if(n>=mg)

{

changecoin(head,p0,n-mg);

\*p=0;

msgbox(1);

return 5;

}

else

{

msgbox(0);

return 5;

}

}

if(mx>=495&&mx<=545&&my>=12&&my<=33&&buttons)

{

return 4;

}

}

}

/////////////////////////////////////////////////

//loadshop7函数说明：当玩家只购买了蓝车时加载的界面及一些后台操作

//参数：p0该玩家用户名 p1为该玩家持有的金币数，但此时为字符串形式

// p2为该玩家持有的车

// p实际代表从主函数传过来的一个标准，\*p=0则玩家此次游戏已经购买了燃油，\*p=1则还未购买

/////////////////////////////////////////////////

int loadshop7(user \*head,char \*p0,char \*p1,char \*p2,int \*p)

{

int buttons, mx, my;

int n;

int mb=3;

int mg=1;

cleardevice();

setviewport(0,0,640,480,1);

drawcar2();

drawgas();

setfillstyle(1,BROWN);

bar(0,0,640,40);

settextstyle(1,0,3);

outtextxy(500,5,"Back");

outtextxy(55,5,"coin");

outtextxy(120,5,p1);

mouseInit(&mx, &my,&buttons);

n=toshuzi(p1);

while (1)

{

newxy(&mx, &my, &buttons);

if(mx>=150&&mx<=250&&my>=350&&my<=370&&buttons)

{

if(n>=mb)

{

changecar(head,p0,"D");

changecoin(head,p0,n-mb);

//outtextxy(192,125,"now you can use the yellow car");

//outtextxy(192,150,"press any key to continue");

//getch();

msgbox(1);

return 5;

}

else

{

//outtextxy(192,125,"your coin is not enough");

//outtextxy(192,150,"press any key to continue");

//getch();

msgbox(0);

return 5;

}

}

if(mx>=80&&mx<=160&&my>=140&&my<=180&&buttons)

{

if(n>=mg)

{

changecoin(head,p0,n-mg);

\*p=0;

msgbox(1);

return 5;

}

else

{

msgbox(0);

return 5;

}

}

if(mx>=495&&mx<=545&&my>=12&&my<=33&&buttons)

{

return 4;

}

}

}

/////////////////////////////////////////////////

//loadshop8函数说明：当玩家只有燃油未购买时加载的界面及一些后台操作

//参数：p0该玩家用户名 p1为该玩家持有的金币数，但此时为字符串形式

// p2为该玩家持有的车

// p实际代表从主函数传过来的一个标准，\*p=0则玩家此次游戏已经购买了燃油，\*p=1则还未购买

/////////////////////////////////////////////////

int loadshop8(user \*head,char \*p0,char \*p1,char \*p2, int \*p )

{

int buttons, mx, my;

int mg=1;

int n;

n=toshuzi(p1);

mouseInit(&mx, &my,&buttons);

cleardevice();

drawgas();

while(1)

{

newxy(&mx, &my, &buttons);

if(mx>=80&&mx<=160&&my>=140&&my<=180&&buttons)

{

if(n>=mg)

{

changecoin(head,p0,n-mg);

\*p=0;

msgbox(1);

return 4;

}

else

{

msgbox(0);

return 4;

}

}

if(mx>=290&&mx<=340&&my>=215&&my<=250&&buttons)

{

return 4;

}

}

}

/////////////////////////////////////////////////

//void drawcar3函数用于画蓝车和它的一些相关信息的背景

/////////////////////////////////////////////////

void drawcar3()

{

int a1[]={415,290,415,280,425,270,450,270,470,250,500,250,520,270,520,290,415,290};

setcolor(BLUE);

setbkcolor(CYAN);

////////////////////////////

//画速度条

setfillstyle(1,WHITE);

settextstyle(3,0,1);

outtextxy(380,322,"Speed");

bar(430,330,535,342);

setfillstyle(1,RED);

bar(430,330,520,342);

////////////////////////////

//画按钮与信息

setfillstyle(1,LIGHTGRAY);

bar(390,350,525,370);

settextstyle(SANS\_SERIF\_FONT ,HORIZ\_DIR,3);

outtextxy(530,345,"$4");

outtextxy(392,345,"Click To Buy");

outtextxy(420,295,"Blue Car");

/////////////////////////////////

//画蓝车

setcolor(DARKGRAY);

drawpoly(9,a1);

arc(445,290,180,360,10);

arc(500,290,180,360,10);

setfillstyle(1,BLUE);

floodfill(440,280,DARKGRAY);

setfillstyle(1,DARKGRAY);

floodfill(445,296,DARKGRAY);

floodfill(500,296,DARKGRAY);

setcolor(BLUE);

}

/////////////////////////////////////////////////

//void drawcar2函数用于画黄车和它的一些相关信息的背景

/////////////////////////////////////////////////

void drawcar2()

{

int a1[]={165,290,165,280,175,270,200,270,220,250,250,250,270,270,270,290,165,290};

setcolor(BLUE);

setbkcolor(CYAN);

////////////////////////////

//画速度条

setfillstyle(1,WHITE);

settextstyle(3,0,1);

outtextxy(130,322,"Speed");

bar(180,330,285,342);

setfillstyle(1,RED);

bar(180,330,245,342);

////////////////////////////

//画按钮与信息

settextstyle(SANS\_SERIF\_FONT ,HORIZ\_DIR,3);

setfillstyle(1,LIGHTGRAY);

bar(140,350,275,370);

outtextxy(280,345,"$3");

outtextxy(142,345,"Click To Buy");

outtextxy(152,295,"Yellow Car");

/////////////////////////

//画黄车

setcolor(DARKGRAY);

drawpoly(9,a1);

arc(195,290,180,360,10);

arc(250,290,180,360,10);

setfillstyle(1,YELLOW);

floodfill(190,280,DARKGRAY);

setfillstyle(1,DARKGRAY);

floodfill(195,296,DARKGRAY);

floodfill(250,296,DARKGRAY);

setcolor(BLUE);

}

///////////////////////////////////////////

//void msgbox函数用来画弹出的提示框

// n=1代表购买成功提示，n=0代表购买失败提示

///////////////////////////////////////////////

void msgbox(int n)

{

int buttons, mx, my;

setfillstyle(1,LIGHTGRAY);

bar(200,100,400,300);

setlinestyle(0,0,3);

rectangle(200,100,400,300);

rectangle(280,270,320,300);

outtextxy(285,270,"ok");

//setlinestyle(1,0,1);

if(n==1)

{

outtextxy(250,150,"Success!");

}

else

{

settextstyle(3,0,1);

outtextxy(220,150,"Coin is not enough");

settextstyle(SANS\_SERIF\_FONT ,HORIZ\_DIR,3);

}

setfillstyle(1,BROWN);

mouseInit(&mx, &my,&buttons);

while(1)

{

newxy(&mx, &my, &buttons);

if(mx>=280&&mx<=320&&my>=270&&my<=300&&buttons)

{

break;

}

}

}

//////////////////////////////////////////////////////////////

//void drawgas函数：用来画燃油的和有关信息

/////////////////////////////////////////////////////////////

void drawgas()

{

int a1[ ]={160,180,260,180,260,50,240,50,220,70,190,70,160,80,160,180};

setcolor(BROWN);

setfillstyle(1,RED);

drawpoly(8,a1);

floodfill(180,120,BROWN);

setcolor(BROWN);

setlinestyle(0,0,3);

line(196,70,210,48);

line(210,48,234,48);

setcolor(BLUE);

setfillstyle(1,BROWN);

rectangle(80,160,140,180);

outtextxy(50,152,"$1");

outtextxy(95,152,"Buy");

}

yonghu.c

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<bios.h>

#include<graphics.h>

#include "jiegouti.h"

#include "yonghu.h"

//////////////////////////////////////////////////

//void userlist函数说明：

//该函数用于将用户信息从文件读入链表中，注意：注册时不能有空格，因为改金币数需要补空格，但金币数并不包含空格，

//所以读入数据时按跳过空格编的代码

//////////////////////////////////////////////////

void userlist(user \*head)

{

user \*current;

FILE \*fp;

char c;

char \*p=NULL;

current=head;

p=current->name;

if ((fp = fopen("..\\TEST1\\user.txt","r+")) == NULL)

{

closegraph();

printf("Can't open user.txt");

getchar();

exit(1);

}

while(!feof(fp))

{

c=fgetc(fp);

if(c=='!')

{

\*p='\0';

p=current->key;

}

else if(c=='@')

{

\*p='\0';

p=current->coin;

}

else if(c=='#')

{

\*p='\0';

p=current->car;

}

else if(c=='$')

{

if((current->next=(user \*)malloc(sizeof(user)))==NULL)

{

closegraph();

printf("\nOverflow");

// getchar();

exit(1);

}

current=current->next;

\*p='\0';

p=current->name;

}

else if(c!=' ')

{

\*p=c;

p++;

}

}

current->next=NULL;

fclose(fp);

}

////////////////////////////////////////////////////

//int searchname函数说明：

//比较注册时用户键入的用户名是否已存在，若存在，则返回0，不存在则返回1

////////////////////////////////////////////////

int searchname( user \*head,char \*p )

{

user \*current=head;

while(current!=NULL)

{

if(strcmp(current->name,p)==0)

return 0;

current=current->next;

}

return 1;

}

///////////////////////////////////////////////////////////

//int judge函数说明

//在登录时，判断用户输入的用户名和密码是否匹配，匹配返回1，否则返回0

////////////////////////////////////

int judge(user \*head,char \*p1,char \*p2)

{

user \*current=head;

while(current!=NULL)

{

if(strcmp(current->name,p1)==0&&strcmp(current->key,p2)==0)

return 1;

current=current->next;

}

return 0;

}

///////////////////////////////////////////////////////////

void newuser(user \*head, char \*p1, char \*p2)

{

user \*current=head;

FILE \*fp;

char \*p;

while(current->next!=NULL)

{

current=current->next;

}

if((current->next=(user \*)malloc(sizeof(user)))==NULL)

{

closegraph();

printf("\nOverflow");

getchar();

exit(1);

}

current=current->next;

strcpy(current->name,p1);

strcpy(current->key,p2);

strcpy(current->coin,"4");

strcpy(current->car,"A");

current->next=NULL;

if ((fp = fopen("..\\TEST1\\user.txt", "r+")) == NULL)

{

closegraph();

printf("Can't open user.txt");

getchar();

exit(1);

}

fseek(fp,0,2);

p=p1;

while(\*p!='\0')

{

putc(\*p,fp);

p++;

}

putc('!',fp);

p=p2;

while(\*p!='\0')

{

putc(\*p,fp);

p++;

}

putc('@',fp);

fprintf(fp,"4 #A$");

fclose(fp); //用完必须关闭文件，否则会出错，且打开状态下很占内存

}

///////////////////////////////////

void tozifu(int number, char \*p)

{

int n=number;

int i=0,j=1;

if(n)

{

while(n)

{

n/=10;

i++;

}

while(number)

{

p[i-j]=number%10+48;

number/=10;

j++;

}

p[i]='\0';

}

else

{ p[0]='0';

p[1]='\0';

}

}

///////////////////////////////////////////////////////

int toshuzi(char \*p)

{

int i=1;

int j;

int sum=0;

j=strlen(p);

while(j-1)

{

i\*=10;

j--;

}

j=0;

while(i)

{

sum+=(p[j]-48)\*i;

j++;

i/=10;

}

return sum;

}

//////////////////////////////////////////

void changecoin(user \*head,char \*p1,int num)

{

user \*current=head;

FILE \*fp;

int i=0;

char pcoin[4];

int length;

char tempname[15];

char ch;

char \*p;

tozifu(num,pcoin);

length = strlen(pcoin);

while(current!=NULL)

{

if(strcmp(current->name,p1)==0)

break;

current=current->next;

}

if ((fp = fopen("..\\TEST1\\user.txt", "r+")) == NULL)

{

closegraph();

printf("Can't open usercode.txt");

getchar();

exit(1);

}

while (!feof(fp))

{

ch = fgetc(fp);

if (ch == '$')

{

ch = fgetc(fp);

for (p = tempname;ch != '!';ch = fgetc(fp))

{

\*p = ch;

\*(p + 1) = '\0';

p++;

}

if (strcmp(tempname, p1) == 0)

{

while(ch!='$')

{ ch=fgetc(fp);}

fseek(fp, -6, SEEK\_CUR);

fputs(pcoin, fp);

for (i = 0;i < 3 - length;i++)

putc(' ', fp);

break;

}

}

}

strcpy(current->coin, pcoin);

fclose(fp);

}

/////////////////////////////////////////////////

void changecar(user \*head,char \*p1,char \*type)

{

user \*current=head;

FILE \*fp;

int length;

char tempname[15];

char ch;

char \*p;

while(current!=NULL)

{

if(strcmp(current->name,p1)==0)

break;

current=current->next;

}

if ((fp = fopen("..\\TEST1\\user.txt", "r+")) == NULL)

{

closegraph();

printf("Can't open usercode.txt");

getch();

exit(1);

}

while (!feof(fp))

{

ch = fgetc(fp);

if (ch == '$')

{

ch = fgetc(fp);

for (p = tempname;ch != '!';ch = fgetc(fp))

{

\*p = ch;

\*(p + 1) = '\0';

p++;

}

if (strcmp(tempname, p1) == 0)

{

while(ch!='$')

{

ch=fgetc(fp);

}

fseek(fp, -2, 1);

fputs(type, fp);

break;

}

}

}

strcpy(current->car,type);

fclose(fp);

}

//////////////////////////////////////////////

void tiqu(user \*head,char \*p0,char \*p1,char \*p2)

{

user \*current=head;

while(current!=NULL)

{

if(strcmp(current->name,p0)==0)

break;

current=current->next;

}

strcpy(p1,current->coin);

strcpy(p2,current->car);

}

zhuce2.c

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<graphics.h>

#include<conio.h>

#include<dos.h>

#include<bios.h>

#include "mouse1.h"

#include "jiemian2.h"

#include "caidan1.h"

#include "zhuce2.h"

#include "jiegouti.h"

int getkey1()

{

int key;

key=bioskey(0);

if(key<<8)

{

key=key&0x00ff;

}

return key;

}

///////////////////////////////////////////

int loadzhuce(user \*head,char \*p0)

{

void \*buffer0;

int size;

int buttons, mx, my;

int j=0;

char temp;

char pkey[10];

cleardevice();

setviewport(0,0,640,480,1);

setcolor(BLUE);

setbkcolor(CYAN);

//setlinestyle(2,0,3);

setfillstyle(1,LIGHTGRAY);

outtextxy(265,195,"register");

bar(150,330,250,350);

bar(390,330,490,350);

outtextxy(172,295,"Name");

outtextxy(420,295,"Key");

outtextxy(500,5,"Back");

mouseInit(&mx, &my,&buttons);

size=imagesize(150,300,255,365);

buffer0=malloc(size);

getimage(150,300,255,365,buffer0);

while (1)

{

if(kbhit()!=0)

{

temp=getkey1();

}

newxy(&mx, &my, &buttons);

if(mx>=150&&mx<=250&&my>=330&&my<=350&&buttons)

{

inputname(p0,buffer0);

if(searchname(head,p0))

{

printf("success");

inputkey(pkey);

newuser(head,p0,pkey);

outtextxy(10,10,"register success");

getch();

return 4;

}

else

outtextxy(10,10,"the name has been used");

outtextxy(10,30,"press any key to continue");

getch();

return 2;

}

if(mx>=495&&mx<=545&&my>=12&&my<=33&&buttons)

{

return 1;

}

}

}

///////////////////////////

//////////////////////////////////////////////////////////////////

void inputname(char \*name,void \*buffer0)

{

int k=0;

FILE \*fp;

//int buttons, mx, my;

void \*buffer1;

int size,size1;

//void \*buffer2;

//char name[11];

//char \*p=NULL;

unsigned int key;

char c[2]={'\0','\0'};

int j=0;

int i=0;

//int flag=0;

// setcolor(BLACK);

putimage(150,300,buffer0,COPY\_PUT); //这句话一定要放在后面的getimage之前，否则可能出现bug

size=imagesize(151,330,162,350);

//size1=imagesize(151,330,249,350);

buffer1=malloc(size);

//buffer2=malloc(size1);

getimage(151,330,162,350,buffer1);

//getimage(151,350,162,370,buffer2);

//putimage(150,300,buffer0,COPY\_PUT);

setlinestyle(0,0,3);

setcolor(YELLOW);

rectangle(148,328,252,352);

setcolor(YELLOW);

while(1)

{

key=getkey1();

if(key==0x0008)

{

if (j>0&&j<=8)

{

name[j-1]='\0';

j--;

//outtextxy(152+12\*j,325," ");//空格并不能将原来的东西覆盖

putimage(152+12\*j,330,buffer1,COPY\_PUT);

//flag=1;

}

if(j>7)

{

j--;

}

}

else if(key==0x000d)

break;

else

{

if(j>7)

{

j++;

}

else

{

c[0]=key;

if(c[0]=='i'||c[0]=='j'||c[0]=='r'||c[0]=='t'||c[0]=='l')//因为rltj这些字母所占空间偏左，按相同方式显示会不好看

{

outtextxy(152+5+12\*j,323,c);

}

else

{

outtextxy(152+12\*j,323,c);

}

name[j]=c[0];

name[j+1]='\0';

j++;

}

}

}

name[j]='\0';

//此时已经完成name的输入

free(buffer1);//很重要，因为对于malloc分配的空间，在函数结束后不会自动消失，所以需要free掉

free(buffer0); //防止多次调用后爆了

}

//////////////////////////////////////////////////////

void inputkey(char \*key)

{

void \*buffer1;

int size;

int k=0;

FILE \*fp;

int key1;

//char key[10];

//char \*p=NULL;

//struct user b[100];

char c[2]={'\0','\0'};

int j=0;

//bar(391,300,392,350);

size=imagesize(393,330,404,350);

buffer1=malloc(size);

getimage(393,330,404,350,buffer1);

//putimage(150,300,buffer0,COPY\_PUT);

setcolor(CYAN);

rectangle(148,328,252,352);

setcolor(YELLOW);

rectangle(388,328,492,352);

while(1)

{

key1=getkey1();

if(key1==0x0008)

{

if (j>0&&j<=8)

{

key[j-1]='\0';

j--;

//outtextxy(152+12\*j,325," "); //空格并不能将原来的东西覆盖

putimage(392+12\*j,330,buffer1,COPY\_PUT);

}

else if (j>7)

{

j--;

}

}

else if(key1==0x000d)

break;

else

{

if(j>7)

{

j++;

}

else

{

c[0]=key1;

if(c[0]=='i'||c[0]=='j'||c[0]=='r'||c[0]=='t'||c[0]=='l')//因为rltj这些字母所占空间偏左，按相同方式显示会不好看

{

outtextxy(392+5+12\*j,323,c);

}

else

{

outtextxy(392+12\*j,323,c);

}

key[j]=c[0];

j++;

}

}

}

key[j]='\0';

free(buffer1);

}

jiemian2.c

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<graphics.h>

#include<conio.h>

#include<math.h>

#include<dos.h>

#include<bios.h>

#include "mouse1.h"

#include "zhuce2.h"

#include "denglu1.h"

#include "jiemian2.h"

#include "jiegouti.h"

#define PI 3.1415926

void jiazai()

{

int i;

settextstyle(1,0,5);

setbkcolor(BLACK);

cleardevice();

setfillstyle(1,WHITE);

bar(100,300,550,330);

setfillstyle(1,RED);

outtextxy(230,200,"Loading");

for(i=0;i<450;i++)

{

bar(100+i,300,101+i,330);

delay(10);

}

}

int jiemian1()

{

char temp;

int buttons, mx, my;

//jiazai();

setcolor(BLUE);

setbkcolor(CYAN);

cleardevice();

//rectangle(0,0,640,480);

setfillstyle(1,BROWN);

bar(150,300,250,350);

bar(390,300,490,350);

settextstyle(3,0,7);

outtextxy(200,150,"welcome");

settextstyle(SANS\_SERIF\_FONT ,HORIZ\_DIR,3);

outtextxy(162,308,"Register");

outtextxy(410,308,"Log In");

outtextxy(500,5,"Quit");

//settextstyle(1,HORIZ\_DIR,1);

mouseInit(&mx, &my,&buttons);

while (1)

{

newxy(&mx, &my, &buttons);

if(kbhit()!=0)

{

temp=getkey();

}

if(mx>=390&&mx<=490&&my>=300&&my<=350&&buttons)

