

# 农田无人机喷洒农药模拟系统

C语言课程设计

**分**

**析**

**报**

**告**

**专业班级：自动化卓越班2301班**

**小组成员：**

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

**上交时间：二〇二四年四月二十七日**

目录

[第一部分 前言 1](#_Toc55022825)

[一、 编写背景 1](#_Toc55022826)

[二、 参考资料 1](#_Toc55022828)

[三、 参考软件 1](#_Toc55022829)

[第二部分 任务概述 2](#_Toc55022830)

[一、 目标功能 2](#_Toc55022831)

[二、 编写规范 2](#_Toc55022832)

[第三部分 运行环境和配置 3](#_Toc55022833)

[一、 硬件接口 3](#_Toc55022834)

[二、 软件接口 3](#_Toc55022835)

[三、 控制 3](#_Toc55022836)

[第四部分 需求分析与系统设计 3](#_Toc55022837)

[一、 需求分析 3](#_Toc55022838)

[二、 系统设计 3](#_Toc55022839)

[第五部分 算法设计 11](#_Toc55022840)

[一、 数据结构设计 11](#_Toc55022841)

[二、 算法说明 12](#_Toc55022842)

[第六部分 界面设计 15](#_Toc55022843)

[一、 界面设计 15](#_Toc55022845)

[第七部分 源代码 29](#_Toc55022847)

[第八部分 个人总结 189](#_Toc55022848)

[一、 组员总结 189](#_Toc55022849)

[二、 组员总结 190](#_Toc55022850)

[第九部分 时间安排与代码分工 191](#_Toc55022851)

[一、 时间安排 191](#_Toc55022852)

[二、 代码分工 192](#_Toc55022853)

# 前言

## 编写背景

多年来，病虫害一直是影响粮食产量的主要问题之一，因此，我国每年需要进行大量的植保作业。无人机能够与农作物的距离保持在1~3米的固定高度，喷洒规模作业能达到每小时80~100亩，其效率要比常规喷洒高出100倍。农用无人机自动飞控导航系统作业最大限度地减少了工作人员接触农药的时间，保证了工作人员的安全。将无人机运用至农田当中具有许多优点。包括但不限于：1.高效率：飞行灵活，可以快速覆盖大面积的农田，提高喷洒效率。相比传统机械喷洒，无人机农药喷洒可以节约大量的时间和人力成本。2.安全高效：传统农药喷洒中，操作人员需要暴露在喷洒现场，面临农药的直接接触和风险。而无人机农药喷洒将农民和操作人员远离喷洒现场，降低了人身安全的风险。同时，无人机可以在复杂环境中精确操控，避免损坏作物和喷洒设备，提高农田管理的效果。3.环保可持续：无人机农药喷洒技术可以准确控制农药剂量，并大幅减少农药的使用量，从而减少环境和生态的影响。喷洒过程中，无人机可以围绕目标区域进行智能喷洒，避免喷洒到非目标区域，减少了对周围生态的影响。

在这次课程设计中，我们利用C语言开发了一个农田无人机喷洒农药模拟系统。它模拟了无人机在农田中的应用，如自动化生长检测，将检测结果自动生成，进行农药配置，智能实现最优路径规划，自动进行农药喷洒等功能。同时，它区别了大小两种农田，适用于更多的地形，环境，还考虑了无人机的避障。此外，我们针对一些点进行了高度模拟，比如生长检测中，不同植物的病的随机性，以及重复生长检测的病虫扩散性，植物自愈等情况。

本软件具有农田无人机喷洒农药的基本功能，同时考虑了实际生产的情况。

## 参考资料

1. 王士元. C高级实用程序设计. 北京: 清华大学出版社. 1996年

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

3. [美] Prata. C Primer Plus（第六版）北京：人民邮电出版社. 2016年

4. 严蔚敏，吴伟民编著. 数据结构（C语言版）北京：清华大学出版社. 2018年

## 参考软件

1. Borland C++ 3.1

2. Dev C++ 5.11

# 任务概述

## 目标功能

该软件可实现模拟农田无人机喷洒农药模拟系统，应该有如下基本功能：

* + - 1. 生长检测功能：对农田进行扫描检测其是否得病。
      2. 结果查看功能：生长检测过后，要能够直观的呈现出生长检测的结果。
      3. 农药配置功能：当检测过后，要能够针对不同的病虫害进行农药配置。
      4. 路径自动规划功能：智能生成最短最优路径，并呈现出来。
      5. 农药喷洒功能：实现避障，以及模拟无人机的喷洒农药。

## 编写规范

1. 相对独立的程序块之间、变量说明之后加空行，较长的语句分成多行书写
2. 大括号独占一行并对齐，缩进采用四个空格
3. 边写代码边注释，注释的量应达到让程序清晰易懂，部分令测试者比较难以理解的算法和流程应该给出相应的注释。
4. 标识符的命名清晰、明了，有明确含义
5. 命名规范：函数命名应该用英文表达其确定含义
6. 文件命名都用小写，并且表达出该文件所包含函数的主要功能。
7. 函数功能都要在函数原型后注明。

# 运行环境和配置

## 硬件接口

处理器：Intel Pentium 166 MX 或以上。

硬盘：空间500MB 以上。

屏幕适配器：VGA 接口。

系统运行内存：要求32MB 以上。

## 软件接口

开发软件工具：Borland C++ 3.1

文字编辑工具：Visual Studio 2019

数据库：文本存储（记事本）或MySQL。

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

## 控制

该系统通过鼠标直接进行控制。用户将鼠标移至需要操作的功能区进行点击。

# 需求分析与系统设计

## 需求分析

使用本软件的目标用户，是生产粮食的农场主，或者是普通的农民用户。

对使用“农田无人机喷洒农药模拟系统”的用户需求进行分析后，我们认为本软件首先应该要有广泛性，适用性。此外，还需要考虑无人机在实际生活生产中的应用。因此，我们加入了两个农田，模拟两种不同的适用对象和地点，地势。还加入了多种植物，考虑了无人机避开障碍物等等实际需要。

## 系统设计

1. **程序执行主流程**

图示

描述已自动生成

程序总体流程：运行本程序后，便是欢迎界面。其次跳转到主界面。主界面可以进入两个农田，以及本应用的说明书。进入农田，首先点击生长检测，进行结果查看。之后便可以进行农药配置，装配农药。自动路径规划后，无人机开始智能喷洒农药。

1. **平坝田模块**

**（1）总体流程**

图示

描述已自动生成

**（2）生长检测规律**

图片包含 图示

描述已自动生成

**生长检测时，如果前面已经生长检测过，而且并未进行农药喷洒，那么部分植物的发病情况会保留（甚至更加严重）。部分植物会自愈。这模拟了自然界的农作物实际的发病。**

**（3）结果查看**

图片包含 日程表

描述已自动生成

**通过生长检测从而获取到患病植物的坐标情况，从而计算出植物患病占比。从而绘制出患病柱形图。**

**（4）准备喷洒**

日程表

描述已自动生成

**进行“生长检测”后进入喷洒页面，可以对标明的对应某一植株的某一病状的农药进行勾选或取消勾选，进行勾选后点击“装配农药”进入页面，可以选择农药浓度，点击相应已经勾选的农药，小车装取农药后运输至无人机，显示农田界面，进入“路径规划”，点击路径规划，系统自动规划农药喷洒路径，点击准备喷洒，无人机根据已经规划的路径喷洒农药，喷洒结束后显示已经清除的病害种类，跳转回相应农田界面，可注意到相应病害已经清除。**

1. **小农田模块**

**（1）总体流程**

图示, 日程表

中度可信度描述已自动生成

**（2）生长检测部分**

图片包含 文本

描述已自动生成

**（3）结果查看部分**

图片包含 图示

描述已自动生成

**（4）准备喷洒部分**

日程表

描述已自动生成

**进行“生长检测”后进入喷洒页面，可以对标明的对应某一植株的某一病状的农药进行勾选或取消勾选，进行勾选后点击“装配农药”进入页面，可以选择农药浓度，点击相应已经勾选的农药，小车装取农药后运输至无人机，显示农田界面，进入“路径规划”，点击路径规划，系统自动规划农药喷洒路径，点击准备喷洒，无人机根据已经规划的路径，自行规避障碍，并喷洒农药，喷洒结束后显示已经清除的病害种类，跳转回相应农田界面，可注意到相应病害已经清除。**

# 算法设计

## 数据结构设计

本程序共定义了2个结构体，用于储存大农田中植物患病的坐标

typedef struct

{

int bfzuobiao[384][2];//

} niang; //储存黏虫的坐标信息；

typedef struct

{

int bfzuobiao[192][2];//

} dbfxy;//储存小麦的白粉病信息以及玉米的棉铃虫信息

这里定义了一个二维数组来储存坐标。其中小麦和玉米的患黏虫共用一个二维数组。该原理简单，故不作过多解释。

本程序共生成了16个文件 有MHMLDIAN ， MHXBDIAN，XMBFDIAN，XMNCDIAN，XTMHMIAN，XTMHXIU，XTMIANHU，XTXIAO，XTXMBAI，XTXMNIAN，XTYMMIAN，XTYMNIAN，XTYUMI，YMMLDIAN，YMMLDIAN，TIME。这16个文件。下面介绍这些文件以及其作用。

TIME文件储存当前时间信息。作用为实现时间每秒更新一次，防止一直调用MHMLDIAN ， MHXBDIAN，XMBFDIAN，XMNCDIAN， YMMLDIAN，YMMLDIAN这六个文件储存的是小农田中不同得病的点列坐标。主要目的是为了实现界面跳转时，植物的患病点保持不变。

XTXIAO，XTYUMI，XTMIANHU，这三个文件储存了小农田中三种植物的坐标。

XTMHMIAN，XTMHXIU，XTXMBAI，XTXMNIAN，XTYMMIAN，XTYMNIAN 这六个文件储存了患病植物的坐标。

## 二、算法说明

核心算法：路径规划

路径规划：

* + - 1. 创建二维数组rount，储存农药对应病害的坐标，数组的头尾储存无人机的初始位置。创建数组dist储存病害间的距离。
      2. int i=0 ; 对于第i个坐标，将目前序号大于i ，且与i 距离最小的病害坐标作为第i+1个坐标，i++,继续循环计算，直到病害坐标全被排序，将最后一个序号给予无人机初始位置。

路径动画：

int i=0; 在第i坐标和第i+1坐标之间画颜色为CYAN的线，无人机坐标为第i坐标，顺时针或逆时针识别周围8点，识别到CYAN则将坐标重置为此点，并用RED覆盖此点和原坐标点，继续循环识别，直到无法识别CYAN，则i++，继续画线并进行运动。

障碍物规避：

以第i坐标为原点建立坐标系，对第i+1坐标处于的象限进行分类，第i+1坐标所处象限决定无人机动画中颜色识别的方向（顺时针，逆时针）以及起始识别点，识别所确定的点，原坐标点，以及从确定的点沿识别方向的两个点均以红点覆盖。

if (rount[i][0] < rount[i + 1][0]&& rount[i][1] >= rount[i + 1][1])//第一象限

{

do

{

k = linemove21(6,x1, y1, buffer1);

} while (k == 1);

}

else if (rount[i][0] >= rount[i + 1][0] && rount[i][1] > rount[i + 1][1])//第二象限

{

do

{

k = linemove22(6, x1, y1, buffer1);

} while (k == 1);

}

else if (rount[i][0] > rount[i + 1][0] && rount[i][1] <= rount[i + 1][1])//第三象限

{

do

{

k = linemove22(0, x1, y1, buffer1);

} while (k == 1);

}

else if (rount[i][0] <= rount[i + 1][0] && rount[i][1] < rount[i + 1][1])//第三象限

{

do

{

k = linemove21(0, x1, y1, buffer1);

} while (k == 1);

}

int linemove21(int judge1, int\* x, int\* y, int\* buffer1)//小农田沿线段运动(顺时针)

{

int a[17][2] = { {0,-1},{1,-1},{1,0},{1,1},{0,1},{-1,1},{-1,0}, {-1,-1},{0,-1},{1,-1},{1,0},{1,1},{0,1},{-1,1},{-1,0}, {-1,-1},{0,-1} };

int b, judge = 0;

int i;

int x0 = 0, y0 = 0;

for (i = judge1; i < judge1+8; i++)

{

b = getpixel(a[i][0] + \*x, a[i][1] + \*y);

if (b == CYAN)

{

x0 = a[i][0];

y0 = a[i][1];

judge = 1;

break;

}

}

getimage(\*x, \*y, \*x + 20, \*y + 10, buffer1);

fushitu(\*x, \*y);

delay(8);

putimage(\*x, \*y, buffer1, COPY\_PUT);

if (judge == 1)

{

putpixel(\*x + a[i][0], \*y + a[i][1], RED);

putpixel(\*x + a[i+1][0], \*y + a[i+1][1], RED);

putpixel(\*x , \*y , RED);

\*x = \*x + x0;

\*y = \*y + y0;

putpixel(\*x,\*y, RED);

}

return judge;

}

int linemove22(int judge1, int\* x, int\* y, int\* buffer1)//小农田沿线段运动(逆时针)

{

int a[17][2] = { {0,-1}, {-1,-1}, {-1,0}, {-1,1}, {0,1} , {1,1}, {1,0}, {1,-1}, {0,-1}, {-1,-1}, {-1,0}, {-1,1}, {0,1} , {1,1}, {1,0}, {1,-1},{0,-1} };

int b, judge = 0;

int i;

int x0 = 0, y0 = 0;

for (i = judge1; i < judge1+8; i++)

{

b = getpixel(a[i][0] + \*x, a[i][1] + \*y);

if (b == CYAN)

{

x0 = a[i][0];

y0 = a[i][1];

judge = 1;

break;

}

}

getimage(\*x, \*y, \*x + 20, \*y + 10, buffer1);

fushitu(\*x, \*y);

delay(8);

putimage(\*x, \*y, buffer1, COPY\_PUT);

if (judge == 1)

{

putpixel(\*x + a[i][0], \*y + a[i][1], RED);

putpixel(\*x + a[i+1][0], \*y + a[i+1][1], RED);

putpixel(\*x, \*y, RED);

\*x = \*x + x0;

\*y = \*y + y0;

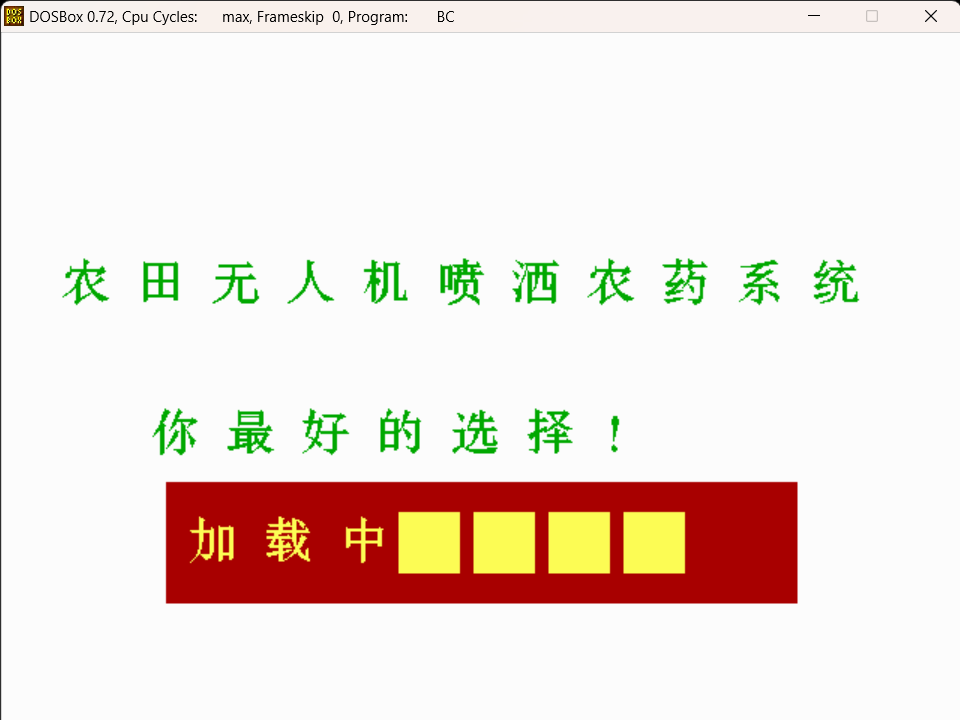
putpixel(\*x, \*y, RED);

}

return judge;

# 界面设计

1. 欢迎界面



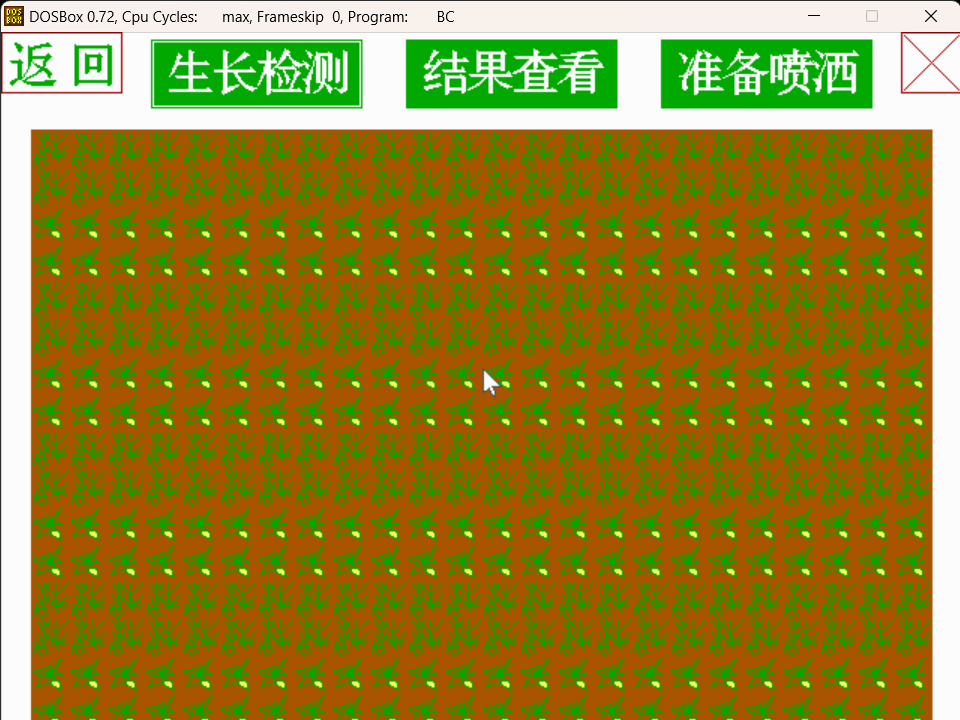
此界面是一个欢迎界面。以朴实无华的界面呈现出直观的功能。并做了一个加载动画。

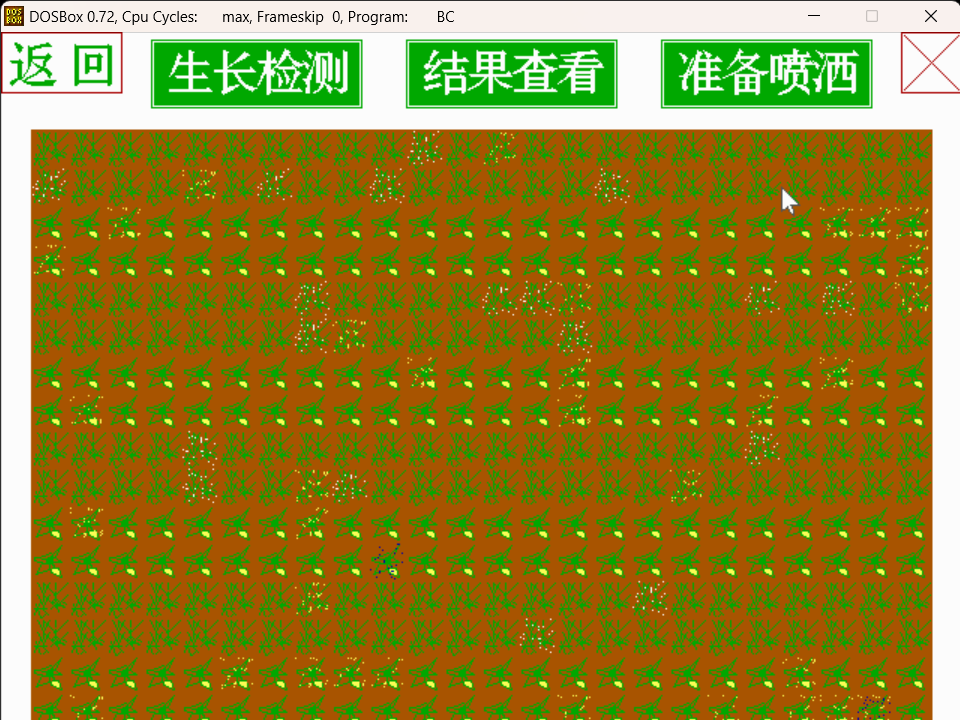
1. 主界面

文本, 表格

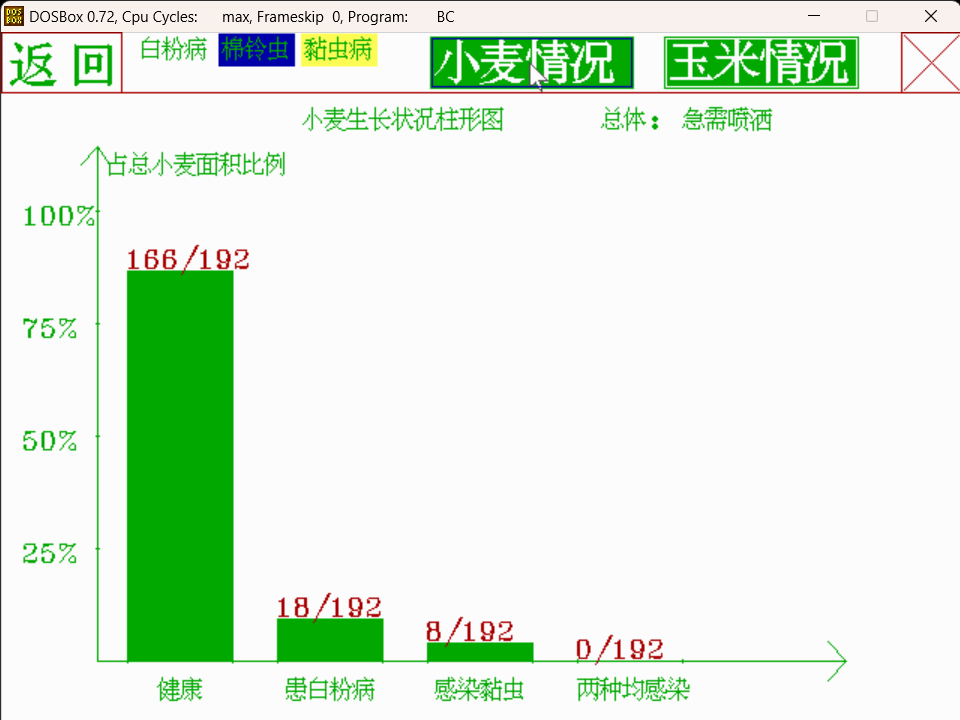
描述已自动生成

主界面是功能选择界面，相当于一个中转界面。能够实现界面跳转，从而进入三种不同的大功能。

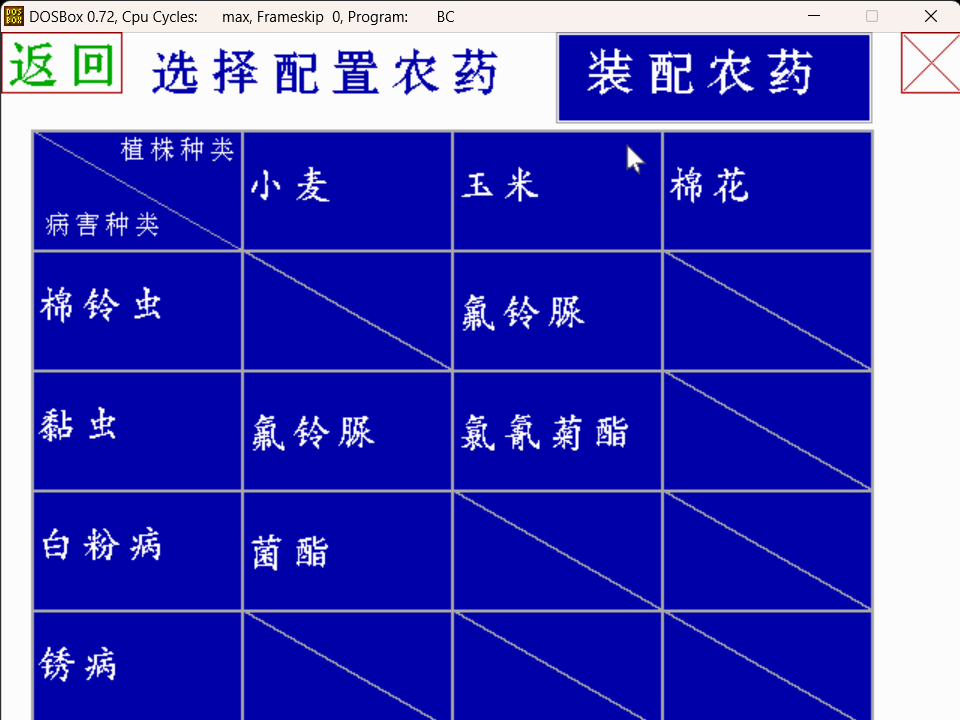
1. 平坝田界面



通过生长检测，直接在本界面生成患病植物。之后可以进行查看具体的患病情况。最后根据情况，进行农药配置。

1. 平坝田结果查看界面

使用图表，直观的展现植物的患病情况。

1. 平坝田配置农药界面

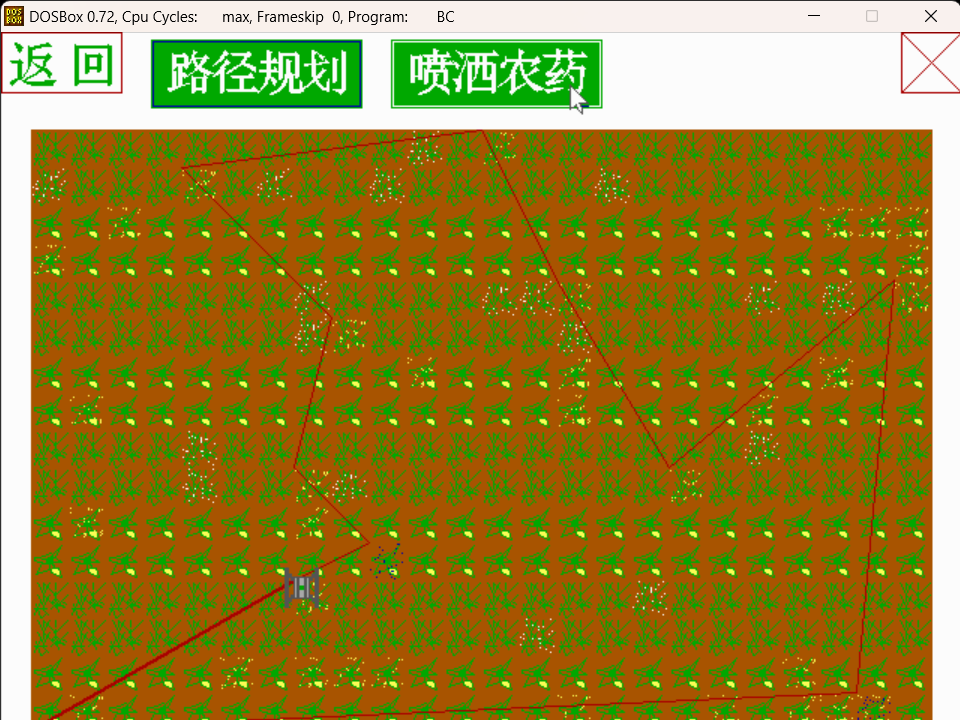
显示相应植株与相应病状所对应的农药种类，对所需要的农药进行勾选，或对已选择的农药取消勾选，点击“装配农药”后跳转至农药装配界面。

图示

描述已自动生成

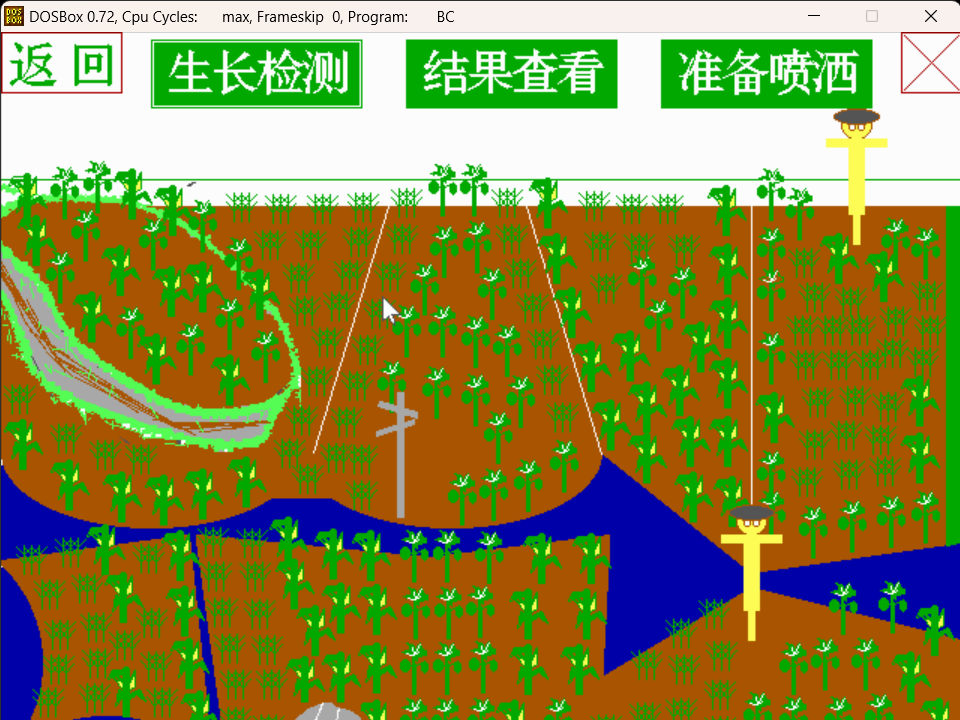
可以选择装配农药的浓度。在农药选择界面选择的农药可以在此界面显示，点击相应农药，则显示小车为无人机装配农药的动画，动画结束后跳转至路径规划界面。