{

return 3;

}

if(mx>=150&&mx<=250&&my>=300&&my<=350&&buttons)

{

return 2;

}

if(mx>=495&&mx<=545&&my>=12&&my<=33&&buttons)

{

cleardevice();

delay(100);

exit(0);

}

}

}

denglu1.c

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<graphics.h>

#include<conio.h>

#include<dos.h>

#include<bios.h>

#include "zhuce2.h"

#include "mouse1.h"

#include "caidan1.h"

#include "denglu1.h"

#include "jiegouti.h"

#include "jiemian2.h"

int loaddenglu(user \*head,char \*p0)

{

void \*buffer0;

int size;

int buttons, mx, my;

int j=0;

//char pname[10];

char pkey[10];

int k=0;

char temp;

FILE \*fp;

// struct user a;

// struct user b[100];

char c[2]={'\0','\0'};

int i=0;

//int sum=0;

cleardevice();

setviewport(0,0,640,480,1);

setcolor(BLUE);

setbkcolor(CYAN);

// setlinestyle(2,0,3);

setfillstyle(1,LIGHTGRAY);

bar(150,330,250,350);

bar(390,330,490,350);

outtextxy(285,195,"Log In");

outtextxy(172,295,"Name");

outtextxy(420,295,"Key");

outtextxy(500,5,"Back");

mouseInit(&mx, &my,&buttons);

size=imagesize(150,300,255,365);

buffer0=malloc(size);

getimage(150,300,255,365,buffer0);

while (1)

{

if(kbhit()!=0)

{

temp=getkey1();

}

newxy(&mx, &my, &buttons);

if(mx>=150&&mx<=250&&my>=330&&my<=350&&buttons)

{

inputname(p0,buffer0);

inputkey(pkey);

if(judge(head,p0,pkey))

{

//printf("666");

outtextxy(10,10,"log in success");

outtextxy(10,30,"press any key to continue");

getch();

//zhucaidan(head,pname);

return 4;

}

else

outtextxy(10,10,"wrong number");

outtextxy(10,30,"press any key to continue");

getch();

return 3;

}

/\*if(mx>=150&&mx<=250&&my>=300&&my<=350&&buttons)

{

inputkey();

}\*/

if(mx>=495&&mx<=545&&my>=12&&my<=33&&buttons)

{

return 1;

}

}

}

chaxun.c

#include<stdio.h>

#include<string.h>

#include<stdlib.h>

#include<graphics.h>

#include<conio.h>

#include<dos.h>

#include<bios.h>

#include "mouse1.h"

#include "jiegouti.h"

#include "yonghu.h"

#include "chaxun.h"

#include "record.h"

/////////////////////////////////////////////

//int loadchaxun函数说明：

//该函数用于加载查询初始界面，并根据用户点击情况进一步加载查询

//mark：用来标识是否需要清屏，若进入了下一级界面，则返回后mark=1，说明下一次循环时需要清屏重画，否则一直清屏会很闪

/////////////////////////////

int loadchaxun(record \*head, char \*p1)

{

int buttons, mx, my;

int mark=0;

huacha();

mouseInit(&mx, &my,&buttons);

while(1)

{

if(mark==1)

{

huacha();

mark=0;

mouseInit(&mx, &my,&buttons);

}

newxy(&mx, &my, &buttons);

if(mx>=100&&mx<=230&&my>=100&&my<=130&&buttons)

{

chaxun(head,p1,1,0);

mark=1;

//return 4;

}

if(mx>=100&&mx<=180&&my>=200&&my<=230&&buttons)

{

chaxun(head,p1,2,0);

mark=1;

//return 4;

}

if(mx>=210&&mx<=290&&my>=285&&my<=310&&buttons)

{

chaxun(head,p1,3,1);

mark=1;

//return 4;

}

if(mx>=210&&mx<=310&&my>=320&&my<=345&&buttons)

{

chaxun(head,p1,3,2);

mark=1;

//return 4;

}

if(mx>=210&&mx<=310&&my>=370&&my<=395&&buttons)

{

chaxun(head,p1,4,1);

mark=1;

//return 4;

}

if(mx>=210&&mx<=310&&my>=405&&my<=430&&buttons)

{

chaxun(head,p1,4,2);

mark=1;

//return 4;

}

if(mx>=210&&mx<=310&&my>=440&&my<=465&&buttons)

{

chaxun(head,p1,4,3);

mark=1;

//return 4;

}

if(mx>=495&&mx<=545&&my>=12&&my<=33&&buttons)

{

return 4;

}

}

}

void chaxun(record \*head, char \*p1,int n1,int n2)

{

int buttons, mx, my;

record b[15];

int i,j,n;

cleardevice();

outtextxy(500,5,"Back");

mouseInit(&mx, &my,&buttons);

outtextxy(50,50,"User");

outtextxy(200,50,"Track");

outtextxy(285,50,"Car");

outtextxy(355,50,"Time");

outtextxy(450,50,"Rank");

if(n1==1)

{

searchname1(head,p1,b);

}

else if(n1==2)

{

n=searchall(head,p1,b);

sort1(b,n);

}

else if(n1==3)

{

searchtrack(head,n2,b);

}

else if (n1==4)

{

searchcar(head,n2,b);

}

for(i=0;i<10&&strcmp(b[i].jiluname,"\0")!=0;i++)

{

outtextxy(50,100+30\*i,b[i].jiluname);

outtextxy(200,100+30\*i,b[i].jilutrack);

outtextxy(295,100+30\*i,b[i].jilucar);

outtextxy(370,100+30\*i,b[i].jilutime);

outtextxy(470,100+30\*i,b[i].jilurank);

}

while(1)

{

newxy(&mx, &my, &buttons);

if(mx>=495&&mx<=545&&my>=12&&my<=33&&buttons)

{

break;

}

}

}

void huacha()

{

cleardevice();

settextstyle(1,0,3);

rectangle(100,100,230,130);

outtextxy(105,100,"My Record");

rectangle(100,200,180,230);

outtextxy(105,200,"TOP10");

rectangle(100,300,180,330);

outtextxy(105,300,"TRACK");

rectangle(100,400,150,430);

outtextxy(105,400,"CAR");

outtextxy(500,5,"Back");

rectangle(210,285,290,310);

outtextxy(215,280,"Square");

rectangle(210,320,310,345);

outtextxy(215,315,"Triangle");

rectangle(210,370,290,395);

outtextxy(215,365,"RED");

rectangle(210,405,310,430);

outtextxy(215,400,"YELLOW");

rectangle(210,440,290,465);

outtextxy(215,435,"BLUE");

}

record.c

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<bios.h>

#include<graphics.h>

#include "jiegouti.h"

#include "yonghu.h"

#include "record.h"

////////////////////////////////////

//函数说明：建立存储比赛所以记录信息的链表，从文件中读取数据装入链表

////////////////////////////////////

void recordlist(record \*head)

{

record \*current;

FILE \*fp;

char c;

char \*p=NULL;

current=head;

p=current->jiluname;

if ((fp = fopen("..\\TEST1\\record.txt","r+")) == NULL)

{

closegraph();

printf("Can't open record.txt");

getchar();

exit(1);

}

while(!feof(fp))

{

c=fgetc(fp);

if(c=='!')

{

\*p='\0';

p=current->jilutrack;

}

else if(c=='@')

{

\*p='\0';

p=current->jilucar;

}

else if(c=='#')

{

\*p='\0';

p=current->jilutime;

}

else if(c=='~')

{

\*p='\0';

p=current->jilurank;

}

else if(c=='$')

{

if((current->next=(record \*)malloc(sizeof(record)))==NULL)

{

closegraph();

printf("\nOverflow");

// getchar();

exit(1);

}

//printf("%s",current->number);////

current=current->next;

\*p='\0';

p=current->jiluname;

}

else if(c!=' ')

{

\*p=c;

p++;

}

}

current->next=NULL; //重要！！！

//current指针置为0没有意义，必须让current->next这个在结构体中的指针指向NULL才能使链表结束

// getch();

fclose(fp);

}

////////////////////////////////////////////////

//voidnewrecord函数说明：用于比赛完成后，将本场比赛的信息加入链表，并同时将信息写入文件

///////////////////////////////////////////////

void newrecord(record \*head, char \*p1,int p2,int p3,int p4,int p5)

{

record \*current=head;

FILE \*fp;

char \*p;

char track1[10];

char car1[10];

char time1[5];

char rank1[3];

//tozifu(p2,track1);

//tozifu(p3,car1);

tozifu(p4,time1);

tozifu(p5,rank1);

//char pnum[3];

//int l;

//int length2=strlen(p2);

//int i = 1;

if(p2==1)

stpcpy(track1,"Square");

if(p2==2)

stpcpy(track1,"Triangle");

if(p3==1)

stpcpy(car1,"Red");

if(p3==2)

stpcpy(car1,"Yellow");

if(p3==3)

stpcpy(car1,"Blue");

while(current->next->next!=NULL) //重要！！！在创建链表时由于文件读写特点，最后一个节点为空

{

current=current->next;

//i++;

}

current->next=NULL; //重要！！！须将之前创建链表时出现的空节点填补上，做到无缝衔接

if((current->next=(record \*)malloc(sizeof(user)))==NULL)

{

closegraph();

printf("\nOverflow");

getchar();

exit(1);

}

//给链表节点赋值

current=current->next;

strcpy(current->jiluname,p1);

strcpy(current->jilutrack,track1);

strcpy(current->jilucar,car1);

strcpy(current->jilutime,time1);

strcpy(current->jilurank,rank1);

////////

current->next=NULL;

if ((fp = fopen("..\\TEST1\\record.txt", "r+")) == NULL)

{

closegraph();

printf("Can't open record.txt");

getchar();

exit(1);

}

fseek(fp,0,2);

p=p1;

while(\*p!='\0')

{

putc(\*p,fp);

p++;

}

putc('!',fp);

p=current->jilutrack;

while(\*p!='\0')

{

putc(\*p,fp);

p++;

}

putc('@',fp);

p=current->jilucar;

while(\*p!='\0')

{

putc(\*p,fp);

p++;

}

putc('#',fp);

p=current->jilutime;

while(\*p!='\0')

{

putc(\*p,fp);

p++;

}

putc('~',fp);

p=current->jilurank;

while(\*p!='\0')

{

putc(\*p,fp);

p++;

}

putc('$',fp);

fclose(fp);

//////////////////////

//看似已经完成加入记录操作，但此时还需要在链表尾部加一个空节点，

//为了保证和创建链表时同样有一个空节点，这样才能使提取元素时判断条件统一

//否则可能导致多提取或少提取一个元素，下面操作为了在链表尾部加入一个空节点

///////////////////////

if((current->next=(record \*)malloc(sizeof(user)))==NULL)

{

closegraph();

printf("\nOverflow");

getchar();

exit(1);

}

current=current->next; //重要！！

current->next=NULL;

}

//////////////////////////////////////////////

//void searchname1函数说明：

//在链表中搜索符合用户名的比赛记录，提取出来，存入一个结构数组

//////////////////////////////////////////////

void searchname1(record \*head, char \*p1,record \*p2)

{

record \*current=head;

int i=0;

while(current!=NULL)

{

if(strcmp(current->jiluname,p1)==0)//因为链表最后一个结点为空，所以提前一个节点结束提取

{

stpcpy(p2[i].jiluname,current->jiluname);

stpcpy(p2[i].jilutrack,current->jilutrack);

stpcpy(p2[i].jilucar,current->jilucar);

stpcpy(p2[i].jilutime,current->jilutime);

stpcpy(p2[i].jilurank,current->jilurank);

i++;

}

p2[i].jiluname[0]='\0'; //不需要的下一个数组的name设为'\0'，在查询中可以用来判断要输出的个数

current=current->next;

}

}

////////////////////////////////////////////

//int searchall函数说明

//该函数用于提取每一个比赛记录信息存入传入的结构数组，并将比赛信息数量作为返回值

////////////////////////////////////////////

int searchall(record \*head, char \*p1,record \*p2)

{

record \*current=head;

int i=0;

while(current->next!=NULL) //重要！！！

{

stpcpy(p2[i].jiluname,current->jiluname);

stpcpy(p2[i].jilutrack,current->jilutrack);

stpcpy(p2[i].jilucar,current->jilucar);

stpcpy(p2[i].jilutime,current->jilutime);

stpcpy(p2[i].jilurank,current->jilurank);

i++;

p2[i].jiluname[0]='\0';

current=current->next;

}

return i;

}

//////////////////////////////////////////////

//void sort1函数说明：

//该函数将传入的结构数组按时间从小到大排序，其中n为此数组有意义的元素的个数

////////////////////////////////////////////

void sort1(record \*p,int n)

{

int i,j,k1,k2,k;

record c;

for(i=0;i<n-1;i++)

{ for(j=0;j<n-i-1;j++)

{

k1=toshuzi(p[j].jilutime);

k2=toshuzi(p[j+1].jilutime);

if(k1>k2)

{

/\*strcpy(c.jilutime,p[j].jilutime);

strcpy(p[j].jilutime,p[j+1].jilutime);

strcpy(p[j+1].jilutime,c.jilutime);\*/

c=p[j];

p[j]=p[j+1];

p[j+1]=c;

}

}

}

// return n;

}

////////////////////////////////////////////

//void searchtrack函数说明：

//参数p1=1代表提取Square型赛道的记录，

// p1=2代表提取triangle型赛道的记录

/////////////////////////////////////////////

void searchtrack(record \*head, int p1,record \*p2)

{

record \*current=head;

int i=0;

char track[10];

if(p1==1)

stpcpy(track,"Square");

if(p1==2)

stpcpy(track,"Triangle");

while(current!=NULL)

{

if(strcmp(current->jilutrack,track)==0)

{

stpcpy(p2[i].jiluname,current->jiluname);

stpcpy(p2[i].jilutrack,current->jilutrack);

stpcpy(p2[i].jilucar,current->jilucar);

stpcpy(p2[i].jilutime,current->jilutime);

stpcpy(p2[i].jilurank,current->jilurank);

i++;

}

p2[i].jiluname[0]='\0';

current=current->next;

}

}

////////////////////////////////////////////////////////

void searchcar(record \*head, int p1,record \*p2)

{

record \*current=head;

int i=0;

char car[10];

if(p1==1)

stpcpy(car,"Red");

if(p1==2)

stpcpy(car,"Yellow");

if(p1==3)

stpcpy(car,"Blue");

while(current!=NULL)

{

if(strcmp(current->jilucar,car)==0)

{

stpcpy(p2[i].jiluname,current->jiluname);

stpcpy(p2[i].jilutrack,current->jilutrack);

stpcpy(p2[i].jilucar,current->jilucar);

stpcpy(p2[i].jilutime,current->jilutime);

stpcpy(p2[i].jilurank,current->jilurank);

i++;

}

p2[i].jiluname[0]='\0';

current=current->next;

}

}

/////////////////////////////////////////////////////////

void searchrank(record \*head, int p1,record \*p2)

{

record \*current=head;

int i=0;

char rank[3];

tozifu(p1,rank);

while(current!=NULL)

{

if(strcmp(current->jiluname,rank)==0)

{

stpcpy(p2[i].jiluname,current->jiluname);

stpcpy(p2[i].jilutrack,current->jilutrack);

stpcpy(p2[i].jilucar,current->jilucar);

stpcpy(p2[i].jilutime,current->jilutime);

stpcpy(p2[i].jilurank,current->jilurank);

i++;

}

//return 0;

current=current->next;

}

}

result.c

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<bios.h>

#include<graphics.h>

#include "jiegouti.h"

#include "yonghu.h"

#include "record.h"

////////////////////////////////////

//函数说明：建立存储比赛所以记录信息的链表，从文件中读取数据装入链表

////////////////////////////////////

void recordlist(record \*head)

{

record \*current;

FILE \*fp;

char c;

char \*p=NULL;

current=head;

p=current->jiluname;

if ((fp = fopen("..\\TEST1\\record.txt","r+")) == NULL)

{

closegraph();

printf("Can't open record.txt");

getchar();

exit(1);

}

while(!feof(fp))

{

c=fgetc(fp);

if(c=='!')

{

\*p='\0';

p=current->jilutrack;

}

else if(c=='@')

{

\*p='\0';

p=current->jilucar;

}

else if(c=='#')

{

\*p='\0';

p=current->jilutime;

}

else if(c=='~')

{

\*p='\0';

p=current->jilurank;

}

else if(c=='$')

{

if((current->next=(record \*)malloc(sizeof(record)))==NULL)

{

closegraph();

printf("\nOverflow");

// getchar();

exit(1);

}

//printf("%s",current->number);////

current=current->next;

\*p='\0';

p=current->jiluname;

}

else if(c!=' ')

{

\*p=c;

p++;

}

}

current->next=NULL; //重要！！！

//current指针置为0没有意义，必须让current->next这个在结构体中的指针指向NULL才能使链表结束

// getch();

fclose(fp);

}

////////////////////////////////////////////////

//voidnewrecord函数说明：用于比赛完成后，将本场比赛的信息加入链表，并同时将信息写入文件

///////////////////////////////////////////////

void newrecord(record \*head, char \*p1,int p2,int p3,int p4,int p5)

{

record \*current=head;

FILE \*fp;

char \*p;

char track1[10];

char car1[10];

char time1[5];

char rank1[3];

//tozifu(p2,track1);

//tozifu(p3,car1);

tozifu(p4,time1);

tozifu(p5,rank1);

//char pnum[3];

//int l;

//int length2=strlen(p2);

//int i = 1;

if(p2==1)

stpcpy(track1,"Square");

if(p2==2)

stpcpy(track1,"Triangle");

if(p3==1)

stpcpy(car1,"Red");

if(p3==2)

stpcpy(car1,"Yellow");

if(p3==3)

stpcpy(car1,"Blue");

while(current->next->next!=NULL) //重要！！！在创建链表时由于文件读写特点，最后一个节点为空

{

current=current->next;

//i++;

}

current->next=NULL; //重要！！！须将之前创建链表时出现的空节点填补上，做到无缝衔接

if((current->next=(record \*)malloc(sizeof(user)))==NULL)

{

closegraph();

printf("\nOverflow");

getchar();

exit(1);

}

//给链表节点赋值

current=current->next;

strcpy(current->jiluname,p1);

strcpy(current->jilutrack,track1);

strcpy(current->jilucar,car1);

strcpy(current->jilutime,time1);

strcpy(current->jilurank,rank1);

////////

current->next=NULL;

if ((fp = fopen("..\\TEST1\\record.txt", "r+")) == NULL)

{

closegraph();

printf("Can't open record.txt");

getchar();

exit(1);

}

fseek(fp,0,2);

p=p1;

while(\*p!='\0')

{

putc(\*p,fp);

p++;

}

putc('!',fp);

p=current->jilutrack;

while(\*p!='\0')

{

putc(\*p,fp);

p++;

}

putc('@',fp);

p=current->jilucar;

while(\*p!='\0')

{

putc(\*p,fp);

p++;

}

putc('#',fp);

p=current->jilutime;

while(\*p!='\0')

{

putc(\*p,fp);

p++;

}

putc('~',fp);

p=current->jilurank;

while(\*p!='\0')

{

putc(\*p,fp);

p++;

}

putc('$',fp);

fclose(fp);

//////////////////////

//看似已经完成加入记录操作，但此时还需要在链表尾部加一个空节点，

//为了保证和创建链表时同样有一个空节点，这样才能使提取元素时判断条件统一

//否则可能导致多提取或少提取一个元素，下面操作为了在链表尾部加入一个空节点

///////////////////////

if((current->next=(record \*)malloc(sizeof(user)))==NULL)

{

closegraph();

printf("\nOverflow");

getchar();

exit(1);

}

current=current->next; //重要！！

current->next=NULL;

}

//////////////////////////////////////////////

//void searchname1函数说明：

//在链表中搜索符合用户名的比赛记录，提取出来，存入一个结构数组

//////////////////////////////////////////////

void searchname1(record \*head, char \*p1,record \*p2)

{

record \*current=head;

int i=0;

while(current!=NULL)

{

if(strcmp(current->jiluname,p1)==0)//因为链表最后一个结点为空，所以提前一个节点结束提取

{

stpcpy(p2[i].jiluname,current->jiluname);

stpcpy(p2[i].jilutrack,current->jilutrack);

stpcpy(p2[i].jilucar,current->jilucar);

stpcpy(p2[i].jilutime,current->jilutime);

stpcpy(p2[i].jilurank,current->jilurank);

i++;