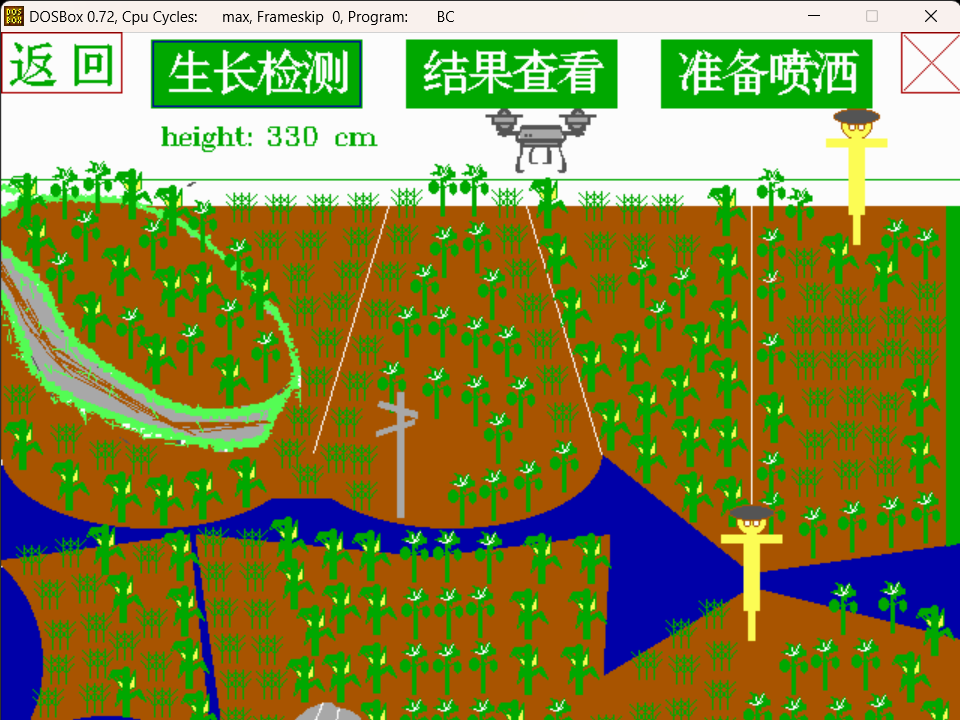
1. 平坝田农药喷洒界面图片包含 图形用户界面

   描述已自动生成

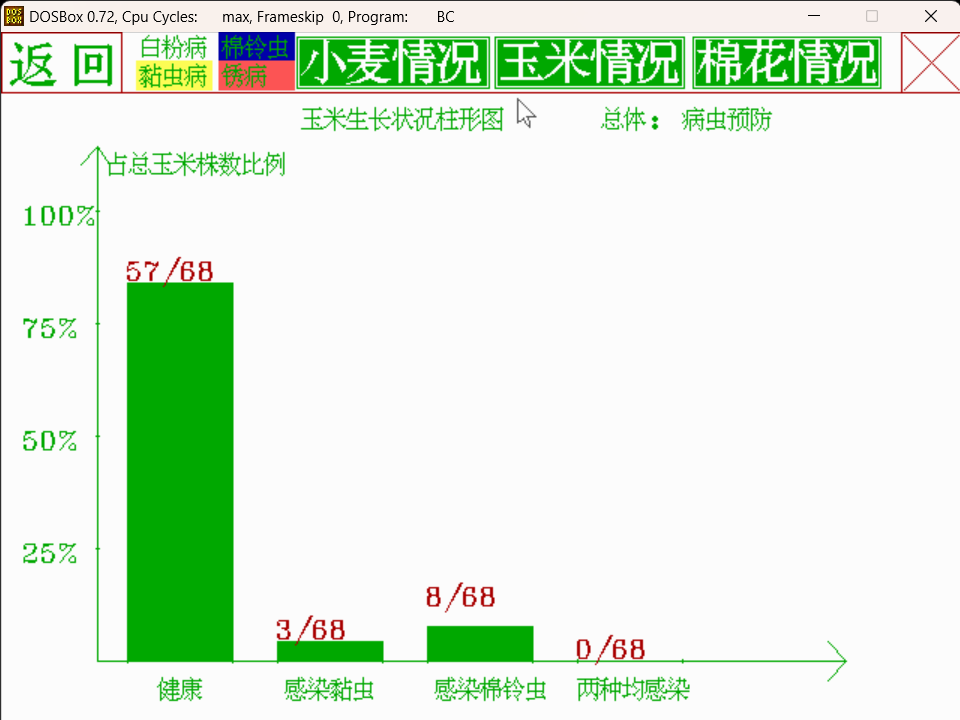
点击路径规划，系统自动规划路径，并在屏幕上显示路径。

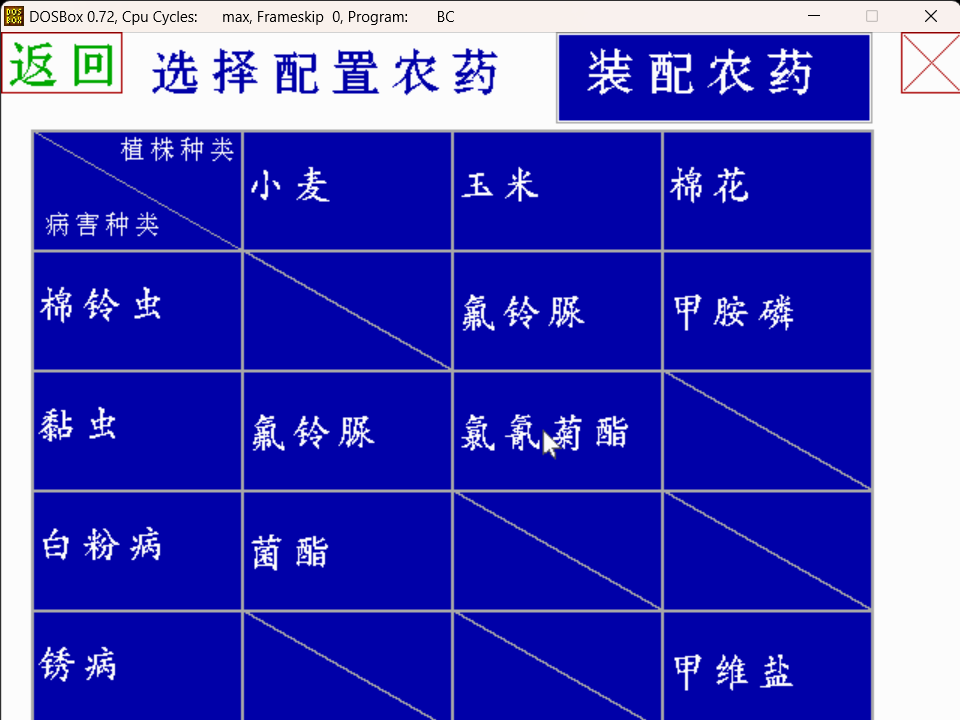
再点击喷洒农药，显示无人机根据路径喷洒农药的动画，农药喷洒后显示已经清除的相应植株的病害种类。跳转回相应农田界面，农田界面上相应病害已经被清除。

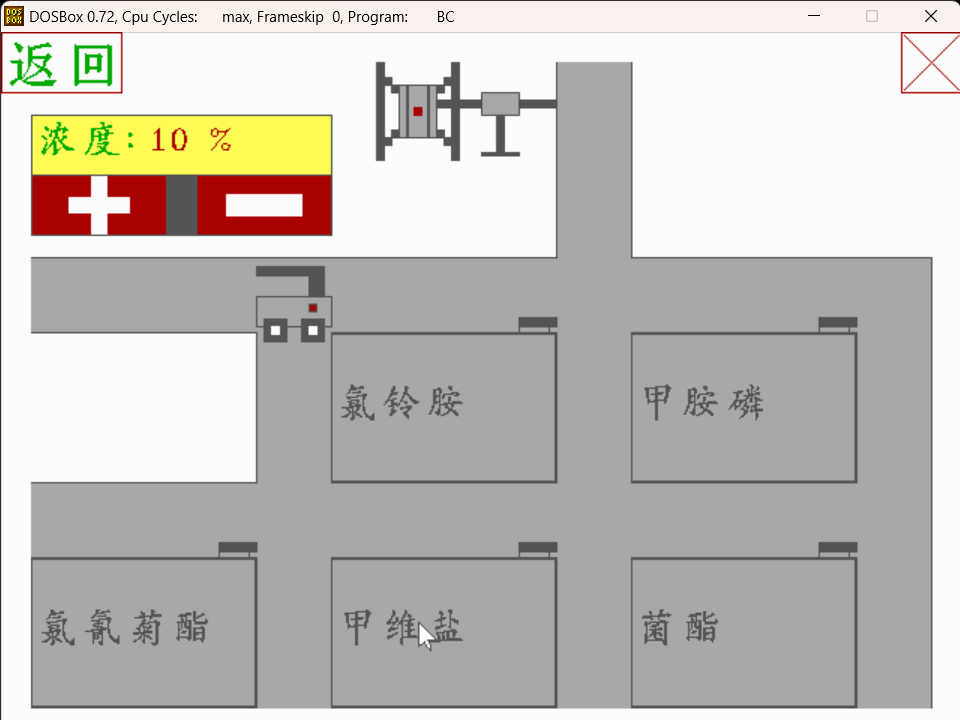
1. 小农田界面



实现了还原小农田。有生长检测和结果查看的功能。之后可以根据实际需求，进行农药喷洒。

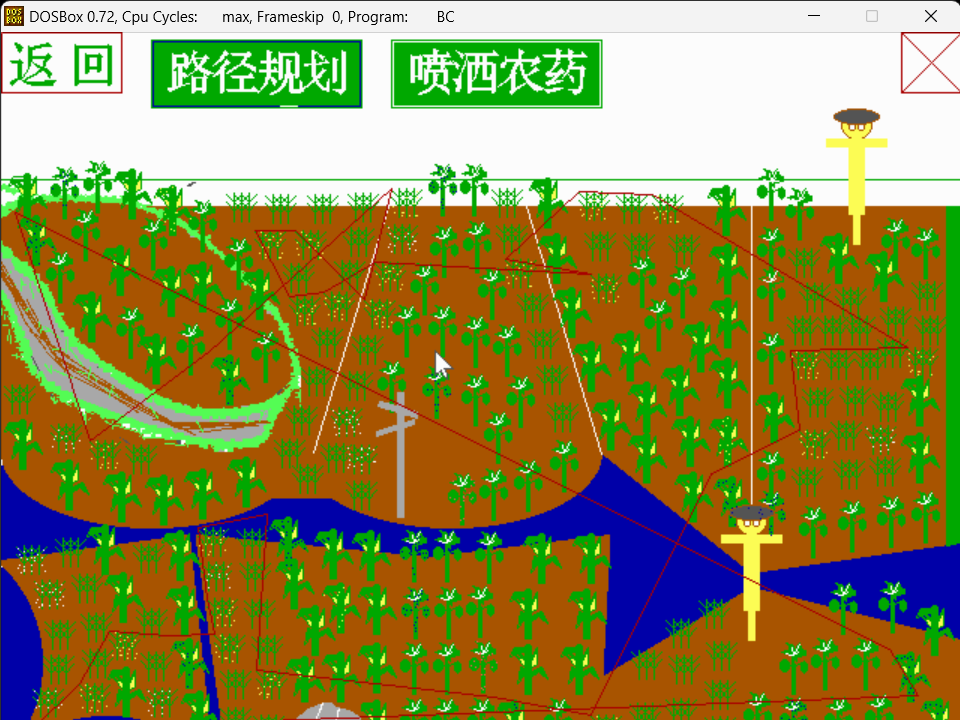
1. 小农田结果查看界面
2. 小农田配置农药界面

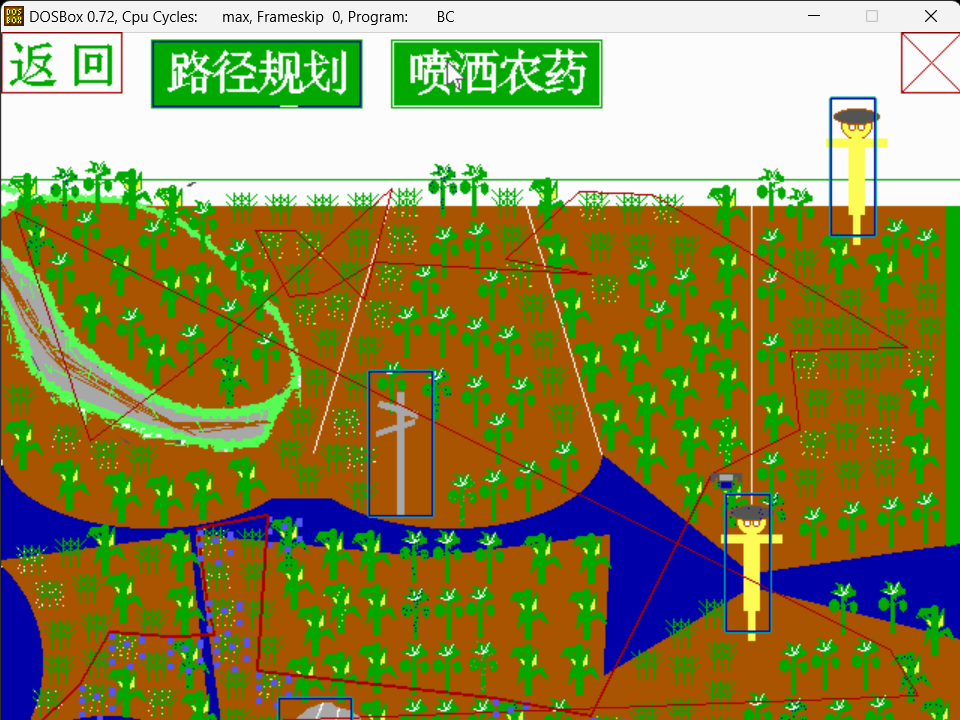




显示相应植株与相应病状所对应的农药种类，对所需要的农药进行勾选，或对已选择的农药取消勾选，点击“装配农药”后跳转至农药装配界面。

可以选择装配农药的浓度。在农药选择界面选择的农药可以在此界面显示，点击相应农药，则显示小车为无人机装配农药的动画，动画结束后跳转至路径规划界面。

1. 小农田农药喷洒界面



点击路径规划，系统自动规划路径，并在屏幕上显示路径。

再点击喷洒农药，显示无人机根据路径喷洒农药的动画，无人机自动翻阅所经过的障碍，农药喷洒后显示已经清除的相应植株的病害种类。跳转回相应农田界面，农田界面上相应病害已经被清除。

1. 系统说明界面文本, 信件

   描述已自动生成

一个简单的说明书，介绍本应用的基本流程。

# 第七部分 源代码

**1.main.c**

#include "common.h"

//程序控制台：负责页面跳转。

//实现各个函数的跳转，调用

void main(void)

{

int a1 = 1;

int a2 = 1;

int a3 = 1;

int a4 = 1;

int a5 = 1;

int a6 = 1;

int a7 = 1;

int a8 = 1;

int a9 = 1;

int a10 = 1;

int i, j;

int page;//页面值

//初始化VGA模式

int gd=VGA;//graphdriver

int gm=VGAHI;//graphmode

initgraph(&gd,&gm,"C:\\BORLANDC\\BGI");//这里后面的C:什么的看放的BGI文件路径

page=0;//初始化为0；

mouseinit();

while (1)

{

switch (page)

{

case 0:

page = moveint(welc, &a1);//欢迎界面，return：welc中的1//可参考 move.c。

break;

case 1:

page = moveint(hello, &a2);//主界面，return各种数字。

for (i = 0; i < 5; i++)

{

nongyao[i] = 0;

}

break;

case 2:

page = moveint1(mainfunc, &a3,diannian\_bitmap,dianbai\_bitmap,dianmian\_bitmap);//大农田界面。简单农田。

nongyao[5] = 0;

break;

case 3:

page = moveint( elsetian, &a4);//另一个奇怪的农田。进阶农田，

nongyao[5] = 1;

break;

case 4:

page = moveint(usepage, &a5);//系统说明界面，主要介绍了本系统的使用方法

break;

case 5:

page = moveint(elseoutcome, &a6);//小农田的结果查看界面。。

break;

case 6:

page = moveint(outcome, &a7);//查看大农田，说明农作物具体的生长状况，，

break;

case 7:

page = moveint2(zhuzhan, &a8, nongyao);

break;

case 8:

page = moveint2(zpjiemian, &a9, nongyao);//装配农药

break;

case 9:

page = moveint3(psdonghua, &a10, nongyao, diannian\_bitmap, dianbai\_bitmap, dianmian\_bitmap, rount, &amtjudge);//喷洒动画

break;

case -1://-1是点了 退出 框框；直接关闭BC

break;

default :break;

}

if (page==-1)

break;

}

delay(1000);//程序结束运行

closegraph();

}

**2.welcome.c**

/\*

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

欢迎界面

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

\*/

#include "common.h"

int welc(int \*a)//欢迎页

{

if (\*a == 1)

{

setbkcolor(WHITE);//background color

setcolor(GREEN);//边界color

setfillstyle(1, GREEN);//

puthz(40, 150, "农田无人机喷洒农药系统", 32, 50, GREEN);

puthz(100, 250, "你最好的选择！", 32, 50, GREEN);

\*a = 0;

}

////-------------加载动画

setfillstyle(1, RED);

bar(110, 300, 530, 380);

puthz(125, 322, "加载中", 32, 50, YELLOW);

setfillstyle(1, YELLOW);

bar(265, 320, 305, 360);

delay(400);

bar(315, 320, 355, 360);

delay(400);

bar(365, 320, 405, 360);

delay(400);

bar(415, 320, 455, 360);

delay(400);

bar(465, 320, 505, 360);

delay(400);

return 1;

}

**3.welcome.h**

#ifndef \_WELCOM\_H\_

#define \_WELCOM\_H\_

int welc(int \*a);//函数声明

#endif

**4.usehelp.c**

#include "common.h"

/\*========================

用户说明界面

简单的操作说明书

=========================\*/

int usepage(int\* a5)

{

if (\*a5 == 1)

{

cleardevice();

setbkcolor(WHITE);//background color

setcolor(RED);//边界color

setfillstyle(1, RED);//意思：设置绿色的填充模式为1（完全填充）

rectangle(600, 0, 640, 40);

line(600, 0, 640, 40);

line(640, 0, 600, 40);

rectangle(0, 0, 80, 40);

puthz(5, 5, "返回", 32, 40, GREEN);

puthz(250,10, "系统说明", 32, 30,GREEN);

setcolor(GREEN);//边界color

rectangle(10, 50, 635, 470);

puthz(10+5, 50+5, "该系统模拟了两个不同的农田，代表了两种不同的用户。平坝田地势平坦，一览无余，适用于种植大户，设置了小麦与玉米间种。小农田地势多样、具有障碍物，植物多种分布不匀杂乱。操作流程：首先进入农田，点击生长检测，此时无人机运行开始检测。结束后，可以在结果查看界面查看农作物的患病情况，并根据需要，选择配置相应的农药。农药装配到无人机后，无人机会自动进行最优路径规划，并进行农药喷洒。", 32, 30, GREEN);

\*a5 = 0;

}

showbiankuang(0, 0, 80, 40);

showbiankuang(600, 0, 640, 40);

//界面的跳转

if (mouse\_press(600, 0, 640, 40) == 1)

{

return -1;//关闭BC。

}

else if (mouse\_press(0, 0, 80, 40) == 1)

{

\*a5 = 1;

return 1;//

}

return 4;

}

**5.usehelp.h**

#ifndef \_USEHELP\_H\_

#define \_USEHELP\_H\_

int usepage(int\* a5);

#endif

**6.showtime.c**

#include "common.h"

#include "shuju.h"

#include "showtime.h"

/\*-----------------------------------------------------

展示时间，具体时间可以自行修改，显示的坐标也可以。

注意这里实现了时间每秒刷新一次。

------------------------------------------------------\*/

void displayCurrentTime(void)

{

int currenttime = 1;//设定现在时间,初始化。

int pasttime = 0;//参考时间

time\_t timep;

struct tm\* p;

char timeString[20]; // 获取当前时间的秒数

time(&timep); // 转换为协调世界时的时间结构体

p = localtime(&timep);

currenttime = p->tm\_sec;//赋值

getshijian1(&pasttime);//实现一段时间内（1s）pasttime=currenttime.

if (pasttime != currenttime)//利用这个实现每过一秒中就刷新一个，具体自己理解）

{

clearTimeArea();// 清除时间显示区域

pasttime = currenttime;

setshijian1(pasttime);

}

// 在窗口中显示转换后的时间信息

setcolor(LIGHTGREEN);//outtextxy的颜色

settextstyle(1, HORIZ\_DIR, 1);//HOR...表示横向显示，第二个1表是字体大小。1为小，0默认。2，3..大

outtextxy(10, 10, "Local Time");

settextstyle(1, HORIZ\_DIR, 1);

sprintf(timeString, "%d-%02d-%02d- %02d:%02d:%02d",//图形化下的数据显示函数。sprintf将这些储存到timestring字符串中

p->tm\_year + 1900, p->tm\_mon + 1, p->tm\_mday,

p->tm\_hour, p->tm\_min, p->tm\_sec);

outtextxy(10, 30, timeString);//显示字符串

}

/\*-----------------------------------------------------

用一个实心矩形（注意改背景色），遮挡。

并不断刷新，实现部分cleardevice()的效果。

------------------------------------------------------\*/

void clearTimeArea(void) {

setcolor(WHITE);

setfillstyle(SOLID\_FILL, WHITE);

bar(10, 30, 300, 55); // 清除时间显示区域

}

**7.showtime.h**

#ifndef \_SHOWTIME\_H\_

#define \_SHOWTIME\_H\_

void displayCurrentTime(void);//showtime

void clearTimeArea(void);

#endif

**8.move.c**

#include "common.h"

/\*--------------

解决了鼠标的闪动问题，

解决了鼠标被 图像遮挡的问题

--------------\*/

//适用于 int类型函数。可参考 welcome.h中的使用

int moveint(int (\*hanshu)(int\*), int\* a)

{

int b;

if (\*a == 1)

{

clrmous(MouseX, MouseY);

}

b= hanshu(a);//顺序有严格要求，不能颠倒，否则出问题

if (\*a != -10)//注意只能if，不能while,而且，这里的\*a!=-10是为了消除warning

{

newmouse(&MouseX, &MouseY, &press);

}

return b;

}

int moveint1(int (\*hanshu)(int\*, unsigned char(\*)[49], unsigned char(\*)[49], unsigned char(\*)[49]), int\* a, unsigned char b[384][49], unsigned char c[192][49], unsigned char d[192][49])

{

int bbb;

if (\*a == 1)

{

clrmous(MouseX, MouseY);

}

bbb = hanshu(a,b,c,d);

if (\*a!=-10)

{

newmouse(&MouseX, &MouseY, &press);

}

return bbb;

}

void showbiankuang(int x1, int y1, int x2, int y2)//用于显使得鼠标在 矩形中显示边框

{

if (mouse\_press( x1, y1, x2, y2)==2|| mouse\_press(x1, y1, x2, y2) == 1)

{

setcolor(BLUE);

rectangle(x1 + 1, y1 + 1, x2 - 1, y2 - 1);

}

else

{

setcolor(WHITE);

rectangle(x1 + 1, y1 + 1, x2 - 1, y2 - 1);

}

}

**9.move.h**

#ifndef \_MOVE\_H\_

#define \_MOVE\_H\_

//界面跳转中转函数

int moveint(int (\*hanshu)(int\*), int\* a);

int moveint1(int (\*hanshu)(int\*, unsigned char(\*)[49], unsigned char(\*)[49], unsigned char(\*)[49]), int\* a, unsigned char b[384][49], unsigned char c[192][49], unsigned char d[192][49]);

void showbiankuang(int x1, int y1, int x2, int y2);//用于显使得鼠标在 矩形中显示边框

#endif

**10.mainpage.c**

#include "common.h"

/\*-----------------------------------------------------

主界面的编写，实现不同功能界面的跳转

可以一览本程序的功能。

------------------------------------------------------\*/

int hello(int\* a2)

{

//绘图,只画一次,绘制主界面

if (\*a2 == 1)

{

int i = -1;

clrmous(MouseX, MouseY);

cleardevice();

setbkcolor(WHITE);//background color

setcolor(RED);

setfillstyle(1,RED );

rectangle(600, 0, 640, 40);

line(600, 0, 640, 40);

line(640, 0, 600, 40);//右上角画 退出

setcolor(GREEN);

puthz(80, 60, "无人机喷洒农药系统", 32, 50, GREEN);

line(80, 100, 520, 100);//画标题

rectangle(220, 120, 420, 200);

rectangle(220, 240, 420, 320);

rectangle(220, 360, 420, 440);

puthz(230, 140, "进入平坝田", 32, 38, GREEN);

puthz(230, 260, "进入小农田", 32, 38, GREEN);

puthz(230, 380, "系统说明", 32, 50, GREEN);

//画三个框框，代表三种功能。

\*a2 = 0;

setshijian1(i);//初始化 time.dat文件

}

//else---------------------------------------------------

displayCurrentTime();//左上角显示时间

showbiankuang(220,120,420,200);//用于显使得鼠标在 矩形中显示边框

showbiankuang(220, 240, 420, 320);

showbiankuang(220, 360, 420, 440);

showbiankuang( 600 ,0 ,640 ,40 );

//-------------------------------------------------------

//界面的跳转

if( mouse\_press(600,0,640,40) == 1)

{

\*a2 = 1;//使得函数参数重置，为了能够实现返回界面

return -1;//关闭BC。

}

else if (mouse\_press(220,120,420,200) == 1)

{

\*a2 = 1;

return 2;//进入大农田

}

else if (mouse\_press(220, 240, 420, 320)== 1)

{

\*a2 = 1;

return 3;//小农田

}

else if (mouse\_press(220, 360, 420, 440) == 1)

{

\*a2=1;

return 4 ;//系统说明

}

return 1;

}

**11.mainpage.h**

#ifndef \_MAINPAGE\_H\_

#define \_MAINPAGE\_H\_

int hello(int\* a2);//主界面的编写。

#endif

**12.mainfunc.c**

#include "common.h"

/\*===========================================

实现了分工，其中准备喷洒会进入另外一个新界面。

绘制了农田和植物的基本样子。

=============================================\*/

int mainfunc(int\* a3, unsigned char diannian[384][49], unsigned char dianbai[192][49], unsigned char dianmian[192][49])

{

int i;

srand(time(NULL));

if (\*a3 == 1)

{

cleardevice();

clrmous(MouseX, MouseY);

setbkcolor(WHITE);//background color

drawBackground();

farmfield();//绘制农田

\*a3 = 0;

}

if (\*a3 ==- 1)//实现界面跳转后，返回改界面的重新绘图

{

cleardevice();

clrmous(MouseX, MouseY);

setbkcolor(WHITE);//background color

drawBackground();

farmfield();

for (i = 0; i < 192; i++)//小麦区域的发病率

{

if (nian.bfzuobiao[i][0] != 0 && nian.bfzuobiao[i][1] != 0)

{

nianchong(nian.bfzuobiao[i][0], nian.bfzuobiao[i][1],diannian[i]);

}

}

for (i=192; i < 384; i++)//玉米区域的发病率

{

if (nian.bfzuobiao[i][0] != 0 && nian.bfzuobiao[i][1] != 0)

{

nianchong(nian.bfzuobiao[i][0], nian.bfzuobiao[i][1],diannian[i]);

}

}

for (i = 0; i< 192; i++)//玉米区域的发病率.

{

if (mian.bfzuobiao[i][0] != 0 && mian.bfzuobiao[i][1] != 0)

{

mianling(mian.bfzuobiao[i][0], mian.bfzuobiao[i][1],dianmian[i]);

}

}

for (i = 0;i < 192; i++)

{

if (bai.bfzuobiao[i][0] != 0 && bai.bfzuobiao[i][1] != 0)

{

baifen(bai.bfzuobiao[i][0], bai.bfzuobiao[i][1],dianbai[i]);

}

}

\*a3 = 2;

}

//====================ELSE

showbiankuang(600, 0, 640, 40);

showbiankuang(0, 0, 80, 40);

showbiankuang(100, 5, 100 + 140, 50);

//界面的跳转

if (mouse\_press(600, 0, 640, 40) == 1)

{

return -1;//关闭BC。

}

else if (mouse\_press(0, 0, 80, 40) == 1)

{

\*a3 = 1;

zero();//重置病虫坐标

delay(100);

return 1;//返回前一个界面

}

else if (mouse\_press(100, 5,240, 50) == 1)

{

\*a3 = 2;

clrmous(MouseX, MouseY);

grow();//无人机动画的实现

recover();//再次点击生长检测，部分区域恢复

//显示三种病

showbaifen(dianbai);

shownianchong(diannian);

showmianling(dianmian);

}

if (\*a3 == 2 || \*a3 == -1)//这里实现了必须先点击一次生长检测，才能点击结果查看和准备喷洒的功能

{

if (mouse\_press(100 + 140 + 30, 5, 100 + 140 + 30 + 140, 50) == 1)

{

\*a3 = -1;

delay(100);

return 6;//结果查看部分

}

else if (mouse\_press(100 + 140 + 140 + 30 + 30, 5, 100 + 140 + 140 + 30 + 30 + 140, 50) == 1)

{

\*a3 = -1;

delay(100);

return 7;//准备喷洒部分

}

showbiankuang(100 + 140 + 30, 5, 100 + 140 + 30 + 140, 50);

showbiankuang(100 + 140 + 140 + 30 + 30, 5, 100 + 140 + 140 + 30 + 30 + 140, 50);

}

return 2;

}

//绘制通用背景

void drawBackground(void)

{

setcolor(RED);//边界color

setfillstyle(1, GREEN);//意思：设置绿色的填充模式为1（完全填充）

rectangle(600, 0, 640, 40);

line(600, 0, 640, 40);

line(640, 0, 600, 40);//关闭按钮

rectangle(0, 0, 80, 40);

puthz(5, 5, "返回", 32, 40, GREEN);

setfillstyle(1, GREEN);

bar(100, 5, 100 + 140, 50);

bar(100 + 140 + 30, 5, 100 + 140 + 30 + 140, 50);

bar(100 + 140 + 140 + 30 + 30, 5, 100 + 140 + 140 + 30 + 30 + 140, 50);

puthz(100 + 10, 5 + 5, "生长检测", 32, 30, WHITE);

puthz(100 + 140 + 30 + 10, 5 + 5, "结果查看", 32, 30, WHITE);

puthz(100 + 140 + 140 + 30 + 30 + 10, 5 + 5, "准备喷洒", 32, 30, WHITE);

}

//绘制农田

void farmfield(void)

{

int x = 0, y = 0;

int j;

int jia1[] = { 25,75,25,75,25,75,25,75 };

setfillstyle(1, BROWN);

setcolor(BROWN);

bar(20, 60+5, 640-20, 480-15);//边框，即农田大小

for (x = 20; x < 620 - 25 + 1; x = x + 25)//小麦区域,

{

j = -1;

for (y = 65; y < 465; y = y + jia1[j])

{

j++;

fxiaomai(x, y);

}

}

for (y = 0; y <= 300; y = y + 100)//玉米的绘制

{

for (x = 20; x <= 620 - 25; x = x + 25)

{

fyumi(x, 65+50 + y);

fyumi(x, 65 +25+50 + y);

}

}

}

**13.mainfunc.h**

#ifndef \_MAINFUNC\_H\_

#define \_MAINFUNC\_H\_

int mainfunc(int\* a3, unsigned char diannian[384][49], unsigned char dianbai[192][49], unsigned char dianmian[192][49]);

void farmfield(void);

void drawBackground(void);

#endif

**14.elsetian.c**

#include "common.h"

// 画植物，均为左上角坐标。

void drawPlants(void)

{

// 绘制固定位置的植物

int i;

//玉米的坐标30\*35

int plantPositions1[][2] =

{

//梯田那部分。11个

{100,100},

{ 3 , 92 },

{ 73 , 89 },

{ 46 , 171 },

{ 138 , 213 },

{ 99 , 154 },

{ 120 , 151 },

{ 9 , 120 },

{ 65 , 140 },

{ 158 , 156 },

{ 89 , 199 },

//最右上角一块 5

{ 501 , 238 },

{ 598 , 265 },

{ 598 , 227 },

{ 574 , 144 },

{ 544 ,132 },

// 第三块 18

{ 350 , 95 },

{ 356 , 132 },

{ 366 , 168 },

{ 379 , 204 },

{ 392 , 243},

{ 469 , 100},

{ 470 , 139 },

{ 470 , 215 },

{ 470 , 175 },

{ 470 , 254 },

{ 473 , 294 },

{ 416 , 265 },

{ 447 , 291 },

{ 405 , 197 },

{ 416 , 231 },

{ 445 , 254 },

{ 438 , 220 },

{ 452 , 191 },

// 最上面区域 6个

{ 0,256 },

{ 31 , 283 },

{ 66 , 291 },

{ 149 , 281 },

{ 120 , 289 },

{ 94 , 293 },

{ 105 , 330 },

{ 108 , 367 },

{ 118 , 437 },

{ 111 , 402 },

{ 55 , 326 },

{ 69 , 421 },

{ 67 , 358 },

{ 177 , 321 },

{ 181 , 349 },

{ 188, 414 },

{ 195 , 380 },

{ 207 , 329 },

{ 212 , 365 },

{ 212, 411 },

{ 232 , 330 },

{ 236 , 366 },

{ 236 , 443 },

{ 236 , 405 },

{ 608 , 380 },

{ 592 , 411 },

{ 611 , 442 },

{ 328 , 443 },

{ 371 , 406 },

{ 376 , 369 },

{ 379 , 332 },

{ 337 , 406 },

{ 337 , 369 },

{ 346 , 332 }

};

//棉花的20\*35

int plantPositions2[][2] =

{

//梯田那部分。12个

{ 31 , 89 },

{ 53 , 85 },

{ 123 , 111 },

{ 75 , 182 },

{ 115 , 193 },

{ 165 , 198 },

{ 147 , 136 },

{ 45 , 117 },

{ 28 , 145 },

{ 90 , 123 },

{ 141 , 176 },

{ 502 , 90 },

{ 502 , 305 },

{ 502 , 129 },

{ 502 , 157 },

{ 502 , 199 },

{ 502 , 278 },

{ 530 , 315 },

{ 605 , 305 },

{ 582 , 308 },

{ 556 , 311 },

{ 521 , 103 },

{ 585 , 123 },

{ 605 , 129 },

//中间 23

{ 283 , 87 },

{ 304 , 87 },

{ 284, 128 },

{ 271 , 153 },

{ 306 , 125 },

{ 316 , 156 },

{ 259 , 181 },

{ 249 , 218},

{ 326, 194 },

{ 283 , 181 },

{ 305 , 188 },

{ 279 , 222 },

{ 305 , 227 },

{ 335 , 228 },

{ 364 , 271 },

{ 340, 284 },

{ 317, 290 },

{ 296 , 293 },

{ 443 , 155 },

{ 416, 148 },

{ 427 , 177},

{ 320 , 252 },

{ 264 , 331 },

{ 265 , 369 },

{ 264 , 443 },

{ 264, 406 },

{ 286 , 331 },

{ 287 , 369 },

{ 286 , 406 },

{ 285 , 443 },

{ 314 , 332 },

{ 309 , 443 },

{ 310 , 405 },

{ 308 , 368 },

{ 583 , 365 },

{ 517 , 406 },

{ 493 , 439 },

{ 517 , 442 },

{ 574 , 440 },

{ 547 , 442 },

{ 538 , 403 },

{ 565 , 403 },

{ 550 , 366 },

{ 360 , 443 }

};

//小麦占地.20\*19 总共104个

int plantPositions3[][2] =

{

//最右上角一块

{ 607 , 166 },

{ 609 , 189 },

{ 604 , 210 },

{ 525 , 144 },

{ 532 , 167 },

{ 524 , 189 },

{ 544 , 189 },

{ 562 , 189 },

{ 586 , 183},

{ 556 , 170 },

{ 526 , 212 },

{ 548 , 212 },

{ 569 , 212 },

{ 584 , 204 },

{ 534 , 237 },

{ 559 , 236 },

{ 532 , 265 },

{ 527 , 290 },

{ 555 , 285 },

{ 579 , 283 },

{ 553 , 259 },

{ 576 , 262 },

{ 581 , 240 },

// 中间

{ 260 , 104 },

{ 257 , 128 },

{ 249 , 153 },

{ 242 , 178 },

{ 234 , 202 },

{ 227 , 226 },

{ 220 , 250 },

{ 213 , 274 },

{ 230 , 275 },

{ 230 , 299},

{ 327 , 104 },

{ 330 , 128 },

{ 336 , 151 },

{ 344 , 174 },

{ 351 , 198 },

{ 357 , 222 },

{ 364 , 247 },

//第三块 8

{ 385 , 106 },

{ 410 , 108 },

{ 434 , 108 },

{ 389 , 133 },

{ 415 , 134 },

{ 445 , 135 },

{ 393, 161 },

// 整个上区域 20

{231 , 107 },

{ 228 , 130 },

{ 222 , 152 },

{ 215 , 173 },

{ 207 , 197 },

{ 200 , 223 },

{ 190 , 249 },

{ 182 , 271 },

{ 194 , 288 },

{ 204 , 108 },

{ 176 , 108 },

{ 150 , 107 },

{ 169 , 132 },

{ 196 , 132 },

{ 188 , 154 },

{ 192 , 176 },

{ 33 , 261 },

{ 59 , 272 },

{ 83 , 275 },

{ 2 , 235 },

//所有下方区域 32

{ 131 , 330 },

{ 133 , 353 },

{ 137, 377 },

{ 142 , 401 },

{ 146 , 425 },

{ 146 , 450 },

{ 157 , 331 },

{ 159 , 353 },

{ 162 , 378 },

{ 167 , 402 },

{ 170 , 425},

{ 168 , 455 },

{ 10 , 342 },

{ 22 , 365 },

{ 29 , 392 },

{ 21 , 437 },

{ 28 , 415 },

{ 36 , 337 },

{ 47 , 361 },

{ 53 , 387 },

{ 54 , 411 },

{ 52 , 434 },

{ 465 , 378 },

{ 470 , 406 },

{ 469 , 432 },

{ 469 , 456 },

{ 443 , 391 },

{ 445 , 415 },

{ 443 , 444 },

{ 420 , 412},

{ 419 , 447 },

{ 393 , 455 },

{ 88 , 341 },

{90,384 },

{72 ,399 },

{93 ,413 },

{ 97,437 }

};

for (i = 0; i < sizeof(plantPositions1) / sizeof(plantPositions1[0]); i++)//画植物

{

yumi(plantPositions1[i][0], plantPositions1[i][1]);

}

for (i = 0; i < sizeof(plantPositions2) / sizeof(plantPositions2[0]); i++)//

{

mianhua(plantPositions2[i][0], plantPositions2[i][1]);

}

for (i = 0; i < sizeof(plantPositions3) / sizeof(plantPositions3[0]); i++)

{

xiaomaikuai(plantPositions3[i][0], plantPositions3[i][1]);

}

setxtzhiwu(plantPositions1, plantPositions2, plantPositions3);//记录植物的坐标

}

//小农田主界面

int elsetian(int\* a5)

{

srand(time(NULL));

if (\*a5 == 1)

{

cleardevice();

clrmous(MouseX, MouseY);

drawBackground();

//梯田

bmp\_convert("5.bmp", "5.dbm");

open\_display();

show\_dbm(0, 100, "5.dbm", 0);//208\*180

tian();//画农田。

bmp\_convert("stone.bmp", "stone.dbm");

open\_display();

drawPlants();//画植物

drawdxg(250, 240);

show\_dbm(194, 447, "stone.dbm", 0);//石头

drawScarecrow(570, 61);//稻草人1

drawScarecrow(500, 325);//稻草人2

\*a5 = 0;

}

//=============

//恢复农田

if (\*a5 == -1)

{

cleardevice();

clrmous(MouseX, MouseY);

drawBackground();

bmp\_convert("5.bmp", "5.dbm");

open\_display();

show\_dbm(0, 100, "5.dbm", 0);//208\*180

tian();//画农田。

bmp\_convert("stone.bmp", "stone.dbm");

open\_display();

drawPlants();//画植物

drawdxg(250, 240);

show\_dbm(194, 447, "stone.dbm", 0);//石头

drawScarecrow(570, 61);//稻草人1

drawScarecrow(500, 325);//稻草人2

binghai();

\*a5 = 2;

}

showbiankuang(600, 0, 640, 40);

showbiankuang(0, 0, 80, 40);

showbiankuang(100, 5, 100 + 140, 50);

//界面的跳转

if (mouse\_press(600, 0, 640, 40) == 1)

{

return -1;//关闭BC。

}

else if (mouse\_press(0, 0, 80, 40) == 1)

{

\*a5 = 1;

return 1;//返回前一个界面

}

else if (mouse\_press(100, 5, 240, 50) == 1)

{

delay(100);

clrmous(MouseX, MouseY);

grow();//无人机动画的实现

if (\*a5 == 2 || \*a5 == -1)

{

delay(200);

cleardevice();

clrmous(MouseX, MouseY);

drawBackground();

bmp\_convert("5.bmp", "5.dbm");

open\_display();

show\_dbm(0, 100, "5.dbm", 0);//208\*180

tian();//画农田。

bmp\_convert("stone.bmp", "stone.dbm");

open\_display();

drawPlants();//画植物

drawdxg(250, 240);

show\_dbm(194, 447, "stone.dbm", 0);//石头

drawScarecrow(570, 61);//稻草人1

drawScarecrow(500, 325);//稻草人2

}

showxtbaifen();

showxtnianchong();

showxtymmian();

showxtymnian();

showxtmhxiu();

showxtmhmian();

\*a5 = 2;

}

if (\*a5 == 2 || \*a5 == -1)

{

showbiankuang(100 + 140 + 30, 5, 100 + 140 + 30 + 140, 50);

showbiankuang(100 + 140 + 140 + 30 + 30, 5, 100 + 140 + 140 + 30 + 30 + 140, 50);

if (mouse\_press(100 + 140 + 30, 5, 100 + 140 + 30 + 140, 50) == 1)

{

\*a5 = -1;

delay(100);

return 5;//结果查看部分

}

else if (mouse\_press(100 + 140 + 140 + 30 + 30, 5, 100 + 140 + 140 + 30 + 30 + 140, 50) == 1)

{

\*a5 = -1;

delay(100);

return 7;//准备喷洒部分

}

}

return 3;

}

// 画稻草人

void drawScarecrow(int x, int y)

{

setcolor(BROWN);

setfillstyle(SOLID\_FILL, YELLOW);

fillellipse(x, y, 10, 10);

setfillstyle(SOLID\_FILL, DARKGRAY);

fillellipse(x, y - 5, 15, 5);

setfillstyle(SOLID\_FILL, BLACK);

fillellipse(x - 3, y + 2, 2, 2);

fillellipse(x + 3, y + 2, 2, 2);

setfillstyle(SOLID\_FILL, YELLOW);

bar(x - 5, y + 10, x + 5, y + 60);

bar(x - 20, y + 10, x - 5, y + 15);

bar(x + 5, y + 10, x + 20, y + 15);

bar(x - 2, y + 60, x + 2, y + 80);

}

void drawdxg(int x, int y)//电线杆

{

setcolor(LIGHTGRAY);

setfillstyle(SOLID\_FILL, LIGHTGRAY);

bar(14 + x, y, 18 + x, 83 + y);

line(x + 0, y +26 , x +25 , y +18 );

line(x +25 , y +18 , x +25 , y +21 );

line(x + 25, y + 21, x +0 , y +29 );

line(x + 0, y +29 , x +0 , y +26 );

floodfill(x+8, y+25, LIGHTGRAY);

floodfill(x+21,y+ 20, LIGHTGRAY);

line(x + 3, y +6 , x + 27, y + 13);

line(x + 27, y +13 , x +27 , y +17 );

line(x + 27, y +17 , x +1 , y +9 );

line(x + 1, y +9 , x + 2, y + 6);

line(x + 2, y +6 , x + 3, y + 6);

floodfill(x + 10, y + 9, LIGHTGRAY);

floodfill(x + 20, y + 13, LIGHTGRAY);

}

// 画不规则形状的农田

void tian(void)

{

int points1[] = { 150,480,130,335, 285,480 - 130,405,335,400,430,500,370,640,405,640,480 };//最下面中间的

int points2[] = { 400,280,500,360, 640,340,640,116,200,116,200,280 };//上面的

setcolor(BROWN);

setfillstyle(1, BROWN);

arc(-62, 420, 0, 360, 90);

arc(70, 560, 0, 360, 105);

line(140, 480, 125, 335);

line(125, 335, 0, 352);

floodfill(100, 420, BROWN);//左下角的

fillpoly(8, points1);//多边形绘制

fillpoly(6, points2);//多边形绘制

line(0, 280, 210, 280);

ellipse(100, 280, 180, 360, 100, 50);//椭圆

floodfill(100, 290, BROWN);

line(0 + 200, 280, 210 + 200 - 10, 280);

ellipse(100 + 200, 280, 180, 360, 100, 50);//椭圆

floodfill(100 + 200, 290, BROWN);

line(150, 280 + 20 + 10, 280, 300 + 10);

floodfill(200, 290, BROWN);

line(400, 280, 500, 360);

line(500, 360, 640, 340);

line(640, 340, 640, 116);

line(110, 116, 640, 116);

line(200, 116, 200, 280);

line(200, 280, 400, 280);

line(400, 280, 500, 360);

line(500, 360, 640, 340);

line(640, 340, 640, 116);

line(110, 116, 640, 116);

line(200, 116, 200, 280);

line(200, 280, 400, 280);

setfillstyle(1, BLUE);

floodfill(630, 350, BROWN);

floodfill(10, 400, BROWN);

floodfill(100, 470, BROWN);

setcolor(WHITE);

line(208, 280, 258, 116);

line(400, 280 + 1, 350, 116);

line(500, 360 - 1, 500, 116);

setfillstyle(1, GREEN);

setcolor(GREEN);

line(0, 98, 640, 98);//分割线

line(630, 116, 640, 116);

line(640, 116, 640, 340);

line(640, 340, 630, 342);

line(630, 342, 630, 116);

floodfill(632, 120, GREEN);

}

void binghai(void)

{

int row;

FILE\* filePointer1;

FILE\* filePointer2;

FILE\* filePointer3;

FILE\* filePointer4;

FILE\* filePointer5;

FILE\* filePointer6;

int a22[68 \* 35][2] = { 0 };

int a11[103 \* 31][2] = { 0 };

filePointer1 = fopen("data//xmbfdian.dat", "r");

for (row = 0; row < 103 \* 31; row++)

{

fscanf(filePointer1, "%d %d", &a11[row][0], &a11[row][1]);

}

fclose(filePointer1);

for (row = 0; row < 103 \* 31; row++)//

{

setcolor(WHITE);

putpixel(a11[row][0], a11[row][1], WHITE);

}

for (row = 0; row < 103 \* 31; row++)

{

a11[row][0] = 0;

a11[row][1] = 0;

}

filePointer2 = fopen("data//xmncdian.dat", "r");

for (row = 0; row < 103 \* 31; row++)

{

fscanf(filePointer2, "%d %d", &a11[row][0], &a11[row][1]);

}

for (row = 0; row < 103 \* 31; row++)

{

setcolor(YELLOW);

putpixel(a11[row][0], a11[row][1], YELLOW);

}

fclose(filePointer2);

filePointer3 = fopen("data//ymncdian.dat", "r");

for (row = 0; row < 68 \* 35; row++)

{

fscanf(filePointer3, "%d %d", &a22[row][0], &a22[row][1]);

}

for (row = 0; row < 68 \* 35; row++)

{

setcolor(YELLOW);

putpixel(a22[row][0], a22[row][1], YELLOW);

}

fclose(filePointer3);

for (row = 0; row < 68 \* 35; row++)

{

a22[row][0] = 0;

a22[row][1] = 0;

}

filePointer4 = fopen("data//ymmldian.dat", "r");

for (row = 0; row < 68 \* 35; row++)

{

fscanf(filePointer4, "%d %d", &a22[row][0], &a22[row][1]);

}

for (row = 0; row < 68 \* 35; row++)

{

setcolor(BLUE);

putpixel(a22[row][0], a22[row][1], BLUE);

}

fclose(filePointer4);

for (row = 0; row < 68 \* 35; row++)

{

a22[row][0] = 0;

a22[row][1] = 0;

}

filePointer5 = fopen("data//mhmldian.dat", "r");

for (row = 0; row < 68 \* 35; row++)

{

fscanf(filePointer5, "%d %d", &a22[row][0], &a22[row][1]);

}

for (row = 0; row < 68 \* 35; row++)

{

setcolor(BLUE);

putpixel(a22[row][0], a22[row][1], BLUE);

}

fclose(filePointer5);

for (row = 0; row < 68 \* 35; row++)

{

a22[row][0] = 0;

a22[row][1] = 0;

}

filePointer6 = fopen("data//mhxbdian.dat", "r");

for (row = 0; row < 68 \* 35; row++)

{

fscanf(filePointer6, "%d %d", &a22[row][0], &a22[row][1]);

}

for (row = 0; row < 68 \* 35; row++)

{

setcolor(LIGHTRED);

putpixel(a22[row][0], a22[row][1], LIGHTRED);

}

fclose(filePointer6);

}

**15.elsetian.h**

#ifndef \_ELSETIAN\_H\_

#define \_ELSETIAN\_H\_

int elsetian(int\* a5);

void drawPlants(void);

void tian(void);

void drawScarecrow(int x, int y);

void drawdxg(int x, int y);

void binghai(void);

#endif

**16.shuju.c**

#include "shuju.h"

#include "common.h"

//==============================================================

//控制时间间隔。原理请配合showtime.c使用

void setshijian1(int cankaotime)

{

FILE\* filePointer1;

filePointer1 = fopen("data//time.dat", "w");

if (filePointer1 == NULL) {

printf("Unable to open the file.\n");

getch();

exit(1);

}

fprintf(filePointer1, "%d\n", cankaotime);//设置使得time.dat 储存cankaotime的数字

fclose(filePointer1);

}

void getshijian1(int \*cankaotime)

{

FILE\* filePointer1;

filePointer1 = fopen("data//time.dat", "r");

if (filePointer1 == NULL) {

printf("Unable to open the file.\n");

getch();

exit(1);

}

fscanf(filePointer1, "%d", cankaotime);//设置使得cankaotime获得time.dat 储存的数字，注意这里用了指针。。

fclose(filePointer1); //fscanf用&,fprintf不用&

}

unsigned char dianmian\_bitmap[192][49];

unsigned char dianbai\_bitmap[192][49];

unsigned char diannian\_bitmap[384][49];

dbfxy bai = { 0 };//小麦地得白粉病 的坐标。 注意当x，y均为0时，表示该区域不得病。

dbfxy mian = { 0 };//得棉铃虫病的坐标。 注意当x，y均为0时，表示该区域不得病。

niang nian = { 0 };//黏虫的坐标，左上角坐标， 注意当x，y均为0时，表示该区域不得病。

int nongyao[6];//氟铃胺，甲胺磷，氯氰菊酯，甲维盐，菌酯, nongyao[6]用于判断大小农田 0:大 1：小

int rount[386][2];//储存需要喷洒农药的植物坐标

int amtjudge;//判断需要喷洒的植物的棵数

//=================================================================================

//小农田

//=============================================

//获得所有植物的坐标，把植物坐标传入到 plantPostions里面。

void getxtxiao(int plantPositions3[][2])

{

int row = 0;

FILE\* filePointer3;

filePointer3 = fopen("data//xtxiao.dat", "r"); // 以读取模式打开文件

for (row = 0; row < 104; row++)//画植物的数量为坐标数

{

fscanf(filePointer3, "%d %d", &plantPositions3[row][0], &plantPositions3[row][1]);

}

fclose(filePointer3);

}

void getxtyumi(int plantPositions3[][2])

{

int row = 0;

FILE\* filePointer3;

filePointer3 = fopen("data//xtyumi.dat", "r"); // 以读取模式打开文件

for (row = 0; row < 68; row++)//画植物的数量为坐标数

{

fscanf(filePointer3, "%d %d", &plantPositions3[row][0], &plantPositions3[row][1]);

}

fclose(filePointer3);

}

void getxtmianhua(int plantPositions3[][2])

{

int row = 0;

FILE\* filePointer3;

filePointer3 = fopen("data//xtmianhua.dat", "r"); // 以读取模式打开文件

for (row = 0; row < 68; row++)//画植物的数量为坐标数

{

fscanf(filePointer3, "%d %d", &plantPositions3[row][0], &plantPositions3[row][1]);

}

fclose(filePointer3);

}

//=================================================

//下面的三个文件对文件进行初始化定义。

void setxtzhiwu(int plantPositions1[][2], int plantPositions2[][2], int plantPositions3[][2])

{

int j;

FILE\* filePointer1; // 文件指针

FILE\* filePointer2;

FILE\* filePointer3;

filePointer1 = fopen("data//xtyumi.dat", "w"); // 以写入模式打开文件

filePointer2 = fopen("data//xtmianhua.dat", "w");

filePointer3 = fopen("data//xtxiao.dat", "w");

if (filePointer1 == NULL) {

printf("Unable to open the file.\n");

getch();

exit(1);

}

if (filePointer2 == NULL) {

printf("Unable to open the file.\n");

getch();

exit(1);

}

if (filePointer3 == NULL) {

printf("Unable to open the file.\n");

getch();

exit(1);

}

for (j = 0; j < 68; j++)

{

fprintf(filePointer1, "%d %d\n", plantPositions1[j][0], plantPositions1[j][1]);

}

for (j = 0; j < 68; j++)

{

fprintf(filePointer2, "%d %d\n", plantPositions2[j][0], plantPositions2[j][1]);

}

for (j = 0; j < 104; j++)

{

fprintf(filePointer3, "%d %d\n", plantPositions3[j][0], plantPositions3[j][1]);

}

fclose(filePointer1);

fclose(filePointer2);

fclose(filePointer3);

}

//下面是大农田里用到的

void zero(void)//重置储存。。

{

int i, j;

for (i = 0, j = 192; i < 192, j < 384; i++, j++)

{

bai.bfzuobiao[i][0] = 0;

bai.bfzuobiao[i][1] = 0;

mian.bfzuobiao[i][0] = 0;

mian.bfzuobiao[i][0] = 0;

nian.bfzuobiao[i][0] = 0;

nian.bfzuobiao[i][0] = 0;

nian.bfzuobiao[j][0] = 0;

nian.bfzuobiao[j][0] = 0;

}

}

// 设置位图中的某个像素值

void setshu(unsigned char bitmap[49], int x, int value)

{

if (value) {

bitmap[x / 8] |= (1 << (x % 8)); // 设置为 1

}

else {

bitmap[x / 8] &= ~(1 << (x % 8)); // 设置为 0

}

}

// 获取位图中的某个像素值

int getshu(unsigned char bitmap[49], int x)

{

return (bitmap[x / 8] >> (x % 8)) & 1;

}

**17.shuju.h**

#ifndef \_SHUJU\_H

#define \_SHUJU\_H

/\*--------------

作为数据库，实现一次定义

--------------\*/

typedef struct

{

int bfzuobiao[384][2];//

} niang; //储存黏虫的坐标信息；

typedef struct

{

int bfzuobiao[192][2];//

} dbfxy;//储存小麦的白粉病信息以及玉米的棉铃虫信息

extern dbfxy bai;

extern dbfxy mian;

extern niang nian;

extern unsigned char dianmian\_bitmap[192][49];//位图，储存颜色点的信息

extern unsigned char dianbai\_bitmap[192][49];

extern unsigned char diannian\_bitmap[384][49];

//初始化位图和获取位图的0，1.

void setshu(unsigned char bitmap[49], int x, int value);

int getshu(unsigned char bitmap[49], int x);

extern int nongyao[6];//氟铃胺，甲胺磷，氯氰菊酯，甲维盐，菌酯, nongyao[5]用于判断大小农田 0:大 1：小

extern int rount[386][2];//储存需要喷洒农药的植物坐标

extern int amtjudge;//判断需要喷洒的植物的棵数

//初始化小农田里的各种植物坐标。以及从文件中获取各个植物的坐标到数组中

void setxtzhiwu(int plantPositions1[][2], int plantPositions2[][2], int plantPositions3[][2]);

void getxtmianhua(int plantPositions3[][2]);

void getxtyumi(int plantPositions3[][2]);

void getxtxiao(int plantPositions3[][2]);

//=====================

//time.dat

//function:控制时间间隔，每隔一秒刷新

void setshijian1(int cankaotime);

void getshijian1(int\* cankaotime);

//=====================

void zero(void);//重置大农田的患病植物坐标储存

#endif

**18.abdrone.c**

#include "common.h"

/\*===========================================

无人机所有函数，包括运动，画图

=============================================\*/

//俯视图 的无人机

//用于大农田

void drfdrone(int x, int y)//xy是左上角坐标

{

setcolor(DARKGRAY);

setfillstyle(1, DARKGRAY); //circle(x + 12, y + 12, 4);

fillellipse(x + 12, y + 12, 4, 4);

rectangle(x + 8, y + 9, x + 15, y + 15);// floodfill(x + 12, y + 12, DARKGRAY);

line(x + 5, y + 4, x + 9, y + 8);

line(x + 3, y + 4, x + 9, y + 9);

line(x + 3, y + 5, x + 8, y + 10);

line(x + 9, y + 1, x + 1, y + 9);

line(x + 8, y + 1, x + 0, y + 9);

line(x + 7, y + 1, x + 0, y + 8);

line(x + 15, y + 8, x + 19, y + 4);

line(x + 15, y + 9, x + 20, y + 4);

line(x + 15, y + 10, x + 20, y + 5);

line(x + 15, y + 0, x + 24, y + 9);

line(x + 16, y + 0, x + 24, y + 8);

line(x + 14, y + 1, x + 23, y + 9);

line(x + 16, y + 15, x + 21, y + 20);

line(x + 15, y + 15, x + 20, y + 20);

line(x + 14, y + 16, x + 20, y + 21);

line(x + 24, y + 15, x + 15, y + 24);

line(x + 24, y + 16, x + 16, y + 24);

line(x + 24, y + 14, x + 14, y + 24);

line(x + 9, y + 15, x + 3, y + 21);

line(x + 8, y + 15, x + 3, y + 20);

line(x + 9, y + 16, x + 4, y + 21);

line(x + 0, y + 15, x + 9, y + 24);

line(x + 0, y + 16, x + 8, y + 24);

line(x + 0, y + 14, x + 10, y + 24);

}

//==================================================

void move11(int\* x, int\* y, int\* buffer1)

{

for (; \*x < 620 - 25; \*x = \*x + 2)//运动到了最左端。。

{

getimage(\*x, \*y, \*x + 25, \*y + 25, buffer1);//保存image到buffer

drfdrone(\*x, \*y);

delay(8);

putimage(\*x, \*y, buffer1, COPY\_PUT);//恢复图像

}

}

void move12(int\* x, int\* y, int\* buffer1, int\* judge)

{

for (; \*y < \*judge + 25; (\*y)++)

{

getimage(\*x, \*y, \*x + 25, \*y + 25, buffer1);

drfdrone(\*x, \*y);

delay(10);

putimage(\*x, \*y, buffer1, COPY\_PUT);

}

\*judge = \*y;

}

void move13(int\* x, int\* y, int\* buffer1)

{

for (; \*x > 20; \*x = \*x - 2)

{

getimage(\*x, \*y, \*x + 25, \*y + 25, buffer1);

drfdrone(\*x, \*y);

delay(8);

putimage(\*x, \*y, buffer1, COPY\_PUT);

}

}

//=============================================

//无人机透视图，用于小农田

void drtdrone(int xx, int y)//xy是左上角坐标 75\*44

{

int x;

x = xx - 25;

setcolor(DARKGRAY);

setfillstyle(1, DARKGRAY);

bar(0 + x, 4 + y, 24 + x, 5 + y);

bar(49 + x, 4 + y, 73 + x, 5 + y);

bar(x + 4, y + 6, x + 5, y + 10);

bar(x + 6, y + 9, x + 20, y + 10);

bar(x + 19, y + 6, x + 20, y + 10);

bar(x + 53, y + 6, x + 54, y + 10);

bar(x + 54, y + 10, x + 69, y + 9);

bar(x + 68, y + 6, x + 69, y + 9);

bar(x + 7, y + 11, x + 9, y + 14);

bar(x + 16, y + 11, x + 18, y + 13);

bar(x + 8, y + 16, x + 22, y + 17);

bar(x + 55, y + 11, x + 57, y + 13);

bar(x + 64, y + 14, x + 66, y + 11);

bar(x + 51, y + 16, x + 65, y + 17);

bar(x + 24, y + 11, x + 49, y + 13);

bar(x + 21, y + 20, x + 51, y + 22);

bar(x + 23, y + 24, x + 50, y + 25);

line(x + 11, y + 18, x + 13, y + 18);

line(x + 60, y + 18, x + 62, y + 18);

line(x + 9, y + 15, x + 23, y + 15);

line(x + 50, y + 15, x + 64, y + 15);

line(x + 10, y + 14, x + 17, y + 14);

line(x + 10, y + 13, x + 15, y + 13);

line(x + 10, y + 12, x + 10, y + 14);

line(x + 15, y + 12, x + 15, y + 13);

line(x + 57, y + 14, x + 63, y + 14);

line(x + 58, y + 13, x + 63, y + 13);

line(x + 58, y + 12, x + 58, y + 13);

line(x + 63, y + 12, x + 63, y + 13);

line(x + 23, y + 12, x + 23, y + 14);

line(x + 50, y + 12, x + 50, y + 14);

line(x + 8, y + 2, x + 10, y + 0);

line(x + 8, y + 3, x + 11, y + 0);

line(x + 14, y + 0, x + 16, y + 2);

line(x + 13, y + 0, x + 16, y + 3);

line(x + 12, y + 0, x + 9, y + 3);

line(x + 13, y + 1, x + 15, y + 3);

line(x + 12, y + 1, x + 10, y + 3);

line(x + 12, y + 1, x + 14, y + 3);

line(x + 21, y + 18, x + 21, y + 19);

line(x + 20, y + 18, x + 20, y + 21);

line(x + 52, y + 18, x + 52, y + 21);

line(x + 53, y + 18, x + 53, y + 20);

line(x + 23, y + 23, x + 25, y + 23);

line(x + 48, y + 23, x + 50, y + 23);

line(x + 57, y + 2, x + 59, y + 0);

line(x + 57, y + 3, x + 60, y + 0);

line(x + 58, y + 3, x + 61, y + 0);

line(x + 62, y + 0, x + 59, y + 3);

line(x + 63, y + 0, x + 65, y + 2);

line(x + 63, y + 1, x + 65, y + 3);

line(x + 62, y + 1, x + 64, y + 3);

line(x + 62, y + 2, x + 63, y + 2);

line(x + 23, y + 26, x + 23, y + 32);

line(x + 24, y + 26, x + 24, y + 30);

line(x + 22, y + 32, x + 22, y + 42);

line(x + 21, y + 32, x + 21, y + 42);

line(x + 20, y + 37, x + 20, y + 41);

line(x + 21, y + 41, x + 25, y + 41);

line(x + 21, y + 42, x + 24, y + 42);

line(x + 49, y + 26, x + 53, y + 37);

line(x + 49, y + 27, x + 53, y + 38);

line(x + 51, y + 34, x + 51, y + 42);

line(x + 48, y + 41, x + 53, y + 41);

line(x + 49, y + 42, x + 52, y + 42);

line(x + 53, y + 37, x + 53, y + 41);

line(x + 51, y + 34, x + 51, y + 42);

line(x + 52, y + 42, x + 52, y + 34);

bar(x + 41, y + 28, x + 43, y + 36);

bar(x + 41, y + 26, x + 42, y + 36);

line(x + 38, y + 36, x + 43, y + 36);

line(x + 31, y + 26, x + 29, y + 30);

line(x + 32, y + 26, x + 29, y + 31);

line(x + 34, y + 36, x + 30, y + 36);

line(x + 30, y + 36, x + 30, y + 27);

line(x + 29, y + 30, x + 29, y + 35);

setcolor(LIGHTGRAY);

setfillstyle(1, LIGHTGRAY);

floodfill(x + 12, y + 3, DARKGRAY);

floodfill(x + 61, y + 3, DARKGRAY);

floodfill(x + 13, y + 12, DARKGRAY);

floodfill(x + 60, y + 11, DARKGRAY);

line(x +26 , y +23 , x +47 , y + 23);

bar(x +6 , y + 6, x +18 , y + 8);

bar(x +55 , y +6, x +67 , y +8 );

bar(x + 24, y +14 , x + 49, y +19 );

line(x +23 , y + 16, x +23 , y + 19);

line(x +50 , y + 16, x +50 , y + 19);

setcolor(DARKGRAY);

setfillstyle(1, DARKGRAY);

bar(x +26 , y + 17, x + 27, y +18 );

bar(x + 29, y + 17, x + 30, y + 18);