}

p2[i].jiluname[0]='\0'; //不需要的下一个数组的name设为'\0'，在查询中可以用来判断要输出的个数

current=current->next;

}

}

////////////////////////////////////////////

//int searchall函数说明

//该函数用于提取每一个比赛记录信息存入传入的结构数组，并将比赛信息数量作为返回值

////////////////////////////////////////////

int searchall(record \*head, char \*p1,record \*p2)

{

record \*current=head;

int i=0;

while(current->next!=NULL) //重要！！！

{

stpcpy(p2[i].jiluname,current->jiluname);

stpcpy(p2[i].jilutrack,current->jilutrack);

stpcpy(p2[i].jilucar,current->jilucar);

stpcpy(p2[i].jilutime,current->jilutime);

stpcpy(p2[i].jilurank,current->jilurank);

i++;

p2[i].jiluname[0]='\0';

current=current->next;

}

return i;

}

//////////////////////////////////////////////

//void sort1函数说明：

//该函数将传入的结构数组按时间从小到大排序，其中n为此数组有意义的元素的个数

////////////////////////////////////////////

void sort1(record \*p,int n)

{

int i,j,k1,k2,k;

record c;

for(i=0;i<n-1;i++)

{ for(j=0;j<n-i-1;j++)

{

k1=toshuzi(p[j].jilutime);

k2=toshuzi(p[j+1].jilutime);

if(k1>k2)

{

/\*strcpy(c.jilutime,p[j].jilutime);

strcpy(p[j].jilutime,p[j+1].jilutime);

strcpy(p[j+1].jilutime,c.jilutime);\*/

c=p[j];

p[j]=p[j+1];

p[j+1]=c;

}

}

}

// return n;

}

////////////////////////////////////////////

//void searchtrack函数说明：

//参数p1=1代表提取Square型赛道的记录，

// p1=2代表提取triangle型赛道的记录

/////////////////////////////////////////////

void searchtrack(record \*head, int p1,record \*p2)

{

record \*current=head;

int i=0;

char track[10];

if(p1==1)

stpcpy(track,"Square");

if(p1==2)

stpcpy(track,"Triangle");

while(current!=NULL)

{

if(strcmp(current->jilutrack,track)==0)

{

stpcpy(p2[i].jiluname,current->jiluname);

stpcpy(p2[i].jilutrack,current->jilutrack);

stpcpy(p2[i].jilucar,current->jilucar);

stpcpy(p2[i].jilutime,current->jilutime);

stpcpy(p2[i].jilurank,current->jilurank);

i++;

}

p2[i].jiluname[0]='\0';

current=current->next;

}

}

////////////////////////////////////////////////////////

void searchcar(record \*head, int p1,record \*p2)

{

record \*current=head;

int i=0;

char car[10];

if(p1==1)

stpcpy(car,"Red");

if(p1==2)

stpcpy(car,"Yellow");

if(p1==3)

stpcpy(car,"Blue");

while(current!=NULL)

{

if(strcmp(current->jilucar,car)==0)

{

stpcpy(p2[i].jiluname,current->jiluname);

stpcpy(p2[i].jilutrack,current->jilutrack);

stpcpy(p2[i].jilucar,current->jilucar);

stpcpy(p2[i].jilutime,current->jilutime);

stpcpy(p2[i].jilurank,current->jilurank);

i++;

}

p2[i].jiluname[0]='\0';

current=current->next;

}

}

/////////////////////////////////////////////////////////

void searchrank(record \*head, int p1,record \*p2)

{

record \*current=head;

int i=0;

char rank[3];

tozifu(p1,rank);

while(current!=NULL)

{

if(strcmp(current->jiluname,rank)==0)

{

stpcpy(p2[i].jiluname,current->jiluname);

stpcpy(p2[i].jilutrack,current->jilutrack);

stpcpy(p2[i].jilucar,current->jilucar);

stpcpy(p2[i].jilutime,current->jilutime);

stpcpy(p2[i].jilurank,current->jilurank);

i++;

}

//return 0;

current=current->next;

}

}

mouse1.c

#include "mouse1.h"

#include<graphics.h>

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<dos.h>

union REGS regs;

int arrowMouse[10][16] = {

{ 1,1,1,1,1,1,1,1,1,1,1,1,1,3,3,3 },

{ 1,0,0,0,0,0,0,0,0,0,0,1,3,3,3,3 },

{ 3,1,0,0,0,0,0,0,0,0,1,3,3,3,3,3 },

{ 3,3,1,0,0,0,0,0,0,1,3,3,3,3,3,3 },

{ 3,3,3,1,0,0,0,0,0,0,1,1,3,3,3,3 },

{ 3,3,3,3,1,0,0,0,0,0,0,0,1,1,3,3 },

{ 3,3,3,3,3,1,0,0,1,1,1,0,0,0,1,3 },

{ 3,3,3,3,3,3,1,0,1,3,3,1,0,0,1,3 },

{ 3,3,3,3,3,3,3,1,1,3,3,3,1,1,3,3 },

{ 3,3,3,3,3,3,3,3,1,3,3,3,3,3,3,3 },

};

int AddMouse[10][16]={

{ 3,3,3,3,3,3,1,1,1,1,1,3,3,3,3,3 },

{ 3,3,3,3,3,3,1,0,0,0,1,3,3,3,3,3 },

{ 3,3,3,3,3,3,1,0,0,0,1,3,3,3,3,3 },

{ 3,3,1,1,1,1,1,0,0,0,1,1,1,1,3,3 },

{ 3,3,1,0,0,0,0,0,0,0,0,0,0,1,3,3 },

{ 3,3,1,0,0,0,0,0,0,0,0,0,0,1,3,3 },

{ 3,3,1,1,1,1,1,0,0,0,1,1,1,1,3,3 },

{ 3,3,3,3,3,3,1,0,0,0,1,3,3,3,3,3 },

{ 3,3,3,3,3,3,1,0,0,0,1,3,3,3,3,3 },

{ 3,3,3,3,3,3,1,1,1,1,1,3,3,3,3,3 },

};

int Mouse[10][16] = {

{ 1,1,1,1,1,1,1,1,1,1,1,1,1,3,3,3 },

{ 1,0,0,0,0,0,0,0,0,0,0,1,3,3,3,3 },

{ 3,1,0,0,0,0,0,0,0,0,1,3,3,3,3,3 },

{ 3,3,1,0,0,0,0,0,0,1,3,3,3,3,3,3 },

{ 3,3,3,1,0,0,0,0,0,0,1,1,3,3,3,3 },

{ 3,3,3,3,1,0,0,0,0,0,0,0,1,1,3,3 },

{ 3,3,3,3,3,1,0,0,1,1,1,0,0,0,1,3 },

{ 3,3,3,3,3,3,1,0,1,3,3,1,0,0,1,3 },

{ 3,3,3,3,3,3,3,1,1,3,3,3,1,1,3,3 },

{ 3,3,3,3,3,3,3,3,1,3,3,3,3,3,3,3 },

};

int MouseSave[10][16] = {0};

/\*mark为0设置为箭头鼠标,mark为1设置为加号鼠标\*/

void setMouseShape(int mark,int mx,int my)

{

int i;

int j;

if (mark == 0)

{

for(i=0;i<10;i++)

for (j = 0;j < 16;j++)

Mouse[i][j] = arrowMouse[i][j];

}

else if (mark == 1)

{

for (i = 0;i<10;i++)

for (j = 0;j < 16;j++)

Mouse[i][j] = AddMouse[i][j];

}

else

{

printf("MouseShape doesn't exit!");

getchar();

exit(1);

}

mousehide(mx,my);

cursor(mx, my);

}

void cursor(int x, int y)

{

int i, j;

for (i = 0;i<10;i++)

for (j = 0;j<16;j++)

{

if (Mouse[i][j] == 0)

putpixel(x + i, y + j, WHITE);

else if (Mouse[i][j] == 1)

putpixel(x + i, y + j, BROWN);

}

}

void getMousebk(int x, int y)

{

int i, j;

for(i=0;i<10;i++)

for(j=0;j<16;j++)

MouseSave[i][j] = getpixel(x + i, y + j);

}

void mousehide(int x, int y)

{

int i, j;

for (i = 0;i<10;i++)

for (j = 0;j<16;j++)

{

putpixel(x + i, y + j, MouseSave[i][j]);

}

}

int init(int xmi, int xma, int ymi, int yma) //鼠标器初始化操作

{

int retcode;

regs.x.ax = 0;

int86(51, &regs, &regs);

retcode = regs.x.ax;

if (retcode == 0)

return 0;

regs.x.ax = 7;

regs.x.cx = xmi;

regs.x.dx = xma;

int86(51, &regs, &regs);

regs.x.ax = 8;

regs.x.cx = ymi;

regs.x.dx = yma;

int86(51, &regs, &regs);

return retcode;

}

void mouseInit(int \*mx,int \*my, int \*mbutt)

{

if (init(2, 638, 8, 477) == 0)

{

closegraph();

printf("Mouse or Mouse Driver Absent,Please Install");

delay(5000);

exit(1);

}

\*mx = 3;

\*my = 460;

\*mbutt = 0;

getMousebk(\*mx, \*my);

cursor(\*mx, \*my);

}

int read(int \*mx, int \*my, int \*mbutt) //读取鼠标的位置

{

static int mark = 0; //按键按松开标志

int xx0 = \*mx, yy0 = \*my, buto = \*mbutt;

int xnew, ynew;

do

{

regs.x.ax = 3;

int86(51, &regs, &regs);

xnew = regs.x.cx;

ynew = regs.x.dx;

if (mark == 0 && regs.x.bx != 0)

{

mark = 1;

//delay(10);

if(regs.x.bx != 0)\*mbutt = regs.x.bx;

}

else if (regs.x.bx == 0)

{

mark = 0;

\*mbutt = 0;

}

else \*mbutt = 0;

} while (xnew == xx0&&ynew == yy0&&\*mbutt == buto);

\*mx = xnew;

\*my = ynew;

if (\*mbutt)

{

\*mx = xnew;

\*my = ynew;

return -1;

}

else

{

\*mx = xnew;

\*my = ynew;

return 1;

}

}

void newxy(int \*mx, int \*my, int \*mbutt) //在新的位置处画鼠标

{

static int i = 0;

int ch, xx0 = \*mx, yy0 = \*my;

int xm, ym;

read(&xm, &ym, mbutt);

if (xm != xx0 || ym != yy0)

{

mousehide(xx0, yy0);

getMousebk(xm, ym);

cursor(xm, ym);

\*mx = xm;

\*my = ym;

}

}

void backgroundChange(int mx, int my,int x1,int y1,int x2,int y2)

{

int i, j;

int mark = 0;

for(i=0;i<10;i++)

for (j = 0;j < 16;j++)

{

if (mx + i >= x1&&mx + i <= x2&&my + j >= y1&&my + j <= y2)

{

MouseSave[i][j] = getpixel(mx + i, my + j);

mark = 1;

}

}

if (mark == 1)

{

mousehide(mx,my);

getMousebk(mx, my);

cursor(mx, my);

}

}

autorun.c

//自定义头文件

#include "mytype.h"

//C运行时头文件

#include <graphics.h>

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <bios.h>

#include <string.h>

#include <dos.h>

#include <math.h>

//自定义头文件

#define MYMACRO extern

#include "common.h"

//自动在路线上跑

void autoruncar(ROADAREA ra[])

{

MYPOINT p1,p2;

double len;

int frompos;

double xingshi,jixu;

get\_automatic\_vlen(&vauto,vautomax,1,&len);

//saveautoold();

p1=pos;

xingshi=len;

frompos=1;//从当前位置

nopos=whereroadarea(ra,pos,2\*roadW);

if(strcmp(ra[nopos].type,"line")==0)

jixu=runline(ra,xingshi,frompos);

else

jixu=runarc(ra,xingshi,frompos);

while(jixu>0)

{

nopos++;

nopos%=2\*segs;

xingshi=jixu;

frompos=0;//从段的开始点

if(strcmp(ra[nopos].type,"line")==0)

jixu=runline(ra,xingshi,frompos);

else

jixu=runarc(ra,xingshi,frompos);

}

p2=pos;

offsetauto=jicheng(p1,p2,&sumauto);

}

//计算自动档时，一步路程和下一步小车速度

void get\_automatic\_vlen(double \*vcur,double vmax,double throttle,double \*len)

{

double vloc=\*vcur;

double a;

double timesec=tick/1000;

double rollResist;

double drag;

double fraction;

double F;

double gearRatio;

//根据速度确定档位

if(vloc<=10)

gearRatio=gear[1];

else

gearRatio=gear[2];

if(vloc>vmax && throttle==1)//若超速且有油门，加速度设为-1

{

a=-1;

}

else

{

//计算摩擦阻力

rollResist=Crr\*weight\*g;

//计算风阻

drag=0.5\*rou\*areaying\*Cd\*vloc\*vloc;

//计算牵引力

fraction=throttle\*torque\*gearRatio\*masterRatio\*eta/rwheel;

//计算合力

F=fraction-rollResist-drag;

//计算加速度

a=F/weight;

}

\*len=vloc\*timesec+0.5\*a\*timesec\*timesec;//行驶一步路程

\*vcur+=a\*timesec;//本步结束时的速度

if(throttle>0)//有油门

{

if(a<0)//超速，不能低于最高速度

{

if(\*vcur<vmax)

\*vcur=vmax;

}

else//正常，不能高于最高速度

{

if(\*vcur>vmax)

\*vcur=vmax;

}

}

else//无油门

{

if(\*vcur<0)

\*vcur=0;

}

}

int whereroadarea(ROADAREA ra[], MYPOINT p1,double offset)

{

int i,no=-1;

LINE line1;

ARC arc1;

for (i = 0;i < 2\*segs;i++)

{

if (strcmp(ra[i].type, "line") == 0)

{

line1=\*(PLINE)(ra[i].pcurve);

if(isAtLineRegion(line1,p1,offset)==1)

{

no=i;

break;

}

}

else

{

arc1=\*(PARC)(ra[i].pcurve);

if(isAtArcRegion(arc1,p1,offset)==1)

{

no=i;

break;

}

}

}

return no;

}

//在直线段跑

double runline(ROADAREA ra[],double xingshi,int frompos)

{

LINE line1;

double ang,toEnd,jixu;

MYPOINT p11,p21,pos0,pos1;

line1=\*(PLINE)(ra[nopos].pcurve);

if(frompos==1)//从当前位置

pos0=pos;

else//从段的起点

pos0=line1.p1;

ang=pp\_angle(line1.p1,line1.p2);

x2xp(line1.p1,ang,line1.p1,&p11);

x2xp(line1.p1,ang,line1.p2,&p21);

x2xp(line1.p1,ang,pos0,&pos1);

pos1.y=0;

toEnd=p21.x-pos1.x;

jixu=xingshi-toEnd;

if(jixu<=0)//未出本段

{

pos1.x+=xingshi;

pos1.y=0;

xp2x(line1.p1,ang,pos1,&pos);

angpos=ang-pi/2;

}

return jixu;

}

//在弧线跑

double runarc(ROADAREA ra[],double xingshi,int frompos)

{

ARC arc1;

double toEnd,jixu,angpos0;

arc1=\*(PARC)(ra[nopos].pcurve);

if(frompos==1)//从当前位置

angpos0=angpos;

else//从段的起点

angpos0=arc1.angstart;

toEnd=arc1.dang-stdangle0(angpos0-arc1.angstart);

xingshi/=arc1.r;//转换为角度

jixu=(xingshi-toEnd)\*arc1.r;//继续的距离

if(jixu<=0)//未出本段

{

angpos+=xingshi;

pos=getarcpnt(arc1,angpos);

}

return jixu;

}

//计算对应中间自动行驶路线的路程，返回与该路线的偏离值

double jicheng(MYPOINT p1, MYPOINT p2, double \*sum)

{

LINE line1;

ARC arc1;

int no1,no2;

double lenstart1,lenend1;

double lenstart2,lenend2;

double ang;

MYPOINT p21,pp;

double angp,ange;

MYPOINT o;

double offset;

no1=whereroadarea(autoroad[1],p1,2\*roadW);

if (strcmp(autoroad[1][no1].type, "line") == 0)

{

line1=\*(PLINE)(autoroad[1][no1].pcurve);

ang=pp\_angle(line1.p1,line1.p2);

x2xp(line1.p1,ang,line1.p2,&p21);

x2xp(line1.p1,ang,p1,&pp);

lenstart1=pp.x;

lenend1=p21.x-pp.x;

}

else

{

arc1=\*(PARC)(autoroad[1][no1].pcurve);

o=arc\_o(arc1);

angp=stdangle0(pp\_angle(o,p1)-arc1.angstart);

ange=arc1.dang;

lenstart1=arc1.r\*angp;

lenend1=arc1.r\*(ange-angp);

}

no2=whereroadarea(autoroad[1],p2,2\*roadW);

if (strcmp(autoroad[1][no2].type, "line") == 0)

{

line1=\*(PLINE)(autoroad[1][no2].pcurve);

ang=pp\_angle(line1.p1,line1.p2);

x2xp(line1.p1,ang,line1.p2,&p21);

x2xp(line1.p1,ang,p2,&pp);

lenstart2=pp.x;

lenend2=p21.x-pp.x;

offset=pp.y;

}

else

{

arc1=\*(PARC)(autoroad[1][no2].pcurve);

o=arc\_o(arc1);

angp=stdangle0(pp\_angle(o,p2)-arc1.angstart);

ange=arc1.dang;

lenstart2=arc1.r\*angp;

lenend2=arc1.r\*(ange-angp);

offset=pp\_dist(o,p2)-arc1.r;

}

if(no1!=no2)//不在同一段

{

if (no1 == 2\*segs-1 && no2 == 0)

{

no2=2\*segs;

}

else if (no1 == 0 && no2 == 2\*segs-1)

{

no2=-1;

}

if (no2 > no1)//前进

{

\*sum+=lenend1+lenstart2;

}

else//后退

{

\*sum-=(lenstart1+lenend2);

}

}

else//在同一段

{

\*sum+=lenstart2-lenstart1;

}

return offset;

}

Collide.c

//自定义头文件

#include "mytype.h"

//C运行时头文件

#include <graphics.h>

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <bios.h>

#include <string.h>

#include <dos.h>

#include <math.h>

//自定义头文件

#define MYMACRO extern

#include "common.h"

void proccarcollide()

{

double vtemp;

if (carcollide() == 1)//两车相撞，两车速度交换，并且前车速度加10，后车速度减10

{

if(sumauto>sumdrive)//自动行驶小车在前，速度较慢

{

vtemp=vauto;

vauto=v+10;

v=vtemp-10;

if(v<0)

v=0;

}

else//交互行驶小车在前，速度较慢

{

vtemp=v;

v=vauto+5;

vauto=vtemp-5;

if(vauto<0)

vauto=0;

}

}

}

int carcollide()

{

MYPOINT pp[4];

int i,collide=0;

//转到全局坐标，再转到交互驾驶汽车坐标

for (i = 0;i < 4;i++)

{

xp2x(pos,angpos,pcar[i],&pp[i]);

x2xp(mypos,myangpos,pp[i],&pp[i]);

if(GEANDLE(-carHW,pp[i].x,carHW,1e-6))

if (GEANDLE(-carL, pp[i].y, 0,1e-6))

{

collide=1;

break;

}

}

return collide;

}

Disp.c

//自定义头文件

#include "mytype.h"

//C运行时头文件

#include <graphics.h>

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <bios.h>

#include <string.h>

#include <dos.h>

#include <math.h>

//自定义头文件

#define MYMACRO extern

#include "common.h"

//在屏幕显示动态数据

void displayinf()

{

char str[30];

int i;

static double sumdrive\_t,sumauto\_t,v\_t,vauto\_t;

finish=clock();

sumdrive\_t=sumdrive;

sumauto\_t=sumauto;

v\_t=v;

vauto\_t=vauto;

duration=(finish-start)/CLOCKS\_PER\_SEC;

setcolor(RED);

i=1;

sprintf(str,"%6.0f",sumdrive\_t);

outtextxy(startclr,(i++)\*16,str);

sprintf(str,"%6.0f",sumauto\_t);

outtextxy(startclr,(i++)\*16,str);

sprintf(str,"%6.1f",v\_t);

outtextxy(startclr,(i++)\*16,str);

sprintf(str,"%6.1f",vauto\_t);

outtextxy(startclr,(i++)\*16,str);

sprintf(str,"%6.0f",duration);

outtextxy(startclr,(i++)\*16,str);

if(geartype==STICK)