}

**19.abdrone.h**

#ifndef \_ABDRONE\_H\_

#define \_ABDRONE\_H\_

//======================================

//大农田的运动，无人机运动的路径

void move11(int\* x, int\* y, int\* buffer1);

void move12(int\* x, int\* y, int\* buffer1, int\* judge);

void move13(int\* x, int\* y, int\* buffer1);

//=================================

//画无人机的俯视图

void drfdrone(int x, int y);//xy是左上角坐标

void drtdrone(int xx, int y);//画无人机的透视图

#endif

**20.drzhiwu.c**

#include"common.h"

/\*========================================

functions: 画植物，包括小麦，玉米，棉花。

=========================================\*/

//=============

// 画棉花，透视图

//==============

void mianhua(int x, int y)//x,y为其的左上角坐标。20\*35

{

setcolor(GREEN);

setfillstyle(SOLID\_FILL, GREEN);

bar(x + 10, y + 15, x + 12, y + 35);

line(x + 7, y + 14, x + 10, y + 17);

line(x + 10, y + 17, x + 16, y + 14);

ellipse(x+18, y+17,10, 360, 2, 4);//画椭圆函数

ellipse(x + 5, y +16 , 20, 360, 3, 4);

floodfill(x + 5, y + 16, GREEN);

floodfill(x + 18, y + 17, GREEN);

line(x+10, y+15, x+10, y+9);

line(x+11, y+15, x+11, y+9);

line(x+10, y+9, x+3, y+7);

line(x+3, y+7, x+9, y+5);

line(x+9, y+5, x+6, y+1);

line(x+6, y+1, x+10, y+2);

line(x+10, y+2, x+12, y+5);

line(x+12, y+5, x+15, y+1);

line(x+15, y+1, x+16, y+5);

line(x+16, y+5, x+19, y+6);

line(x+19, y+6, x+19, y+9);

line(x+19, y+9, x+13, y+10);

line(x+13, y+10, x+13, y+13);

line(x+13, y+13, x+11, y+10);

floodfill(x + 9, y + 3, GREEN);

setcolor(WHITE);

line(x+10, y+9, 6+x, y+8);

line(x+11,y+ 9, 10+x, 4+y);

line(11+x, y+9, x+15, y+2);

line(x+12, y+8, x+14, y+7);

}

//=============

// 画玉米，透视图

//==============

void yumi(int x, int y) //xy为左上角35\*35

{

setcolor(GREEN);

setfillstyle(SOLID\_FILL, GREEN);

bar(x + 12, y + 3, x + 16, y + 35);

arc(x+14, y+4, 0, 180, 2);

floodfill(x + 13, y + 3, GREEN);

line(x + 12 , y +4 , x +5 , y +4 );

line(x +5 , y + 4, x +2 , y +10 );

line(x +2 , y +10 , x +6 , y +8 );

line(x + 6, y +8 , x +12 , y +7 );

line(x + 12, y +18 , x +9 , y + 17);

line(x + 9, y + 17, x +7 , y + 28);

line(x +7 , y +28 , x +12 , y + 22);

line(x + 16, y +24 , x + 21, y +20 );

line(x + 21, y + 20, x +27 , y +25 );

line(x + 27, y +25 , x +23 , y +17 );

line(x +23 , y +17 , x + 16, y + 20);

floodfill(x + 20, y + 19, GREEN);

floodfill(x + 10, y + 20, GREEN);

floodfill(x + 5, y + 7, GREEN);

line(x + 16, y +17 , x + 20, y +15 );

line(x +20 , y +15 , x +20 , y +4 );

line(x + 20, y +4 , x +16 , y +12 );

floodfill(x + 19, y + 19, GREEN);

line(x + 16, y + 13, x + 20, y + 6);

line(x +12 , y +10 , x + 7, y +8 );

line(x + 7, y + 8, x +12 , y + 12);

line(x + 7, y +8 , x +6 , y +10 );

line(x + 6, y + 10, x +12 , y + 17);

line(x + 6, y +10 , x +12 , y +12 );

setfillstyle(SOLID\_FILL, YELLOW);

floodfill(x + 10, y + 12, GREEN);

floodfill(x + 18, y + 12, GREEN);

}

//=======

//小麦块

//=======

void xiaomaikuai(int x, int y)

{

xiaomai(x, y);

xiaomai(x+5, y);

xiaomai(x-5, y);

}

//画小麦透视图,,,其大小为 20\*19.

void xiaomai(int x, int y)

{

x = x +10;//修正小麦坐标

//xy是最顶点

setcolor(GREEN);

putpixel(x, y, GREEN);

line(x, y, x, y + 19);

line(x, y + 6, x + 5, y + 6 - 5);

line(x, y + 6, x - 5, y + 6 - 5);

line(x, y + 12, x - 5, y + 6);

line(x, y + 12, x + 5, y + 6 );

}

//画小麦俯视图

void fxiaomai(int x, int y)//xy是最左边的点//25\*25

{

setcolor(GREEN);

line(x + 2, y + 10 + 6 + 4 + 3, x + 6, y + 2 + 6 + 4 + 3);

line(x + 2, y + 6 + 6 + 4 + 3, x + 10, y + 6 + 6 + 4 + 3);

line(x + 6, y + 12 + 6 + 4 + 3, x + 3, y + 3 + 6 + 4 + 3);

line(x + 10, y + 9 + 6 + 4 + 3, x + 7, y + 12 + 6 + 4 + 3);

line(x + 11, y + 10 + 6 + 4 + 3, x + 8, y + 7 + 6 + 4 + 3);

line(x + 13, y + 10 + 6 + 3, x + 6 + 12, y + 2 + 6 + 3 + 3);

line(x + 2 + 12, y + 6 + 6 + 3, x + 10 + 12, y + 6 + 6 + 3 + 3);

line(x + 6 + 12, y + 12 + 5 + 4, x + 3 + 12, y + 3 + 6 + 3 + 3);

line(x + 10 + 12, y + 9 + 6 + 3, x + 7 + 12, y + 12 + 4);

line(x + 11 + 12, y + 10 + 3 + 3, x + 8 + 12, y + 7 + 4 + 3);

line(x + 6 + 10, y + 12, x + 8 + 10, y + 12 + 2);

line(x + 6 + 10, y + 12, x + 6 - 2 + 10, y + 12 + 2);

line(x + 6 + 10, y + 12, x + 6 + 10, y + 12 - 4);

line(x + 6 + 10, y + 12, x + 6 - 2 + 10, y + 12 - 4);

line(x + 4, y + 2, x + 8, y + 10);

line(x + 8, y + 6, x + 8, y + 2);

line(x + 8, y + 10, x + 12, y + 2);

line(x + 4, y + 10, x + 8, y + 18);

line(x + 8, y + 14, x + 8, y + 10);

line(x + 8, y + 18, x + 12, y + 10);

line(x + 8, y + 2, x + 16, y + 10);

line(x + 16, y + 6, x + 16, y + 2);

line(x + 16, y + 10, x + 24, y + 2);

line(x + 8, y + 10, x + 16, y + 18);

line(x + 16, y + 14, x + 16, y + 10);

line(x + 16, y + 18, x + 24, y + 10);

}

//=============

// 画玉米，俯视图

//==============

void fyumi(int x, int y)

{

setcolor(GREEN);

circle(x + 12, y + 12, 2);

line(x + 12, y + 10, x + 20, y + 2);

line(x + 20, y + 2, x + 16, y + 9);

line(x + 16, y + 9, x + 13, y + 11);

line(x + 14, y + 9, x + 20, y + 13);

line(x + 20, y + 13, x + 13, y + 13);

putpixel(x + 13, y + 13, GREEN);

putpixel(x + 13, y + 14, GREEN);

putpixel(x + 14, y + 14, GREEN);

putpixel(x + 14, y + 15, GREEN);

putpixel(x + 15, y + 15, GREEN);

putpixel(x + 15, y + 16, GREEN);

putpixel(x + 14, y + 17, GREEN);

putpixel(x + 15, y + 17, GREEN);

putpixel(x + 16, y + 17, GREEN);

putpixel(x + 17, y + 17, GREEN);

putpixel(x + 18, y + 17, GREEN);

line(x + 18, y + 17, x + 20, y + 23);

line(x + 20, y + 23, x + 13, y + 20);

line(x + 13, y + 20, x + 12, y + 15);

line(x + 11, y + 11, x + 2, y + 10);

line(x + 2, y + 10, x + 2, y + 12);

line(x + 2, y + 12, x + 5, y + 13);

line(x + 5, y + 13, x + 11, y + 14);

line(x + 7, y + 14, x + 3, y + 21);

line(x + 3, y + 21, x + 12, y + 18);

line(x + 5, y + 10, x + 13, y + 7);

setfillstyle(SOLID\_FILL, GREEN);

floodfill(x + 12, y + 12, GREEN); // 遇到边界颜色停止

setfillstyle(SOLID\_FILL, YELLOW);

floodfill(x + 15, y + 18, GREEN);

}

**21.drzhiwu.h**

#ifndef \_DRZHIWU\_H\_

#define \_DRZHIWU\_H\_

//画植物

void xiaomai(int x, int y);

void fxiaomai(int x, int y);

void yumi(int x, int y);

void mianhua(int x, int y);

void fyumi(int x, int y);

void xiaomaikuai(int x, int y);

#endif

**22.growlook.c**

#include "common.h"

/\*===========================================

生长检测部分1。这里仅仅只是实现了动画效果。

如，无人机的动画，植物范围的变色，不同的颜色代表了

不同的病虫害。

具体数据要到结果查看部分。

=============================================\*/

void grow(void)//将两个农田的无人机运动合为一体化

{

if (getpixel(20, 65) == BROWN)//根据这个判断两个农田.这个是平地农田

{

dronemove1();

}

else// (getpixel(20,65) == WHITE) //这个是小农田

{

dronemove2();

}

}

//小农田无人机动画

void move21(int points[][2], int num\_points, int x, int y, int\* buffer)

{

int i;

int barrier[2][2] = { 0 };//障碍物

int barrier1[2][2] = { 0 };

int barrier2[2][2] = { 0 };

int xiao[103][2];

int yu[68][2];

int mian[68][2];

getxtxiao(xiao);

getxtyumi(yu);

getxtmianhua(mian);

barrier[0][0] = 249;

barrier[0][1] = 239;

barrier[1][0] = 278;

barrier[1][1] = 324;

barrier1[0][0] = 480;

barrier1[0][1] = 326 - 10;

barrier1[1][0] = 519;

barrier1[1][1] = 415 ;

barrier2[0][0] = 549;

barrier2[0][1] = 49;

barrier2[1][0] = 589;

barrier2[1][1] = 141;

srand(time(NULL)); // 初始化随机种子

for (i = 0; i < num\_points; i++)

{

int target\_x = points[i][0]; // 获取当前目标点的x坐标

int target\_y = points[i][1]; // 获取当前目标点的y坐标

int j;

int dx = 0;

int dy = 0;

int xxx = 2;

int yyy = 2;

int judge;

int xmax, ymax, xmin, ymin;

int randx[400] = { 0 };

int randy[400] = { 0 };

int randfuzhi;

int rand\_numbers[400] = { 0 }; // 数组用于存储生成的随机数

int canshu1;

int canshu2;

int judgee;

int movebar[2][2] = { 0 };

if ((target\_x - x) >= 0)//判断目标点

{

xxx = 0;

}

else if ((target\_x - x) < 0)

{

xxx = 1;

}

if ((target\_y - y) >= 0)

{

yyy = 0;

}

else if ((target\_y - y) < 0)

{

yyy = 1;

}

if (abs(target\_x - x) >= abs(target\_y - y))

{

for (randfuzhi = 0; randfuzhi < abs((target\_x - x) / 2); randfuzhi++)

{

randx[randfuzhi] = 2;//控制速度 ======================================

randy[randfuzhi] = 0;

}

for (randfuzhi = 0; randfuzhi < abs((target\_y - y) / 2); randfuzhi++)

{

while (1)

{

judgee = 1;

canshu1 = rand() % (abs(target\_y - y) / 2) + 1;

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

{

if (rand\_numbers[j] == canshu1)

{

judgee = 0;

}

}

if (judgee == 1)

{

break;

}

}

rand\_numbers[randfuzhi] = canshu1; // 将生成的随机数标记为已使用

randy[canshu1 - 1] = 2;

}

}

else if (abs(target\_x - x) < abs(target\_y - y))

{

for (randfuzhi = 0; randfuzhi < abs((target\_y - y) / 2); randfuzhi++)

{

randx[randfuzhi] = 0;//控制速度 ======================================

randy[randfuzhi] = 2;

}

for (randfuzhi = 0; randfuzhi < abs((target\_x - x) / 2); randfuzhi++)

{

while (1)

{

judgee = 1;

canshu2 = rand() % (abs(target\_x - x) / 2) + 1;

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

{

if (rand\_numbers[j] == canshu2)

{

judgee = 0;

}

}

if (judgee == 1)

{

break;

}

}

rand\_numbers[randfuzhi] = canshu2; // 将生成的随机数标记为已使用

randx[canshu2 - 1] = 2;

randx[randfuzhi] = 2;

}

}

while (x != target\_x || y != target\_y)

{

int jj = 0;

int aa, bb, cc, dd;

judge = rand() % 2;

movebar[0][0] = x - 25;

movebar[0][1] = y - 44;

movebar[1][0] = x + 50;

movebar[1][1] = y;

if (abs(target\_x - x) < 10 || abs(target\_y - y) < 10)

{

if (x != target\_x)//

{

if (judge == 0)

{

dx = 1;

dy = 0;

}

}

if (y != target\_y)//

{

if (judge == 1)

{

dx = 0;

dy = 1;

}

}

}

else

{

dx = randx[jj];

dy = randy[jj];

jj++;

}

if (x != target\_x && y == target\_y && abs(target\_x - x) < 10)

{

dx = 1;

dy = 0;

}

else if (x != target\_x && y == target\_y && abs(target\_x - x) >= 10)

{

dx = 2;

dy = 0;

}

if (y != target\_y && x == target\_x && abs(target\_y - y) < 10)

{

dx = 0;

dy = 1;

}

else if (y != target\_y && x == target\_x && abs(target\_y - y) >= 10)

{

dx = 0;

dy = 2;

}

aa = skipthing(barrier, movebar, &dx, &dy);

bb = skipthing(barrier1, movebar, &dx, &dy);

cc = skipthing(barrier2, movebar, &dx, &dy);

if (aa == 1 && bb == 1 && cc == 1)

{

if ((target\_x - x) >= 0)

{

xxx = 0;

}

else if ((target\_x - x) < 0)

{

xxx = 1;

}

if ((target\_y - y) >= 0)

{

yyy = 0;

}

else if ((target\_y - y) < 0)

{

yyy = 1;

}

}

xmax = (x + 50) > 639 ? 639 : (x + 50);

ymax = (y + 0) > 479 ? 479 : (y + 0);

xmin = (x - 25) < 0 ? 0 : (x - 25);

ymin = (y - 44) < 20 ? 0 : (y - 44);

showheight(x, y, xiao, yu, mian);

getimage(xmin, ymin, xmax, ymax, buffer);

drtdrone(x, y - 44);

//无人机速度控制

delay(15);//15

if (abs(x - target\_x) < 2 && abs(y - target\_y) < 2)

{

delay(50);//100

}

putimage(xmin, ymin, buffer, COPY\_PUT);

if (xxx == 0)

{

x += dx; // 更新x坐标

}

else if (xxx == 1)

{

x -= dx; // 更新x坐标

}

if (yyy == 0)

{

y += dy; // 更新y坐标

}

else if (yyy == 1)

{

y -= dy; // 更新y坐标

}

}

}

}

int skipthing(int barrier[][2], int movebar[][2], int\* dx, int\* dy)

{

//barrier 储存的是障碍物坐标，movebar 是无人机坐标。dx,dy是运动的 值。targetxy是终点坐标。

if (barrier[0][0] - 5 <= movebar[1][0] &&

barrier[1][0] - 5 >= movebar[0][0] &&

barrier[0][1] + 5 <= movebar[1][1] &&

barrier[1][1] + 5 >= movebar[0][1]

)//障碍物与无人机 相交：：：判断两矩形是否相交。，这里扩充了障碍物，边界+10

{

if (abs(movebar[0][1] - barrier[1][1]) < 4)//无人机 上边与 障碍物下边 相交

{//无人机从 上方碰见障碍物

\*dy = 0;// 1;

\*dx = 1;

}

else if (abs(movebar[1][1] - barrier[0][1]) < 4)//无人机 下边与 障碍物上边 相交

{//无人机从 下方碰见障碍物

\*dy = 0;// 1;

\*dx = 1;

}

else if (abs(movebar[0][0] - barrier[1][0]) < 4)//无人机 左边与 障碍物右边 相交

{//无人机从 右方碰见障碍物

\*dx = 0;// 1;

\*dy = -1;

}

else if (abs(movebar[1][0] - barrier[0][0]) < 4)//无人机 右边与 障碍物左边 相交

{//无人机从 左方碰见障碍物

\*dx = 0; // -1;

\*dy = 1;

}

return 0;

}

return 1;

}void dronemove2(void)

{

void\* buffer1 = malloc(imagesize(0, 0, 75, 44));//定义缓存区

int x1, y1;

int a[239][2];

int i, j, k;

int temp[2];

int quyu[12] = { 77,110,137,157,185,219,265,316,355,384,417,480 };//将农田分成11区域，有12个分界线无人机扫描11次

int count[11] = { 0 };//计算每个区域有多少植物

int ii[11] = { 0 };

int barrier[2][2] = { 0 };

int barrier1[2][2] = { 0 };

int barrier2[2][2] = { 0 };

getxtxiao(a);

getxtyumi(&a[103]);

getxtmianhua(&a[171]);

barrier[0][0] = 249;

barrier[0][1] = 239;

barrier[1][0] = 278;

barrier[1][1] = 324;

barrier1[0][0] = 480;

barrier1[0][1] = 326 - 10;

barrier1[1][0] = 519;

barrier1[1][1] = 415 - 10;

barrier2[0][0] = 549;

barrier2[0][1] = 49;

barrier2[1][0] = 589;

barrier2[1][1] = 141;

paipai(a, barrier2);

paipai(a, barrier1);

paipai(a, barrier);

for (i = 0; i < 238; i++)

{

for (j = i + 1; j < 239; j++)

{

if (a[i][1] > a[j][1])

{

temp[0] = a[i][0];

temp[1] = a[i][1];

a[i][0] = a[j][0];

a[i][1] = a[j][1];

a[j][0] = temp[0];

a[j][1] = temp[1];

}

}

}

for (i = 0; i < 239; i++)

{

for (k = 0; k < 11; k++)

{

if (a[i][1] >= quyu[k] && a[i][1] < quyu[k + 1])//判断植物在第几区域内部

{

count[k]++;

}

}

}

for (i = 1; i < 11; i++)

{

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

{

ii[i] += count[j];

}

}

ii[0] = 0;

i = 0;

for (k = 0; k < 11; k++)

{

if ((k % 2) == 0)//向右移动，x小的 在前

{

for (i = ii[k]; i < ii[k] + count[k] - 1; i++)

{

for (j = i + 1; j < ii[k] + count[k]; j++)

{

if (a[i][0] > a[j][0])

{

temp[0] = a[i][0];

temp[1] = a[i][1];

a[i][0] = a[j][0];

a[i][1] = a[j][1];

a[j][0] = temp[0];

a[j][1] = temp[1];

}

}

}

}

else

{

for (i = ii[k]; i < ii[k] + count[k] - 1; i++)

{

for (j = i + 1; j < ii[k] + count[k]; j++)

{

if (a[i][0] < a[j][0])

{

temp[0] = a[i][0];

temp[1] = a[i][1];

a[i][0] = a[j][0];

a[i][1] = a[j][1];

a[j][0] = temp[0];

a[j][1] = temp[1];

}

}

}

}

}

x1 = 0;//初始化无人机坐标

y1 = 60;

move21(a, 239, x1, y1, buffer1);

free(buffer1);

setcolor(WHITE);

setfillstyle(SOLID\_FILL, WHITE);

bar(100, 56, 250, 88); // 清除时间显示区域

delay(100);

}

void paipai(int a[][2], int barrier2[][2])//排序

{

int i;

for (i = 0; i < 238; i++)

{

if (a[i][0] >= barrier2[0][0] - 52 && a[i][0] <= barrier2[1][0] + 26 && a[i][1] >= barrier2[0][1] && a[i][1] <= barrier2[1][1] + 46)

{

if (a[i][0] <= ((barrier2[1][0] + barrier2[0][1]) / 2))

{

a[i][0] = barrier2[1][0] - 53;

a[i][1] = barrier2[0][1];

}

else

{

a[i][0] = barrier2[1][0] + 27;

a[i][1] = barrier2[1][1];

}

}

}

}

//大农田的无人机运动

void dronemove1(void)

{

//-========================================================

void\* buffer1 = malloc(imagesize(0, 0, 25, 25));

int\* x1 = malloc(sizeof(int));

int\* y1 = malloc(sizeof(int));

int\* judge = malloc(sizeof(int));

int i;

\*x1 = 0;

\*judge = 65;

\*y1 = 65;

//====================================

//整个运动的实现

for (i = 0; i < 7; i++)

{

move11(x1, y1, buffer1);

move12(x1, y1, buffer1, judge);

move13(x1, y1, buffer1);

move12(x1, y1, buffer1, judge);

//这三个运动为一个循环。。。

}

move11(x1, y1, buffer1);

move12(x1, y1, buffer1, judge);

for (; \*x1 > 0; \*x1 = \*x1 - 2)//运动到了最左端。。

{

getimage(\*x1, \*y1, \*x1 + 25, \*y1 + 25, buffer1);//保存image到buffer

drfdrone(\*x1, \*y1);

delay(8);

putimage(\*x1, \*y1, buffer1, COPY\_PUT);//恢复图像

}

for (; \*y1 > 65; \*y1 = \*y1 - 1)//运动到了最左端。。

{

getimage(\*x1, \*y1, \*x1 + 25, \*y1 + 25, buffer1);//保存image到buffer

drfdrone(\*x1, \*y1);

delay(5);

putimage(\*x1, \*y1, buffer1, COPY\_PUT);//恢复图像

}

free(buffer1);

free(x1);

free(y1);

free(judge);

delay(100);

}

void showheight(int x, int y, int xiao[][2], int yu[][2], int mian[][2])//显示无人机高度

{

char string[20];

int high = 0;

int i;

int judge = 0;

int randshu;

if (x > 1 && x < 203 && y < 278 && y>76)//梯田区域

{

for (i = 0; i < 103; i++)

{

if (abs(x - xiao[i][0]) < 4 && abs(y - xiao[i][1]) < 4)

{

high = rand() % 30 + 150 + 300;

judge = 1;

}

}

for (i = 0; i < 68; i++)

{

if (abs(x - yu[i][0]) < 4 && abs(y - yu[i][1]) < 4)

{

high = rand() % 50 + 330 + 300;

judge = 1;

}

if (abs(x - mian[i][0]) < 4 && abs(y - mian[i][1]) < 4)

{

high = rand() % 50 + 210 + 300;

judge = 1;

}

}

}

else

{

for (i = 0; i < 103; i++)

{

if (abs(x - xiao[i][0]) < 4 && abs(y - xiao[i][1]) < 4)

{

high = rand() % 30 + 150;

judge = 1;

}

}

for (i = 0; i < 68; i++)

{

if (abs(x - yu[i][0]) < 4 && abs(y - yu[i][1]) < 4)

{

high = rand() % 50 + 330;

judge = 1;

}

if (abs(x - mian[i][0]) < 4 && abs(y - mian[i][1]) < 4)

{

high = rand() % 50 + 210;

judge = 1;

}

}

}

if (judge == 0)

{

randshu = rand() % 100;

if (randshu == 1)

{

if (getpixel(x, y) == BLUE)

{

high = rand() % 40 + 300;

}

else if (x > 1 && x < 203 && y < 278 && y>76)//梯田区域

{

high = rand() % 40 + 100+300;

}

else if (high < 150)

{

high = rand() % 40 + 200;

}

}

else high = 0;

}

if (high >= 150)

{

setcolor(WHITE);

setfillstyle(SOLID\_FILL, WHITE);

bar(177, 56, 230, 88); // 清除高度

setcolor(GREEN);

settextstyle(1, HORIZ\_DIR, 1);

sprintf(string, "height: %d cm", high);

outtextxy(107, 57, string);//显示字符串

}

}

**23.growlook.h**

#ifndef \_GROWLOOK\_H\_

#define \_GROWLOOK\_H\_

void grow(void);

void dronemove1(void);

void dronemove2(void);

void move21(int points[][2], int num\_points, int x, int y, int\* buffer);

int skipthing(int barrier[][2], int movebar[][2], int\* dx, int\* dy);

void paipai(int a[][2], int barrier2[][2]);

void showheight(int x, int y, int xiao[][2], int yu[][2], int mian[][2]);

#endif

**24.outcome.c**

#include"common.h"

/\*=========================

大农田小麦结果查看

===========================\*/

void xiaomaioutcome(void)

{

int y2, y3, y4;

int i;

double yt;

char string[20];

yt = 0;

y2 = 0;

y3 = 0;

y4 = 0;

draw111();

setcolor(GREEN);

line(64, 419, 563, 419);

line(556-5, 406, 563, 419);

line(563, 419, 556-5, 432);

line(64, 419, 64, 76);

line(64, 76, 53, 88);

line(64, 76, 72, 88);

puthz(200, 50, "小麦生长状况柱形图", 16, 15, GREEN);

puthz(400, 50, "总体：", 16, 15, GREEN);

puthz(70, 80, "占总小麦面积比例", 16, 15, GREEN);

puthz(200, 50, "小麦生长状况柱形图", 16, 15, GREEN);

line(63, 119, 65, 119);

settextstyle(1, HORIZ\_DIR, 1);

outtextxy(15, 110, "100%");

line(63, 419-150, 65, 419 - 150);

outtextxy(15, 419 - 150-9, "50%");

line(63, 419-75, 65, 419 - 75);

outtextxy(15, 419 - 75-9, "25%");

line(63, 419 - 225, 65, 419 - 225);

outtextxy(15, 419 - 225-9, "75%");

line(84, 418, 84, 420);

line(84+70, 418, 84+70, 420);

puthz(84+20, 430, "健康", 16, 15, GREEN);

line(84+70+30, 418, 84+70+30, 420);

line(84 + 70 + 30+70, 418,84 + 70 + 30+70, 420);

puthz(84+70+30+5, 430, "患白粉病", 16, 15, GREEN);

line(84 + 70 + 30 + 70+30, 418, 84 + 70 + 30 + 70+30, 420);

line(84 + 70 + 30 + 70 + 30+70, 418, 84 + 70 + 30 + 70+70 + 30, 420);

puthz(84 + 70 + 30 + 70 + 30+5, 430, "感染黏虫", 16, 15, GREEN);

line(84 + 70 + 30 + 70 + 30 + 70+30, 418, 84 + 70 + 30 + 70 +30+ 70 + 30, 420);

line(84 + 70 + 30 + 70 + 30 + 70+70 + 30, 418, 84 + 70 + 30 + 70 + 30 + 70 +70+ 30, 420);

puthz(84 + 70 + 30 + 70 + 30 + 70 + 30, 430, "两种均感染", 16, 15, GREEN);

for (i = 0; i < 192; i++)

{

if (bai.bfzuobiao[i][0] == 0 && bai.bfzuobiao[i][1] == 0&& nian.bfzuobiao[i][0] == 0 && nian.bfzuobiao[i][1]==0)

{

yt+=1;

}

if (bai.bfzuobiao[i][0] != 0 && bai.bfzuobiao[i][1] != 0)

{

y2++;

}

if (nian.bfzuobiao[i][0] != 0 && nian.bfzuobiao[i][1] != 0)

{

y3++;

}

if(bai.bfzuobiao[i][0] != 0 && bai.bfzuobiao[i][1] != 0&& nian.bfzuobiao[i][0] != 0 && nian.bfzuobiao[i][1] != 0)

{

y4++;

}

}

setcolor(RED);

settextstyle(1, HORIZ\_DIR, 1);

bar(84, 419 - (yt \* 300) / 192, 84 + 70, 419);

sprintf(string, "%.lf/192", yt);

outtextxy(84, 419 - (yt \* 300) / 192-20, string);

bar(84 + 100 + 70,419 - (y2 \* 300) / 192, 84 + 100, 419 );

sprintf(string, "%d/192", y2);

outtextxy(84 + 100 , 419 - (y2 \* 300) / 192-20, string);

bar( 84 + 270, 419 - (y3 \* 300) / 192, 84 + 200, 419);

sprintf(string, "%d/192", y3);

outtextxy(84 + 200, 419 - (y3 \* 300) / 192-20, string);

bar(84 +370, 419 - (y4 \* 300) / 192,84 + 300, 419 );

sprintf(string, "%d/192", y4);

outtextxy(84 + 300, 419 - (y4 \* 300) / 192-20, string);

if (yt >= 192 - 8)

{

puthz(370 + 84, 50, "健康", 16, 15, GREEN);//

}

else if (yt >= 192 - 15)

{

puthz(370 + 84, 50, "正常", 16, 15, GREEN);//

}

else if (yt >= 192 - 25)

{

puthz(370 + 84, 50, "病虫预防", 16, 15, GREEN);//

}

else if (yt >= 192 - 35)

{

puthz(370 + 84, 50, "急需喷洒", 16, 15, GREEN);//

}

else if (yt >= 192 - 60)

{

puthz(370 + 84, 50, "严重病害", 16, 15, GREEN);//

}

else

{

puthz(370 + 84, 50, "重大病害", 16, 15, GREEN);//

}

}

/\*=========================

大农田结果查看

===========================\*/

int outcome(int\* a3)

{

if (\*a3 == 1)

{

cleardevice();

draw111();

\*a3 = 0;

}

showbiankuang(600, 0, 640, 40);

showbiankuang(0, 0, 80, 40);

showbiankuang(311 - 25, 3, 421, 37);

showbiankuang(442, 3, 546 + 25, 37);

//界面的跳转

if (mouse\_press(600, 0, 640, 40) == 1)

{

return -1;//关闭BC。

}

else if (mouse\_press(0, 0, 80, 40) == 1)

{

\*a3 = 1;

delay(100);

return 2;//返回前一个界面

}

else if (mouse\_press(311 - 25, 3, 421, 35) == 1&&( \*a3==0 || \*a3== 3))

{

\*a3 = 2;

cleardevice();

delay(100);

xiaomaioutcome();

//小麦的

}

else if (mouse\_press(442, 3, 546 + 25, 35) == 1 && (\*a3 == 0 || \*a3 == 2))

{

\*a3 = 3;

cleardevice();

delay(100);

yumioutcome();

//玉米的

}

return 6;

}

/\*=========================

小农田结果查看

===========================\*/

int elseoutcome(int\* a3)

{

if (\*a3 == 1)

{

cleardevice();

draw222();

\*a3 = 0;

}

showbiankuang(600, 0, 640, 40);

showbiankuang(0, 0, 80, 40);

showbiankuang(197, 3, 325, 37);

showbiankuang(329, 3, 455, 37);

showbiankuang(461, 3, 586, 37);

//界面的跳转

if (mouse\_press(600, 0, 640, 40) == 1)

{

return -1;//关闭BC。

}

else if (mouse\_press(0, 0, 80, 40) == 1)

{

\*a3 = 1;

delay(100);

return 3;//返回前一个界面

}

else if (mouse\_press(197, 3, 325, 37) == 1 && (\*a3 == 0 || \*a3 == 3 || \*a3 == 4))

{

\*a3 = 2;

cleardevice();

delay(100);

elsexiaomaioutcome();

//小麦的

}

else if (mouse\_press(329, 3, 455, 37) == 1 && (\*a3 == 0 || \*a3 == 2 || \*a3 == 4))

{

\*a3 = 3;

cleardevice();

delay(100);

elseyumioutcome();

//玉米的

}

else if (mouse\_press(461, 3, 586, 37) == 1 && (\*a3 == 0 || \*a3 == 2 || \*a3 == 3))

{

\*a3 = 4;

cleardevice();

delay(100);

elsemianhuaoutcome();

//棉花的

}

return 5;

}

/\*=========================

小农田小麦结果查看

===========================\*/

void elsexiaomaioutcome(void)

{

int y2, y3, y4;

int i, j;

double yt;

char string[20];

int baifen[103][2];

int nianchong[103][2] ;

FILE\* filePointer11;

FILE\* filePointer12;

filePointer11 = fopen("data//xtxmbai.dat", "r"); // 以读取模式打开文件

filePointer12 = fopen("data//xtxmnian.dat", "r"); // 以读取模式打开文件

yt = 0;

y2 = 0;

y3 = 0;

y4 = 0;

draw222();

//坐标轴绘制

setcolor(GREEN);

line(64, 419, 563, 419);

line(556 - 5, 406, 563, 419);

line(563, 419, 556 - 5, 432);

line(64, 419, 64, 76);

line(64, 76, 53, 88);

line(64, 76, 72, 88);

puthz(200, 50, "小麦生长状况柱形图", 16, 15, GREEN);

puthz(400, 50, "总体：", 16, 15, GREEN);

puthz(70, 80, "占总小麦面积比例", 16, 15, GREEN);

puthz(200, 50, "小麦生长状况柱形图", 16, 15, GREEN);

line(63, 119, 65, 119);

settextstyle(1, HORIZ\_DIR, 1);

outtextxy(15, 110, "100%");

line(63, 419 - 150, 65, 419 - 150);

outtextxy(15, 419 - 150 - 9, "50%");

line(63, 419 - 75, 65, 419 - 75);

outtextxy(15, 419 - 75 - 9, "25%");

line(63, 419 - 225, 65, 419 - 225);

outtextxy(15, 419 - 225 - 9, "75%");

line(84, 418, 84, 420);

line(84 + 70, 418, 84 + 70, 420);

puthz(84 + 20, 430, "健康", 16, 15, GREEN);

line(84 + 70 + 30, 418, 84 + 70 + 30, 420);

line(84 + 70 + 30 + 70, 418, 84 + 70 + 30 + 70, 420);

puthz(84 + 70 + 30 + 5, 430, "患白粉病", 16, 15, GREEN);

line(84 + 70 + 30 + 70 + 30, 418, 84 + 70 + 30 + 70 + 30, 420);

line(84 + 70 + 30 + 70 + 30 + 70, 418, 84 + 70 + 30 + 70 + 70 + 30, 420);

puthz(84 + 70 + 30 + 70 + 30 + 5, 430, "感染黏虫", 16, 15, GREEN);

line(84 + 70 + 30 + 70 + 30 + 70 + 30, 418, 84 + 70 + 30 + 70 + 30 + 70 + 30, 420);

line(84 + 70 + 30 + 70 + 30 + 70 + 70 + 30, 418, 84 + 70 + 30 + 70 + 30 + 70 + 70 + 30, 420);

puthz(84 + 70 + 30 + 70 + 30 + 70 + 30, 430, "两种均感染", 16, 15, GREEN);

//统计植物病虫害情况。

for (i = 0; i < 103; i++)

{

int a, b;

a=fscanf(filePointer11, "%d %d", &baifen[i][0], &baifen[i][1]) ;

b=fscanf(filePointer12, "%d %d", &nianchong[i][0], &nianchong[i][1]) ;

if (a==2)

{

y2++;

}

if ( b== 2)

{

y3++;

}

}

for (i = 0; i <y2; i++)

{

for (j = 0; j < y3; j++)

{

if (baifen[i][0] == nianchong[j][0] && baifen[i][1] == nianchong[j][1])

{

y4++;

}

}

}

yt = 103 - y2 - y3 + y4;

fclose(filePointer11);

fclose(filePointer12);

//坐标轴。柱形图

setcolor(RED);

settextstyle(1, HORIZ\_DIR, 1);

bar(84, 419 - (yt \* 300) / 103, 84 + 70, 419);

sprintf(string, "%.lf/103", yt);

outtextxy(84, 419 - (yt \* 300) / 103 - 20, string);

bar(84 + 100 + 70, 419 - (y2 \* 300) / 103, 84 + 100, 419);

sprintf(string, "%d/103", y2);

outtextxy(84 + 100, 419 - (y2 \* 300) / 103 - 20, string);

bar(84 + 270, 419 - (y3 \* 300) / 103, 84 + 200, 419);

sprintf(string, "%d/103", y3);

outtextxy(84 + 200, 419 - (y3 \* 300) / 103 - 20, string);

bar(84 + 370, 419 - (y4 \* 300) / 103, 84 + 300, 419);

sprintf(string, "%d/103", y4);

outtextxy(84 + 300, 419 - (y4 \* 300) / 103 - 20, string);

//显示总体评价

if (yt >= 103 - 7)

{

puthz(370 + 84, 50, "健康", 16, 15, GREEN);//

}

else if (yt >= 103 - 13)

{

puthz(370 + 84, 50, "正常", 16, 15, GREEN);//

}

else if (yt >= 103 - 20)

{

puthz(370 + 84, 50, "病虫预防", 16, 15, GREEN);//

}

else if (yt >= 103 - 30)

{

puthz(370 + 84, 50, "急需喷洒", 16, 15, GREEN);//

}

else if (yt >= 103 - 50)

{

puthz(370 + 84, 50, "严重病害", 16, 15, GREEN);//

}

else

{

puthz(370 + 84, 50, "重大病害", 16, 15, GREEN);//

}

}

/\*=========================

小农田玉米结果查看

===========================\*/

void elseyumioutcome(void)

{

int y2, y3, y4;

int i, j;

double yt;

char string[20];

int baifen[103][2];

int nianchong[103][2];

FILE\* filePointer11;

FILE\* filePointer12;

filePointer11 = fopen("data//xtymnian.dat", "r"); // 以读取模式打开文件

filePointer12 = fopen("data//xtymmian.dat", "r");

yt = 0;

y2 = 0;

y3 = 0;

y4 = 0;

draw222();

setcolor(GREEN);

line(64, 419, 563, 419);

line(556 - 5, 406, 563, 419);

line(563, 419, 556 - 5, 432);

line(64, 419, 64, 76);

line(64, 76, 53, 88);

line(64, 76, 72, 88);

puthz(200, 50, "玉米生长状况柱形图", 16, 15, GREEN);

puthz(400, 50, "总体：", 16, 15, GREEN);

puthz(70, 80, "占总玉米株数比例", 16, 15, GREEN);

line(63, 119, 65, 119);

settextstyle(1, HORIZ\_DIR, 1);

outtextxy(15, 110, "100%");

line(63, 419 - 150, 65, 419 - 150);

outtextxy(15, 419 - 150 - 9, "50%");

line(63, 419 - 75, 65, 419 - 75);

outtextxy(15, 419 - 75 - 9, "25%");

line(63, 419 - 225, 65, 419 - 225);

outtextxy(15, 419 - 225 - 9, "75%");

line(84, 418, 84, 420);

line(84 + 70, 418, 84 + 70, 420);

puthz(84 + 20, 430, "健康", 16, 15, GREEN);

line(84 + 70 + 30, 418, 84 + 70 + 30, 420);

line(84 + 70 + 30 + 70, 418, 84 + 70 + 30 + 70, 420);

puthz(84 + 70 + 30 + 5, 430, "感染黏虫", 16, 15, GREEN);

line(84 + 70 + 30 + 70 + 30, 418, 84 + 70 + 30 + 70 + 30, 420);

line(84 + 70 + 30 + 70 + 30 + 70, 418, 84 + 70 + 30 + 70 + 70 + 30, 420);

puthz(84 + 70 + 30 + 70 + 30 + 5, 430, "感染棉铃虫", 16, 15, GREEN);

line(84 + 70 + 30 + 70 + 30 + 70 + 30, 418, 84 + 70 + 30 + 70 + 30 + 70 + 30, 420);

line(84 + 70 + 30 + 70 + 30 + 70 + 70 + 30, 418, 84 + 70 + 30 + 70 + 30 + 70 + 70 + 30, 420);

puthz(84 + 70 + 30 + 70 + 30 + 70 + 30, 430, "两种均感染", 16, 15, GREEN);

for (i = 0; i < 68; i++)

{

int a, b;

a = fscanf(filePointer11, "%d %d", &baifen[i][0], &baifen[i][1]);

b = fscanf(filePointer12, "%d %d", &nianchong[i][0], &nianchong[i][1]);

if (a == 2)

{

y2++;

}

if (b == 2)

{

y3++;

}

}

for (i = 0; i < y2; i++)

{

for (j = 0; j < y3; j++)

{

if (baifen[i][0] == nianchong[j][0] && baifen[i][1] == nianchong[j][1])

{

y4++;

}

}

}

yt = 68 - y2 - y3 + y4;

fclose(filePointer11);

fclose(filePointer12);

setcolor(RED);

settextstyle(1, HORIZ\_DIR, 1);

bar(84, 419 - (yt \* 300) / 68, 84 + 70, 419);

sprintf(string, "%.lf/68", yt);

outtextxy(84, 419 - (yt \* 300) / 68 - 20, string);

bar(84 + 100 + 70, 419 - (y2 \* 300) / 68, 84 + 100, 419);

sprintf(string, "%d/68", y2);

outtextxy(84 + 100, 419 - (y2 \* 300) / 68 - 20, string);

bar(84 + 270, 419 - (y3 \* 300) / 103, 84 + 200, 419);

sprintf(string, "%d/68", y3);

outtextxy(84 + 200, 419 - (y3 \* 300) / 68 - 20, string);

bar(84 + 370, 419 - (y4 \* 300) / 103, 84 + 300, 419);

sprintf(string, "%d/68", y4);

outtextxy(84 + 300, 419 - (y4 \* 300) / 68 - 20, string);

if (yt >= 68 -5)

{

puthz(370 + 84, 50, "健康", 16, 15, GREEN);//

}

else if (yt >= 68 - 10)

{

puthz(370 + 84, 50, "正常", 16, 15, GREEN);//

}

else if (yt >= 68 - 14)

{

puthz(370 + 84, 50, "病虫预防", 16, 15, GREEN);//

}

else if (yt >= 68 - 20)

{

puthz(370 + 84, 50, "急需喷洒", 16, 15, GREEN);//

}

else if (yt >= 68 - 40)

{

puthz(370 + 84, 50, "严重病害", 16, 15, GREEN);//

}

else

{

puthz(370 + 84, 50, "重大病害", 16, 15, GREEN);//

}

}

/\*=========================

小农田棉花结果查看

===========================\*/

void elsemianhuaoutcome(void)

{

int y2, y3, y4;

int i, j;

double yt;

char string[20];

int baifen[103][2];

int nianchong[103][2];

FILE\* filePointer11;

FILE\* filePointer12;

filePointer11 = fopen("data//xtmhmian.dat", "r");

filePointer12 = fopen("data//xtmhxiu.dat", "r");

yt = 0;

y2 = 0;

y3 = 0;

y4 = 0;

draw222();

setcolor(GREEN);

line(64, 419, 563, 419);

line(556 - 5, 406, 563, 419);

line(563, 419, 556 - 5, 432);

line(64, 419, 64, 76);

line(64, 76, 53, 88);

line(64, 76, 72, 88);

puthz(200, 50, "棉花生长状况柱形图", 16, 15, GREEN);

puthz(400, 50, "总体：", 16, 15, GREEN);

puthz(70, 80, "占总棉花株数比例", 16, 15, GREEN);

line(63, 119, 65, 119);

settextstyle(1, HORIZ\_DIR, 1);

outtextxy(15, 110, "100%");

line(63, 419 - 150, 65, 419 - 150);

outtextxy(15, 419 - 150 - 9, "50%");

line(63, 419 - 75, 65, 419 - 75);

outtextxy(15, 419 - 75 - 9, "25%");

line(63, 419 - 225, 65, 419 - 225);

outtextxy(15, 419 - 225 - 9, "75%");

line(84, 418, 84, 420);

line(84 + 70, 418, 84 + 70, 420);

puthz(84 + 20, 430, "健康", 16, 15, GREEN);

line(84 + 70 + 30, 418, 84 + 70 + 30, 420);

line(84 + 70 + 30 + 70, 418, 84 + 70 + 30 + 70, 420);

puthz(84 + 70 + 30 + 5, 430, "感染棉铃虫", 16, 15, GREEN);

line(84 + 70 + 30 + 70 + 30, 418, 84 + 70 + 30 + 70 + 30, 420);

line(84 + 70 + 30 + 70 + 30 + 70, 418, 84 + 70 + 30 + 70 + 70 + 30, 420);

puthz(84 + 70 + 30 + 70 + 30 + 5, 430, "患上锈病", 16, 15, GREEN);

line(84 + 70 + 30 + 70 + 30 + 70 + 30, 418, 84 + 70 + 30 + 70 + 30 + 70 + 30, 420);

line(84 + 70 + 30 + 70 + 30 + 70 + 70 + 30, 418, 84 + 70 + 30 + 70 + 30 + 70 + 70 + 30, 420);

puthz(84 + 70 + 30 + 70 + 30 + 70 + 30, 430, "两种均感染", 16, 15, GREEN);

for (i = 0; i < 68; i++)

{

int a, b;

a = fscanf(filePointer11, "%d %d", &baifen[i][0], &baifen[i][1]);

b = fscanf(filePointer12, "%d %d", &nianchong[i][0], &nianchong[i][1]);

if (a == 2)

{

y2++;

}

if (b == 2)

{

y3++;

}

}

for (i = 0; i < y2; i++)

{

for (j = 0; j < y3; j++)

{

if (baifen[i][0] == nianchong[j][0] && baifen[i][1] == nianchong[j][1])

{

y4++;

}

}

}

yt = 68 - y2 - y3 + y4;

fclose(filePointer11);

fclose(filePointer12);

setcolor(RED);

settextstyle(1, HORIZ\_DIR, 1);

bar(84, 419 - (yt \* 300) / 68, 84 + 70, 419);

sprintf(string, "%.lf/68", yt);

outtextxy(84, 419 - (yt \* 300) / 68 - 20, string);

bar(84 + 100 + 70, 419 - (y2 \* 300) / 68, 84 + 100, 419);

sprintf(string, "%d/68", y2);

outtextxy(84 + 100, 419 - (y2 \* 300) / 68 - 20, string);

bar(84 + 270, 419 - (y3 \* 300) / 103, 84 + 200, 419);

sprintf(string, "%d/68", y3);

outtextxy(84 + 200, 419 - (y3 \* 300) / 68 - 20, string);

bar(84 + 370, 419 - (y4 \* 300) / 103, 84 + 300, 419);

sprintf(string, "%d/68", y4);

outtextxy(84 + 300, 419 - (y4 \* 300) / 68 - 20, string);

if (yt >= 68 - 5)

{

puthz(370 + 84, 50, "健康", 16, 15, GREEN);//

}

else if (yt >= 68 - 9)

{

puthz(370 + 84, 50, "正常", 16, 15, GREEN);//

}

else if (yt >= 68- 15)

{

puthz(370 + 84, 50, "病虫预防", 16, 15, GREEN);//

}

else if (yt >= 68 - 25)

{

puthz(370 + 84, 50, "急需喷洒", 16, 15, GREEN);//

}

else if (yt >= 68 - 34)

{

puthz(370 + 84, 50, "严重病害", 16, 15, GREEN);//

}

else

{

puthz(370 + 84, 50, "重大病害", 16, 15, GREEN);//

}

}

/\*=========================

绘制大农田普适背景

===========================\*/

void draw111(void)

{

setbkcolor(WHITE);

setcolor(RED);

setfillstyle(1, GREEN);

rectangle(600, 0, 640, 40);

line(600, 0, 640, 40);

line(640, 0, 600, 40);//关闭按钮

rectangle(0, 0, 80, 40);

puthz(5, 5, "返回", 32, 40, GREEN);

line(0, 40, 640, 40);

setfillstyle(1, WHITE);

bar(90, 1, 140, 22);

setcolor(GREEN);

puthz(90 + 2, 1 + 2, "白粉病", 16, 15, GREEN);

setfillstyle(1, BLUE);

bar(145, 1, 195, 22);

puthz(145 + 2, 1 + 2, "棉铃虫", 16, 15, GREEN);

setfillstyle(1, YELLOW);

bar(200, 1, 250, 22);

puthz(200 + 2, 1 + 2, "黏虫病", 16, 15, GREEN);

setfillstyle(1, GREEN);

bar(311 - 25, 3, 421, 37);

bar(442, 3, 546 + 25, 37);

puthz(311 - 25 + 2, 3, "小麦情况", 32, 30, WHITE);

puthz(442 + 2, 3, "玉米情况", 32, 30, WHITE);

}

/\*=========================

绘制小农田普适背景

===========================\*/

void draw222(void)

{

setbkcolor(WHITE);

setcolor(RED);

setfillstyle(1, GREEN);

rectangle(600, 0, 640, 40);

line(600, 0, 640, 40);

line(640, 0, 600, 40);//关闭按钮

rectangle(0, 0, 80, 40);

puthz(5, 5, "返回", 32, 40, GREEN);

line(0, 40, 640, 40);

setfillstyle(1, WHITE);

bar(90, 0, 140, 19);

setcolor(GREEN);

puthz(90 + 2, 0 + 2, "白粉病", 16, 15, GREEN);

setfillstyle(1, BLUE);

bar(145, 0, 195, 19);

puthz(145 + 2, 0 + 2, "棉铃虫", 16, 15, GREEN);

setfillstyle(1, YELLOW);

bar(90, 19, 140, 38);

puthz(90 + 2, 19 + 2, "黏虫病", 16, 15, GREEN);

setfillstyle(1, LIGHTRED);

bar(145, 19, 195, 38);

puthz(145 + 2, 19 + 2, "锈病", 16, 15, GREEN);

setfillstyle(1, GREEN);

bar(197, 3, 325, 37);

bar(329, 3, 455, 37);

bar(461, 3, 586, 37);

puthz(197 + 2, 3, "小麦情况", 32, 30, WHITE);

puthz(329 + 2, 3, "玉米情况", 32, 30, WHITE);

puthz(461 + 2, 3, "棉花情况", 32, 30, WHITE);

}/\*=========================

大农田玉米结果查看

===========================\*/

void yumioutcome(void)

{

int y2, y3, y4;

int i;

double y1;

char string[20];

int j;

y1 = 0;

y2 = 0;

y3 = 0;

y4 = 0;

draw111();

setcolor(GREEN);

line(64, 419, 563, 419);

line(556 - 5, 406, 563, 419);

line(563, 419, 556 - 5, 432);

line(64, 419, 64, 76);

line(64, 76, 53, 88);

line(64, 76, 72, 88);

puthz(200, 50, "玉米生长状况柱形图", 16, 15, GREEN);

puthz(400, 50, "总体：", 16, 15, GREEN);

puthz(70, 80, "占玉米总株数比例", 16, 15, GREEN);

line(63, 119, 65, 119);

settextstyle(1, HORIZ\_DIR, 1);

outtextxy(15, 110, "100%");

line(63, 419 - 150, 65, 419 - 150);

outtextxy(15, 419 - 150 - 9, "50%");

line(63, 419 - 75, 65, 419 - 75);

outtextxy(15, 419 - 75 - 9, "25%");

line(63, 419 - 225, 65, 419 - 225);

outtextxy(15, 419 - 225 - 9, "75%");

line(84, 418, 84, 420);

line(84 + 70, 418, 84 + 70, 420);

puthz(84 + 20, 430, "健康", 16, 15, GREEN);

line(84 + 70 + 30, 418, 84 + 70 + 30, 420);

line(84 + 70 + 30 + 70, 418, 84 + 70 + 30 + 70, 420);

puthz(84 + 70 + 30, 430, "感染棉铃虫", 16, 15, GREEN);

line(84 + 70 + 30 + 70 + 30, 418, 84 + 70 + 30 + 70 + 30, 420);

line(84 + 70 + 30 + 70 + 30 + 70, 418, 84 + 70 + 30 + 70 + 70 + 30, 420);

puthz(84 + 70 + 30 + 70 + 30 + 5, 430, "感染黏虫", 16, 15, GREEN);

line(84 + 70 + 30 + 70 + 30 + 70 + 30, 418, 84 + 70 + 30 + 70 + 30 + 70 + 30, 420);

line(84 + 70 + 30 + 70 + 30 + 70 + 70 + 30, 418, 84 + 70 + 30 + 70 + 30 + 70 + 70 + 30, 420);

puthz(84 + 70 + 30 + 70 + 30 + 70 + 30, 430, "两种均感染", 16, 15, GREEN);

//统计各个植物的患病情况

for (i = 0, j = 192; i < 192, j < 384; i++, j++)

{

if (mian.bfzuobiao[i][0] == 0 && mian.bfzuobiao[i][1] == 0 && nian.bfzuobiao[j][0] == 0 && nian.bfzuobiao[j][1] == 0)

{

y1 += 1;//无病

}

if (mian.bfzuobiao[i][0] != 0 && mian.bfzuobiao[i][1] != 0)

{

y2++;

}

if (nian.bfzuobiao[j][0] != 0 && nian.bfzuobiao[j][1] != 0)

{

y3++;

}

if (mian.bfzuobiao[i][0] != 0 && mian.bfzuobiao[i][1] != 0 && nian.bfzuobiao[j][0] != 0 && nian.bfzuobiao[j][1] != 0)

{

y4++;//二者均患

}

}

//绘制坐标轴，以及显示柱形图

setcolor(RED);

settextstyle(1, HORIZ\_DIR, 1);

bar(84, 419 - (y1 \* 300) / 192, 84 + 70, 419);

sprintf(string, "%.lf/192", y1);

outtextxy(84, 419 - (y1 \* 300) / 192 - 20, string);

bar(84 + 100 + 70, 419 - (y2 \* 300) / 192, 84 + 100, 419);

sprintf(string, "%d/192", y2);

outtextxy(84 + 100, 419 - (y2 \* 300) / 192 - 20, string);

bar(84 + 270, 419 - (y3 \* 300) / 192, 84 + 200, 419);

sprintf(string, "%d/192", y3);

outtextxy(84 + 200, 419 - (y3 \* 300) / 192 - 20, string);

bar(84 + 370, 419 - (y4 \* 300) / 192, 84 + 300, 419);

sprintf(string, "%d/192", y4);

outtextxy(84 + 300, 419 - (y4 \* 300) / 192 - 20, string);

//显示生长状态

if (y1 >= 192 - 8)//8

{

puthz(370 + 84, 50, "健康", 16, 15, GREEN);

}

else if (y1 >= 192 - 15)//15

{

puthz(370 + 84, 50, "正常", 16, 15, GREEN);

}

else if (y1 >= 192 - 25)//25

{

puthz(370 + 84, 50, "病虫预防", 16, 15, GREEN);

}

else if (y1 >= 192 - 35)//35

{

puthz(370 + 84, 50, "急需喷洒", 16, 15, GREEN);

}

else if (y1 >= 192 - 60)//60

{

puthz(370 + 84, 50, "严重病害", 16, 15, GREEN);

}

else

{

puthz(370 + 84, 50, "重大病害", 16, 15, GREEN);

}

}

**25.outcome.h**

#ifndef \_OUTCOME\_H\_

#define \_OUTCOME\_H\_

//=大农田的 结果查看

int outcome(int\* a3);//核心

void draw111(void);//普通绘图

void xiaomaioutcome(void);//小麦结果

void yumioutcome(void);//玉米结果

//=====================================

int elseoutcome(int\* a3);

void draw222(void);

void elseyumioutcome(void);

void elsexiaomaioutcome(void);

void elsemianhuaoutcome(void);

#endif

**26.illness.c**

#include "common.h"

/\*==========================

植物患病的概率

===========================\*/

int randill(int k)//这里j代表系数,注意0，1，2，3.

{

int judge;

int i, j;

int a[4][60] =

{

//棉铃虫对于玉米的

{40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40, 40,40,40,40,5,5,5,5,7,7,7,7,7,8,8,8,8,10,10,10,10,11,11,11,12,13,13,14,14,14,14,14,15,15,16,17,18,19,20,20,20,20,20,20 },

//黏虫对于小麦的

{40,40,40,40,40,40,40,40,40,40,40,40,40,40,40, 40, 40,40,40,40,5,5,5,6,6,6,6,8,8,8,8,9,9,9,9,10,10,10,10,11,11,11,12,12,12,13,14,14,15,16,17,18,19,20,21,22,22,22,22,22 },

//黏虫对于玉米的

{40,40,40,40,40,40,40,40,40,40,40,40,40,40,40, 40, 40,40,40,40,5,5,6,6,6,6,7,7,7,9,9,9,10,10,10,11,11,11,12,12,12,13,13,13,13,13,14,14,15,16,16,17,18,19,20,21,22,22,22,22 },

//白粉对于小麦的

{40,40,40,40,40,40,40,40,40,40,40,40,40,40,40,40, 40,40,40,40,5,5,5,5,5,5,7,7,7,7,7,8,8,8,8,10,10,10,10,11,11,11,12,13,13,14,14,14,14,14,16,17,18,19,20,20,20,20,20,20 },

};

int b[4][60] = { 0 };

judge = rand() % 8;

if (judge != 0)//有8分之一的概率全部是正常

{

for (i = 0; i < 4; i++)

{

for (j = 0; j < 60; j++)

{

b[i][j] = a[i][j];

}

}

}

else

{

for (i = 0; i < 4; i++)

{

for (j = 0; j < 60; j++)

{

b[i][j] = 999;

}

}

}

i = rand() % 60;

return b[k][i];

}

/\*==========================

显示棉铃虫，大农田

===========================\*/

void showmianling(unsigned char aaa[192][49])//棉铃虫。感染玉米和棉花。

{

// int aaa[49] = { 0 };//

int temp;

int judge = 0;

int j;

int jia2[] = { 25,75,25,75,25,75,25,75 };//控制y的加的数，使其跳过小麦区域。

int x;

int y;

int i;

x = 0;

y = 0;

i = 0;

temp = randill(0);//控制发病率的调整比例

for (x = 20; x < 620 - 25 + 1; x = x + 25)//玉米区域的发病率

{

j = -1;

for (y = 65 + 50; y < 465; y = y + jia2[j])//3行

{

ifmianling(aaa[i]);//控制点的形成

j++;

judge = rand() % temp;

if (judge == 0)//

{

mianling(x, y,aaa[i]);

mian.bfzuobiao[i][0] = x;

mian.bfzuobiao[i][1] = y;

i++;

}

else

{

i++;

}

}

}

}

/\*==========================

显示棉铃虫点列

===========================\*/

void mianling(int x, int y, unsigned char aaa[49])//

{

int i;

int j;

int jjj;

int mlxy[49][2] =

{

//蓝色点所有可能的位置...49个点

{ 7 , 3 },

{ 19 , 1 },

{ 20 ,1 },

{ 6 , 5 },

{ 9, 5 },

{ 18 , 4},

{ 18 , 5},

{ 11 , 6 },

{ 14 , 6 },

{ 5 ,7 },

{ 2 , 8 },

{ 7 ,8 },

{ 13 ,7 },

{ 17 , 7 },

{ 22 ,7 },

{ 19 , 8 },

{ 20 , 8 },

{ 6 , 10 },

{ 4 , 11 },

{ 1 ,13 },

{ 3 , 16 },

{ 1 ,17 },

{ 1 , 18 },

{ 2 , 18 },

{ 7 , 19 },

{ 7 , 20 },

{ 5 ,23 },

{ 7 , 23 },

{ 10 , 12 },

{ 10 , 13 },

{ 9 , 17 },

{ 11 , 15 },

{ 12 , 14 },

{ 19 , 12 },

{ 22 , 12 },

{ 22 ,13 },

{ 22 , 15 },

{ 18 , 15 },

{ 19 , 16 },

{ 13 , 17 },

{ 15 , 16 },

{ 15 , 19 },

{ 16 , 19 },

{ 14 , 21 },

{ 15 , 23 },

{ 16 , 22 },

{ 17 , 24},

{ 19 , 21 },

{ 21 , 23 },

};

for (i = 0; i < 49; i++)

{

mlxy[i][0] = mlxy[i][0] + x;

mlxy[i][1] = mlxy[i][1] + y;

}

for (j = 0; j < 49; j++)

{

jjj = getshu(aaa, j);

if ( jjj== 1)

{

setcolor(BLUE);

putpixel(mlxy[j][0], mlxy[j][1], BLUE);

}

}

}

/\*==========================

使得能够随机显示棉铃虫点列

===========================\*/

void ifmianling(unsigned char aaa[49])//随机显示

{

int judge[49];

int i;

for (i = 0; i < 49; i++)

{

judge[i] = rand() % 2;//显示的概率为 1/2..

if (judge[i] == 1)

{

setshu(aaa, i, 0);//不显示

}

else

{

setshu(aaa, i, 1);//显示

}

}

}

void shownianchong(unsigned char aaa[384][49])//黏虫。两者均患

{

int judge = 0;

int j;

int jia1[] = { 25,75,25,75,25,75,25,75 };

int jia2[] = { 25,75,25,75,25,75,25,75 };

int x;

int y;

int i;

int temp1, temp2;

temp1 = randill(1);

temp2 = randill(2);

x = 0;

y = 0;

i = 0;

for (x = 20; x < 620 - 25 + 1; x = x + 25)//小麦区域的发病率

{

j = -1;

for (y = 65; y < 465; y = y + jia1[j])

{

ifnianchong(aaa[i]);

j++;

judge = rand() % temp1;//

if (judge == 0)//

{

nianchong(x, y,aaa[i]);

nian.bfzuobiao[i][0] = x;

nian.bfzuobiao[i][1] = y;

i++;

}

else

{

i++;

}

}

}

i = 192;//就是说，nian.bfzuobiao[i][],第1--192个是小麦区域的得了黏虫病的坐标，第193-384个是玉米的

//注意， 进行了农药喷洒后，请注意把对应的坐标 重新赋值为0

for (x = 20; x < 620 - 25 + 1; x = x + 25)//玉米区域的发病率

{

j = -1;

for (y = 65+50; y < 465; y = y + jia2[j])//3行

{

ifnianchong(aaa[i]);

j++;

judge = rand() % temp2;// 1/4的发病率

if (judge == 0)//

{

nianchong(x, y,aaa[i]);

nian.bfzuobiao[i][0] = x;

nian.bfzuobiao[i][1] = y;

i++;

}

else

{

i++;

}

}

}

}

void nianchong(int x, int y, unsigned char aaa[49])//

{

int i;

int j;

int jjj;

int ncxy[43][2] =

{

//洋红色点所有可能的位置...43个点

{ 1 , 2 },

{ 1 , 5 },

{ 3 , 3 },

{ 2 ,20 },

{ 2 , 22 },

{ 4 , 20 },

{ 6 , 13 },

{ 6 , 14 },

{ 6 , 16 },

{ 7 , 14 },

{ 10 , 9 },

{ 10 , 11 },

{ 11 , 10 },

{ 13 ,11 },

{ 15 , 9 },

{ 16 , 10 },

{ 16 , 11 },

{ 17 , 10 },

{ 18 , 9 },

{ 11 , 18 },

{ 11 ,19 },

{ 11 , 20 },

{ 10 ,19 },

{ 10 , 21 },

{ 17 , 17},

{ 16 , 18 },

{ 17 , 18 },

{ 18 , 18 },

{ 22 ,17 },

{ 22 , 19 },

{ 21 , 18 },

{ 21 , 19 },

{ 21 , 20 },

{ 11 , 2 },

{ 13 , 2 },

{ 13 , 3 },

{ 15 , 3 },

{ 16 , 2 },

{ 20 , 3},

{ 20 , 4 },

{ 21 , 5 },

{ 22 , 3 },

{ 22 , 4 }

};

for (i = 0; i < 43; i++)