{

sprintf(str,"%6d",myshift);

outtextxy(startclr,(i++)\*16,str);

}

setcolor(WHITE);

}

//在屏幕左边显示不变的文本

void displayinf0()

{

char str[30];

int i;

static double sumdrive\_t,sumauto\_t,v\_t,vauto\_t;

finish=clock();

sumdrive\_t=sumdrive;

sumauto\_t=sumauto;

v\_t=v;

vauto\_t=vauto;

duration=(finish-start)/CLOCKS\_PER\_SEC;

setfillstyle(SOLID\_FILL,LIGHTCYAN);

setcolor(LIGHTCYAN);

bar(0,0,(scrW-scrH)/2,scrH/2);//静态数字区清屏

setcolor(RED);

i=1;

outtextxy(0,(i++)\*16,"me(m):");

outtextxy(0,(i++)\*16,"auto(m):");

outtextxy(0,(i++)\*16,"me(m/s):");

outtextxy(0,(i++)\*16,"auto(m/s):");

outtextxy(0,(i++)\*16,"time(s):");

if(geartype==STICK) outtextxy(0,(i++)\*16,"me stick:");

setfillstyle(SOLID\_FILL,WHITE);//恢复为白色

setcolor(WHITE);

}

Drive.c

//自定义头文件

#include "mytype.h"

//C运行时头文件

#include <graphics.h>

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <bios.h>

#include <string.h>

#include <dos.h>

#include <math.h>

//自定义头文件

#define MYMACRO extern

#include "common.h"

void drivecar()

{

MYPOINT mypos1;

double newang,yidong;

LINE line2;

ARC arc2;

double len;

if(geartype==AUTO)

get\_automatic\_vlen(&v,vdrivemax,throttledrive,&len);

else

get\_stick\_vlen(&v,throttledrive,&len);

if (len != 0)

{

//savedriveold();

if(R==0)

twowheelstraight(len,&mypos1,&newang,&yidong,&line2);

else

twowheelturn(len,&mypos1,&newang,&yidong,&arc2);

offsetdrive=jicheng(mypos,mypos1,&sumdrive);

mypos=mypos1;

myangpos=newang;

}

}

void get\_stick\_vlen(double \*vcur,double throttle,double \*len)

{

double vold=\*vcur,vmax,vnew;

double a;

double timesec=tick/1000,t1;

double rollResist;

double drag;

double fraction;

double F;

double gearRatio;

//得到档位对应的变速比和最大速度

gearRatio=gear[myshift];

vmax=vmaxshift[myshift];

if(vold>vmax && throttle==1)//若超速且有油门，加速度设为-1

{

a=-1;

}

else

{

//计算摩擦阻力

rollResist=Crr\*weight\*g;

//计算风阻

drag=0.5\*rou\*areaying\*Cd\*vold\*vold;

//计算牵引力

fraction=throttle\*torque\*gearRatio\*masterRatio\*eta/rwheel;

//计算合力

F=fraction-rollResist-drag;

//计算加速度

a=F/weight;

}

vnew=vold+a\*timesec;//本步结束时的速度

if(throttle>0)//有油门

{

if(a<0)//超速，不能低于最高速度

{

if(vnew<vmax)

vnew=vmax;

}

else//正常，不能高于最高速度

{

if(vnew>vmax)

vnew=vmax;

}

}

else//无油门，最慢停下来

{

if(vnew<0)

vnew=0;

}

\*len=0.5\*(vold+vnew)\*timesec;//行驶一步路程

//加速有速度不变段，或减速有速度不变段

t1=(vnew-vold)/a;

if(t1<timesec)

\*len=0.5\*(vold+vnew)\*t1+vnew\*(timesec-t1);

\*vcur=vnew;

}

//双轮在内外环线上转弯到下一个位置，mypos1,myangpos1,yidong均对应轴中点，arc2为先碰轮子的轨迹

void twowheelturn(double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong, ARC \*arc2)

{

MYPOINT mypos1L,mypos1R;

double myangpos1L, myangpos1R;

double yidongL,yidongR;

ARC arc2L,arc2R;

onewheelturn(-carHW, len, &mypos1L, &myangpos1L, &yidongL,&arc2L);

onewheelturn(carHW, len, &mypos1R, &myangpos1R, &yidongR,&arc2R);

if (yidongL <= yidongR)

{

\*mypos1=mypos1L;

\*myangpos1=myangpos1L;

\*yidong=yidongL;

\*arc2=arc2L;

}

else

{

\*mypos1=mypos1R;

\*myangpos1=myangpos1R;

\*yidong=yidongR;

\*arc2=arc2R;

}

}

//双轮在内外环线上直行到下一个位置，mypos1,myangpos1,yidong均对应轴中点，line2为先碰轮子的轨迹

void twowheelstraight(double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong, LINE \*line2)

{

MYPOINT mypos1L,mypos1R;

double myangpos1L, myangpos1R;

double yidongL,yidongR;

LINE line2L,line2R;

onewheelstraight(-carHW, len, &mypos1L, &myangpos1L, &yidongL,&line2L);

onewheelstraight(carHW, len, &mypos1R, &myangpos1R, &yidongR,&line2R);

if (yidongL <= yidongR)

{

\*mypos1=mypos1L;

\*myangpos1=myangpos1L;

\*yidong=yidongL;

\*line2=line2L;

}

else

{

\*mypos1=mypos1R;

\*myangpos1=myangpos1R;

\*yidong=yidongR;

\*line2=line2R;

}

}

//单轮在内外环线上转弯到下一个位置，dr为轮子相对轴中心的距离，左为负。mypos1,myangpos1,yidong均对应轴中点，arc2为轮子的轨迹

void onewheelturn(double dr, double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong, ARC \*arc2)

{

MYPOINT mypos1N,mypos1W;

double myangpos1N, myangpos1W;

double yidongN,yidongW;

ARC arc2N,arc2W;

oneloopturn(roadareain,dr, len, &mypos1N, &myangpos1N, &yidongN,&arc2N);

oneloopturn(roadareaout,dr, len, &mypos1W, &myangpos1W, &yidongW,&arc2W);

if (yidongN <= yidongW)

{

\*mypos1=mypos1N;

\*myangpos1=myangpos1N;

\*yidong=yidongN;

\*arc2=arc2N;

}

else

{

\*mypos1=mypos1W;

\*myangpos1=myangpos1W;

\*yidong=yidongW;

\*arc2=arc2W;

}

}

//单轮在内外环线上直行到下一个位置，dr为轮子相对轴中心的距离，左为负。mypos1,myangpos1,yidong均对应轴中点，line2为轮子的轨迹

void onewheelstraight(double dr, double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong, LINE \*line2)

{

MYPOINT mypos1N,mypos1W;

double myangpos1N, myangpos1W;

double yidongN,yidongW;

LINE line2N,line2W;

oneloopstraight(roadareain,dr, len, &mypos1N, &myangpos1N, &yidongN,&line2N);

oneloopstraight(roadareaout,dr, len, &mypos1W, &myangpos1W, &yidongW,&line2W);

if (yidongN <= yidongW)

{

\*mypos1=mypos1N;

\*myangpos1=myangpos1N;

\*yidong=yidongN;

\*line2=line2N;

}

else

{

\*mypos1=mypos1W;

\*myangpos1=myangpos1W;

\*yidong=yidongW;

\*line2=line2W;

}

}

//单轮在环线上转弯到下一个位置，dr为轮子相对轴中心的距离，左为负。mypos1,myangpos1,yidong均对应轴中点，arc2为轮子的轨迹

int oneloopturn(ROADAREA ra[],double dr, double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong,ARC \*arc2)

{

int err=0;

MYPOINT pcol;

double R1,angin,newang;

int collide,no;

LINE liner;

ARC arcr,arc0,arc1;

arc0=getrotarc(len);

R1=R-dr;

arc1=arc0;

arc1.r=fabs(R1);

collide=checkarccollide(ra,arc1,R1, &pcol);

if (collide == 1)

{

no=whereroadarea(ra,pcol,roadW);

if (no < 0)

{

err=1;

return err;

}

if(R>0)

angin=pp\_angle(pcol,arc\_o(arc1));

else

angin=pp\_angle(arc\_o(arc1),pcol);

if (strcmp(ra[no].type, "line") == 0)

{

liner=\*(PLINE)(ra[no].pcurve);

newang=linereflect(liner,angin);

}

else

{

arcr=\*(PARC)(ra[no].pcurve);

newang=arcreflect(arcr,pcol,angin);

}

if (R > 0)

\*yidong=arc0.r\*stdangle0(arc1.angstart+arc1.dang-pp\_angle(arc\_o(arc1),pcol));

else

\*yidong=arc0.r\*stdangle0(pp\_angle(arc\_o(arc1),pcol)-arc1.angstart);

\*arc2=getrotarc(\*yidong);

arc2->r=arc1.r;

\*mypos1=getpangpnt(pcol,newang,-dr);

\*myangpos1=newang;

}

else//未碰

{

\*yidong=len;

\*arc2=arc1;

if (R > 0)

{

\*mypos1=getarcpnt(arc0,arc1.angstart);

\*myangpos1=pp\_angle(\*mypos1,arc\_o(arc1));

}

else

{

\*mypos1=getarcpnt(arc0,arc1.angstart+arc1.dang);

\*myangpos1=pp\_angle(arc\_o(arc1),\*mypos1);

}

}

return err;

}

//单轮在环线上直行到下一个位置，dr为轮子相对轴中心的距离，左为负。mypos1,myangpos1,yidong均对应轴中点，line2为轮子的轨迹

int oneloopstraight(ROADAREA ra[], double dr, double len, MYPOINT \*mypos1, double \*myangpos1, double \*yidong, LINE \*line2)

{

int err=0;

MYPOINT pcol;

double newang;

int collide,no;

LINE line1,liner;

ARC arcr;

line1=getline(dr,len);

collide=checklinecollide(ra,line1, &pcol);

if (collide == 1)

{

no=whereroadarea(ra,pcol,roadW);

if (no < 0)

{

err=1;

return err;

}

if (strcmp(ra[no].type, "line") == 0)

{

liner=\*(PLINE)(ra[no].pcurve);

newang=linereflect(liner,myangpos);

}

else

{

arcr=\*(PARC)(ra[no].pcurve);

newang=arcreflect(arcr,pcol,myangpos);

}

\*mypos1=getpangpnt(pcol,newang,-dr);

\*myangpos1=newang;

\*line2=line1;

line2->p2=pcol;

\*yidong=pp\_dist(line1.p1,pcol);

}

else//未碰

{

\*mypos1=getpangpnt(mypos,myangpos+pi/2,len);

\*myangpos1=myangpos;

\*yidong=len;

\*line2=line1;

}

return err;

}

//计算行驶线段与路径的交点

int checklinecollide(ROADAREA ra[], LINE line1, MYPOINT \*pout)

{

int collide=0;

LINE line2;

ARC arc1;

int i,m,sum=0;

MYPOINT ptmp[4];

double ang,x[4];

for (i = 0;i < 2\*segs;i++)

{

if (strcmp(ra[i].type, "line") == 0)

{

line2=\*(PLINE)(ra[i].pcurve);

line\_intersection(line1.p1, line1.p2, line2.p1, line2.p2,&m, &ptmp[sum]);

if(m>0)

sum+=m;

}

else

{

arc1=\*(PARC)(ra[i].pcurve);

arcline\_intersection(line1.p1, line1.p2, arc1, &ptmp[sum],&m);

if(m>0)

sum+=m;

}

}

if (sum > 0)

{

ang=pp\_angle(line1.p1,line1.p2);

for (i = 0;i < sum;i++)

{

x2xp(line1.p1,ang,ptmp[i],&ptmp[i]);

x[i]=ptmp[i].x;

}

sort(sum,x);

for (i = 0;i < sum;i++)

{

if (fabs(x[i]) >= 1e-6)

{

pout->x=x[i];

collide=1;

break;

}

}

if(collide==1)

{

pout->y=0;

xp2x(line1.p1,ang,\*pout,pout);

}

}

return collide;

}

//计算转弯圆弧与路径的交点，rot为负表示左转，为正表示右转

int checkarccollide(ROADAREA ra[], ARC arc1, double rot, MYPOINT \*pout)

{

int collide=0;

LINE line1;

ARC arc2;

int i,m,sum=0;

MYPOINT ptmp[16],o;

double ang[16];

for (i = 0;i < 2\*segs;i++)

{

if (strcmp(ra[i].type, "line") == 0)

{

line1=\*(PLINE)(ra[i].pcurve);

arcline\_intersection(line1.p1,line1.p2,arc1,&ptmp[sum],&m);

if(m>0)

sum+=m;

}

else

{

arc2=\*(PARC)(ra[i].pcurve);

arc\_intersection(arc1,arc2,&m,&ptmp[sum]);

if(m>0)

sum+=m;

}

}

if (sum > 0)

{

o=arc\_o(arc1);

for (i = 0;i < sum;i++)

{

if(rot<0)//左转

ang[i]=stdangle0(pp\_angle(o,ptmp[i])-arc1.angstart);

else//右转

ang[i]=stdangle0((arc1.angstart+arc1.dang)-pp\_angle(o,ptmp[i]));

}

sort(sum,ang);

for (i = 0;i < sum;i++)

{

if (fabs(ang[i]) >= 1e-8)

{

if(rot<0)//左转

\*pout=getarcpnt(arc1,ang[i]+arc1.angstart);

else

\*pout=getarcpnt(arc1,(arc1.angstart+arc1.dang)-ang[i]);

collide=1;

break;

}

}

}

return collide;

}

ARC getrotarc(double len)

{

ARC arc1;

double dang;

MYPOINT pnt;

arc1.r=fabs(R);

dang=-len/R;//R左为正，右为负

if (R > 0)

{

pnt=getpangpnt(mypos,myangpos,R);

arc1.angstart=myangpos+pi+dang;

arc1.dang=-dang;

}

else

{

pnt=getpangpnt(mypos,myangpos,R);

arc1.angstart=myangpos;

arc1.dang=dang;

}

arc1.x=pnt.x;

arc1.y=pnt.y;

return arc1;

}

//len为前车轴中点单步行驶路程,dr为轮子相对轴中心的距离，左为负。

LINE getline(double dr, double len)

{

MYPOINT p0,p1,p2;

LINE line1;

p0=mypos;

p1=getpangpnt(p0,myangpos,dr);

p2=getpangpnt(p1,myangpos+pi/2,len);

line1.p1=p1;

line1.p2=p2;

return line1;

}

Geo.c

//自定义头文件

#include "mytype.h"

//C运行时头文件

#include <graphics.h>

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <bios.h>

#include <string.h>

#include <dos.h>

#include <math.h>

//自定义头文件

#define MYMACRO extern

#include "common.h"

//几何计算函数

//计算直线线段与线段的交点

void line\_intersection(MYPOINT p1, MYPOINT p2, MYPOINT p3, MYPOINT p4, int \*count, MYPOINT pout[])

{

MYPOINT p11 = { 0,0 }, p21, p31, p41, pout1[4], ptemp;

double dx, dy, a;

int n = 0;

int i, j;

dx = p2.x - p1.x;

dy = p2.y - p1.y;

a = atan2(dy, dx);

//a = a / pi \* 180;

x2xp(p1, a, p2, &p21);

if (fabs(p21.y) < 1e-6) p21.y = 0;//y坐标很接近0时，调整为0

x2xp(p1, a, p3, &p31);

if (fabs(p31.y) < 1e-6) p31.y = 0;//y坐标很接近0时，调整为0

x2xp(p1, a, p4, &p41);

if (fabs(p41.y) < 1e-6) p41.y = 0;//y坐标很接近0时，调整为0

if (fabs(p31.y) < 1e-6 && fabs(p41.y) < 1e-6)//p3、p4都在直线p1p2上

{

MYPOINT temp;

if (p31.x > p41.x)

{

temp = p31;

p31 = p41;

p41 = temp;

}

if (p31.x >= p11.x && p31.x <= p21.x)

{

pout1[n] = p31;

n++;

}

if (p41.x >= p11.x && p41.x <= p21.x)

{

pout1[n] = p41;

n++;

}

if (p21.x > p31.x && p21.x <p41.x)

{

pout1[n] = p21;

n++;

}

if (p11.x > p31.x && p11.x <p41.x)

{

pout1[n] = p11;

n++;

}

if (n == 2)

{

if (fabs(pout1[0].x - pout1[1].x)<1e-6) n = 1;

}

else if (n >= 3)

{

for (i = 1;i < n;i++)

{

for (j = 0;j < n - i;j++)

{

if (pout1[j].x > pout1[j + 1].x)

{

ptemp = pout1[j];

pout1[j] = pout1[j + 1];

pout1[j + 1] = ptemp;

}

}

}

pout1[1] = pout1[n - 1];

n = 2;

}

}

else if (fabs(p31.y) < 1e-6)//仅p3在直线p1p2上

{

if (p31.x >= p11.x && p31.x <= p21.x)

{

pout1[0] = p31;

n = 1;

}

}

else if (fabs(p41.y) < 1e-6)//仅p4在直线p1p2上

{

if (p41.x >= p11.x && p41.x <= p21.x)

{

pout1[0] = p41;

n = 1;

}

}

else if (p31.y\*p41.y < 0)///p3、p4分别在直线p1p2两侧

{

double angtem;

dx = p41.x - p31.x;

dy = p41.y - p31.y;

angtem = atan2(dy, dx);

pout1[0].y = 0;

pout1[0].x = p31.x + (0.0 - p31.y) / sin(angtem)\*cos(angtem);

if (pout1[0].x >= p11.x && pout1[0].x <= p21.x) n = 1;

}

if (n>0)//将交点转回全局坐标

{

for (i = 0;i < n;i++)

{

xp2x(p1, a, pout1[i], &pout[i]);

}

}

\*count = n;

}

void arcline\_intersection(MYPOINT p1, MYPOINT p2, ARC arc1, MYPOINT intersection[], int \*n)

{

//O为圆心，E为圆心到p1p2的垂足，D为交点

int count = 0,i;

double dx, dy, angp1p2, angOP1, angOP2, angOP1E, angOP2E, angOD, OP1, OP2, OE, PE, DE, ox, oy;

arc1.angstart = stdangle0(arc1.angstart);//转换到大于等于0，小于2pi

//转换到圆弧圆心为原点的局部坐标系

ox = arc1.x;//保存圆弧圆心全局x坐标

oy = arc1.y;//保存圆弧圆心全局y坐标

//p1、p2、圆弧圆心转换到局部坐标系

p1.x -= ox;

p1.y -= oy;

p2.x -= ox;

p2.y -= oy;

arc1.x = 0;

arc1.y = 0;

dx = p2.x - p1.x;

dy = p2.y - p1.y;

angp1p2 = atan2(dy, dx);//p1p2的角度

angOP1 = atan2(p1.y, p1.x);//OP1的角度

angOP1E = angp1p2 - angOP1;//两个矢量夹角，与角OP1E相等或互补

angOP1E = stdangle(angOP1E);//转换到 - pi到pi之间

angOP2 = atan2(p2.y, p2.x);//OP2的角度

angOP2E = angp1p2 - angOP2;//两个矢量夹角，与角OP2E相等或互补

angOP2E = stdangle(angOP2E);//转换到 - pi到pi之间

OP1 = sqrt(p1.x\*p1.x + p1.y\*p1.y);

OP2 = sqrt(p2.x\*p2.x + p2.y\*p2.y);

OE = fabs(OP1\*sin(angOP1E));

PE = fabs(OP1\*cos(angOP1E));

if(fabs(arc1.r - OE)<1e-6 || arc1.r < OE)//圆弧半径等于OE，或小于OE

DE = 0;

else//圆弧半径大于OE

DE = sqrt(arc1.r\*arc1.r - OE \* OE);

if (fabs(OE - arc1.r)<1e-6)//相切

{

if ((fabs(angOP1E - pi / 2)<1e-6 || fabs(angOP1E) > pi / 2) && ((fabs(angOP2E - pi / 2)<1e-6 || fabs(angOP1E) < pi / 2)))//p1不超过E,且p2至少从E开始

{

intersection[0].x = p1.x + PE \* cos(angp1p2);

intersection[0].y = p1.y + PE \* sin(angp1p2);

count = 1;

}

}

else if (OE < arc1.r)//与圆可能有两个交点

{

if ((fabs(OP1 - arc1.r)<1e-6 || OP1 > arc1.r) && fabs(angOP1E) > pi / 2) //(GREATER\_EQUVAL(OP1, arc1.r))//p1在圆上或圆外,且p1向着E运动

{

intersection[0].x = p1.x + (PE - DE)\*cos(angp1p2);

intersection[0].y = p1.y + (PE - DE)\*sin(angp1p2);

if ((fabs(OP2 - arc1.r)<1e-6 && fabs(angOP2E) > pi / 2) || OP2 < arc1.r)//p2在圆上，且向着E运动，或p2在圆内

count = 1;

else if ((fabs(OP2 - arc1.r) < 1e-6 || OP2 > arc1.r) && fabs(angOP2E) < pi / 2) //p2在圆上或圆外,且p2离开E运动

{

count = 2;

intersection[1].x = p1.x + (PE + DE)\*cos(angp1p2);

intersection[1].y = p1.y + (PE + DE)\*sin(angp1p2);

}

}

else if ((OP1 < arc1.r || (fabs(OP1 - arc1.r) < 1e-6 && fabs(angOP1E) < pi / 2)) && ((fabs(OP2 - arc1.r)<1e-6 && fabs(angOP2E) < pi / 2) || OP2 > arc1.r))//p1在圆内，或圆上离开E的方向；且p2在圆上离开E的方向，或圆外

{

count = 1;

if (fabs(angOP1E) >= pi / 2)//p1向着E运动

{

intersection[0].x = p1.x + (PE + DE)\*cos(angp1p2);

intersection[0].y = p1.y + (PE + DE)\*sin(angp1p2);

}

else//p1离开E运动

{

intersection[0].x = p1.x + (DE - PE)\*cos(angp1p2);

intersection[0].y = p1.y + (DE - PE)\*sin(angp1p2);

}

}

}

if (count == 1)

{

angOD = atan2(intersection[0].y, intersection[0].x);

angOD -= arc1.angstart;//以圆弧起始角为0，OD的角度

angOD = stdangle0(angOD);//转换到大于等于0，小于2pi

if (angOD >= 0 && ((fabs(angOD - arc1.dang)<1e-6 || angOD < arc1.dang)))

count = 1;

else

count = 0;

}

else if (count == 2)

{

angOD = atan2(intersection[0].y, intersection[0].x);

angOD -= arc1.angstart;//以圆弧起始角为0，OD的角度

angOD = stdangle0(angOD);//转换到大于等于0，小于2pi

if (angOD >= 0 && ((fabs(angOD - arc1.dang)<1e-6 || angOD < arc1.dang)))//第1个是交点

{

count = 1;

angOD = atan2(intersection[1].y, intersection[1].x);

angOD -= arc1.angstart;//以圆弧起始角为0，OD的角度

angOD = stdangle0(angOD);//转换到大于等于0，小于2pi

if (angOD >= 0 && ((fabs(angOD - arc1.dang)<1e-6 || angOD < arc1.dang)))//第2个是交点

count = 2;

}

else//第1个不是交点

{

angOD = atan2(intersection[1].y, intersection[1].x);

angOD -= arc1.angstart;//以圆弧起始角为0，OD的角度

angOD = stdangle0(angOD);//转换到大于等于0，小于2pi

if (angOD >= 0 && ((fabs(angOD - arc1.dang)<1e-6 || angOD < arc1.dang)))//第2个是交点

{

count = 1;

intersection[0] = intersection[1];

}

else

count = 0;

}

}

//交点转回全局坐标

for (i = 0;i < count;i++)

{

intersection[i].x += ox;

intersection[i].y += oy;

}

\*n = count;

}

void arc\_intersection(ARC arc1, ARC arc2, int \*count, MYPOINT pout[])

{

double oo,angoo,r1,r2,x,y,DOO,ang[4];

int n,m,i;

MYPOINT point[4],ptmp[4],po;

oo=oo\_dist(arc1,arc2);

angoo=oo\_angle(arc1,arc2);

po=arc\_o(arc1);

r1=arc1.r;

r2=arc2.r;

n=0;//没有交点，n=0

//半径相等且同心

if (fabs(r1 - r2) < 1e-6 && oo < 1e-6)

{

point[0]=getarcpnt(arc1,arc1.angstart);

point[1]=getarcpnt(arc1,arc1.angstart+arc1.dang);

point[3]=getarcpnt(arc2,arc2.angstart);

point[4]=getarcpnt(arc2,arc2.angstart+arc2.dang);

n=4;

}

//外切

else if (fabs(r1 + r2 - oo) < 1e-6)

{

point[0]=getarcpnt(arc1,angoo);

n=1;

}

//内切

else if (fabs(fabs(r1 - r2) - oo) < 1e-6)

{

n=1;

if (r1 > r2)

point[0]=getarcpnt(arc1,angoo);

else

point[0]=getarcpnt(arc1,angoo+pi);

}

else if (r1 + r2 > oo)

{

x=(r1\*r1+oo\*oo-r2\*r2)/(2\*r1\*oo); //cos(DOO)

if(1<x\*x)//一个圆在另一个圆内，无交点

n=0;

else//有两个交点

{

y=sqrt(1-x\*x);//sin(DOO)

DOO=atan2(y,x);

point[0]=getarcpnt(arc1,angoo+DOO);

point[1]=getarcpnt(arc1,angoo-DOO);

n=2;

}

}

//是否在两个圆弧上

m=0;

for(i=0;i < n;i++)

{

if(isAtArc(arc1,point[i]) && isAtArc(arc2,point[i]))

ptmp[m++]=point[i];

}

if (m > 0)

{

for(i=0;i<m;i++)

ang[i]=stdangle0(pp\_angle(po,ptmp[i])-arc1.angstart);

}

//排序

sort(m,ang);

//排除重合点

if (m == 2)

{

if(fabs(ang[0]-ang[1])<1e-8)

m=1;

}

else if (m > 2)

{

ang[1]=ang[m-1];

m=2;

}

//转换为点

for (i = 0;i < m;i++)

pout[i]=getarcpnt(arc1,ang[i]+arc1.angstart);

\*count=m;

}

//找点在路线的哪一段

//确定点是否在线段两边一定距离范围内

int isAtLineRegion(LINE line1, MYPOINT p,double hw)

{

int at=0;

double ang;

MYPOINT p21,pp;

ang=pp\_angle(line1.p1,line1.p2);

x2xp(line1.p1,ang,line1.p2,&p21);

x2xp(line1.p1,ang,p,&pp);

if(GEANDLE(0.0,pp.x,p21.x,1e-6))

if(fabs(pp.y)<hw)

at=1;

return at;

}

//确定点是否在圆弧两边一定距离范围内

int isAtArcRegion(ARC arc1, MYPOINT p,double hw)

{

int at=0;

double angp,ange;

MYPOINT o;

o=arc\_o(arc1);

angp=stdangle0(pp\_angle(o,p)-arc1.angstart);

ange=arc1.dang;

if(GEANDLE(0.0,angp,ange,1e-8))

if(fabs(pp\_dist(o,p)-arc1.r)<hw)

at=1;

return at;

}

//根据角度确定点是否在圆弧上

int isAtArc(ARC arc1, MYPOINT p)

{

double ang,ang1;

int i;

ang1=pp\_angle(arc\_o(arc1),p),

ang=stdangle0(ang1-arc1.angstart);

i=(GEANDLE(0.0,ang,arc1.dang,1e-8))?1:0;

return i;

}

//计算对线的反射角，ain为车右方与x轴夹角

double linereflect(LINE line1, double ain)

{

double a0;

a0=pp\_angle(line1.p1,line1.p2)-pi/2;//法线角度

return stdangle(2\*a0-ain);

}

//计算对圆弧的反射角

double arcreflect(ARC arc1, MYPOINT p,double ain)

{

double a0;

MYPOINT o;

o=arc\_o(arc1);

a0=pp\_angle(o,p);//法线角度

return stdangle(2.0\*a0-ain);

}

double pp\_angle(MYPOINT p1,MYPOINT p2)

{

double dx,dy;

dx=p2.x-p1.x;

dy=p2.y-p1.y;

return atan2(dy,dx);

}

double oo\_angle(ARC arc1, ARC arc2)

{

double dx,dy;

dx=arc2.x-arc1.x;

dy=arc2.y-arc1.y;

return atan2(dy,dx);

}

MYPOINT arc\_o(ARC arc1)

{

MYPOINT p;

p.x=arc1.x;

p.y=arc1.y;

return p;

}

double pp\_dist(MYPOINT p1,MYPOINT p2)

{

double dx,dy;

dx=p2.x-p1.x;

dy=p2.y-p1.y;

return sqrt(dx\*dx+dy\*dy);

}

double oo\_dist(ARC arc1, ARC arc2)

{

double dx,dy;

dx=arc2.x-arc1.x;

dy=arc2.y-arc1.y;

return sqrt(dx\*dx+dy\*dy);

}

MYPOINT getarcpnt(ARC arc1,double ang)

{

MYPOINT p;

p.x=arc1.x+arc1.r\*cos(ang);

p.y=arc1.y+arc1.r\*sin(ang);

return p;

}

//由起点p1，按矢量移动（矢量的方向角ang和长度len），得到新的点

MYPOINT getpangpnt(MYPOINT p0, double ang, double len)

{

MYPOINT p1;

p1.x=p0.x+len\*cos(ang);

p1.y=p0.y+len\*sin(ang);

return p1;

}

//转换到 - pi到pi之间，用于锐角、钝角判断

double stdangle(double angle)

{

//转换到-2pi到2pi之间

if (angle > 2 \* pi) angle -= (int)(angle / (2 \* pi)) \* 2 \* pi;

else if (angle < -2 \* pi) angle += (int)(angle / (2 \* pi)) \* 2 \* pi;

//转换到 - pi到pi之间

if (angle > pi) angle -= 2 \* pi;

else if (angle < -pi) angle += 2 \* pi;

return angle;

}

//将弧度角转换到大于等于0，小于2pi，用于角度大小比较

double stdangle0(double angle)

{

//转换到-2pi到2pi之间

if (angle > 2 \* pi) angle -= (int)(angle / (2 \* pi)) \* 2 \* pi;

else if (angle < -2 \* pi) angle += (int)(angle / (2 \* pi)) \* 2 \* pi;

//转换到 - pi到pi之间

if (angle > pi) angle -= 2 \* pi;

else if (angle < -pi) angle += 2 \* pi;

if (fabs(angle) < 1e-6) angle = 0;//与0夹角很小，则为0。

else if (angle < 0) angle += 2 \* pi;//转换到大于等于0，小于2pi。

return angle;

}

//xy二维平面全局到局部坐标系转换

void x2xp(MYPOINT pntsight, double alpha0, MYPOINT pnt, MYPOINT \*pnt\_p)

{

double alpha, x, y;

//alpha = alpha0 / 180.0\*pi;

alpha = alpha0;

x = pnt.x - pntsight.x;

y = pnt.y - pntsight.y;//平移

pnt\_p->x = x \* cos(alpha) + y \* sin(alpha);

pnt\_p->y = -x \* sin(alpha) + y \* cos(alpha);//旋转

}

//xy二维平面局部到全局坐标系转换

void xp2x(MYPOINT pntsight, double alpha0, MYPOINT pnt\_p, MYPOINT \*pnt)

{

double alpha, x, y;

//alpha = -alpha0 / 180.0\*pi;

alpha = -alpha0;

x = pnt\_p.x \* cos(alpha) + pnt\_p.y \* sin(alpha);

y = -pnt\_p.x \* sin(alpha) + pnt\_p.y \* cos(alpha);

pnt->x = x + pntsight.x;

pnt->y = y + pntsight.y;

}

//圆心移动，yz平面转动，三维点全局到局部坐标系转换

void y2yp(MY3DPOINT pntsight, double alpha0, MY3DPOINT pnt, MY3DPOINT \*pnt\_p)

{

double alpha, x, y, z;

alpha = alpha0;

x = pnt.p.x - pntsight.p.x;

y = pnt.p.y - pntsight.p.y;

z = pnt.z - pntsight.z;//平移

pnt\_p->p.y = y \* cos(alpha) + z \* sin(alpha);

pnt\_p->z = -y \* sin(alpha) + z \* cos(alpha);//旋转

}

//圆心移动，yz平面转动，三维点局部到全局坐标系转换

void yp2y(MY3DPOINT pntsight, double alpha0, MY3DPOINT pnt\_p, MY3DPOINT \*pnt)

{

double alpha, x, y, z;

alpha = -alpha0;

y = pnt\_p.p.y \* cos(alpha) + pnt\_p.z \* sin(alpha);

z = -pnt\_p.p.y \* sin(alpha) + pnt\_p.z \* cos(alpha);

pnt->p.x += pntsight.p.x;

pnt->p.y = y + pntsight.p.y;

pnt->z = z + pntsight.z;

}

LINE scaleline(double k, LINE line1)

{

MYPOINT p1,p2;

LINE li;

p1.x=k\*line1.p1.x;

p1.y=k\*line1.p1.y;

p2.x=k\*line1.p2.x;

p2.y=k\*line1.p2.y;

li.p1=p1;

li.p2=p2;

return li;

}

ARC scalearc(double k, ARC arc1)

{

ARC ar;

ar=arc1;

ar.r\*=k;

ar.x\*=k;

ar.y\*=k;

return ar;

}

//全局双精度单点转换为窗口坐标，并移到窗口中心

POINT x2scr(MYPOINT p1)

{

POINT p2;

p2.x=(long)(p1.x+xcen);

p2.y=(long)(-p1.y+ycen);

return p2;

}

//长整数点数组转换为窗口坐标，并移到窗口中心

void toscreen(POINT p1[])

{

int i;

for(i=0;i<5;i++)

{

p1[i].x+=xcen;

p1[i].y=-p1[i].y+ycen;

}

}

void sort(int n, double a[])

{

int i,j,jh=1;

double tmp;

for (i=n-1;i>0 && jh==1;i--)

{

jh=0;//若没有交换，则jh保持为0

for(j=0;j<i;j++)

{

if (a[j]>a[j+1])

{

tmp=a[j];

a[j]=a[j+1];

a[j+1]=tmp;

jh=1;

}

}

}

}

Init.c

//自定义头文件

#include "mytype.h"

//C运行时头文件

#include <graphics.h>

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <bios.h>

#include <string.h>

#include <dos.h>

#include <math.h>

//自定义头文件

#define MYMACRO extern

#include "common.h"

void roadinit()

{

int i,pathno=1;

double r0,r1;

r0=road\_r-0.5\*roadW;

r1=road\_r-roadW;

if(segs==4)//四边形跑道

{

//跑道中间虚线

for(i=0;i<2;i++)

{

setrecroadloop(r0+i\*roadW, road\_linedash[i], road\_arcdash[i], roadareadash[i]);

}

//跑道外边界

setrecroadloop(road\_r+1.5\*roadW, road\_lineout, road\_arcout, roadareaout);

//跑道内边界

setrecroadloop(road\_r-1.5\*roadW, road\_linein, road\_arcin, roadareain);

//自动行驶路线

for(i=0;i<3;i++)

{

setrecroadloop(r1+i\*roadW, auto\_line[i], auto\_arc[i],autoroad[i]);

}

}

else if(segs==3)//三角形跑道

{

//跑道中间虚线

for(i=0;i<2;i++)

{

settriroadloop(r0+i\*roadW, road\_linedash[i], road\_arcdash[i], roadareadash[i]);

}

//跑道外边界

settriroadloop(road\_r+1.5\*roadW, road\_lineout, road\_arcout, roadareaout);

//跑道内边界

settriroadloop(road\_r-1.5\*roadW, road\_linein, road\_arcin, roadareain);

//自动行驶路线

for(i=0;i<3;i++)

{

settriroadloop(r1+i\*roadW, auto\_line[i], auto\_arc[i],autoroad[i]);

}

}

else//其他段数跑道

{

//跑道中间虚线

for(i=0;i<2;i++)

{

setroadloop(r0+i\*roadW, road\_linedash[i], road\_arcdash[i], roadareadash[i]);

}

//跑道外边界

setroadloop(road\_r+1.5\*roadW, road\_lineout, road\_arcout, roadareaout);

//跑道内边界

setroadloop(road\_r-1.5\*roadW, road\_linein, road\_arcin, roadareain);

//自动行驶路线

for(i=0;i<3;i++)

{

setroadloop(r1+i\*roadW, auto\_line[i], auto\_arc[i],autoroad[i]);

}

}

offsetauto=r0+pathno\*roadW-road\_r;

pos.x=road\_halfo+road\_r+offsetauto;

pos.y=0;

sumauto=pos.y;

//sumautoold=sumauto;

offsetdrive=-roadW;

mypos.x=road\_halfo+road\_r+offsetdrive;

mypos.y=0;

sumdrive=mypos.y;

//sumdriveold=sumdrive;

}

//初始化四边形跑道

void setrecroadloop(double r, LINE li[], ARC ar[], ROADAREA ra[])

{

MYPOINT p1,p2;

int i;

double road\_halfl= road\_halfo+r;

p1.x=road\_halfl;

p1.y=-road\_halfo;

p2.x=p1.x;

p2.y=-p1.y;

li[0].p1=p1;

li[0].p2=p2;

p1.x=road\_halfo;

p1.y=road\_halfl;

p2.x=-p1.x;

p2.y=p1.y;

li[1].p1=p1;

li[1].p2=p2;

p1.x=-road\_halfl;

p1.y=road\_halfo;

p2.x=p1.x;

p2.y=-p1.y;

li[2].p1=p1;

li[2].p2=p2;

p1.x=-road\_halfo;

p1.y=-road\_halfl;

p2.x=-p1.x;

p2.y=p1.y;

li[3].p1=p1;

li[3].p2=p2;

ar[0].x=road\_halfo;

ar[0].y=ar[0].x;

ar[0].r=r;

ar[0].angstart=0;

ar[0].dang=pi/2;

ar[1].x=-road\_halfo;

ar[1].y=-ar[1].x;

ar[1].r=r;

ar[1].angstart=pi/2;

ar[1].dang=pi/2;

ar[2].x=-road\_halfo;

ar[2].y=ar[2].x;

ar[2].r=r;

ar[2].angstart=pi;

ar[2].dang=pi/2;

ar[3].x=road\_halfo;

ar[3].y=-ar[3].x;

ar[3].r=r;

ar[3].angstart=3\*pi/2;

ar[3].dang=pi/2;

for (i = 0;i < 4;i++)

{

strcpy(ra[2\*i].type,"line");

ra[2\*i].pcurve=&li[i];

strcpy(ra[2\*i+1].type,"arc");

ra[2\*i+1].pcurve=&ar[i];

}

}

//初始化三角形跑道

void settriroadloop(double r, LINE li[], ARC ar[], ROADAREA ra[])

{

int i;

MYPOINT p1,p2,pc0,pc1,pc2;

double theta0=2\*pi/3,theta;

double road\_halfl= road\_halfo+r;

p1.x=road\_halfl;

p1.y=-road\_halfo;

p2.x=p1.x;

p2.y=-p1.y;

li[0].p1=p1;

li[0].p2=p2;

pc0.x=road\_halfo;

pc0.y=road\_halfo;

pc2.x=pc0.x;

pc2.y=-pc0.y;

theta=theta0;

p1.x=pc0.x+r\*cos(theta);

p1.y=pc0.y+r\*sin(theta);

theta+=pi/2;

p2.x=p1.x+2\*road\_halfo\*cos(theta);

p2.y=p1.y+2\*road\_halfo\*sin(theta);

li[1].p1=p1;

li[1].p2=p2;

pc1.x=pc0.x+2\*road\_halfo\*cos(theta);

pc1.y=0;

p1=li[1].p2;

p1.y=-p1.y;

p2=li[1].p1;

p2.y=-p2.y;

li[2].p1=p1;

li[2].p2=p2;

ar[0].x=pc0.x;

ar[0].y=pc0.y;

ar[0].r=r;

ar[0].angstart=0;

ar[0].dang=theta0;

ar[1].x=pc1.x;

ar[1].y=pc1.y;

ar[1].r=r;

ar[1].angstart=theta0;

ar[1].dang=theta0;

ar[2].x=pc2.x;

ar[2].y=pc2.y;

ar[2].r=r;

ar[2].angstart=2\*theta0;

ar[2].dang=theta0;

for (i = 0;i < 3;i++)

{

strcpy(ra[2\*i].type,"line");

ra[2\*i].pcurve=&li[i];

strcpy(ra[2\*i+1].type,"arc");

ra[2\*i+1].pcurve=&ar[i];