{

ncxy[i][0] = ncxy[i][0] + x;

ncxy[i][1] = ncxy[i][1] + y;

}

for (j = 0; j < 43; j++)

{

jjj = getshu(aaa, j);

if (jjj == 1)

{

setcolor(YELLOW);

putpixel(ncxy[j][0], ncxy[j][1], YELLOW);

}

}

}

void ifnianchong(unsigned char aaa[49])//随机显示

{

int judge[43];

int i;

for (i = 0; i < 43; i++)

{

judge[i] = rand() % 2;//显示的概率为 1/2.

if (judge[i] == 1)

{

setshu(aaa, i, 0);//不显示

}

else

{

setshu(aaa, i, 1);//显示

}

}

}

void showbaifen(unsigned char abc[192][49])//白粉病的发病率

{

int judge = 0;

int j;

int jia[] = { 25,75,25,75,25,75,25,75 };//控制y的加的数，使其跳过玉米区域。

int x;

int y;

int i;

int temp;

temp = randill(3);

x = 0;

y = 0;

i = 0;

for (x = 20; x < 620 - 25 + 1; x = x + 25)//96\*2片区域,,这是24列

{

j = -1;

for (y = 65; y < 465; y = y + jia[j])//6行

{

ifbaifen(abc[i]);

j++;

judge = rand() % temp;

if (judge == 0)// 1/4的概率得病

{

baifen(x, y,abc[i]);

bai.bfzuobiao[i][0] = x;////储存区域生病的坐标；

bai.bfzuobiao[i][1] = y;

i++;

}

else

{

i++;

}

}

}

}

void baifen(int x, int y, unsigned char abc[49])//画出白粉病，适用于俯视图。25\*25大小

{

int i;

int j;

int jjj;

int bfxy[34][2] =

{

//白色点所有可能的位置

{5, 5},

{ 7, 4},

{ 9, 5},

{ 7, 9},

{ 4, 9},

{ 4, 14},

{ 3, 14},

{ 8, 18},

{ 8, 19},

{ 9, 14},

{ 13, 5},

{ 13, 6},

{ 13, 7},

{ 13, 8},

{ 13, 14},

{ 14, 14},

{ 14, 17},

{ 14, 20},

{ 18, 21},

{ 19, 15},

{ 20, 11},

{ 16, 6},

{ 17, 6},

{ 18, 3},

{ 5, 1 },

{ 21 , 1 },

{ 21 , 23 },

{ 5 ,22 },

{ 10 ,22 },

{ 1 , 13 },

{ 1 , 18 },

{ 3 , 11 },

{22 , 9 },

{ 23 , 19}//34个点

};

for (i = 0; i < 34; i++)

{

bfxy[i][0] = bfxy[i][0] + x;

bfxy[i][1] = bfxy[i][1] + y;

}

for (j = 0; j < 34; j++)

{

jjj = getshu(abc, j);

if (jjj == 1)

{

setcolor(WHITE);

putpixel(bfxy[j][0], bfxy[j][1], WHITE);

}

}

}

void ifbaifen(unsigned char abc[49])//使得白粉病 设置白点的随机显示

{

int judge[34];

int i;

for (i = 0; i < 34; i++)

{

judge[i] = rand() % 3;//显示的概率为 2/3.。。

if (judge[i] == 1)

{

setshu(abc, i, 0);//不显示

}

else

{

setshu(abc, i, 1);

}

}

}

//============================================================================================================

//============================================================================================================

//下面为小农田

void showxtbaifen(void)//小农田显示白粉病

{

FILE\* filePointer4;

FILE\* filePointer5;

int plantPositions3[104][2];

int judge = 0;

int j;

int x;

int i;

int temp;

temp = randill(3);

x = 0;

i = 0;

filePointer4 = fopen("data//xtxmbai.dat", "w");

filePointer5 = fopen("data//xmbfdian.dat", "w");

getxtxiao(plantPositions3);//使得小麦坐标到了plantpositions3

for (x = 0; x < 104; x++)

{

judge = rand() % temp;

if (judge == 0)

{

xtbaifen(plantPositions3[i][0], plantPositions3[i][1], filePointer5);

fprintf(filePointer4, "%d %d\n", plantPositions3[i][0], plantPositions3[i][1]);

i++;

}

else

{

i++;

}

}

fclose(filePointer4); //

fclose(filePointer5); // 关闭文件

}

void xtbaifen(int x, int y, FILE\* filePointer5)

{

int i;

int j;

int bfxy[31][2] =

{

//白色点所有可能的位置 31

{ 1 , 2 },

{ 2 , 6 },

{ 2 , 10 },

{ 2 , 13 },

{ 1 , 15 },

{ 3 , 8 },

{ 3 , 18 },

{ 4 , 12 },

{ 5 , 4 },

{ 5 , 8 },

{ 5 , 14 },

{ 5 , 16 },

{ 5 , 17 },

{ 7 , 9 },

{ 10 , 2 },

{ 10 , 3 },

{ 10 , 8 },

{ 10 , 13 },

{ 10 , 17 },

{ 12 , 8 },

{ 12 , 10 },

{ 14 , 6 },

{ 14 , 13 },

{ 15 , 2 },

{ 15 , 16 },

{ 18 , 3 },

{ 17 , 8 },

{ 17 , 13 },

{ 18 , 11 },

{ 19 , 11 },

{ 19 , 17 }

};

int ttemp;

for (i = 0; i < 31; i++)

{

bfxy[i][0] = bfxy[i][0] + x;

bfxy[i][1] = bfxy[i][1] + y;

}

for (j = 0; j < 31; j++)

{

ttemp = rand() % 2;

if (ttemp == 1)

{

setcolor(WHITE);

putpixel(bfxy[j][0], bfxy[j][1], WHITE);

fprintf(filePointer5, "%d %d\n", bfxy[j][0], bfxy[j][1]);

}

}

}

void showxtnianchong(void)//小农田

{

FILE\* filePointer4;

FILE\* filePointer5;

int plantPositions3[104][2];

int judge = 0;

int j;

int x;

int i;

int temp;

temp = randill(3);

x = 0;

i = 0;

filePointer4 = fopen("data//xtxmnian.dat", "w"); // 以读取模式打开文件

filePointer5 = fopen("data//xmncdian.dat", "w"); // 以读取模式打开文件

getxtxiao(plantPositions3);//使得小麦坐标到了plantpositions3

for (x = 0; x < 104; x++)//

{

judge = rand() % temp;

if (judge == 0)

{

xtnianchong(plantPositions3[i][0], plantPositions3[i][1], filePointer5);

fprintf(filePointer4, "%d %d\n", plantPositions3[i][0], plantPositions3[i][1]);

i++;

}

else

{

i++;

}

}

fclose(filePointer4); // 关闭文件

fclose(filePointer5);

}

void xtnianchong(int x, int y, FILE\* filePointer5)

{

int i;

int j;

int bfxy[27][2] =

{

//黄色点所有可能的位置 27

{ 4, 1 },

{ 1 , 4 },

{ 1 , 8 },

{ 1 , 12 },

{ 2 , 16 },

{ 3 , 15 },

{ 7 , 3 },

{ 7 , 7 },

{ 5 , 11 },

{ 7 , 13 },

{ 7 , 16 },

{ 8 , 16 },

{ 7 , 18 },

{ 9 , 6 },

{ 10 , 6 },

{ 9 , 11 },

{ 12 , 4 },

{ 12 , 15 },

{ 12 , 18 },

{ 13, 12 },

{ 15 , 8 },

{ 15 , 10 },

{ 17 , 2 },

{ 17 , 6 },

{ 16, 18 },

{ 17 , 18 },

{ 18 , 15 }

};

int ttemp;

for (i = 0; i < 27; i++)

{

bfxy[i][0] = bfxy[i][0] + x;

bfxy[i][1] = bfxy[i][1] + y;

}

for (j = 0; j < 27; j++)

{

ttemp = rand() % 2;

if (ttemp == 1)

{

setcolor(YELLOW);

putpixel(bfxy[j][0], bfxy[j][1], YELLOW);

fprintf(filePointer5, "%d %d\n", bfxy[j][0], bfxy[j][1]);

}

}

}

void showxtymmian(void)//小农田玉米的棉虫病坐标

{

FILE\* filePointer4;

FILE\* filePointer5;

int plantPositions3[68][2];

int judge = 0;

int j;

int x;

int i;

int temp;

temp = randill(2);

x = 0;

i = 0;

filePointer4 = fopen("data//xtymmian.dat", "w");

filePointer5 = fopen("data//ymmldian.dat", "w");

getxtyumi(plantPositions3);

for (x = 0; x < 68; x++)

{

judge = rand() % temp;

if (judge == 0)

{

xtymmian(plantPositions3[i][0], plantPositions3[i][1], filePointer5);

fprintf(filePointer4, "%d %d\n", plantPositions3[i][0], plantPositions3[i][1]);

i++;

}

else

{

i++;

}

}

fclose(filePointer4); // 关闭文件

fclose(filePointer5); // 关闭文件

}

void xtymmian(int x, int y,FILE\* filePointer5)//画出

{

int i;

int j;

int bfxy[27][2] =

{

//蓝色点所有可能的位置 27

{ 5 , 6 },

{ 5 , 8 },

{ 8 , 6 },

{ 10 ,5 },

{ 12 , 6 },

{ 14 , 5 },

{ 15 , 7 },

{ 14 , 9 },

{ 13 , 12 },

{ 13 , 14 },

{ 10 , 19 },

{ 11 , 20 },

{ 9 , 23 },

{ 14 , 18 },

{ 14 , 23 },

{ 14 , 26 },

{ 13 , 29 },

{ 13 , 34 },

{ 16 , 29 },

{ 16 , 34 },

{ 17 , 11 },

{ 17 , 22 },

{ 19 , 20 },

{ 21 , 19 },

{ 23 , 18 },

{ 24 , 21 },

{ 25 , 23 }

};

int ttemp;

for (i = 0; i < 27; i++)

{

bfxy[i][0] = bfxy[i][0] + x;

bfxy[i][1] = bfxy[i][1] + y;

}

for (j = 0; j < 27; j++)

{

ttemp = rand() % 2;

if (ttemp == 1)

{

setcolor(BLUE);

putpixel(bfxy[j][0], bfxy[j][1], BLUE);

fprintf(filePointer5, "%d %d\n", bfxy[j][0], bfxy[j][1]);

}

}

}

void showxtymnian(void)

{

FILE\* filePointer4;

FILE\* filePointer5;

int plantPositions3[68][2];

int judge = 0;

int j;

int x;

int i;

int temp;

temp = randill(2);

x = 0;

i = 0;

filePointer4 = fopen("data//xtymnian.dat", "w");

filePointer5 = fopen("data//ymncdian.dat", "w");

getxtyumi(plantPositions3);

for (x = 0; x < 68; x++)

{

judge = rand() % temp;

if (judge == 0)

{

xtymnian(plantPositions3[i][0], plantPositions3[i][1], filePointer5);

fprintf(filePointer4, "%d %d\n", plantPositions3[i][0], plantPositions3[i][1]);

i++;

}

else

{

i++;

}

}

fclose(filePointer4);

fclose(filePointer5);

}

void xtymnian(int x, int y, FILE\* filePointer5)

{

int i;

int j;

int bfxy[26][2] =

{

//蓝色点所有可能的位置 26

{ 3 , 10 },

{ 4 , 7 },

{ 6 , 4 },

{ 6 , 6 },

{ 8 , 8 },

{ 11 , 5 },

{ 10, 6 },

{ 14 , 3 },

{ 16 , 5 },

{ 13 , 8 },

{ 12 , 10 },

{ 24 , 19 },

{ 23 , 20 },

{ 26 , 24 },

{ 16 , 20 },

{ 17 , 20 },

{ 14 , 20 },

{ 9 , 21 },

{ 9 , 26 },

{ 13 , 25 },

{ 15 , 27 },

{ 15 , 32 },

{ 13 , 31 },

{ 14 , 21 },

{ 15 , 23 },

{ 11 , 21 }

};

int ttemp;

for (i = 0; i < 26; i++)

{

bfxy[i][0] = bfxy[i][0] + x;

bfxy[i][1] = bfxy[i][1] + y;

}

for (j = 0; j < 26; j++)

{

ttemp = rand() % 2;

if (ttemp == 1)

{

setcolor(YELLOW);

putpixel(bfxy[j][0], bfxy[j][1], YELLOW);

fprintf(filePointer5, "%d %d\n", bfxy[j][0], bfxy[j][1]);

}

}

}

void showxtmhmian(void)//小农田 棉花的棉铃虫病坐标

{

FILE\* filePointer4;

FILE\* filePointer5;

int plantPositions3[68][2];

int judge = 0;

int j;

int x;

int i;

int temp;

temp = randill(1);

x = 0;

i = 0;

filePointer4 = fopen("data//xtmhmian.dat", "w");

filePointer5 = fopen("data//mhmldian.dat", "w");

getxtmianhua(plantPositions3);

for (x = 0; x < 68; x++)

{

judge = rand() % temp;

if (judge == 0)

{

xtmhmian(plantPositions3[i][0], plantPositions3[i][1],filePointer5);

fprintf(filePointer4, "%d %d\n", plantPositions3[i][0], plantPositions3[i][1]);

i++;

}

else

{

i++;

}

}

fclose(filePointer4);

fclose(filePointer5);

}

void xtmhmian(int x, int y, FILE\* filePointer5)

{

int i;

int j;

int bfxy[33][2] =

{

// 33

{ 9 , 2 },

{ 9 , 5 },

{ 8 , 6 },

{ 11 , 3 },

{ 12 , 6 },

{ 11 , 9 },

{ 5 , 12 },

{ 4 , 14 },

{ 6 , 15 },

{ 9 , 15 },

{ 5 , 18 },

{ 11 , 16 },

{ 12 , 13 },

{ 13 , 20 },

{ 12 , 23 },

{ 11 , 25 },

{ 13 , 25 },

{ 12 , 27 },

{ 12 , 29 },

{ 16 , 2 },

{ 16 , 4 },

{ 14 , 5 },

{ 15 , 8 },

{ 13 , 9 },

{ 13 , 11 },

{ 17 , 6 },

{ 19 , 6 },

{ 20 , 8 },

{ 19 , 12 },

{ 18 , 15 },

{ 20 , 15 },

{ 19 , 18 },

{ 21 , 18 }

};

int ttemp;

for (i = 0; i < 33; i++)

{

bfxy[i][0] = bfxy[i][0] + x-1;

bfxy[i][1] = bfxy[i][1] + y+1;

}

for (j = 0; j < 33; j++)

{

ttemp = rand() % 2;

if (ttemp == 1)

{

setcolor(BLUE);

putpixel(bfxy[j][0], bfxy[j][1], BLUE);

fprintf(filePointer5, "%d %d\n", bfxy[j][0], bfxy[j][1]);

}

}

}

void showxtmhxiu(void)

{

FILE\* filePointer4;

FILE\* filePointer5;

int plantPositions3[68][2];

int judge = 0;

int j;

int x;

int i;

int temp;

temp = randill(1);

x = 0;

i = 0;

filePointer4 = fopen("data//xtmhxiu.dat", "w");

filePointer5 = fopen("data//mhxbdian.dat", "w");

getxtmianhua(plantPositions3);

for (x = 0; x < 68; x++)

{

judge = rand() % temp;

if (judge == 0)

{

xtmhxiu(plantPositions3[i][0], plantPositions3[i][1], filePointer5);

fprintf(filePointer4, "%d %d\n", plantPositions3[i][0], plantPositions3[i][1]);

i++;

}

else

{

i++;

}

}

fclose(filePointer4);

fclose(filePointer5);

}

void xtmhxiu(int x, int y, FILE\* filePointer5)//

{

int i;

int j;

int bfxy[35][2] =

{

//洋红色点所有可能的位置 35

{ 8 , 0 },

{ 6 , 5 },

{ 5 , 6 },

{ 4 , 16 },

{ 6 , 13 },

{ 8 , 14 },

{ 7 , 17 },

{ 11 , 1 },

{ 10 , 2 },

{ 10 , 8 },

{ 10 , 7 },

{ 11 , 12 },

{ 11 , 15 },

{ 11 , 18 },

{ 11 , 20 },

{ 11 , 22 },

{ 11 , 31 },

{ 12 , 32 },

{ 12 , 10 },

{ 13 , 15 },

{ 13 , 16 },

{ 13 , 18 },

{ 13 , 22 },

{ 13 , 30 },

{ 14 , 10 },

{ 16 , 9 },

{ 18 , 4 },

{ 18 , 8 },

{ 20 , 6 },

{ 17 , 13 },

{ 18 , 17 },

{ 19 , 14 },

{ 19 , 16 },

{ 21 , 14 },

{ 21 , 17 }

};

int ttemp;

for (i = 0; i < 35; i++)

{

bfxy[i][0] = bfxy[i][0] + x-1;

bfxy[i][1] = bfxy[i][1] + y+1;

}

for (j = 0; j < 35; j++)

{

ttemp = rand() % 2;

if (ttemp == 1)

{

setcolor(LIGHTRED);

putpixel(bfxy[j][0], bfxy[j][1], LIGHTRED);

fprintf(filePointer5, "%d %d\n", bfxy[j][0], bfxy[j][1]);

}

}

}

**27.illness.h**

#ifndef \_ILLNESS\_H\_

#define \_ILLNESS\_H\_

//大农田显示

void showbaifen(unsigned char abc[192][49]);

void baifen(int x, int y, unsigned char abc[49]);

void ifbaifen(unsigned char abc[49]);

void nianchong(int x, int y, unsigned char aaa[49]);//画黏虫

void ifnianchong(unsigned char aaa[49]);//随机点列

void shownianchong(unsigned char aaa[384][49]);

void ifmianling(unsigned char aaa[49]);//随机显示

void mianling(int x, int y, unsigned char aaa[49]);

void showmianling(unsigned char aaa[192][49]);

int randill(int j);//控制不同的病的不同的发病率。就是使得一种病，每次检测时，的发病率不同

//小农田显示

void showxtbaifen(void);

void xtbaifen(int x, int y, FILE\* filePointer5);//画出白粉病，适用于俯视图。25\*25大小

void showxtnianchong(void);

void xtnianchong(int x, int y, FILE\* filePointer5);//画出白粉病，适用于俯视图。25\*25大小

void xtymmian(int x, int y, FILE\* filePointer5);

void showxtymmian(void);

void showxtymnian(void);

void xtymnian(int x, int y, FILE\* filePointer5);

void showxtmhmian(void);

void xtmhmian(int x, int y, FILE\* filePointer5);

void showxtmhxiu(void);

void xtmhxiu(int x, int y, FILE\* filePointer5);

#endif

**28.recover.c**

#include "common.h"

//===========================

//大农田里面的植物自愈

//===========================

void recover(void)//完全随机重新刷新；；

{

int j;//跳过区域用

int jia1[] = { 25,75,25,75,25,75,25,75 };//跳过玉米区域。

int jia2[] = { 25,75,25,75,25,75,25,75 };//跳过小麦区域。

int x, y;//循环

int i = 0;//调用患病植株坐标

int ii;//同上

int judge = 0;//恢复比例的分子

int temp1, temp2;//自愈比例分母

temp1 = randrecover(0);//小麦自愈比例

temp2 = randrecover(1);//玉米自愈比例

for (x = 20; x < 620 - 25 + 1; x = x + 25)//小麦区域,

{

j = -1;

for (y = 65; y < 465; y = y + jia1[j])//

{

j++;

judge = rand() % temp1;//

if (judge == 0)//

{

setcolor(BROWN);

setfillstyle(1, BROWN);

bar(x, y, x + 25, y + 25);//矩形覆盖原生病植物。再画健康植物

fxiaomai(x, y);

nian.bfzuobiao[i][0] = 0;

nian.bfzuobiao[i][1] = 0;

bai.bfzuobiao[i][0] = 0;

bai.bfzuobiao[i][1] = 0;//使其为0；表示其恢复健康。

i++;

}

else i++;

}

}

i = 192;

ii = 0;

for (x = 20; x < 620 - 25 + 1; x = x + 25)//玉米区域

{

j = -1;

for (y = 65 + 50; y < 465; y = y + jia2[j])//

{

j++;

judge = rand() % temp2;//

if (judge == 0)//

{

setcolor(BROWN);

setfillstyle(1, BROWN);

bar(x, y, x + 25, y + 25);

fyumi(x,y);

mian.bfzuobiao[ii][0] = 0;

mian.bfzuobiao[ii][1] = 0;//使其为0，表示健康

nian.bfzuobiao[i][0] = 0;

nian.bfzuobiao[i][1] = 0;//

i++;

ii++;

}

else

{

i++;//表示仍然患病，跳过

ii++;

}

}

}

}

int randrecover(int k)//恢复比例分配

{

int i;

int a[2][10] =

{

{1,1,2,2,2,2, 2,3,3,4},//小麦

{1,1,2,2,2,2, 2,3,3,4}//玉米的

};

i = rand() % 10;

return a[k][i];

}

**29.recover.h**

#ifndef \_RECOVER\_H\_

#define \_RECOVER\_H\_

//实现农作物的自我痊愈功能就是指，每次连续点击生长检测时，会有一部分植物自愈

void recover(void);//大农田植物自愈功能

int randrecover(int k);

#endif

**30.pssj.c**

#include "pssj.h"

#include "common.h"

int moveint2(int (\*hanshu)(int\*,int\* ), int\* a, int b[6])

{

int bbbb;

if (\*a == 1)

{

clrmous(MouseX, MouseY);

}

bbbb = hanshu(a, b);

if (\*a != -10)

{

newmouse(&MouseX, &MouseY, &press);

}

return bbbb;

}

int moveint3(int (\*hanshu)(int\*, int\*, unsigned char(\*)[49], unsigned char(\*)[49], unsigned char(\*)[49], int(\*)[2], int\*), int\* a, int b[6], unsigned char c[384][49], unsigned char d[192][49], unsigned char e[192][49],int f[386][2], int\* g)

{

int bbbbbb;

if (\*a == 1)

{

clrmous(MouseX, MouseY);

}

bbbbbb = hanshu(a, b, c, d, e,f,g);

if (\*a != -10)

{

newmouse(&MouseX, &MouseY, &press);

}

return bbbbbb;

}

**31.pssj.h**

#ifndef \_PSSJ\_H

#define \_PSSJ\_H

int moveint2(int (\*hanshu)(int\*, int\*), int\* a, int b[6]);

int moveint3(int (\*hanshu)(int\*, int\*, unsigned char(\*)[49], unsigned char(\*)[49], unsigned char(\*)[49], int(\*)[2], int\*), int\* a, int b[6], unsigned char c[384][49], unsigned char d[192][49], unsigned char e[192][49], int f[386][2], int\* g);

#endif

**32.zpjm.c**

#include "common.h"

int zpjiemian(int\* a9, int nongyao[6])

{

if (\*a9 == 1)

{

chejian();

\*a9 = 0;

}

if (\*a9 == 0 || (\*a9 > 1 && \*a9 < 10))

if (mouse\_press(20, 95, 110, 135) == 1)

{

if (\*a9 == 0)

\*a9 = 2;

else if (\*a9 > 1 && \*a9 < 10)

\*a9 = \*a9 + 1;

nongdu(\*a9);

delay(400);

}

if (\*a9 > 1 && \*a9 < 11)

if (mouse\_press(130, 95, 220, 135) == 1)

{

if (\*a9 == 2)

\*a9 = 0;

else if (\*a9 > 2 && \*a9 < 11)

\*a9 = \*a9 - 1;

nongdu(\*a9);

delay(400);

}

if (mouse\_press(220, 200, 370, 300) == 1)

{

if (nongyao[0] == 1)

{

nongyao[0] = 2;

zpdh(0);

\*a9 = 1;

return 9;

}

if (nongyao[0] == 0)

{

return 8;

}

}

else if (mouse\_press(420, 200, 570, 300) == 1)

{

if (nongyao[1] == 1)

{

nongyao[1] = 2;

zpdh(1);

\*a9 = 1;

return 9;

}

if (nongyao[1] == 0)

{

return 8;

}

}

else if (mouse\_press(20, 350, 170, 450) == 1)

{

if (nongyao[2] == 1)

{

nongyao[2] = 2;

zpdh(2);

\*a9 = 1;

return 9;

}

if (nongyao[2] == 0)

{

return 8;

}

}

else if (mouse\_press(220, 350, 370, 450) == 1)

{

if (nongyao[3] == 1)

{

nongyao[3] = 2;

zpdh(3);

\*a9 = 1;

return 9;

}

if (nongyao[3] == 0)

{

return 8;

}

}

else if (mouse\_press(420, 350, 570, 450) == 1)

{

if (nongyao[4] == 1)

{

nongyao[4] = 2;

zpdh(4);

\*a9 = 1;

return 9;

}

if (nongyao[4] == 0)

{

return 8;

}

}

else if (mouse\_press(600, 0, 640, 40) == 1)

{

return -1;//关闭BC。

}

else if (mouse\_press(0, 0, 80, 40) == 1)

{

\*a9 = 1;

return 7;//返回前一个界面

}

showbiankuang(600, 0, 640, 40);

showbiankuang(0, 0, 80, 40);

return 8;

}

void chejian(void)//绘画车间图像

{

cleardevice();

clrmous(MouseX, MouseY);

setcolor(RED);

rectangle(600, 0, 640, 40);

line(600, 0, 640, 40);

line(640, 0, 600, 40);

rectangle(0, 0, 80, 40);

puthz(5, 5, "返回", 32, 40, GREEN);

setfillstyle(1, LIGHTGRAY);

bar(20, 150, 620, 200);

bar(20, 300, 620, 350);

bar(170, 200, 220, 450);

bar(370, 200, 420, 450);

bar(570, 200, 620, 450);

bar(370, 20, 420, 150);

setcolor(DARKGRAY);

line(370, 20, 370, 150);

line(420, 20, 420, 150);

line(20, 150, 370, 150);

line(20, 200, 170, 200);

line(420, 150, 620, 150);

line(620, 150, 620, 450);

line(170, 200, 170, 300);

line(20, 300, 170, 300);

jizhan();

xiangzi(220, 200, 0, nongyao[0]);

xiangzi(420, 200, 1, nongyao[1]);

xiangzi(20, 350, 2, nongyao[2]);

xiangzi(220, 350, 3, nongyao[3]);

xiangzi(420, 350, 4, nongyao[4]);

drcar(20, 150,0);

nydrone(265, 40, 0);

setfillstyle(1, YELLOW);

bar(20, 55, 220, 135);

setfillstyle(1, RED);

bar(20, 95, 220, 135);

setfillstyle(1, WHITE);

bar(60, 95, 70, 135);

bar(45, 110, 85, 120);

bar(150, 108, 200, 122);

rectangle(20, 55, 220, 135);

line(20, 95, 220, 95);

setfillstyle(1, DARKGRAY);

bar(110, 95, 130, 135);

puthz(25, 60, "浓度", 24, 30, GREEN);

settextstyle(1, HORIZ\_DIR, 2);

setcolor(GREEN);

outtextxy(85, 56, ":");

setcolor(RED);

outtextxy(140, 58, "%");

nongdu(0);

}

void nongdu(int judge)//浓度显示

{

char a[3];

setcolor(RED);

setfillstyle(1, YELLOW);

bar(100, 58, 138, 94);

if(judge==0)

sprintf(a, "%d", 10);

if (judge == 2)

sprintf(a, "%d", 20);

if (judge == 3)

sprintf(a, "%d", 30);

if (judge == 4)

sprintf(a, "%d", 40);

if (judge == 5)

sprintf(a, "%d", 50);

if (judge == 6)

sprintf(a, "%d", 60);

if (judge == 7)

sprintf(a, "%d", 70);

if (judge == 8)

sprintf(a, "%d", 80);

if (judge == 9)

sprintf(a, "%d", 90);

if (judge == 10)

sprintf(a, "%d", 100);

outtextxy(100, 58, a);

}

void jizhan(void)//无人机基站图像

{

setcolor(DARKGRAY);

setfillstyle(1, DARKGRAY);

bar(290, 45, 370, 50);

bar(330, 55, 335, 80);

bar(320, 80, 345, 82);

setfillstyle(1, LIGHTGRAY);

bar(320, 40, 345, 55);

rectangle(320, 40, 345, 55);

}

void xiangzi(int x, int y ,int num,int judge2)//农药储存箱图像

{

setcolor(DARKGRAY);

setfillstyle(1,LIGHTGRAY);

bar(x, y, x + 150, y + 100);

bar(x+125, y-10, x+145 , y);

rectangle(x + 125, y - 10, x + 145, y);

rectangle(x, y, x + 150, y + 100);

line(x, y + 1, x + 150, y + 1);

line(x, y + 99, x + 150, y + 99);

line(x + 149, y, x + 149, y + 100);

setfillstyle(1, DARKGRAY);

bar(x + 125, y -10, x + 150, y - 4);

if (judge2 == 1)

{

if (num == 0)

{

puthz(x + 5, y + 35, "氯铃胺", 24, 30, DARKGRAY);

}

else if (num == 1)

{

puthz(x + 5, y + 35, "甲胺磷", 24, 30, DARKGRAY);

}

else if (num == 2)

{

puthz(x + 5, y + 35, "氯氰菊酯", 24, 30, DARKGRAY);

}

else if (num == 3)

{

puthz(x + 5, y + 35, "甲维盐", 24, 30, DARKGRAY);

}

else if (num == 4)

{

puthz(x + 5, y + 35, "菌酯", 24, 30, DARKGRAY);

}

}

else

{

line(x, y, x + 150, y + 100);

line(x+150, y, x , y + 100);

line(x+1, y, x + 151 ,y + 100);

line(x + 151, y, x+1, y + 100);

}

}

void drcar(int x, int y,int judge1)//小车

{

setcolor(DARKGRAY);

setfillstyle(1, LIGHTGRAY);

bar(x, y + 20, x + 50, y + 40);

rectangle(x, y + 20, x + 50, y + 40);

setfillstyle(1, DARKGRAY);

bar(x + 35, y, x + 45, y + 20);

bar(x, y, x + 40, y + 6);

bar(x + 5, y + 35, x + 20, y + 50);

bar(x + 30, y + 35, x + 45, y + 50);

setfillstyle(1, WHITE);

bar(x + 10, y + 40, x + 15, y + 45);

bar(x + 35, y + 40, x + 40, y + 45);

if (judge1 == 0)

{

setfillstyle(1, RED);

bar(x + 35, y + 25, x + 40, y + 30);

rectangle(x + 35, y + 25, x + 40, y + 30);

}

else if (judge1 == 1)

{

setfillstyle(1, GREEN);

bar(x + 35, y + 25, x + 40, y + 30);

rectangle(x + 35, y + 25, x + 40, y + 30);

}

}

void nydrone(int x , int y , int judge4)//车间内无人机图像

{

setcolor(DARKGRAY);

setfillstyle(1, LIGHTGRAY);

bar(x, y - 5, x + 25, y + 30);

rectangle (x, y - 5, x + 25, y + 30);

line(x + 5, y - 5, x + 5, y + 30);

line(x + 6, y - 5, x + 6, y + 30);

line(x + 20, y - 5, x + 20, y + 30);

line(x + 19, y - 5, x + 19, y + 30);

setfillstyle(1, DARKGRAY);

bar(x - 5, y - 5, x, y);

bar(x - 10, y - 10, x - 5, y - 5);

bar(x + 25, y - 5, x + 30, y);

bar(x + 30, y - 10, x + 35, y-5);

bar(x - 5, y +25, x, y+30);

bar(x - 10, y + 30, x-5, y + 35);

bar(x + 25, y + 25, x + 30, y + 30);

bar(x + 30, y + 30, x + 35, y + 35);

bar(x - 15, y - 20, x - 10, y + 45);

bar(x + 35 , y - 20, x + 40, y + 45);

if (judge4 == 0)