}

}

void setroadloop(double r, LINE li[], ARC ar[], ROADAREA ra[])

{

int i;

MYPOINT p0,p1={100,-100},p2={100,100};

double a=0;//a为直线段平移方向

double b=pi/segs;//b为圆心连线第2点的极坐标角度

double len=100.0/sin(b);//正多边形外接圆半径

p0.y=0;

p0.x=100.0-100.0/tan(b);

for(i=0;i<segs;i++)

{

li[i].p1=getpangpnt(p1,a,r);

li[i].p2=getpangpnt(p2,a,r);

ar[i].x=p2.x;

ar[i].y=p2.y;

ar[i].r=r;

ar[i].angstart=a;

ar[i].dang=2\*pi/segs;

a+=2\*pi/segs;

b+=2\*pi/segs;

p1=p2;

p2=getpangpnt(p0,b,len);

}

for (i = 0;i < segs;i++)

{

strcpy(ra[2\*i].type,"line");

ra[2\*i].pcurve=&li[i];

strcpy(ra[2\*i+1].type,"arc");

ra[2\*i+1].pcurve=&ar[i];

}

}

//全局变量初始化

void GlobalInit()

{

double theta;

int n;

int i,j;

//平面图视点和视角选择

//sight=Global;

//平面图显示选择

//pingmian=0;//为1时显示平面图

//透视图显示选择

toushi=1;//为1时显示透视图

//kroad=0.8;//道路平面图放大系数

//====车行、档，跑道选择变量====

//segs=6;//确定跑道为三角形，还是四边形

//交互行驶车型选择，改高度，颜色

//mecolor=YELLOW;

myshift=0;//手动档档位

//geartype= STICK;//STICK; //AUTO

oldstate=0;//单击键前一状态，1为按下

trig=0;//为1表示单击键经过了一次松开到按下的转变

//posping.x=0;

//posping.y=0;

//angposping=0;

dly=1;

throttledrive=0;//交互驾驶车油门

throttleauto=1;//自动驾驶车油门

rwheel=0.310;//轮胎半径

areaying=2.45;//迎风面积

weight=1750;//车质量

g=9.8;//重力加速度

rou=1.29;//空气密度

Cd=1.5;//风阻系数

Crr=0.05;//滚动摩擦系数

//各档传动比1.7~1.8，如六档变速器7.640,4.835,2.857,1.895,1.337,1

masterRatio=3.3;//主减速器变速比，3.5-5

eta=0.9;//传动效率

torque=170;//扭矩

//各档对应的变速箱变速比，1档对应下标1,2档对应下标2

gear[0]=0;

gear[1]=4;

gear[2]=3;

//vdrivemax=20;//20交互行驶最大速度

//各档对应的变速箱变速比

vmaxshift[0]=0;

vmaxshift[1]=vdrivemax/2;

vmaxshift[2]=vdrivemax;

cmd = 0;

keystate=0;

itext=0;//每几帧显示速度等信息的控制变量

//视点高、水平面位置、视场坐标相对全局坐标转动角度

camH = 3;//单位m

w = 0.1;//线半宽，单位m

carHW=0.9;//汽车半宽

carL=4;//汽车长度

v = 0;//交互行驶速度

R = 0;//判断是否转弯

vautomax=19;//自动行驶最大速度

vauto=0;//自动行驶速度

tick = 1000.0/CLOCKS\_PER\_SEC;

rdrive=60;//转弯半径

//跑道中心线参数

roadW=5;//3.5;

kdash=1;//虚线周期放大系数

road\_halfo=100;//道路中间线圆弧圆心与坐标轴距离

road\_r=153;//道路中心线弯道圆弧半径

//ROADAREA \*pathauto;

autoH=1.4;//自动行驶车高度

nopos=0;//自动行驶当前段号

pathno=1;//自动行驶当前路线号

sumauto=0;//自动行驶对应自动行驶路线路程

//sumautoold=0;

angpos=0;//自动行驶车右方向与x轴的当前夹角

//angposold=0;//自动行驶车右方向与x轴的前一夹角

sumdrive=0;//交互行驶对应自动行驶路线路程

//sumdriveold=0;

myangpos=0;//-pi/6;//交互行驶右方向与x轴的夹角

//myangposold=0;//交互行驶车右方向与x轴的前一夹角

//屏幕参数

scrW = 640.0;

scrH=480.0;

startclr=(scrW-scrH)/2;

pi = 4.0\*atan(1.0);

vmaxshift[2]=vdrivemax;

//ohstep=1.5\*vauto/1000.0\*tick;

oneauto=vauto/1000.0\*tick;

dashlen=4\*kdash;//虚线周期长

theta=2\*pi/segs;

n=(int)(theta\*road\_r/dashlen);//按直虚线周期数求等分数

dashang=theta/n;//pi/2.0/(60.0/kdash);

endline.p1.x=road\_r+road\_halfo-1.5\*roadW;

endline.p1.y=0;

endline.p2.x=road\_r+road\_halfo+1.5\*roadW;

endline.p2.y=0;

k = scrW / scrH;//屏幕宽高比

xcen= scrW-scrH/2;

ycen=scrH/2;

angvmax = pi/6;//视场的垂向半视角

yminsight =2\*camH, ymaxsight = 200+yminsight;//视场的近距、远距

anghmax = angvmax;//视场的水平半视角，与垂直半视角相等，屏幕两边有没有用到的地方

a2hsignt = pi / 2 + anghmax;//视场左边界角

a1hsignt = pi / 2 - anghmax;//视场右边界角

p[0].x=-yminsight \* tan(anghmax);

p[0].y=yminsight;

p[1].x=yminsight \* tan(anghmax);

p[1].y=yminsight;

p[2].x=ymaxsight \* tan(anghmax);

p[2].y=ymaxsight;

p[3].x=-ymaxsight \* tan(anghmax);

p[3].y=ymaxsight;

p[4]=p[0];

//车子自身坐标系下的轮廓点坐标,右为x，车头中点为原点

//侧面头底到尾底

pcarright[0].p.y=0;

pcarright[0].z=0.3100;

pcarright[1].p.y=-0.0686;

pcarright[1].z=0.8117;

pcarright[2].p.y=-0.8571;

pcarright[2].z=0.9197;

pcarright[3].p.y=-1.7056;

pcarright[3].z=1.4;

pcarright[4].p.y=-3.5426;

pcarright[4].z=1.4;

pcarright[5].p.y=-4.0;

pcarright[5].z=0.8739;

pcarright[6].p.y=-4.0;

pcarright[6].z=0.31;

for(i=0;i<7;i++)

{

pcarleft[i]=pcarright[i];

pcarright[i].p.x=0.9;

pcarleft[i].p.x=-0.9;

}

pcar[0]=pcarleft[0].p;

pcar[1]=pcarright[0].p;

pcar[2]=pcarright[6].p;

pcar[3]=pcarleft[6].p;

pcar[4]=pcar[0];

/\*pcar[0].x=-carHW;

pcar[0].y=0;

pcar[1].x=carHW;

pcar[1].y=0;

pcar[2].x=carHW;

pcar[2].y=-carL;

pcar[3].x=-carHW;

pcar[3].y=-carL;

pcar[4]=pcar[0];\*/

//前后轮外侧中心

pcarcenright[0].p.y=-0.7;

pcarcenright[0].z=0.31;

pcarcenright[1].p.y=-3.3;

pcarcenright[1].z=0.31;

for(i=0;i<2;i++)

{

pcarcenleft[i]=pcarcenright[i];

pcarcenright[i].p.x=0.9;

pcarcenleft[i].p.x=-0.9;

}

//前后轮内侧中心

for(i=0;i<2;i++)

{

pcarcenrightin[i]=pcarcenright[i];

pcarcenleftin[i]=pcarcenright[i];

pcarcenrightin[i].p.x=0.9-0.215;

pcarcenleftin[i].p.x=-(0.9-0.215);

}

//车窗

pcarwinright[0][0].p.y=-1.0632;

pcarwinright[0][0].z=0.9167;

pcarwinright[0][1].p.y=-1.7684;

pcarwinright[0][1].z=1.3159;

pcarwinright[0][2].p.y=-2.3720;

pcarwinright[0][2].z=1.3159;

pcarwinright[0][3].p.y=-2.2180;

pcarwinright[0][3].z=0.8999;

pcarwinright[1][0].p.y=-2.4497;

pcarwinright[1][0].z=0.8965;

pcarwinright[1][1].p.y=-2.5407;

pcarwinright[1][1].z=1.3159;

pcarwinright[1][2].p.y=-2.9585;

pcarwinright[1][2].z=1.3159;

pcarwinright[1][3].p.y=-3.1788;

pcarwinright[1][3].z=0.8858;

pcarwinright[2][0].p.y=-3.3241;

pcarwinright[2][0].z=0.8837;

pcarwinright[2][1].p.y=-3.1027;

pcarwinright[2][1].z=1.3159;

pcarwinright[2][2].p.y=-3.4861;

pcarwinright[2][2].z=1.3159;

pcarwinright[2][3].p.y=-3.8688;

pcarwinright[2][3].z=0.8758;

for(i=0;i<3;i++)

{

for(j=0;j<4;j++)

{

pcarwinleft[i][j]=pcarwinright[i][j];

pcarwinright[i][j].p.x=0.9;

pcarwinleft[i][j].p.x=-0.9;

}

}

}

Judgeend.c

//自定义头文件

#include "mytype.h"

//C运行时头文件

#include <graphics.h>

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <bios.h>

#include <string.h>

#include <dos.h>

#include <math.h>

//自定义头文件

#define MYMACRO extern

#include "common.h"

void judgetoendline()

{

char str[50];

double time1=(clock()-start)/CLOCKS\_PER\_SEC;

setcolor(RED);

if(sumdrive/sumpathauto>=1)

{

timeme=time1;

if(sumdrive>=sumauto)

{

sprintf(str,"No1,%6.1fs:me",timeme);

//outtextxy(160,4\*16,str);

outtextxy(0,10\*16,str);

timeauto=timeme+(sumpathauto-sumauto)/vauto;

sprintf(str,"No2,%6.1fs:auto",timeauto);

//outtextxy(160+1\*240,4\*16,str);

outtextxy(0,11\*16,str);

}

else

{

sprintf(str,"No1,%6.1fs:auto",timeauto);

//outtextxy(160,4\*16,str);

outtextxy(0,10\*16,str);

sprintf(str,"No2,%6.1fs:me",timeme);

//outtextxy(160+1\*240,4\*16,str);

outtextxy(0,11\*16,str);

}

cmd |= CMD\_QUIT;

}

else

{

if (sumauto / sumpathauto <= 1)

{

timeauto=(finish-start)/CLOCKS\_PER\_SEC;

}

}

setcolor(WHITE);

}

Key.c

#include "mytype.h"

//C运行时头文件

#include <graphics.h>

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <bios.h>

#include <string.h>

#include <dos.h>

#include <math.h>

//自定义头文件

#define MYMACRO extern

#include "common.h"

int getkey()

{

R=0;

throttledrive=0;

cmd = GetCommand();

if (cmd & CMD\_QUIT)

{

return 1;

}

if (cmd & CMD\_LEFT)

{

R= -rdrive;

}

if (cmd & CMD\_RIGHT)

{

R= rdrive;

}

//油门

if (cmd & CMD\_UP)

{

throttledrive=1;

}

if (cmd&CMD\_DOWN)

{

throttledrive=-1;

}

//按下、松开空格键，trig为1，则换档

if ((trig & OLD\_SPACE)==1)

{

trig &=(~OLD\_SPACE);//因为松开的时间比动画间时间长，需要trig清零

myshift++;

if(myshift>2)

myshift=2;

}

//按z降档

if ((trig&OLD\_Z) == OLD\_Z)

{

trig &=(~OLD\_Z);

myshift--;

if(myshift<0)

myshift=0;

}

return 0;

}

//返回键盘按下状态

int GetCommand()

{

return keystate;

}

//键盘中断处理程序

void interrupt newint9()

{

unsigned char scan\_code,PB;

//开屏蔽中断

asm sti

//取扫描码

scan\_code=inportb(0x60);

if(scan\_code & 0x80)//是松开按键（最高位为1）

switch(scan\_code & 0x7F)//低7位

{

case SCAN\_LEFT:

keystate &=(~CMD\_LEFT);

break;

case SCAN\_RIGHT:

keystate &=(~CMD\_RIGHT);

break;

case SCAN\_UP:

keystate &=(~CMD\_UP);

break;

case SCAN\_DOWN:

keystate &=(~CMD\_DOWN);

break;

case SCAN\_SPACE:

oldstate &=(~OLD\_SPACE);//空格前一状态清零

break;

case SCAN\_Z:

oldstate &=(~OLD\_Z);//Z前一状态清零

break;

case SCAN\_ESCAPE:

keystate &=(~CMD\_QUIT);

break;

}

else//是按下按键

switch(scan\_code)

{

case SCAN\_LEFT:

keystate |=CMD\_LEFT;

break;

case SCAN\_RIGHT:

keystate |=CMD\_RIGHT;

break;

case SCAN\_UP:

keystate |=CMD\_UP;

break;

case SCAN\_DOWN:

keystate |=CMD\_DOWN;

break;

case SCAN\_SPACE:

if((oldstate & OLD\_SPACE) == 0)//前一状态是松开，则触发

{

trig |= OLD\_SPACE;//空格触发位置1

oldstate |= OLD\_SPACE;//空格前一状态置1

}

else

trig &= (~OLD\_SPACE);//空格触发位清零

break;

case SCAN\_Z:

if((oldstate & OLD\_Z) == 0)//前一状态是松开，则触发

{

trig |= OLD\_Z;//Z触发位置2

oldstate |= OLD\_Z;//Z前一状态置2

}

else

trig &= (~OLD\_Z);//Z触发位清零

break;

case SCAN\_ESCAPE:

keystate |=CMD\_QUIT;

break;

}

//发EOI

outportb(0x20,0x20);

}

//设新键盘中断

void setint9()

{

oldint9=getvect(9);

setvect(9,newint9);

}

//回复原键盘中断

void recoverint9()

{

setvect(9,oldint9);

}

Race.c

#include "mytype.h"

#include <graphics.h>

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <bios.h>

#include <string.h>

#include <dos.h>

#include <math.h>

#define MYMACRO extern

#include "common.h"

void carRace(int\*x)

{

setint9();

Game\_Init();

displayinf0();

do

{

\*x=Game\_Main();

mydelay(dly);

} while (!(cmd & CMD\_QUIT));

//恢复原键盘中断处理程序

recoverint9();

}

void Game\_Init()

{

GlobalInit();

roadinit();

start=clock();

finish=start;

sumpathauto=getlooplen(autoroad[pathno]);//计算自动行驶小车路线周长

displayinf();

setcamera(mypos,myangpos);//根据汽车位置设相机参数

toushitu();

}

int Game\_Main()

{

int x;

x=getkey();

autoruncar(autoroad[pathno]);

drivecar();

proccarcollide();

setcamera(mypos,myangpos);//根据汽车位置设相机参数

clearclient();

//pingmiantu();

toushitu();

if(itext==0) displayinf();

judgetoendline();

itext++;

itext%=4;//每4帧变零

return x;

}

//清屏

void clearclient()

{

setfillstyle(SOLID\_FILL,LIGHTCYAN);

setcolor(LIGHTCYAN);

bar(scrW-scrH,0,scrW,scrH/2-1);//动画区天空清屏

setfillstyle(SOLID\_FILL,GREEN);

setcolor(GREEN);

bar(0,scrH/2,scrW,scrH);//动画区地面清屏

if(itext==0)

{

setfillstyle(SOLID\_FILL,LIGHTCYAN);

setcolor(LIGHTCYAN);

bar(startclr,0,scrW-scrH-1,scrH/2-1);//动态数字区清屏

}

else

{

//动态数字区靠尾20像素区清屏

setfillstyle(SOLID\_FILL,LIGHTCYAN);

setcolor(LIGHTCYAN);

bar(140,0,159,239);

}

setfillstyle(SOLID\_FILL,WHITE);//恢复为白色

setcolor(WHITE);

}//延迟，clock函数的时间单位，对8086是1000/18.2=55ms

void mydelay(clock\_t td)

{

clock\_t start1;

start1=clock();

while(clock()<start1+td);

}

double getlooplen(ROADAREA ra[])

{

int i;

double sum=0;

LINE line1;

ARC arc1;

for (i = 0;i < 2\*segs;i++)

{

if (strcmp(ra[i].type, "line") == 0)

{

line1=\*(PLINE)(ra[i].pcurve);

sum+=pp\_dist(line1.p1,line1.p2);

}

else

{

arc1=\*(PARC)(ra[i].pcurve);

sum+=arc1.r\*arc1.dang;

}

}

return sum;

}

Setcam.c

//自定义头文件

#include "mytype.h"

//C运行时头文件

#include <graphics.h>

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <bios.h>

#include <string.h>

#include <dos.h>

#include <math.h>

//自定义头文件

#define MYMACRO extern

#include "common.h"

//让相机在车后3m处

void setcamera(MYPOINT p1,double ang)

{

angsight=ang;

pntsight.x=p1.x+(yminsight+carL+3)\*cos(angsight-pi/2);

pntsight.y=p1.y+(yminsight+carL+3)\*sin(angsight-pi/2);

}

Toushi.c

//自定义头文件

#include "mytype.h"

//C运行时头文件

#include <graphics.h>

#include <stdio.h>

#include <conio.h>

#include <time.h>

#include <bios.h>

#include <string.h>

#include <dos.h>

#include <math.h>

//自定义头文件

#define MYMACRO extern

#include "common.h"

void toushitu()

{

int i;

if(toushi==1)

{

draw\_solidloop(2\*segs,roadareaout);//绘制跑道外边界透视图

for(i=0;i<2;i++)

{

draw\_dashloop(2\*segs,roadareadash[i]);//绘制跑道中心线透视图

}

draw\_solidloop(2\*segs,roadareain);//绘制跑道内边界透视图

drawendline();

draw\_lanloop(2\*segs,roadareaout);//绘制跑道外边界栏杆

draw\_lanloop(2\*segs,roadareain);//绘制跑道内边界栏杆

//draw\_treeloop(2\*segs,roadareaout);//绘制跑道外边界树

//draw\_treeloop(2\*segs,roadareain);//绘制跑道内边界树

drawautocarface();

drawdrivecarface();

}

}

void drawendline()

{

LINE lineout;

int n;

n=seeline(endline.p1,endline.p2,&lineout);

if(n>0) draw\_line(lineout);

}

//透视实线公路边界线

void draw\_solidloop(int nseg,ROADAREA ra[])

{

LINE linein,lineout;

ARC arcin,arcout[9];

int n,i,j;

for(i=0;i<nseg;i++)

{

if(strcmp(ra[i].type,"line")==0)//直线段

{

linein=\*(PLINE)(ra[i].pcurve);

n=seeline(linein.p1,linein.p2,&lineout);

if(n>0) draw\_line(lineout);

}

else//圆弧

{

arcin=\*(PARC)(ra[i].pcurve);

n=seearc(arcin,arcout);

for(j=0;j<n;j++)

draw\_solidarc(arcout[j]);

}

}

}

//透视公路虚线

void draw\_dashloop(int nseg, ROADAREA ra[])

{

LINE linein,lineout;

ARC arcin,arcout[9];

int n,i,j;

for(i=0;i<nseg;i++)

{

if(strcmp(ra[i].type,"line")==0)//直线段

{

linein=\*(PLINE)(ra[i].pcurve);

n=seeline(linein.p1,linein.p2,&lineout);

if(n>0) draw\_dashline(linein,lineout);

}

else//圆弧

{

arcin=\*(PARC)(ra[i].pcurve);

n=seearc(arcin,arcout);

for(j=0;j<n;j++)

draw\_dasharc(arcin,arcout[j]);

}

}

}

void draw\_solidarc(ARC arc1)

{

double ang1,dang1;

int ndiv,i;

ndiv=(int)(arc1.dang\*arc1.r/2.0)+1;

ang1=arc1.angstart;

dang1=arc1.dang/ndiv;

arc1.dang=dang1;

for (i = 0;i < ndiv;i++)

{

arc1.angstart=ang1;

draw\_arc(arc1);

ang1+=dang1;

}

}

//画道路虚直线，虚线在后半段

//点1在虚线段，

void draw\_dashline(LINE linearea,LINE line)

{

int at1,at2,n1,n2,n,m,i,nstart;

LINE line1;

double ang;

ang=pp\_angle(linearea.p1,linearea.p2);

at1=wheredashline(linearea,line.p1,&n1);

at2=wheredashline(linearea,line.p2,&n2);

if (n2 == n1)//在同一段

{

if (at1 == 1)//点1在虚线上

{

if (at2 == 1)//点2在虚线上

{

draw\_line(line);

}

}

else//点1在间隔上

{

if (at2 == 1)//点2在虚线上

{

line1.p1=getpangpnt(linearea.p1,ang,(n1+0.5)\*dashlen);

line1.p2=line.p2;

draw\_line(line1);

}

}

}

else//在不同段

{

if (at1 == 1)//首不完整虚线

{

line1.p1=line.p1;

line1.p2=getpangpnt(linearea.p1,ang,(n1+1)\*dashlen);

draw\_line(line1);

}

if(at2==1)//尾不完整虚线

{

line1.p1=getpangpnt(linearea.p1,ang,(n2+0.5)\*dashlen);

line1.p2=line.p2;

draw\_line(line1);

}

//完整虚线

n=n2-n1-1;//中间完整虚线数

if(at1==1)//点1在虚线段，中间完整段从n1+1开始

{

nstart=n1+1;

}

else//点1不在虚线段，中间连续段数加1

{

n++;

nstart=n1;//从可视起点段开始

}

for (i = 0;i < n;i++)//中间完整虚线

{

m=nstart+i;

line1.p1=getpangpnt(linearea.p1,ang,(m+0.5)\*dashlen);

line1.p2=getpangpnt(linearea.p1,ang,(m+1)\*dashlen);

draw\_line(line1);

}

}

}

//画道路虚弧线，虚线在后半段

void draw\_dasharc(ARC arcarea,ARC arc)

{

int at1,at2,n1,n2,n,m,i,nstart;

ARC arc1;

double ang1,ang2;

arc1=arcarea;

ang1=arc.angstart;

ang2=ang1+arc.dang;

at1=wheredasharc(arcarea,ang1,&n1);

at2=wheredasharc(arcarea,ang2,&n2);

if (n2 == n1)//在同一段

{

if (at1 == 1)//点1在虚线上

{

if (at2 == 1)//点2在虚线上

{

draw\_solidarc(arc);

}

}

}

else//在不同段

{

if (at1 == 1)//首不完整虚线

{

arc1.angstart=ang1;

arc1.dang=stdangle0(arcarea.angstart+(n1+1)\*dashang-ang1);

draw\_solidarc(arc1);

}

if(at2==1)//尾不完整虚线

{

arc1.angstart=arcarea.angstart+(n2+0.5)\*dashang;

arc1.dang=stdangle0(ang2-arc1.angstart);

draw\_solidarc(arc1);

}

//完整虚线

n=n2-n1-1;//中间完整虚线数

if(at1==1)

{

nstart=n1+1;

}

else//可视起点段加入中间连续段

{

n++;

nstart=n1;//从可视起点段开始

}

for (i = 0;i < n;i++)//中间完整虚弧线

{

m=nstart+i;

arc1.angstart=arcarea.angstart+(m+0.5)\*dashang;

arc1.dang=dashang/2;

draw\_solidarc(arc1);

}

}

}

//画有宽度的直线段

void draw\_line(LINE line1)

{

MYPOINT point[4];

int i;

int poly[8];

double a;

if(pp\_dist(line1.p1,line1.p2)<0.01) return;//若直线长度过短，则退出

a=atan2(line1.p2.y-line1.p1.y,line1.p2.x-line1.p1.x);

point[0].x = line1.p1.x + w \* cos(a - pi / 2);

point[0].y = line1.p1.y + w \* sin(a - pi / 2);

point[3].x = line1.p1.x + w \* cos(a + pi / 2);

point[3].y = line1.p1.y + w \* sin(a + pi / 2);

point[1].x = line1.p2.x + w \* cos(a - pi / 2);

point[1].y = line1.p2.y + w \* sin(a - pi / 2);

point[2].x = line1.p2.x + w \* cos(a + pi / 2);

point[2].y = line1.p2.y + w \* sin(a + pi / 2);

for (i = 0;i < 4;i++)

{

x2xp(pntsight, angsight, point[i], &point[i]);//转到相机坐标系

Pnt\_Toushi(point[i], &point[i]);//计算单位屏幕高透视点

//透视坐标值乘以屏幕高，存入poly数组

poly[2\*i] =(int)(scrH\*point[i].x+xcen);

poly[2\*i+1] = (int)(scrH\*point[i].y+ycen);

}

fillpoly(4, poly);//画线段

}

//三等分画宽度为2w的圆弧透视图

void draw\_arc(ARC arc2)

{

int poly[16],i;

double x0,y0,r,theta, dang;

MYPOINT p,pnt\_p,pnt\_t;

if(arc2.dang\*arc2.r<0.01) return;//若直线长度过短，则退出

x0=arc2.x;

y0=arc2.y;

r=arc2.r;

theta=arc2.angstart;

dang=arc2.dang/3.0;//圆弧ndiv-1等分

for (i = 0;i < 4;i++)

{

p.x = x0+(r + w)\*cos(theta);

p.y = y0+(r + w)\*sin(theta);

x2xp(pntsight, angsight, p, &pnt\_p);

Pnt\_Toushi(pnt\_p, &pnt\_t);

//透视坐标值乘以屏幕高，存入poly数组

poly[2 \* i] = (int)(pnt\_t.x \*scrH+xcen);

poly[2 \* i + 1] =(int)( pnt\_t.y \* scrH+ycen);

p.x = x0+(r - w)\*cos(theta);

p.y = y0+(r - w)\*sin(theta);

x2xp(pntsight, angsight, p, &pnt\_p);

Pnt\_Toushi(pnt\_p, &pnt\_t);

//透视坐标值乘以屏幕高，存入poly数组

poly[2 \* (7 - i)] = (int)(pnt\_t.x \* scrH+xcen);

poly[2 \* (7 - i) + 1] =(int)(pnt\_t.y \* scrH+ycen);

theta += dang;

}

fillpoly(8, poly);

}

//计算点在虚线上第n个周期，返回是否在虚线上（在后半段则在）

int wheredashline(LINE line, MYPOINT pnt, int \*n)

{

int atline=0,m;

double d;

double ang=pp\_angle(line.p1,line.p2);

MYPOINT pp,p21;

x2xp(line.p1,ang,pnt,&pp);

x2xp(line.p1,ang,line.p2,&p21);

d=pp.x/dashlen;

m=(int)d;

//后半部分为虚线的线区

if(d-m>=0.5) atline=1;

\*n=m;

return atline;

}

//计算点在虚圆弧线上第n个周期，返回是否在虚线上（在后半段则在）

int wheredasharc(ARC arc1,double ang, int \*n)

{

int atarc=0,m;

double d;

double ang1;

ang1=stdangle0(ang-arc1.angstart);

d=ang1/dashang;

m=(int)d;

if(d-m>=0.5) atarc=1;

\*n=m;

return atarc;

}

//计算一线段（全局坐标）与视场边界交点及在视场内部端点，返回值为点数，参数里还返回点（全局坐标）。最多两个点，一段。

//以下版本是转到p1p2局部坐标，交点y坐标为0，便于比较是否重合，容易看懂

int seeline(MYPOINT p1, MYPOINT p2, LINE \*line1)

{

int n = 0,i,j,j1,m;//在视场内部端点或交点总数

MYPOINT p11={0,0}, p21;//p1、p2在p1p2局部坐标系的坐标

MYPOINT pp1[5];

MYPOINT pseg[2];//存放内部端点或交点

double dx,dy,a;

MYPOINT ptmp[2],ptemp;//临时记录交点

int nn;//临时记录交点数

dx = p2.x - p1.x;

dy = p2.y - p1.y;

a = atan2(dy, dx);

x2xp(p1, a, p2, &p21);//p2转到p1p2局部坐标（原点是p1，转动角度是a）

//视场点转到p1p2局部坐标

for (j = 0;j < 4;j++)

{

xp2x(pntsight, angsight, p[j], &pp1[j]);//视场点转到全局坐标

x2xp(p1, a, pp1[j], &pp1[j]);//视场点转到p1p2局部坐标

}

pp1[4] = pp1[0];

//计算视场内的点

if (judgein(p1) && judgein(p2))//都在视场里，结束函数

{

line1->p1 = p1;

line1->p2 = p2;

return 1;

}

else if (judgein(p1))//只p1在视场里

{

pseg[0] = p11;

n = 1;

}

else if (judgein(p2))//只p2在视场里

{

pseg[0] = p21;

n = 1;

}

//计算交点

for (i = 0;i < 4;i++)//与视场每条边进行计算

{

line\_intersection(p11, p21, pp1[i], pp1[i + 1], &nn, ptmp);

for(j=0;j<nn && n<2;j++)//对每个交点进行判断是否存入pseg数组，n小于2才循环

{

double x=ptmp[j].x;

j1=0;

for (m = 0;m < n;m++)//与pseg数组里的点都不重合的交点才存入pseg数组

{

if(fabs(x-pseg[m].x)<1e-6)

{

j1=1;//重合，j1置1

break;

}

}

if(j1==0) pseg[n++]=ptmp[j];//与pseg数组里的点都不重合，存入pseg数组

}

}

if (n == 2)//有两个点，说明视场里有一段直线

{

if (pseg[0].x > pseg[1].x)//排序

{

ptemp=pseg[0];

pseg[0]=pseg[1];

pseg[1]=ptemp;

}

for (j = 0;j < 2;j++)

{

xp2x(p1, a, pseg[j], &pseg[j]); //交点转回全局坐标

}

line1->p1 = pseg[0];

line1->p2 = pseg[1];

return 1;

}

else

return 0;

}

//计算一圆弧（全局坐标）在视场内弧段，返回值为弧段数，参数里还返回弧段ARC结构值（全局坐标）

int seearc(ARC arc2, ARC arc3[])

{

int n,i,j,m,j1;//交底或内部点总数

int nn,num;//

MYPOINT intersection[2], p1[5], middle, ps, pe, point[10];

double a, ang[10], dang[9], temp;

double x0,y0,ang0,R;

x0=arc2.x;

y0=arc2.y;

ang0=arc2.angstart;

R=arc2.r;

//视场点转到全局坐标

for (i = 0;i < 4;i++)

{

xp2x(pntsight, angsight, p[i], &p1[i]);

}

p1[4] = p1[0];

//计算圆弧端点

ps.x = x0 + R\*cos(ang0);

ps.y = y0 + R\*sin(ang0);

a = ang0 + arc2.dang;

pe.x = x0 + R\*cos(a);

pe.y = y0 + R\*sin(a);

//圆弧端点若在视场内，存入point数组，并且与圆弧起始角的夹角存入ang数组

n = 0;

if (judgein(ps))

{

point[n] = ps;

ang[n++]=0.0;

}

if (judgein(pe))

{

point[n] = pe;

ang[n++]=arc2.dang;

}

//视场四条边与圆弧的交点存入point数组

for (i = 0;i < 4;i++)

{

arcline\_intersection(p1[i], p1[i + 1], arc2, intersection, &nn);

for(j=0;j<nn;j++)//对每个交点进行判断是否存入pseg数组

{

a=atan2(intersection[j].y-y0,intersection[j].x-x0)-ang0;

a=stdangle0(a);

j1=0;

for (m = 0;m < n;m++)//与point数组里的点都不重合的交点才存入point数组

{

if(fabs(a-ang[m])<1e-8)

{

j1=1;//重合，j1置1

break;

}

}

if(j1==0)

{

point[n]=intersection[j];//与point数组里的点都不重合，存入point数组，

ang[n++]=a;//与圆弧起始角的夹角存入ang数组

}

}

}

//夹角排序

for (i = 0;i<n - 1;i++)

for (j = i + 1;j < n;j++)

{

if (ang[i] > ang[j])

{

temp = ang[i];

ang[i] = ang[j];

ang[j] = temp;

}

}

//计算视场内的弧段

num=0;

if(n>=2)

{

//计算各弧段扫过的角度，存入dang数组

for (i = 0;i < n-1;i++)

{

dang[i]=ang[i+1]-ang[i];

}

for (i = 0;i < n-1;i++)

{

temp = ang[i] + ang0;

a = temp + dang[i] / 2;

middle.x = x0 + R\*cos(a);

middle.y = y0 + R\*sin(a);

if (judgein(middle))

{

arc3[num] = arc2;

arc3[num].angstart = temp;

arc3[num++].dang = dang[i];

}

}

}

return num;

}

//地面点转为透视点，透视点坐标对应屏幕高度为1的值，相机距离地面camH，垂向为z轴，向前看为y轴，向右为x轴，

//透视坐标系原点在屏幕左上角，x轴向右为正，y轴向下为正。

//点在地面

void Pnt\_Toushi(MYPOINT pnt, MYPOINT \*pnt\_t)

{

double x, dist, fact;

x = pnt.x;

dist = pnt.y;

fact = 0.5 / (dist\*tan(angvmax));

pnt\_t->x = x \* fact;

pnt\_t->y = camH \*fact;

}

//地面点转为透视点，透视点坐标对应屏幕高度为1的值，相机距离地面camH，垂向为z轴，向前看为y轴，向右为x轴，

//透视坐标系原点在屏幕左上角，x轴向右为正，y轴向下为正。

//z为点距离地面的距离

void Pnt\_ToushiZ(MYPOINT pnt,double z, MYPOINT \*pnt\_t)

{

double x, dist, fact,H;

x = pnt.x;

dist = pnt.y;

H=dist\*tan(angvmax);

fact = 0.5 / H;

pnt\_t->x = x \* fact;

pnt\_t->y = (camH-z) \*fact;

}

//判断1个点是否在视场内

//需要视点位置pntsight、视场坐标相对全局坐标转动角度angsight，视场的近距yminsight、远距ymaxsight、

//视场左边界角a2hsight、视场右边界角a1hsight为全局变量，输入参数为p1

int judgein(MYPOINT p1)

{

MYPOINT p2;

double b;

int flag = 0;

x2xp(pntsight, angsight, p1, &p2);//转到视场局部坐标

if (p2.y > yminsight && p2.y < ymaxsight)

{

b = atan2(p2.y, p2.x);

if (b > a1hsignt && b < a2hsignt)

{

flag = 1;

}

}

return flag;

}

//计算一三维线段（全局坐标）与视场边界交点及在视场内部端点，返回值为点数，参数里返回视场内三维线段（全局坐标）。

//适用于铅垂线，水平线，一般空间线

int see3Dline(MY3DPOINT p1, MY3DPOINT p2, LINE3D \*line3D)

{

int n = 0;

MYPOINT p2D1,p2D2,p2D3,p2D4;

LINE line;

double dx,dy,dz,z1,z3,z4;

MY3DPOINT p3,p4;

double p1p2,p1p3,p1p4;

//线段端点p1、p2转换为二维点

p2D1=p1.p;

p2D2=p2.p;

p1p2=pp\_dist(p2D1,p2D2);

if(p1p2<1e-6)//铅垂线

{

if(judgein(p2D1))//在视场内

{

n=2;

line3D->p1=p1;

line3D->p2=p2;

}

}

else//水平线，或一般空间线

{

//求视场内二维线段p3p4

n=seeline(p2D1,p2D2,&line);

if(n>0)//在视场内有线段

{

//计算p1p3、p1p4长度

p2D3=line.p1;

p2D4=line.p2;

p1p3=pp\_dist(p2D1,p2D3);

p1p4=pp\_dist(p2D1,p2D4);

//计算p3、p4的z坐标

z1=p1.z;

dz=p2.z-p1.z;

z3=z1+p1p3/p1p2\*dz;

z4=z1+p1p4/p1p2\*dz;

//返回三维线段

p3.p=p2D3;

p3.z=z3;

p4.p=p2D4;

p4.z=z4;

line3D->p1=p3;

line3D->p2=p4;

}

}

return n;

}

//画三维线段的透视图

void drawcar3Dline(LINE3D line3D)

{

MYPOINT pp1,pp2;

MYPOINT p1,p2;

int x1,y1,x2,y2;

int x11,y11,x22,y22;

//转端点p1

//二维点

p1=line3D.p1.p;

x2xp(pntsight, angsight, p1, &pp1);//转到相机坐标系

Pnt\_ToushiZ(pp1,line3D.p1.z, &p1);//计算单位屏幕高透视点

//透视坐标值乘以屏幕高，移到屏幕中心

x1 =(int)(scrH\*p1.x+xcen);

y1 = (int)(scrH\*p1.y+ycen);

//转端点p2

//二维点

p2=line3D.p2.p;

x2xp(pntsight, angsight, p2, &pp2);//转到相机坐标系

Pnt\_ToushiZ(pp2,line3D.p2.z, &p2);//计算单位屏幕高透视点

//透视坐标值乘以屏幕高，移到屏幕中心

x2 =(int)(scrH\*p2.x+xcen);

y2 = (int)(scrH\*p2.y+ycen);

//画透视线

line(x1,y1,x2,y2);

}

//画轮子透视图，pcen为轮子中心在车子自身坐标系下的三维坐标

void drawwheel(MY3DPOINT pcen)

{

int i,n;

double y0,z0,r,theta,dang;

MYPOINT pp;

MY3DPOINT pnt,p3D[21];

LINE3D lineout;

//在车子自身坐标系下

y0=pcen.p.y;

z0=pcen.z;

r=rwheel;

theta=0;

dang=2\*pi/20;//20等分

pnt.p.x = pcen.p.x;

for (i = 0;i <= 20;i++)

{

//轮子圆上20等分点在车子自身坐标系下的三维坐标

pnt.p.y =y0+r\*cos(theta);

pnt.z = z0+r\*sin(theta);

//转到全局坐标

xp2x(pos,angpos, pnt.p, &pp);

p3D[i].p=pp;

p3D[i].z=pnt.z;

theta += dang;

}

//20等分折线透视

for (i = 0;i < 20;i++)

{

n=see3Dline(p3D[i],p3D[i+1],&lineout);

if(n>0)

drawcar3Dline(lineout);

}

}

//画三维多边面，n为顶点数

void draw3Dfill(int n,MY3DPOINT p3D[],int poly[])

{

int i;

MYPOINT pp,p1;

for(i=0;i<n;i++)

{

x2xp(pntsight, angsight,p3D[i].p, &pp);//转到相机坐标系

Pnt\_ToushiZ(pp,p3D[i].z, &p1);//计算单位屏幕高透视点

//透视坐标值乘以屏幕高，移到屏幕中心

poly[2\*i] =(int)(scrH\*p1.x+xcen);

poly[2\*i+1] = (int)(scrH\*p1.y+ycen);

}

//画透视四边形

fillpoly(n, poly);

}

//画车轮面，输入为车位置，右侧与x轴夹角，外侧中心和内侧中心在小车自身坐标系下的三维点

void drawwheelface(MYPOINT pos1,double angpos1,MY3DPOINT pcenout,MY3DPOINT pcenin)

{

int poly[20];

int i,n;

double y0,z0,r,theta,dang;

MYPOINT pp;

MY3DPOINT pnt,p3Dout[11],p3Din[11],p3D[4];

LINE3D lineout;

//在车子自身坐标系下

y0=pcenout.p.y;

z0=pcenout.z;

r=rwheel;

theta=0;

dang=2\*pi/10;

//车轮外侧面等分点

pnt.p.x = pcenout.p.x;

for (i = 0;i <= 10;i++)

{

//轮子圆上等分点在车子自身坐标系下的三维坐标

pnt.p.y =y0+r\*cos(theta);

pnt.z = z0+r\*sin(theta);

//转到全局坐标

xp2x(pos1,angpos1, pnt.p, &pp);

p3Dout[i].p=pp;

p3Dout[i].z=pnt.z;

theta += dang;

}

p3Dout[10]=p3Dout[0];

//车轮内侧面等分点

//y0=pcenin.p.y;

//z0=pcenin.z;

//r=rwheel;

theta=0;

pnt.p.x = pcenin.p.x;

for (i = 0;i <= 10;i++)

{

//轮子圆上等分点在车子自身坐标系下的三维坐标

pnt.p.y =y0+r\*cos(theta);

pnt.z = z0+r\*sin(theta);

//转到全局坐标

xp2x(pos1,angpos1, pnt.p, &pp);

p3Din[i].p=pp;

p3Din[i].z=pnt.z;

theta += dang;

}

p3Din[10]=p3Din[0];

//画内侧面

setfillstyle(SOLID\_FILL,BLACK);

setcolor(WHITE);

draw3Dfill(10,p3Din,poly);