{

setfillstyle(1, RED);

bar(x + 10,y+10 , x + 15, y+15);

}

else if (judge4 == 1)

{

setfillstyle(1, GREEN);

bar(x + 10, y + 10, x + 15, y + 15);

}

}

void fushitu(int x, int y)//小农田农药无人机图

{

setcolor(DARKGRAY);

setfillstyle(1, BLUE);

bar(x+6, y, x + 14, y + 10);

rectangle(x+6 , y, x + 14, y + 10);

setfillstyle(1, DARKGRAY);

bar(x , y, x + 20, y + 2);

bar(x , y, x +2, y + 5);

bar(x + 18, y, x + 20, y + 5);

setfillstyle(1, LIGHTGRAY);

bar(x+5, y, x + 15, y + 5);

rectangle(x+5, y, x + 15, y + 5);

}

void suofangtu(int x, int y, int judge4)//大农田农药无人机缩放图

{

setcolor(DARKGRAY);

setfillstyle(1, LIGHTGRAY);

bar(x, y - 2, x + 10, y + 12);

rectangle(x, y - 2, x + 10, y + 12);

line(x + 2, y - 2, x + 2, y + 12);

line(x + 3, y - 2, x + 3, y + 12);

line(x + 8, y - 2, x + 8, y + 12);

line(x + 7, y - 2, x + 7, y + 12);

setfillstyle(1, DARKGRAY);

bar(x - 2, y - 2, x, y);

bar(x - 4, y - 4, x - 2, y - 2);

bar(x + 10, y - 2, x + 12, y);

bar(x + 12, y - 4, x + 14, y - 2);

bar(x - 2, y + 10, x, y + 12);

bar(x - 4, y + 12, x - 2, y + 14);

bar(x + 10, y + 10, x + 12, y + 12);

bar(x + 12, y + 12, x + 14, y + 14);

bar(x - 6, y - 8, x - 4, y + 18);

bar(x + 14, y - 8, x + 16, y + 18);

if (judge4 == 0)

{

setfillstyle(1, RED);

bar(x + 4, y + 4, x + 6, y + 6);

}

else if (judge4 == 1)

{

setfillstyle(1, GREEN);

bar(x + 4, y + 4, x + 6, y + 6);

}

}

void zpdh(int judge3)//农药装配动画

{

void\* buffer2 = malloc(imagesize(0, 0, 50, 50));

int\* x2 = malloc(sizeof(int));

int\* y2 = malloc(sizeof(int));

int i;

\*x2 = 20;

\*y2 = 150;

setcolor(DARKGRAY);

setfillstyle(1, LIGHTGRAY);

bar(20, 150, 70, 200);

line(20, 150, 370, 150);

line(20, 200, 170, 200);

moveps1(0, x2, y2, buffer2);

if (judge3 == 0)

{

moveps2(0, x2, y2, buffer2);

moveps7(0, x2, y2, buffer2);

moveps8(1, x2, y2, buffer2);

moveps9(1, x2, y2, buffer2);

moveps0(0, x2, y2, buffer2);

}

else if (judge3 == 1)

{

moveps2(0, x2, y2, buffer2);

moveps2(0, x2, y2, buffer2);

moveps7(0, x2, y2, buffer2);

moveps8(1, x2, y2, buffer2);

moveps6(1, x2, y2, buffer2);

moveps9(1, x2, y2, buffer2);

moveps0(0, x2, y2, buffer2);

}

else if (judge3 == 2)

{

moveps4(0, x2, y2, buffer2);

moveps7(0, x2, y2, buffer2);

moveps8(1, x2, y2, buffer2);

moveps3(1, x2, y2, buffer2);

moveps2(1, x2, y2, buffer2);

moveps9(1, x2, y2, buffer2);

moveps0(0, x2, y2, buffer2);

}

else if (judge3 == 3)

{

moveps4(0, x2, y2, buffer2);

moveps2(0, x2, y2, buffer2);

moveps7(0, x2, y2, buffer2);

moveps8(1, x2, y2, buffer2);

moveps3(1, x2, y2, buffer2);

moveps9(1, x2, y2, buffer2);

moveps0(0, x2, y2, buffer2);

}

else if (judge3 == 4)

{

moveps4(0, x2, y2, buffer2);

moveps2(0, x2, y2, buffer2);

moveps2(0, x2, y2, buffer2);

moveps7(0, x2, y2, buffer2);

moveps8(1, x2, y2, buffer2);

moveps3(1, x2, y2, buffer2);

moveps6(1, x2, y2, buffer2);

moveps9(1, x2, y2, buffer2);

moveps0(0, x2, y2, buffer2);

}

drcar(20, 150, 0);

free(buffer2);

free(x2);

free(y2);

delay(100);

}

void moveps0(int judge1, int\* x, int\* y, int\* buffer2)//向左特殊运动

{

for (; \*x >20; \*x = \*x - 2)

{

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, judge1);

delay(8);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

}

void moveps1(int judge1, int\* x, int\* y, int\* buffer2)//向右特殊运动

{

for (; \*x <170; \*x = \*x + 2)

{

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y,judge1);

delay(8);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

}

void moveps2(int judge1, int\* x, int\* y, int\* buffer2)//向右进一区

{

int xx;

xx = \*x;

for (; \*x-xx < 200; \*x = \*x + 2)

{

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, judge1);

delay(8);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

}

void moveps3(int judge1, int\* x, int\* y, int\* buffer2)//向上进一区

{

int yy;

yy = \*y;

for (; yy - \*y < 150; \*y = \*y - 2)

{

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, judge1);

delay(8);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

}

void moveps4(int judge1, int\* x, int\* y, int\* buffer2)//向下进一区

{

int yy;

yy = \*y;

for (; \*y - yy < 150; \*y = \*y + 2)

{

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, judge1);

delay(8);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

}

void moveps6(int judge1, int\* x, int\* y, int\* buffer2)//向左进一区

{

int xx;

xx = \*x;

for (; xx - \*x < 200; \*x = \*x - 2)

{

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, judge1);

delay(8);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

}

void moveps7(int judge1, int\* x, int\* y, int\* buffer2)//向下取药

{

int yy;

yy = \*y;

for (; \*y - yy < 40; \*y = \*y + 2)

{

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, judge1);

delay(20);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, judge1);

delay(200);

drcar(\*x, \*y, 1);

delay(200);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

void moveps8(int judge1, int\* x, int\* y, int\* buffer2)//向上回位

{

int yy;

yy = \*y;

for (; yy-\*y < 40; \*y = \*y - 2)

{

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, judge1);

delay(8);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

}

void moveps9(int judge1, int\* x, int\* y, int\* buffer2)//装药，回位

{

int yy;

yy = \*y;

for (; yy - \*y < 50; \*y = \*y - 2)

{

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, judge1);

delay(16);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

for (; yy - \*y < 105; \*y = \*y - 2)

{

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, judge1);

delay(20);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, judge1);

delay(200);

drcar(\*x, \*y, 0);

nydrone(265, 40, 1);

delay(200);

putimage(\*x, \*y, buffer2, COPY\_PUT);

for (; yy - \*y >50; \*y = \*y + 2)

{

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, 0);

delay(20);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

for (; yy - \*y >0; \*y = \*y + 2)

{

getimage(\*x, \*y, \*x + 50, \*y + 50, buffer2);

drcar(\*x, \*y, 0);

delay(16);

putimage(\*x, \*y, buffer2, COPY\_PUT);

}

}

**33.zpjm.h**

#ifndef \_ZPJM\_H\_

#define \_ZPJM\_H\_

int zpjiemian(int\* a9, int nongyao[6]);

void chejian(void);//绘画车间图像

void nongdu(int judge);//浓度显示

void jizhan(void);//无人机基站图像

void xiangzi(int x, int y , int num,int judge2);//农药储存箱图像

void drcar(int x, int y,int judge1);//小车

void nydrone(int x, int y ,int judge4);//车间内无人机图像

void fushitu(int x, int y);//小农田农药无人机图

void suofangtu(int x, int y , int judge4);//大农田农药无人机缩放图

void zpdh(int judge3);//农药装配动画

void moveps0(int judge1, int\* x, int\* y, int\* buffer2);//向左特殊运动

void moveps1(int judge1, int\* x, int\* y, int\* buffer2);//向右特殊运动

void moveps2(int judge1, int\* x, int\* y, int\* buffer2);//向右进一区

void moveps3(int judge1, int\* x, int\* y, int\* buffer2);//向上进一区

void moveps4(int judge1, int\* x, int\* y, int\* buffer2);//向下进一区

void moveps6(int judge1, int\* x, int\* y, int\* buffer2);//向左进一区

void moveps7(int judge1, int\* x, int\* y, int\* buffer2);//向下取药

void moveps8(int judge1, int\* x, int\* y, int\* buffer2);//向上回位

void moveps9(int judge1, int\* x, int\* y, int\* buffer2);//向上装药

#endif

**34.pensa.c**

#include "common.h"

int zhuzhan(int\* a8,int nongyao[6])

{

int i;

if (\*a8 == 1)

{

peizhi();

\*a8 = 0;

}

showbiankuang(600, 0, 640, 40);

showbiankuang(0, 0, 80, 40);

showbiankuang(370, 0, 580, 60);

if (mouse\_press(600, 0, 640, 40) == 1)

{

return -1;//关闭BC。

}

else if (mouse\_press(0, 0, 80, 40) == 1)

{

\*a8 = 1;

if (nongyao[5] == 0)

return 2;

else if (nongyao[5] == 1)

return 3;

}

if (nongyao[0] == 0)

{

if (mouse\_press(300, 145, 440, 225) == 1 || mouse\_press(160, 225, 300, 305) == 1)//氟铃脲

{

\*a8 = 1;

nongyao[0] = 1;

}

}

if (nongyao[2] == 0)

{

if (mouse\_press(300, 225, 440, 305) == 1)//氯氰菊酯

{

\*a8 = 1;

nongyao[2] = 1;

}

}

if (nongyao[4] == 0)

{

if (mouse\_press(160, 305, 300, 385) == 1)//菌酯

{

\*a8 = 1;

nongyao[4] = 1;

}

}

if (nongyao[5] == 1)

{

if (nongyao[1] == 0)

{

if (mouse\_press(440, 145, 580, 225) == 1)//甲胺磷

{

\*a8 = 1;

nongyao[1] = 1;

}

}

if (nongyao[3] == 0)

{

if (mouse\_press(440, 385, 580, 465) == 1)//甲维盐

{

\*a8 = 1;

nongyao[3] = 1;

}

}

}

if (nongyao[0] == 1)

{

if (mouse\_press(410, 145, 440, 175) == 1 || mouse\_press(270, 225, 300, 255) == 1)//氟铃脲

{

\*a8 = 1;

nongyao[0] = 0;

delay(400);

}

}

if (nongyao[2] == 1)

{

if (mouse\_press(410, 225, 440, 255) == 1)//氯氰菊酯

{

\*a8 = 1;

nongyao[2] = 0;

delay(400);

}

}

if (nongyao[4] == 1)

{

if (mouse\_press(270, 305, 300, 335) == 1)//菌酯

{

\*a8 = 1;

nongyao[4] = 0;

delay(400);

}

}

if (nongyao[5] == 1)

{

if (nongyao[1] == 1)

{

if (mouse\_press(550, 145, 580, 175) == 1)//甲胺磷

{

\*a8 = 1;

nongyao[1] = 0;

delay(400);

}

}

if (nongyao[3] == 1)

{

if (mouse\_press(550, 385, 580, 415) == 1)//甲维盐

{

\*a8 = 1;

nongyao[3] = 0;

delay(400);

}

}

}

if (mouse\_press(370, 0, 580, 60) == 1)//装配农药

{

\*a8 = 1;

return 8;

}

return 7;

}

void peizhi(void)//配置界面绘画

{

int i;

int x,y;

cleardevice();

clrmous(MouseX, MouseY);

setcolor(RED);

rectangle(600, 0, 640, 40);

line(600, 0, 640, 40);

line(640, 0, 600, 40);//关闭按钮

rectangle(0, 0, 80, 40);

puthz(5, 5, "返回", 32, 40, GREEN);

setfillstyle(1, BLUE);

bar(20, 65, 580, 465);

bar(370, 0, 580, 60);

puthz(390, 10, "装配农药", 32, 40, WHITE);

puthz(100, 10, "选择配置农药", 32, 40, BLUE);

setcolor(LIGHTGRAY);

rectangle(370, 0, 580, 60);

line(20, 65, 160, 145);

for (i = 0; i < 6; i++)

{

x = 20 + 140 \* i;

line(x, 65, x, 465);

line(x+1, 65, x+1, 465);

y = 65 + 80 \* i;

line(20, y, 580, y);

line(20, y+1, 580, y+1);

}

puthz(80, 70, "植株种类", 16, 20, WHITE);

puthz(30, 120, "病害种类", 16, 20, WHITE);

puthz(25, 170, "棉铃虫", 24, 30, WHITE);

puthz(25, 250, "黏虫", 24, 30, WHITE);

puthz(25, 330, "白粉病", 24, 30, WHITE);

puthz(25, 410, "锈病", 24, 30, WHITE);

puthz(165, 90, "小麦", 24, 30, WHITE);

puthz(165, 255, "氟铃脲", 24, 30, WHITE);

puthz(165, 335, "菌酯", 24, 30, WHITE);

puthz(305, 90, "玉米", 24, 30, WHITE);

puthz(305, 175, "氟铃脲", 24, 30, WHITE);

puthz(305, 255, "氯氰菊酯", 24, 30, WHITE);

puthz(445, 90, "棉花", 24, 30, WHITE);

puthz(445, 175, "甲胺磷", 24, 30, WHITE);

puthz(445, 415, "甲维盐", 24, 30, WHITE);

drline(160, 145);

drline(160, 385);

drline(300, 305);

drline(300, 385);

drline(440, 225);

drline(440, 305);

if (nongyao[0] == 1)

{

setfillstyle(1, WHITE);

bar(300, 145, 440, 225);

bar(160, 225, 300, 305);

puthz(165, 255, "氟铃脲", 24, 30, BLUE);

puthz(305, 175, "氟铃脲", 24, 30, BLUE);

drhook(300, 145);

drhook(160, 225);

setfillstyle(1, RED);

bar(410, 145, 440, 175);

bar(270, 225, 300, 255);

cross(410, 145);

cross(270, 225);

rectangle(300, 145, 440, 225);

rectangle(160, 225, 300, 305);

}

if (nongyao[1] == 1)

{

setfillstyle(1, WHITE);

bar(440, 145, 580, 225);

puthz(445, 175, "甲胺磷", 24, 30, BLUE);

drhook(440, 145);

setfillstyle(1, RED);

bar(550, 145, 580, 175);

cross(550, 145);

rectangle(440, 145, 580, 225);

}

if (nongyao[2] == 1)

{

setfillstyle(1, WHITE);

bar(300, 225, 440, 305);

puthz(305, 255, "氯氰菊酯", 24, 30, BLUE);

drhook(300, 225);

setfillstyle(1, RED);

bar(410, 225, 440, 255);

cross(410, 225);

rectangle(300, 225, 440, 305);

}

if (nongyao[3] == 1)

{

setfillstyle(1, WHITE);

bar(440, 385, 580, 465);

puthz(445, 415, "甲维盐", 24, 30, BLUE);

drhook(440, 385);

setfillstyle(1, RED);

bar(550, 385, 580, 415);

cross(550, 385);

rectangle(440, 385, 580, 465);

}

if (nongyao[4] == 1)

{

setfillstyle(1, WHITE);

bar(160, 305, 300, 385);

puthz(165, 335, "菌酯", 24, 30, BLUE);

drhook(160, 305);

setfillstyle(1, RED);

bar(270, 305, 300, 335);

cross(270, 305);

rectangle(160, 305, 300, 385);

}

if (nongyao[5] == 0)

{

setfillstyle(1, BLUE);

bar(442, 147, 578, 223);

bar(442, 387, 578, 463);

drline(440, 145);

drline(440, 385);

}

}

void drhook(int x, int y)

{

setcolor(RED);

line(x + 110, y + 70, x + 125, y + 80);

line(x + 111, y + 70, x + 126, y + 80);

line(x + 124, y + 80, x + 139, y + 50);

line(x + 125, y + 80, x + 140, y + 50);

}

void cross(int x, int y)

{

setcolor(WHITE);

line(x, y, x + 30, y + 30);

line(x, y + 1, x + 29, y + 30);

line(x + 1, y, x + 30, y + 29);

line(x, y + 30, x + 30, y);

line(x, y + 29, x + 29, y);

line(x + 1, y + 30, x + 30, y + 1);

}

void drline(int x, int y)

{

setcolor(LIGHTGRAY);

line(x, y, x + 140, y + 80);

line(x+1, y, x + 141, y + 80);

}

**35.pensa.h**

#ifndef \_PENSA\_H\_

#define \_PENSA\_H\_

int zhuzhan(int\* a8,int nongyao[6]);

void peizhi(void);

void drhook(int x, int y);

void cross(int x, int y);

void drline(int x, int y);

#endif

**36.pensadh.c**

int psdonghua(int\* a10, int nongyao[6], unsigned char diannian[384][49], unsigned char dianbai[192][49], unsigned char dianmian[192][49], int rount[386][2],int\* amtjudge)

{

FILE\* filePointer1;

FILE\* filePointer2;

FILE\* filePointer3;

FILE\* filePointer4;

FILE\* filePointer5;

FILE\* filePointer6;

int i,j;

int grandjudge = 0, jg0=0, jg1 = 0, jg2 = 0, jg3 = 0, jg4 = 0;//granjudge:确定农药使用，jg用于贮存相应得病植株数量

int ny0[384][2] = {0};//ny用于贮存坐标

int ny1[192][2] = { 0 };

int ny2[192][2] = { 0 };

int ny3[192][2] = { 0 };

int ny4[192][2] = { 0 };

if (\*a10==1)

{

rount[0][0] = 20;

rount[0][1] = 465;

cleardevice();

clrmous(MouseX, MouseY);

setcolor(RED);

setfillstyle(1, GREEN);

rectangle(600, 0, 640, 40);

line(600, 0, 640, 40);

line(640, 0, 600, 40);

bar(260, 5, 400, 50);

puthz(270, 10, "喷洒农药", 32, 30, WHITE);

bar(100, 5, 240, 50);

puthz(110, 10, "路径规划", 32, 30, WHITE);

rectangle(0, 0, 80, 40);

puthz(5, 5, "返回", 32, 40, GREEN);

if (nongyao[5] == 0)//大农田，已经排除了棉花用的农药，关注0，2，4

{

farmfield();

for (i = 0; i < 192; i++)//小麦，黏虫

{

if (nian.bfzuobiao[i][0] != 0 && nian.bfzuobiao[i][1] != 0)

{

nianchong(nian.bfzuobiao[i][0], nian.bfzuobiao[i][1], diannian[i]);

ny0[jg0][0] = nian.bfzuobiao[i][0];

ny0[jg0][1] = nian.bfzuobiao[i][1];

jg0++;

}

}

for (i = 192; i < 384; i++)//玉米，黏虫

{

if (nian.bfzuobiao[i][0] != 0 && nian.bfzuobiao[i][1] != 0)

{

nianchong(nian.bfzuobiao[i][0], nian.bfzuobiao[i][1], diannian[i]);

ny2[jg2][0] = nian.bfzuobiao[i][0];

ny2[jg2][1] = nian.bfzuobiao[i][1];

jg2++;

}

}

for (i = 0; i < 192; i++)//玉米，棉铃虫

{

if (mian.bfzuobiao[i][0] != 0 && mian.bfzuobiao[i][1] != 0)

{

mianling(mian.bfzuobiao[i][0], mian.bfzuobiao[i][1], dianmian[i]);

ny0[jg0][0] = mian.bfzuobiao[i][0];

ny0[jg0][1] = mian.bfzuobiao[i][1];

jg0++;

}

}

for (i = 0; i < 192; i++)//小麦，白粉病

{

if (bai.bfzuobiao[i][0] != 0 && bai.bfzuobiao[i][1] != 0)

{

baifen(bai.bfzuobiao[i][0], bai.bfzuobiao[i][1], dianbai[i]);

ny4[jg4][0] = bai.bfzuobiao[i][0];

ny4[jg4][1] = bai.bfzuobiao[i][1];

jg4++;

}

}

}

else if (nongyao[5] == 1)//小农田

{

bmp\_convert("5.bmp", "5.dbm");

open\_display();

show\_dbm(0, 100, "5.dbm", 0);

tian();

bmp\_convert("stone.bmp", "stone.dbm");

open\_display();

drawPlants();

drawdxg(250, 240);

show\_dbm(194, 447, "stone.dbm", 0);

drawScarecrow(570, 61);

drawScarecrow(500, 325);

binghai();

filePointer1 = fopen("data//xtxmnian.dat", "r");

for (i = 0; i < 192; i++)

{

if (fscanf(filePointer1, "%d %d", &ny0[jg0][0], &ny0[jg0][1]) != -1)

jg0++;

else

break;

}

fclose(filePointer1);

filePointer2 = fopen("data//xtymmian.dat", "r");

for (i = 0; i < 192; i++)

{

if (fscanf(filePointer2, "%d %d", &ny0[jg0][0], &ny0[jg0][1]) != -1)

jg0++;

else

break;

}

fclose(filePointer2);

filePointer3 = fopen("data//xtmhmian.dat", "r");

for (i = 0; i < 192; i++)

{

if (fscanf(filePointer3, "%d %d", &ny1[jg1][0], &ny1[jg1][1]) != -1)

jg1++;

else

break;

}

fclose(filePointer3);

filePointer4 = fopen("data//xtymnian.dat", "r");

for (i = 0; i < 192; i++)

{

if (fscanf(filePointer4, "%d %d", &ny2[jg2][0], &ny2[jg2][1]) != -1)

jg2++;

else

break;

}

fclose(filePointer4);

filePointer5 = fopen("data//xtmhxiu.dat", "r");

for (i = 0; i < 192; i++)

{

if (fscanf(filePointer5, "%d %d", &ny3[jg3][0], &ny3[jg3][1]) != -1)

jg3++;

else

break;

}

fclose(filePointer5);

filePointer6 = fopen("data//xtxmbai.dat", "r");

for (i = 0; i < 192; i++)

{

if (fscanf(filePointer6, "%d %d", &ny4[jg4][0], &ny4[jg4][1]) != -1)

jg4++;

else

break;

}

fclose(filePointer6);

}

if (nongyao[0] == 2)

{

if (jg0 == 0)

{

nongyao[0] = 0;

huigui();

return 7;

}

if (jg0 > 0)

{

grandjudge = 0;

\*amtjudge = jg0;

}

}

if (nongyao[1] == 2)

{

if (jg1 == 0)

{

nongyao[1] = 0;

huigui();

return 7;

}

if (jg1 > 0)

{

grandjudge = 1;

\*amtjudge = jg1;

}

}

if (nongyao[2] == 2)

{

if (jg2 == 0)

{

nongyao[2] = 0;

huigui();

return 7;

}

if (jg2 > 0)

{

grandjudge = 2;

\*amtjudge = jg2;

}

}

if (nongyao[3] == 2)

{

if (jg3 == 0)

{

nongyao[3] = 0;

huigui();

return 7;

}

if (jg3 > 0)

{

grandjudge = 3;

\*amtjudge = jg3;

}

}

if (nongyao[4] == 2)

{

if (jg4 == 0)

{

nongyao[4] = 0;

huigui();

return 7;

}

if (jg4 > 0)

{

grandjudge = 4;

\*amtjudge = jg4;

}

}

ljgh(grandjudge, amtjudge, ny0, ny1, ny2, ny3, ny4, rount);

\*a10 = 0;

}

if (\*a10 == 2)

{

therount(rount, amtjudge);

\*a10 = 3;

}

if (mouse\_press(600, 0, 640, 40) == 1)

{

return -1;//关闭BC。

}

else if (mouse\_press(0, 0, 80, 40) == 1)

{

\*a10 = 1;

for (i = 0; i < 5; i++)

{

if (nongyao[i] > 1)

{

nongyao[i] = 0;

}

}

showbiankuang(600, 0, 640, 40);

showbiankuang(0, 0, 80, 40);

return 8;//返回农药选择界面

}

if (\*a10 == 0)

{

if (mouse\_press(100, 5, 240, 50) == 1)

{

\*a10 = 2;

}

showbiankuang(100, 5, 240, 50);

}

if (\*a10 == 3)

{

if (mouse\_press(260, 5, 400, 50) == 1)

{

if (nongyao[5] == 0)

{

anima1(rount, amtjudge);

\*a10 = 4;

}

else if (nongyao[5] == 1)

{

anima2(rount, amtjudge);

\*a10 = 5;

}

}

showbiankuang(260, 5, 400, 50);

}

if (\*a10 == 4)

{

for (i = 0; i < 5; i++)

if (nongyao[i] == 2)

break;

if (i == 0)

for (j = 0; j < 192; j++)

{

nian.bfzuobiao[j][0] = 0;

nian.bfzuobiao[j][1] = 0;

mian.bfzuobiao[j][0] = 0;

mian.bfzuobiao[j][1] = 0;

}

if (i == 2)

{

for (j =192; j < 384; j++)

{

nian.bfzuobiao[j][0] = 0;

nian.bfzuobiao[j][1] = 0;

}

}

if (i == 4)

{

for (j = 0; j < 192; j++)

{

bai.bfzuobiao[j][0] = 0;

bai.bfzuobiao[j][1] = 0;

}

}

nongyao[i] = 0;

\*a10 = 1;

huigui2(i);

return 2;

}

if(\*a10==5)

{

for (i = 0; i < 5; i++)

if (nongyao[i] == 2)

break;

if (i == 0)

{

filePointer1 = fopen("data//xtxmnian.dat", "w");

fclose(filePointer1);

filePointer1 = fopen("data//xmncdian.dat", "w");

fclose(filePointer1);

filePointer2 = fopen("data//xtymmian.dat", "w");

fclose(filePointer2);

filePointer2 = fopen("data//ymmldian.dat", "w");

fclose(filePointer2);

}

if (i == 1)

{

filePointer3 = fopen("data//xtmhmian.dat", "w");

fclose(filePointer3);

filePointer3 = fopen("data//mhmldian.dat", "w");

fclose(filePointer3);

}

if (i == 2)

{

filePointer4 = fopen("data//xtymnian.dat", "w");

fclose(filePointer4);

filePointer4 = fopen("data//ymncdian.dat", "w");

fclose(filePointer4);

}

if (i == 3)

{

filePointer5 = fopen("data//xtmhxiu.dat", "w");

fclose(filePointer5);

filePointer5 = fopen("data//mhxbdian.dat", "w");

fclose(filePointer5);

}

if (i == 4)

{

filePointer6 = fopen("data//xtxmbai.dat", "w");

fclose(filePointer6);

filePointer6 = fopen("data//xmbfdian.dat", "w");

fclose(filePointer6);

}

nongyao[i] = 0;

\*a10 = 1;

huigui2(i);

return 3;

}

return 9;

}

void ljgh(int grandjudge, int \*judge, int ny0[384][2], int ny1[192][2], int ny2[192][2], int ny3[192][2], int ny4[192][2],int rount[386][2])//路径规划

{

int i = 0, j = 0, k = \*judge-1,K = 1 + \*judge;

int len1 = 0, len2 = 0;

int dist1 = 0, dist2 = 0;

int temp0 = 0, temp1 = 0;

if (grandjudge == 0)

{

for (i = 0; i < \*judge; i++)

{

j = i + 1;

rount[j][0] = ny0[i][0];

rount[j][1] = ny0[i][1];

}

}

if (grandjudge == 1)

{

for (i = 0; i < \*judge; i++)

{

j = i + 1;

rount[j][0] = ny1[i][0];

rount[j][1] = ny1[i][1];

}

}

if (grandjudge == 2)

{

for (i = 0; i < \*judge; i++)

{

j = i + 1;

rount[j][0] = ny2[i][0];

rount[j][1] = ny2[i][1];

}

}

if (grandjudge == 3)

{

for (i = 0; i < \*judge; i++)

{

j = i + 1;

rount[j][0] = ny3[i][0];

rount[j][1] = ny3[i][1];

}

}

if (grandjudge == 4)

{

j = 1;

for (i = 0; i < \*judge; i++)

{

j = i + 1;

rount[j][0] = ny4[i][0];

rount[j][1] = ny4[i][1];

}

}

rount[K][0] = rount[0][0];

rount[K][1] = rount[0][1];

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

{

len1 = rount[i + 1][0] - rount[i][0];

len2 = rount[i + 1][1] - rount[i][1];

len1 = len1 / 25;

len2 = len2 / 25;

dist1 = len1 \* len1 + len2 \* len2;

for (j = i + 2; j < K; j++)

{

len1 = rount[j][0] - rount[i][0];

len2 = rount[j][1] - rount[i][1];

len1 = len1 / 25;

len2 = len2 / 25;

dist2 = len1 \* len1 + len2 \* len2;

if (dist2 < dist1)

{

dist1 = dist2;

temp0 = rount[j][0];

temp1 = rount[j][1];

rount[j][0] = rount[i + 1][0];

rount[j][1] = rount[i + 1][1];

rount[i+1][0] = temp0;

rount[i+1][1] = temp1;

}

}

}

}

void therount(int rount[386][2], int \* amtjudge)//路径画面

{

int i = 0;

for (i=0;i<\* amtjudge+1;i++)

{

setcolor(RED);

line(rount[i][0], rount[i][1], rount[i + 1][0], rount[i + 1][1]);

}

line(rount[\*amtjudge][0], rount[\*amtjudge][1], rount[0][0], rount[0][1]);

}

void huigui(void)//回归

{

cleardevice();

clrmous(MouseX, MouseY);

puthz(150, 200, "未发现相关病状", 32, 40, RED);

setcolor(BLUE);

rectangle(110, 300, 530, 380);

puthz(125, 322, "回归中", 32, 50, RED);

setfillstyle(1, BLUE);

bar(265, 320, 305, 360);

delay(400);

bar(315, 320, 355, 360);

delay(400);

bar(365, 320, 405, 360);

delay(400);

bar(415, 320, 455, 360);

delay(400);

bar(465, 320, 505, 360);

delay(1000);

}

void huigui2(int i)//喷洒农药后回归

{

cleardevice();

clrmous(MouseX, MouseY);

puthz(100, 150, "已清除下列植物的下列病状", 32, 40, BLUE);

if (i == 0)

{

puthz(100, 250, "小麦黏虫", 32, 40, RED);

puthz(300, 250, "玉米棉铃虫", 32, 40, RED);

}

if (i == 1)

puthz(100, 250, "棉花棉铃虫", 32, 40, RED);

if (i == 2)

puthz(100, 250, "玉米黏虫", 32, 40, RED);

if (i == 3)

puthz(100, 250, "棉花锈病", 32, 40, RED);

if (i == 4)

puthz(100, 250, "小麦白粉病", 32, 40, RED);

setcolor(BLUE);

rectangle(90, 350, 550, 430);

puthz(100, 372, "回归农田", 32, 50, RED);

setfillstyle(1, BLUE);

bar(285, 370, 325, 410);

delay(400);

bar(335, 370, 375, 410);

delay(400);

bar(385, 370, 425, 410);

delay(400);

bar(435, 370, 475, 410);

delay(400);

bar(485, 370, 525, 410);

delay(1000);

}

void nyphoto(int x, int y)//农药图像

{

setfillstyle(1, LIGHTBLUE);

bar(x, y+2, x + 3, y + 5);

bar(x + 10, y + 7, x + 13, y + 10);

bar(x + 20, y+3, x + 23, y + 6);

bar(x, y + 10, x + 3, y + 13);

bar(x + 20, y + 5, x + 23, y + 8);

bar(x + 20, y+15, x + 23, y + 18);

bar(x, y + 19, x + 3, y + 22);

bar(x + 12, y + 22, x + 15, y + 25);