//画滚动面

setcolor(BLACK);

for(i=0;i<10;i++)

{

p3D[0]=p3Din[i];

p3D[1]=p3Dout[i];

p3D[2]=p3Dout[i+1];

p3D[3]=p3Din[i+1];

draw3Dfill(4,p3D,poly);

}

//画外侧面

setcolor(WHITE);

draw3Dfill(10,p3Dout,poly);

}

//获得轮子上半圆的三维等点，从后到前，n为等分数

void wheelup(MY3DPOINT pcen,int n,MY3DPOINT p3Dup[])

{

int i;

double y0,z0,r,theta,dang;

MYPOINT pp;

MY3DPOINT pnt;

//在车子自身坐标系下

y0=pcen.p.y;

z0=pcen.z;

r=rwheel;

theta=pi;

dang=pi/n;

//车轮外侧面等分点

pnt.p.x = pcen.p.x;

for (i = 0;i <= n;i++)

{

//轮子圆上等分点在车子自身坐标系下的三维坐标

pnt.p.y =y0+r\*cos(theta);

pnt.z = z0+r\*sin(theta);

//转到全局坐标

xp2x(pos,angpos, pnt.p, &pp);

p3Dup[i].p=pp;

p3Dup[i].z=pnt.z;

theta -= dang;

}

}

//画自动小车面，在相机前1米才画

void drawautocarface()

{

enum COLORS autocolor=LIGHTMAGENTA;

MYPOINT pp;

MYPOINT p0={0,0},p1,p2,p3,p4,ptmp;

MY3DPOINT p3Dleft[20],p3Dright[20];//车子侧面的全局三维点

MY3DPOINT p3Dwin[4];

MY3DPOINT p3D[10];

LINE3D lineout;

LINE line1;

int poly[40];

int n;

int i,j;

//侧轮廓点转到全局坐标

for (i = 0;i < 7;i++)

{

xp2x(pos,angpos,pcarleft[i].p,&pp);

p3Dleft[i].p=pp;

p3Dleft[i].z=pcarleft[i].z;

xp2x(pos,angpos,pcarright[i].p,&pp);

p3Dright[i].p=pp;

p3Dright[i].z=pcarright[i].z;

}

//左轮上半圆等分点全局坐标

wheelup(pcarcenleft[1],5,&p3Dleft[7]);

wheelup(pcarcenleft[0],5,&p3Dleft[13]);

//右轮上半圆等分点全局坐标

wheelup(pcarcenright[1],5,&p3Dright[7]);

wheelup(pcarcenright[0],5,&p3Dright[13]);

//检查是否在相机前1米，用0、6点

x2xp(pntsight, angsight,p3Dleft[0].p,&p1);//转到相机坐标系

x2xp(pntsight, angsight,p3Dleft[6].p,&p2);//转到相机坐标系

x2xp(pntsight, angsight,p3Dright[0].p,&p3);//转到相机坐标系

x2xp(pntsight, angsight,p3Dright[6].p,&p4);//转到相机坐标系

if(p1.y<1 || p2.y<1 || p3.y<1 || p4.y<1)

return;

setfillstyle(SOLID\_FILL,autocolor);

//画顶面

for(i=0;i<6;i++)

{

//四边形顶点存到数组

p3D[0]=p3Dleft[i];

p3D[1]=p3Dright[i];

p3D[2]=p3Dright[i+1];

p3D[3]=p3Dleft[i+1];

draw3Dfill(4,p3D,poly);

}

//判断是否左侧可见

x2xp(pntsight, angsight,p3Dleft[0].p,&p1);//转到相机坐标系

x2xp(pntsight, angsight,p3Dleft[6].p,&p2);//转到相机坐标系

if (p1.y > p2.y)

{

ptmp=p1;

p1=p2;

p2=ptmp;

}

if (stdangle(pp\_angle(p0, p2) - pp\_angle(p0, p1)) > 0)//左侧可见

{

//画轮子,两个左轮，内在前外在后；右后轮，外在前内在后

for(i=0;i<2;i++)

{

drawwheelface(pos,angpos,pcarcenleft[i],pcarcenleftin[i]);

}

drawwheelface(pos,angpos,pcarcenrightin[1],pcarcenright[1]);

//画左侧

setfillstyle(SOLID\_FILL,autocolor);

draw3Dfill(19,p3Dleft,poly);

//画左窗

//左窗点转到全局坐标

for(i=0;i<3;i++)

{

for (j = 0;j < 4;j++)

{

xp2x(pos,angpos,pcarwinleft[i][j].p,&pp);

p3Dwin[j].p=pp;

p3Dwin[j].z=pcarwinleft[i][j].z;

}

draw3Dfill(4,p3Dwin,poly);

}

}

else//右侧可见

{

//画轮子,两个右轮，内在前外在后；左后轮外在前内在后

for(i=0;i<2;i++)

{

drawwheelface(pos,angpos,pcarcenright[i],pcarcenrightin[i]);

}

drawwheelface(pos,angpos,pcarcenleftin[1],pcarcenleft[1]);

//画右侧

setfillstyle(SOLID\_FILL,autocolor);

draw3Dfill(19,p3Dright,poly);

//画右窗

//右窗点转到全局坐标

for(i=0;i<3;i++)

{

for (j = 0;j < 4;j++)

{

xp2x(pos,angpos,pcarwinright[i][j].p,&pp);

p3Dwin[j].p=pp;

p3Dwin[j].z=pcarwinright[i][j].z;

}

draw3Dfill(4,p3Dwin,poly);

}

}

setfillstyle(SOLID\_FILL,WHITE);//恢复为白色

}

//画交互小车面

void drawdrivecarface()

{

MYPOINT pp;

MYPOINT p0={0,0},p1,p2,p3,p4,ptmp;

MY3DPOINT p3Dleft[20],p3Dright[20];//车子侧面的全局三维点

MY3DPOINT p3D[10];

int poly[20];

int i;

//侧轮廓点转到全局坐标

for (i = 0;i < 7;i++)

{

xp2x(mypos,myangpos,pcarleft[i].p,&pp);

p3Dleft[i].p=pp;

p3Dleft[i].z=pcarleft[i].z;

xp2x(mypos,myangpos,pcarright[i].p,&pp);

p3Dright[i].p=pp;

p3Dright[i].z=pcarright[i].z;

}

//画后轮

drawwheelface(mypos,myangpos,pcarcenleftin[1],pcarcenleft[1]);

drawwheelface(mypos,myangpos,pcarcenright[1],pcarcenrightin[1]);

setcolor(LIGHTMAGENTA);

//画顶面

setfillstyle(SOLID\_FILL,mecolor);

for(i=3;i<6;i++)

{

//四边形顶点存到数组

p3D[0]=p3Dleft[i];

p3D[1]=p3Dright[i];

p3D[2]=p3Dright[i+1];

p3D[3]=p3Dleft[i+1];

draw3Dfill(4,p3D,poly);

}

setfillstyle(SOLID\_FILL,WHITE);//恢复为白色

setcolor(WHITE);

}

//画栏杆

void draw\_lanloop(int nseg,ROADAREA ra[])

{

LINE linein,lineout;

ARC arcin,arcout[9];

int n,i,j;

setfillstyle(SOLID\_FILL,DARKGRAY);

setcolor(DARKGRAY);

for(i=0;i<nseg;i++)

{

if(strcmp(ra[i].type,"line")==0)//直线段

{

linein=\*(PLINE)(ra[i].pcurve);

n=seeline(linein.p1,linein.p2,&lineout);

if(n>0)

{

//画栏杆水平线

//draw\_lanline(lineout);

//画栏杆垂直线

draw\_lanvline(linein,lineout,draw\_lanvbase);

}

}

else//圆弧

{

arcin=\*(PARC)(ra[i].pcurve);

n=seearc(arcin,arcout);

for(j=0;j<n;j++)

{

//画栏杆水平线

//draw\_lanarc(arcout[j]);

//画栏杆垂直线

draw\_lanvarc(arcin,arcout[j],draw\_lanvbase);

}

}

}

setfillstyle(SOLID\_FILL,WHITE);

setcolor(WHITE);

}

//画圆弧栏杆

void draw\_lanarc(ARC arc1)

{

MY3DPOINT p3D[4];

int poly[8];

double ang1,dang1;

int ndiv,i;

LINE line;

//约2m等分

ndiv=(int)(arc1.dang\*arc1.r/2.0)+1;

ang1=arc1.angstart;

dang1=arc1.dang/ndiv;

arc1.dang=dang1;

for (i = 0;i < ndiv;i++)

{

arc1.angstart=ang1;

//画每一段圆弧

line.p1=getarcpnt(arc1,ang1);

line.p2=getarcpnt(arc1,ang1+dang1);

draw\_lanline(line);

ang1+=dang1;

}

}

//画水平线栏杆，输入为二维线段

void draw\_lanline(LINE line1)

{

MY3DPOINT p3D[4];

int poly[8];

p3D[0].p=line1.p1;

p3D[0].z=0.5;

p3D[1].p=line1.p2;

p3D[1].z=0.5;

p3D[2]=p3D[1];

p3D[2].z+=0.05;

p3D[3]=p3D[0];

p3D[3].z+=0.05;

draw3Dfill(4,p3D,poly);

}

//画直线上的垂直栏杆

void draw\_lanvline(LINE linearea,LINE line,BASEFUNCPTR funptr)

{

int at1,at2,n1,n2,n,m,i,nstart;

double ang,angx;

MYPOINT pnt;

ang=pp\_angle(linearea.p1,linearea.p2);

angx=ang-pi/2;

//判断在后半段则为1

at1=wheredashline(linearea,line.p1,&n1);

at2=wheredashline(linearea,line.p2,&n2);

//转换为在前半段为1

at1=(at1==0?1:0);

at2=(at1==0?1:0);

if (n2 == n1)//在同一段

{

if (at1 == 1)//点1在本段中点前

{

if (at2 == 0)//点2在本段中点后

{

pnt=getpangpnt(linearea.p1,ang,(n1+0.5)\*dashlen);

//画垂直栏杆

funptr(pnt,angx);

}

}

}

else//在不同段

{

//点1在中点前，则起始段为n1，否则为n1+1

//点1在中点前，栏杆数在中间栏杆数基础上加1

//点2在中点后，栏杆数在中间栏杆数基础上加1

n=n2-n1-1;//中间栏杆数

if(at1==1)//点1在本段中点前

{

nstart=n1;

n++;

if (at2 == 0)//点2在本段中点后

n++;

}

else//点1在本段中点后

{

nstart=n1+1;

if (at2 == 0)//点2在本段中点后

n++;

}

for (i = 0;i < n;i++)//画栏杆

{

m=nstart+i;

pnt=getpangpnt(linearea.p1,ang,(m+0.5)\*dashlen);

funptr(pnt,angx);

}

}

}

//画圆弧线上的垂直栏杆，按角度等分为多段，求可见圆弧段所包围的栏杆点

void draw\_lanvarc(ARC arcarea,ARC arc,BASEFUNCPTR funptr)

{

int at1,at2,n1,n2,n,m,i,nstart;

ARC arc1;

double ang1,ang2,ang;

MYPOINT pnt;

arc1=arcarea;

ang1=arc.angstart;

ang2=ang1+arc.dang;

//判断在后半段则为1

at1=wheredasharc(arcarea,ang1,&n1);

at2=wheredasharc(arcarea,ang2,&n2);

//转换为在前半段为1

at1=(at1==0?1:0);

at2=(at1==0?1:0);

if (n2 == n1)//在同一段

{

if (at1 == 1)//点1本段中点前

{

if (at2 == 0)//点2在本段中点后

{

ang=arcarea.angstart+(n1+0.5)\*dashang;//栏杆处的径向

pnt=getarcpnt(arcarea,ang);

funptr(pnt,ang);

}

}

}

else//在不同段

{

//点1在中点前，则起始段为n1，否则为n1+1

//点1在中点前，栏杆数在中间栏杆数基础上加1

//点2在中点后，栏杆数在中间栏杆数基础上加1

n=n2-n1-1;//中间栏杆数

if(at1==1)//点1在本段中点前

{

nstart=n1;

n++;

if (at2 == 0)//点2在本段中点后

n++;

}

else//点1在本段中点后

{

nstart=n1+1;

if (at2 == 0)//点2在本段中点后

n++;

}

for (i = 0;i < n;i++)//画栏杆

{

m=nstart+i;

ang=arcarea.angstart+(m+0.5)\*dashang;//栏杆处的径向

pnt=getarcpnt(arcarea,ang);

funptr(pnt,ang);

}

}

}

//由二维点和路线的法线方向，画一根垂直栏杆

void draw\_lanvbase(MYPOINT pnt,double ang)

{

MY3DPOINT p3D[4];

int poly[8];

LINE line1;

line1.p1=getpangpnt(pnt,ang+pi/2,0.025);

line1.p2=getpangpnt(pnt,ang-pi/2,0.025);

p3D[0].p=line1.p1;

p3D[0].z=0;

p3D[1].p=line1.p2;

p3D[1].z=0;

p3D[2]=p3D[1];

p3D[2].z+=0.5;

p3D[3]=p3D[0];

p3D[3].z+=0.5;

draw3Dfill(4,p3D,poly);

}

//画树

void draw\_treeloop(int nseg,ROADAREA ra[])

{

LINE linein,lineout;

ARC arcin,arcout[9];

int n,i,j;

setfillstyle(SOLID\_FILL,DARKGRAY);

setcolor(DARKGRAY);

for(i=0;i<nseg;i++)

{

if(strcmp(ra[i].type,"line")==0)//直线段

{

linein=\*(PLINE)(ra[i].pcurve);

n=seeline(linein.p1,linein.p2,&lineout);

if(n>0)

{

draw\_lanvline(linein,lineout,draw\_treebase);

}

}

else//圆弧

{

arcin=\*(PARC)(ra[i].pcurve);

n=seearc(arcin,arcout);

for(j=0;j<n;j++)

{

//画栏杆水平线

//draw\_lanarc(arcout[j]);

//画栏杆垂直线

draw\_lanvarc(arcin,arcout[j],draw\_treebase);

}

}

}

setfillstyle(SOLID\_FILL,WHITE);

setcolor(WHITE);

}

//由二维点和路线的法线方向，画一颗树

void draw\_treebase(MYPOINT pnt,double ang)

{

MY3DPOINT p3D[4];

int poly[8];

LINE line1;

//画树干

line1.p1=getpangpnt(pnt,ang+pi/2,0.025);

line1.p2=getpangpnt(pnt,ang-pi/2,0.025);

p3D[0].p=line1.p1;

p3D[0].z=0;

p3D[1].p=line1.p2;

p3D[1].z=0;

p3D[2]=p3D[1];

p3D[2].z+=1;

p3D[3]=p3D[0];

p3D[3].z+=1;

draw3Dfill(4,p3D,poly);

//画树冠

line1.p1=getpangpnt(pnt,ang+pi/2,0.75);

line1.p2=getpangpnt(pnt,ang-pi/2,0.75);

p3D[0].p=line1.p1;

p3D[0].z=1;

p3D[1].p=line1.p2;

p3D[1].z=1;

p3D[2].p=pnt;

p3D[2].z=3;

draw3Dfill(3,p3D,poly);

}

# 总结

曾文正总结

自己之前也学过c语言，但是这是我第一次运用c语言，写出一个可以使用的程序，和以前做做题相比，这是一个质的飞跃。

课设还没开始时就听说学院的课设是一个艰巨的任务，但当自己真正开始做课设的时候，感觉其实并没有那么的恐怖。在暑假的时候，我阅读了C高级上的许多章节，书中有许多许多我不知道的东西，不论是较为底层的代码，还是一些功能的实现等。看着书，仿佛开启了新世界的大门。渐渐地，我对C语言的理解又变深了。

我写代码习惯用整段时间编写比如节假日几整天，或者平时一整个晚上，利用连续的时间，才能有充分地考虑各种问题。中秋节我两整天坐在寝室里，凭借自己的思路不断地敲着代码，这种感觉真的很好。需求分析验收时，老师说我们并没有把整体的程序架构建立清楚。我吸取了教训，整个国庆节将自己要编写的功能梳理了一遍，将自己的代码改进、增加，形成了体系。这让我明白了需求分析阶段设计的重要性，有一个好的框架，实现代码会变得清晰。开始的第一步并不是直接开始想到什么就敲什么代码，而是应该想全面，并模块化，这样才能使后续编写代码的过程更顺畅的进行。

我是一个不熬夜的人，但做课设的时候，不免经常遇到bug，有时候调试很久才能解决。如果晚上发现了bug，那么我一定会知道解决为止才回去睡觉，仿佛自己在和bug战斗，直到战胜它为止，我才会心满意足，我知道，思路是不应该间断的，一定要解决后，才能休息。作为一个程序员，我绝不会放弃战斗。

不得不说，许多bug不好解决，很多问题真的是前一发动全身，有的被自己忽略的地方，往往会出现问题，所以要做到每一行代码都没有漏洞。我也学会了使用debug来调试，这大大提高了效率。然后有些东西虽然知道了问题出在了哪里，但还是不知道为什么会出现问题，很多问题真的是冥思苦想之后，才找到了解决方案。不得不说，会写代码重要，会解决bug更重要，bug的种类各种各样。每次解决不同的bug，都会让我收获很多，这无疑让我的思维更加严谨，思路更清晰。改bug的过程真的是忘不掉！

写代码是一方面，将代码精简又是一方面。开始我写的代码，很多地方结构不够清晰。虽然有模块，但很多地方还是有重复。后来，我一步步精简了自己的代码，让结构清晰了起来，所谓的代码量并没有意义，能让代码层次清晰，才重要。

几个月的课设让我收获的不仅是C语言的知识，还有许多方面的经验。并且，在过程中，我也享受了身为一个程序员不断开辟思路，与bug作斗争的过程。通过自己的努力，付出自己的心血与热情去完成一件事，是一个很享受的过程。

许可晗总结

本以为，上半年一个学期的c语言学习已经让我得以窥探到c语言的全貌，然而，在拿到c语言课程设计的题目的时候我才意识到，我也只是如同沙滩上玩沙的孩子，在我的前方，还有着广袤无垠的大海等待着我的探索。

编写课程设计让我对原先学的c语言知识有了更深入的理解。在学习c语言的过程中，我曾有些疑惑。比如，我不明白本来可以在主函数里用一些语句就能解决的问题为什么非要编成函数，并且进行麻烦的参数传递，还有函数的参数传递为什么要用那么多种形式等等。同时，对于c语言中的指针我也很费解，平时编程序也用得很少。或许在上个学期中，我仅仅是了解到了那些语法规则和它们如何使用。而我并不清楚应该在什么情况下用这些学的知识，用这些东西到底有什么好处。而当开始编写这样相对来说比较庞大的工程的时候，我体会到了当很多时候要用到一些共性的，重复的代码时，我可以把它编成函数，即可以让我少写一些代码，也可以增强程序的可读性。而用了很多的函数之后我也体会到了函数参数用地址传递的好处，这样在可以在函数中修改函数之外的参数，而且可以用来返回多个值。很多时候编写都会用指针作为参数。通过编写程序，让我对数据类型，条件，循环语句，函数，指针，数组有了更好的了解。在编程的过程中，我也用到了一些曾经用得不太多甚至没有接触到的知识。比如说宏定义，结构和枚举，绘图函数。一开始运用的时候我还需要对着书上的例子作为参考，而且即使这样也会有一些错误存在。经过了多次地应用，我最终也能掌握它们。

同时，在这个过程之中我也体会到了c语言编程的思想—模块化。编写一个程序最开始不应直接就开始对着电脑敲代码，而是去分析问题，将一个大问题划归为很多小问题，也就是分成很多步骤，很多模块，再一步一步，一个一个模块的问题去解决。这样的话，即使再难得问题也能被划分成很多简单的问题，从而得以解决。其实，这不仅仅是编程的思想，在解决实际问题当中，我们也会遇到很多看似难以解决的问题，而通过这种编程的思想方法，我们也可以将其解决。因此，编写课程设计让我们了解的不仅仅是c语言本身，还让我们掌握了一种分析，处理，解决问题的效思想和手段。

此外，由于在编程过程中的诸多错误，为了解决它们，我也学会了调试。我了解到了编程不仅仅是写代码，调试也是其中很重要的一个环节。对于编程中一些不尽如人意的结果，我们要学会调试，确定错误产生的范围，再逐步确认到错误产生的语句，从而解决它们。