}

void anima1(int rount[386][2], int\* amtjudge)//大农田路径动画

{

void\* buffer1 = malloc(imagesize(0, 0, 22, 26));

int\* x1 = malloc(sizeof(int));

int\* y1 = malloc(sizeof(int));

int i,k;

\*x1 = rount[0][0];

\*y1 = rount[0][1];

for (i = 0; i < \*amtjudge+1; i++)

{

setcolor(CYAN);

line(rount[i][0], rount[i][1], rount[i + 1][0], rount[i + 1][1]);

do

{

k = linemove1(x1, y1, buffer1);

} while (k == 1);

getimage(\*x1 - 6, \*y1 - 8, \*x1 + 16, \*y1 + 18, buffer1);

suofangtu(\*x1, \*y1, 1);

delay(100);

putimage(\*x1 - 6, \*y1 - 8, buffer1, COPY\_PUT);

if (i < \*amtjudge)

nyphoto(rount[i + 1][0], rount[i + 1][1]);

}

getimage(\*x1 - 6, \*y1 - 8, \*x1 + 16, \*y1 + 18, buffer1);

suofangtu(\*x1, \*y1, 1);

delay(500);

putimage(\*x1 - 6, \*y1 - 8, buffer1, COPY\_PUT);

free(buffer1);

free(x1);

free(y1);

}

void anima2(int rount[386][2], int\* amtjudge)//小农田路径动画

{

void\* buffer1 = malloc(imagesize(0, 0, 20, 10));

int\* x1 = malloc(sizeof(int));

int\* y1 = malloc(sizeof(int));

int i, k ;

for (i = 0; i < \*amtjudge + 1; i++)

{

\*x1 = rount[i][0];

\*y1 = rount[i][1];

setcolor(CYAN);

rectangle(185, 443, 233, 480);

rectangle(184, 444, 234, 479);

rectangle(245, 225, 287, 323);

rectangle(244, 226, 288, 322);

rectangle(553, 43, 582, 136);

rectangle(552, 44, 583, 135);

rectangle(483, 307, 512, 400);

rectangle(482, 308, 513, 399);

line(rount[i][0], rount[i][1], rount[i + 1][0], rount[i + 1][1]);

setcolor(BLUE);

rectangle(185, 444, 233, 479);

rectangle(245, 226, 287, 322);

rectangle(553, 44, 582, 135);

rectangle(483, 308, 512, 399);

if (rount[i][0] < rount[i + 1][0]&& rount[i][1] >= rount[i + 1][1])//第一象限

{

do

{

k = linemove21(6,x1, y1, buffer1);

} while (k == 1);

}

else if (rount[i][0] >= rount[i + 1][0] && rount[i][1] > rount[i + 1][1])//第二象限

{

do

{

k = linemove22(6, x1, y1, buffer1);

} while (k == 1);

}

else if (rount[i][0] > rount[i + 1][0] && rount[i][1] <= rount[i + 1][1])//第三象限

{

do

{

k = linemove22(0, x1, y1, buffer1);

} while (k == 1);

}

else if (rount[i][0] <= rount[i + 1][0] && rount[i][1] < rount[i + 1][1])//第三象限

{

do

{

k = linemove21(0, x1, y1, buffer1);

} while (k == 1);

}

getimage(\*x1, \*y1, \*x1 + 20, \*y1 + 10, buffer1);

fushitu(\*x1, \*y1);

delay(100);

putimage(\*x1 , \*y1 , buffer1, COPY\_PUT);

setcolor(RED);

line(rount[i][0], rount[i][1], rount[i + 1][0], rount[i + 1][1]);

if (i < \*amtjudge)

nyphoto(rount[i + 1][0], rount[i + 1][1]);

}

getimage(\*x1, \*y1, \*x1 + 20, \*y1 +10, buffer1);

fushitu(\*x1, \*y1);

delay(500);

putimage(\*x1 , \*y1, buffer1, COPY\_PUT);

free(buffer1);

free(x1);

free(y1);

}

int linemove1( int\* x, int\* y, int\* buffer1)//大农田沿线段运动

{

int a[10][2] = { {-1,0}, {-1,-1},{0,-1},{1,-1},{1,0},{1,1},{0,1},{-1,1},{-1,0}, {-1,-1} };

int b, judge = 0;

int i;

int x0 = 0, y0 = 0;

for (i = 1; i < 9; i++)

{

b = getpixel(a[i][0] + \*x, a[i][1] + \*y);

if (b == CYAN)

{

x0 = a[i][0];

y0 = a[i][1];

judge = 1;

break;

}

}

getimage(\*x-6, \*y-8, \*x + 16, \*y + 18, buffer1);

suofangtu(\*x, \*y,1);

delay(8);

putimage(\*x-6, \*y-8, buffer1, COPY\_PUT);

if (judge == 1)

{

putpixel(\*x + a[i-1][0], \*y + a[i-1][1], RED);

putpixel(\*x + a[i][0], \*y + a[i][1], RED);

putpixel(\*x + a[i+1][0], \*y + a[i+1][1], RED);

putpixel(\*x , \*y , RED);

\*x = \*x + x0;

\*y = \*y + y0;

putpixel(\*x,\*y, RED);

}

return judge;

}

int linemove21(int judge1, int\* x, int\* y, int\* buffer1)//小农田沿线段运动(顺时针)

{

int a[17][2] = { {0,-1},{1,-1},{1,0},{1,1},{0,1},{-1,1},{-1,0}, {-1,-1},{0,-1},{1,-1},{1,0},{1,1},{0,1},{-1,1},{-1,0}, {-1,-1},{0,-1} };

int b, judge = 0;

int i;

int x0 = 0, y0 = 0;

for (i = judge1; i < judge1+8; i++)

{

b = getpixel(a[i][0] + \*x, a[i][1] + \*y);

if (b == CYAN)

{

x0 = a[i][0];

y0 = a[i][1];

judge = 1;

break;

}

}

getimage(\*x, \*y, \*x + 20, \*y + 10, buffer1);

fushitu(\*x, \*y);

delay(8);

putimage(\*x, \*y, buffer1, COPY\_PUT);

if (judge == 1)

{

putpixel(\*x + a[i][0], \*y + a[i][1], RED);

putpixel(\*x + a[i+1][0], \*y + a[i+1][1], RED);

putpixel(\*x , \*y , RED);

\*x = \*x + x0;

\*y = \*y + y0;

putpixel(\*x,\*y, RED);

}

return judge;

}

int linemove22(int judge1, int\* x, int\* y, int\* buffer1)//小农田沿线段运动(逆时针)

{

int a[17][2] = { {0,-1}, {-1,-1}, {-1,0}, {-1,1}, {0,1} , {1,1}, {1,0}, {1,-1}, {0,-1}, {-1,-1}, {-1,0}, {-1,1}, {0,1} , {1,1}, {1,0}, {1,-1},{0,-1} };

int b, judge = 0;

int i;

int x0 = 0, y0 = 0;

for (i = judge1; i < judge1+8; i++)

{

b = getpixel(a[i][0] + \*x, a[i][1] + \*y);

if (b == CYAN)

{

x0 = a[i][0];

y0 = a[i][1];

judge = 1;

break;

}

}

getimage(\*x, \*y, \*x + 20, \*y + 10, buffer1);

fushitu(\*x, \*y);

delay(8);

putimage(\*x, \*y, buffer1, COPY\_PUT);

if (judge == 1)

{

putpixel(\*x + a[i][0], \*y + a[i][1], RED);

putpixel(\*x + a[i+1][0], \*y + a[i+1][1], RED);

putpixel(\*x, \*y, RED);

\*x = \*x + x0;

\*y = \*y + y0;

putpixel(\*x, \*y, RED);

}

return judge;

}

**37.pensadh.h**

#ifndef \_PENSADH\_H\_

#define \_PENSADH\_H\_

int psdonghua(int \*a10,int nongyao[6], unsigned char diannian[384][49], unsigned char dianbai[192][49], unsigned char dianmian[192][49], int rount[386][2],int\* amtjudge);

void ljgh(int grandjudge, int \*judge, int ny0[384][2], int ny1[192][2], int ny2[192][2], int ny3[192][2], int ny4[192][2], int rount[386][2]);//路径规划

void therount(int rount[386][2], int \*amtjudge);//路径画面

void huigui(void);//未发现病害，回归

void huigui2(int i);//喷洒农药后回归

void nyphoto(int x, int y);//农药图像

void anima1(int rount[386][2], int\* amtjudge);//大农田路径动画

void anima2(int rount[386][2], int\* amtjudge);//小农田路径动画

int linemove1(int\* x, int\* y, int\* buffer1);//大农田沿线段运动

int linemove21(int judge1 ,int\* x, int\* y, int\* buffer1);//小农田沿线段运动(顺时针) 21，22合并后，动画卡顿且容易把系统卡爆

int linemove22(int judge1 ,int\* x, int\* y, int\* buffer1);//小农田沿线段运动(逆时针)

#endif

**38.common.h**

#ifndef \_COMMON\_H\_

#define \_COMMON\_H\_

/\*

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

包含所有的头文件，

以后引用头文件只需要 include "common.h"；

更加方便，代码简洁

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

\*/

#include<graphics.h>

#include<stdio.h>

#include<stdlib.h>

#include<bios.h>

#include<string.h>

#include<dos.h>

#include<math.h>

#include<time.h>

#include <conio.h>

//上面是BC自带的头文件，下面为自己编写的

//----必备

#include"mouse.h"//鼠标库

#include"hz.h"//汉字库

//--界面

#include "welcome.h"//开屏动画

#include "mainpage.h"//主界面

#include "mainfunc.h"//农田界面

#include "elsetian.h"//另一个农田界面

#include "usehelp.h"//系统说明界面

#include "growlook.h"//生长检测动画

#include "outcome.h"//大农田的结果查看界面

//---func:功能文件

#include "move.h"//配合鼠标使用的鼠标move函数

#include "showtime.h"//显示时间函数

#include "shuju.h"//数据库

#include "IMAGE.h"

#include "drzhiwu.h"

#include "abdrone.h"

#include"illness.h"

#include "recover.h"

#include "pensa.h"

#include "pensadh.h"

#include "pssj.h"

#include "zpjm.h"

#endif

**39.image.c**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

该图像转换及输出函数，要求输入的图片分辨率在640\*480之内

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include <dos.h>

#include "image.h"

#include <graphics.h>

#include <conio.h>

int bmp\_convert(char \*bmp,char \*dbm) //将 \*.bmp 变为 \*.dbm,字符串bmp中为bmp所在地址，字dbmp为输出地址

{

// static int color[16]={0,4,2,6,1,5,3,7,8,12,10,14,9,13,11,15}; 上win下ps

static int color[16]={0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15};//索引与bc内颜色序号对应转换（建议用win画图转化）

unsigned char dbmline[640],bmpline[640],pixel[8],Vbyte;

int ImageW,ImageH,i,j,k,n,now,bmpcom,gg ;

long hangsize;

FILE \*fbmp,\*fdbm;

union {

unsigned char val;

struct {

unsigned cl:4;

unsigned ch:4;

}color; //用位域储存颜色信息，分拆颜色信息

}MyColor;

if((fbmp=fopen(bmp,"rb"))==NULL)

{

printf("%s\n",bmp);

printf("Open bmp error!");

return 1;

}

fseek(fbmp,18,SEEK\_SET); //bmp图像的宽高在文件头18字节后

fread(&gg,4,1,fbmp);

// fread(&ImageW,4,1,fbmp);

fread(&ImageH,4,1,fbmp);

ImageW=gg;

// ImageH\*=2;

// printf("%d ",ImageW);

// printf("%d",ImageH);

if(ImageW==0||ImageH==0||ImageW>641||ImageH>481)/////注意注意！！根据实际修改/

{

printf("The image is too large!Please input image below 640\*480.");

fclose(fbmp);

return 1;

}

if((fdbm=fopen(dbm,"wb"))==NULL)

{

printf("Creat \"%s\" error!",dbm);

return 1;

}

bmpcom=(ImageW-1)/8+1; //行处理单元数

hangsize=bmpcom\*4; //每行字节数（DIB数据要求每行字节为4整数倍，已计算补零）

fwrite(&ImageW,sizeof(int),1,fdbm);

fwrite(&ImageH,sizeof(int),1,fdbm);

fseek(fbmp,-hangsize,SEEK\_END);

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

{

now=0;

fread(bmpline,hangsize,1,fbmp);

fseek(fbmp,-hangsize\*2,SEEK\_CUR);

for(n=3;n>=0;n--) //每行拆成四个位面的数据输出

// for(n=0;n<=3;n++)

{

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

{

Vbyte=0;

for(k=0;k<4;k++)

{

MyColor.val=bmpline[j\*4+k];

pixel[k\*2]=color[MyColor.color.ch];

pixel[k\*2+1]=color[MyColor.color.cl];

}

for(k=0;k<8;k++)

{

Vbyte+=(pixel[k]>>n&1)<<(7-k);

}

dbmline[now++]=Vbyte;

}

}

fwrite(dbmline,hangsize,1,fdbm);

}

fclose(fdbm);

fclose(fbmp);

return 0;

}

int show\_dbm(int x,int y,char \*dbm,int nowpage) //输入dbmp的路径,在(x,y)输出图片(x需为8的倍数)

{

int ImageW,ImageH,n,i,j;

long hangsize;

FILE \* fdbm;

char far \* per;

if(nowpage==0)

per=(char far \*)0xA0000000L;

else

per=(char far \*)0xA8000000L;

// \_VideoBusy=YES; //hanenv系统内的显示寄存器保护变量

if((fdbm=fopen(dbm,"rb"))==NULL)

{

// printf("%s",dbm);

// getchar();

// printf("Open Error!");

return 1;

}

fread(&ImageW,sizeof(int),1,fdbm);

fread(&ImageH,sizeof(int),1,fdbm);

// printf("%d %d",ImageW,ImageH);

hangsize=(ImageW-1)/8;

per=per+x/8+y\*80;

if((ImageH+y)>480) ImageH=480-y; //VGAMED 640\*480

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

{

for(n=8;n>=1;n>>=1) //从第三位颜色平面写到第0

{

outportb(0x3c4,2);

outportb(0x3c5,n);

fread(per,hangsize,1,fdbm);

fseek(fdbm,1,SEEK\_CUR);

}

// fseek(fdbm,hangsize\*2,SEEK\_CUR);

per+=80;

}

fclose(fdbm);

// \_VideoBusy=NO;

outportb(0x3c5,0xf);

return 0;

}

void cir\_bar(int x1,int y1,int x2,int y2,int color) //在指定位置画出一个圆角矩形框

{

int w=x2-x1,h=y2-y1;

setcolor(color);

setlinestyle(SOLID\_LINE,0,THICK\_WIDTH);

ellipse(5+x1,10+y1,90,180,5,10);

line(0+x1,10+y1,0+x1,10+h-20+y1);

ellipse(5+x1,10+h-20+y1,180,270,5,10);

line(5+x1,0+y1,5+w-10+x1,0+y1);

ellipse(5+w-10+x1+1,10+y1,0,90,5,10);

line(w+x1,10+y1,w+x1,y2-10);

ellipse(w-5+x1+1,10+h-20+y1,270,360,5,10);

line(5+x1,h+y1,5+w-10+x1,h+y1);

}

void movetopage(int x1, int y1, int x2, int y2,int page1) //显示页间局部图像数据传输，输入左上角坐标及右上角坐标，从page1移动到page2

{

int per1, per2;

int hangsize = (x2 - x1) / 8;

int high = y2 - y1 + 1;

register int i, p1, p2;

if (page1 == 0)

{

per2 = 0xA0000000L;

per1 = 0xA8000000L;

}

else

{

per1 = 0xA0000000L;

per2 = 0xA8000000L;

}

per1 = per1 + (x1) / 8 + (y1 ) \* 80;

per2 = per2 + (x1) / 8 + (y1 ) \* 80;

for (i = 0; i <high; i++) //传输视频图像

{

for (p1 = 8, p2 = 3; p1 >= 1; p1 >>= 1, p2--) //从第三位颜色平面写到第0

{

outportb(0x3c4, 2); //颜色位面写寄存器

outportb(0x3c5, p1);

outportb(0x3ce, 4); //颜色位面读寄存器

outportb(0x3cf, p2);

if (page1 == 0)

{

quick\_move\_2(per2, per1, hangsize);

}

else

{

quick\_move\_1(per2, per1, hangsize);

}

}

per1 += 80;

per2 += 80;

}

outportb(0x3cf,0); //恢复设置

outportb(0x3c5,0xf);

}

void set\_color(int color\_no,int red,int green,int blue) //指定颜色编号，修改其对应的rgb值

{

if(color\_no<16)

{

\_AX=0x1007; //选择10H显示中断中的第7号功能

\_BL=color\_no;

geninterrupt(0x10);

color\_no=\_BH;

}

else

color\_no = 0xff; //将屏幕边缘色指向调色板255

//修改调色板寄存器

\_DH=red;

\_CH=green;

\_CL=blue;

\_BX=color\_no;

\_AX=0x1010;

geninterrupt(0x10);

//处理屏幕边缘颜色

if(color\_no==0xff)

{

\_BH=0xff;

\_AX=0x1001;

geninterrupt(0x10);

}

}

/\*此为用汇编编写的支持跨段寻址的数据传输函数，来自《VGA页面图形数据移动技术在特殊显示效果中的应用》一文\*/

int quick\_move\_1(int Start,int End,int Size) //Start为原处首地址，End为目标位置首地址 (1 to 0)

{

asm{

push es

push ds

push di

push si

push ax

push cx

push dx

mov ax,0a800h

mov ds,ax

mov ax,Start

mov si,ax

mov ax,0a000h

mov es,ax

mov ax,End

mov di,ax

mov cx,Size

rep movsb

pop dx

pop cx

pop ax

pop si

pop di

pop ds

pop es

}

return 0;

}

int quick\_move\_2(int Start,int End,int Size) //Start为原处首地址，End为目标位置首地址 (0 to 1)

{

asm{

push es

push ds

push di

push si

push ax

push cx

push dx

mov ax,0a000h

mov ds,ax

mov ax,Start

mov si,ax

mov ax,0a800h

mov es,ax

mov ax,End

mov di,ax

mov cx,Size

rep movsb

pop dx

pop cx

pop ax

pop si

pop di

pop ds

pop es

}

return 0;

}

void close\_display(void) //关闭屏幕显示

{

\_BL=0x32;

\_AH=0x12;

\_AL=0x01;

geninterrupt(0x10);

}

void open\_display(void) //打开屏幕显示

{

\_BL=0x32;

\_AH=0x12;

\_AL=0x00;

geninterrupt(0x10);

}

**40.image.h**

#ifndef IMAGE\_H

#define IMAGE\_H

int bmp\_convert(char \*bmp,char \*dbm); //将 \*.bmp 变为 \*.dbm,字符串bmp中为bmp所在地址，字dbmp为输出地址

int show\_dbm(int x,int y,char \*dbm,int nowpage); //输入dbmp的路径,在(x,y)输出图片(x需为8的倍数)

void cir\_bar(int x1,int y1,int x2,int y2,int color); //在指定位置画出一个圆角矩形框

void movetopage(int x1, int y1, int x2, int y2,int page1);

void set\_color(int color\_no,int red,int green,int blue); //指定颜色编号，修改其对应的rgb值

int quick\_move\_2(int Start, int End, int Size);

int quick\_move\_1(int Start, int End, int Size);

void close\_display(void); //关闭屏幕显示

void open\_display(void); //打开屏幕显示

#endif

**41.mouse.c**

#include<conio.h>

#include<graphics.h>

#include<dos.h>

#include<stdio.h>

#include<stdlib.h>

#include "mouse.h"

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MOUSE.c

UPDATER: dengshuumin

FUNCTION: mouse action

ABSTRACT:

A.mread

B.newmouse

VERSION: 3.0

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int MouseX;

int MouseY;

int MouseS;

int press;//形状用 &press

void \*buffer;//

union REGS regs;

int flag=0;

void mouseinit()//初始化

{

int retcode;

int xmin,xmax,ymin,ymax,x\_max=625,y\_max=480;

int size;

xmin=2;

xmax=x\_max-1;

ymin=8;

ymax=y\_max-2;

regs.x.ax=0;

int86(51,&regs,&regs);

retcode=regs.x.ax;

if(retcode==0)

{

printf("Mouse or Mouse Driver Obsent,Please Install!");

delay(5000);

}

else

{

regs.x.ax=7;

regs.x.cx=xmin;

regs.x.dx=xmax;

int86(51,&regs,&regs);

regs.x.ax=8;

regs.x.cx=ymin;

regs.x.dx=ymax;

int86(51,&regs,&regs);

}

MouseS = 0;

MouseX=320,MouseY=240;

save\_bk\_mou(320,240);

mouse(MouseX,MouseY);

flag=1;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION: mouse

DESCRIPTION: 画不同形态的鼠标

INPUT: x,y

RETURN: 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void mouse(int x,int y)

{

switch(MouseS)

{

case 1: //手势鼠标

{

setcolor(WHITE);

setlinestyle(0,0,1);

line(x-1,y+9,x-1,y+8);

line(x,y+7,x,y+11);

line(x+1,y+6,x+1,y+13);

line(x+2,y+8,x+2,y+14);

line(x+3,y-1,x+3,y+15);

arc(x+4,y-1,0,180,1);

line(x+4,y-2,x+4,y+15);

line(x+5,y-1,x+5,y+16);

arc(x+6,y+3,0,180,1);

line(x+6,y+2,x+6,y+16);

line(x+7,y+3,x+7,y+17);

arc(x+8,y+5,0,180,1);

line(x+8,y+4,x+8,y+17);

line(x+9,y+5,x+9,y+16);

arc(x+10,y+7,0,180,1);

line(x+10,y+6,x+10,y+16);

line(x+11,y+7,x+11,y+13);

setcolor(DARKGRAY);

line(x-1,y+9,x-1,y+8);

line(x-1,y+8,x+1,y+6);

line(x+1,y+6,x+3,y+10);

line(x+3,y+10,x+3,y-1);

arc(x+4,y-1,0,180,1);

line(x+5,y-1,x+5,y+5);

arc(x+6,y+3,0,180,1);

line(x+7,y+3,x+7,y+7);

arc(x+8,y+5,0,180,1);

line(x+9,y+5,x+9,y+9);

arc(x+10,y+7,0,180,1);

line(x+11,y+7,x+11,y+13);

arc(x+7,y+13,-90,0,4);

line(x+7,y+17,x+3,y+15);

line(x+3,y+15,x+1,y+13);

line(x+1,y+13,x-1,y+9);

}

break;

case 2: //光标

{

setcolor(DARKGRAY);

setlinestyle(0,0,1);

line(x+1,y-1,x+9,y-1);

line(x+1,y+15,x+9,y+15);

line(x+5,y-1,x+5,y+15);

}

break;

case 3: //十字

{

setcolor(WHITE);

setlinestyle(0,0,1);

line(x-1,y+7,x+11,y+7);

line(x+5,y-1,x+5,y+15);

}

break;

default: //默认鼠标

{

setlinestyle(0,0,1);

setcolor(WHITE);

line(x,y,x,y+13);

line(x+1,y+1,x+1,y+12);

line(x+2,y+2,x+2,y+11);

line(x+3,y+3,x+3,y+10);

line(x+4,y+4,x+4,y+12);

line(x+5,y+5,x+5,y+9);

line(x+5,y+11,x+5,y+14);

line(x+6,y+6,x+6,y+9);

line(x+6,y+13,x+6,y+15);

line(x+7,y+7,x+7,y+9);

line(x+8,y+8,x+8,y+9);

line(x+9,y+9,x+9,y+9);

setcolor(DARKGRAY);

line(x-1,y-1,x-1,y+14);

line(x-1,y+14,x+3,y+11);

line(x+3,y+11,x+3,y+12);

line(x+3,y+12,x+4,y+13);

line(x+4,y+13,x+4,y+14);

line(x+4,y+14,x+7,y+17);

line(x+7,y+17,x+7,y+13);

line(x+7,y+13,x+6,y+12);

line(x+6,y+12,x+6,y+11);

line(x+6,y+11,x+5,y+10);

line(x+5,y+10,x+11,y+10);

line(x+11,y+10,x-1,y-2);

}

break;

}

}

void mou\_pos(int \*nx,int \*ny,int\*nbuttons)//更改鼠标位置

{

int x0=\*nx,y0=\*ny;

mread(nx,ny,nbuttons);

clrmous(x0,y0);

save\_bk\_mou(\*nx,\*ny);

drawmous(\*nx,\*ny);

}

void mread(int \*nx,int \*ny,int\*nbuttons)//改坐标不画

{

int x0=\*nx,y0=\*ny,buttons0=\*nbuttons;

int xnew,ynew,buttonsnew;

do{

regs.x.ax=3;

int86(51,&regs,&regs);

buttonsnew=regs.x.bx;

delay(10);

regs.x.ax=3;

int86(51,&regs,&regs);

if(regs.x.bx==buttonsnew)

\*nbuttons=regs.x.bx;

else

\*nbuttons=buttons0;

xnew=regs.x.cx;

ynew=regs.x.dx;

}while(xnew==x0&&ynew==y0&&\*nbuttons==0);

\*nx=xnew;

\*ny=ynew;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION: mread

DESCRIPTION: 获取新的寄存器信息

INPUT: nx,ny,nbuttons

RETURN: 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//void mread(int \*nx,int \*ny,int \*nbuttons)

//{

// regs.x.ax=3;

// int86(51,&regs,&regs);

// \*nx = regs.x.cx;

// \*ny = regs.x.dx;

// \*nbuttons = regs.x.bx;

//}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

FUNCTION: newmouse

DESCRIPTION: 鼠标状态发生变化则更新鼠标

INPUT: nx,ny,nbuttons

RETURN: 无

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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

{

int xn,yn,buttonsn;

int x0=\*nx,y0=\*ny,buttons0=\*nbuttons;

mread(&xn,&yn,&buttonsn);

\*nx = xn;

\*ny = yn;

\*nbuttons = buttonsn;

if(buttons0 == \*nbuttons)

\*nbuttons = 0; //使得能连续按键

if(xn == x0 && yn == y0 && buttonsn == buttons0)

return; //鼠标状态不变则直接返回S

clrmous(x0,y0); //说明鼠标状态发生了改变

save\_bk\_mou(\*nx,\*ny);

drawmous(\*nx,\*ny);

}

void save\_bk\_mou(int nx,int ny)//存鼠标背景

{

int size;

size=imagesize(nx-1,ny-2,nx+11,ny+17);

buffer=malloc(size);

if(buffer!=NULL)

getimage(nx-1,ny-2,nx+11,ny+17,buffer);

else

printf("Error");

}

void clrmous(int nx,int ny)//清除鼠标

{

if(flag==1)

{

setwritemode(XOR\_PUT);

mouse(nx,ny);

putimage(nx-1,ny-2,buffer,COPY\_PUT);

free(buffer);

flag=0;

setwritemode(COPY\_PUT);

}

}

void drawmous(int nx,int ny)

{

if(flag==0)

{

setwritemode(COPY\_PUT);

mouse(nx,ny);

flag=1;

}

}

//如果在框中点击，则返回1；在框中未点击，则返回2；不在框中则返回0

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

{

//在框中点击，则返回1

if(MouseX > x1

&&MouseX < x2

&&MouseY > y1

&&MouseY < y2

&&press == 1)

{

return 1;

}

//在框中未点击，则返回2

else if(MouseX > x1

&&MouseX < x2

&&MouseY > y1

&&MouseY < y2

&&press == 0)

{

return 2;

}

//在框中点击右键，则返回3

else if(MouseX > x1

&&MouseX < x2

&&MouseY > y1

&&MouseY < y2

&&press == 2)

{

return 3;

}

else

{

return 0;

}

}

**42.mouse.h**

#ifndef \_mouse\_h\_

#define \_mouse\_h\_

int mouse\_press(int x1, int y1, int x2, int y2);//如果在框中点击，则返回1；在框中未点击，则返回2；不在框中则返回0

void mouse(int,int);//设计鼠标

void mouseinit(void);//初始化

void mou\_pos(int\*,int\*,int\*);//更改鼠标位置

void mread(int \*,int \*,int\*);//改坐标不画

void save\_bk\_mou(int x,int y);//存鼠标背景

void clrmous(int x,int y);//清除鼠标

void drawmous(int x,int y);//画鼠标

void newmouse(int \*nx,int \*ny,int \*nbuttons); //更新鼠标

extern int MouseX;

extern int MouseY;

extern int MouseS;

extern int press;

extern union REGS regs;

#endif

**43.hz.c**

#include <graphics.h>

#include<conio.h>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include"hz.h"

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

{

FILE \*hzk\_p=NULL; //定义汉字库文件指针

unsigned char quma,weima; //定义汉字的区码和位码

unsigned long offset; //定义汉字在字库中的偏移量

unsigned char mask[] = {0x80,0x40,0x20,0x10,0x08,0x04,0x02,0x01}; //功能数组，用于显示汉字点阵中的亮点

int i,j,pos;

switch(flag) //不同的flag对应不同的汉字库，实现了汉字的大小可根据需要改变

{

case 16 :

{

char mat[32]; //16\*16的汉字需要32个字节的数组来存储

int y0=y;

int x0=x;

hzk\_p = fopen("HZK\\HZ16","rb"); //使用相对路径

if(hzk\_p==NULL)

{

settextjustify(LEFT\_TEXT,TOP\_TEXT); //左部对齐，顶部对齐

settextstyle(GOTHIC\_FONT,HORIZ\_DIR,1); //黑体笔划输出，水平输出，24\*24点阵

outtextxy(10,10,"Can't open hzk16 file!Press any key to quit...");

getch();

exit(1);

}

while(\*s!=NULL)

{

while (x<640-flag && (\*s!=NULL))

{

y=y0;

quma=s[0]-0xa0; //求出区码

weima=s[1]-0xa0; //求出位码

offset=(94\*(quma-1)+(weima-1))\*32L; //求出要显示的汉字在字库文件中的偏移

fseek(hzk\_p,offset,SEEK\_SET); //重定位文件指针

fread (mat,32,1,hzk\_p); //读出该汉字的具体点阵数据,1为要读入的项数

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

{

pos=2\*i; //16\*16矩阵中有每一行有两外字节

for(j=0;j<16;j++) //一行一行地扫描，将位上为了1的点显示出来

{

if((mask[j%8]&mat[pos+j/8])!=NULL) //j%8只能在0—8之间循环，j/8在0，1之间循环

{

putpixel(x+j,y,color);

}

}

y++;

}

/\*====================================================

以上是一个汉字显示完

====================================================\*/

x+=part; //给x 一个偏移量part

s+=2; //汉字里存放的是内码，2个字节，所以要加2

}

x=x0;y0+=flag+10; //一行汉字显示完后,重新从左侧开始输出汉字，给y一个偏移量

}

break;

}

case 24 :

{

char mat[72]; //24\*24矩阵要72个字节来存储

int y0=y;

int x0=x;

hzk\_p = fopen("HZK\\Hzk24k","rb");

if (hzk\_p==NULL)

{

settextjustify(LEFT\_TEXT,TOP\_TEXT); //左部对齐，顶部对齐

settextstyle(GOTHIC\_FONT,HORIZ\_DIR,3); //黑体笔划输出，水平输出，24\*24点阵

outtextxy(10,10,"Can't open hzk24 file!Press any key to quit...");

getch();

exit(1);

}

while(\*s!=NULL)

{

while(x<640-flag && (\*s!=NULL))

{

y=y0;

quma=s[0]-0xa0; //求出区码

weima=s[1]-0xa0; //求出位码

offset=(94\*(quma-1)+(weima-1))\*72L;

fseek(hzk\_p,offset,SEEK\_SET);

fread (mat,72,1,hzk\_p);

for (i=0;i<24;i++)

{

pos=3\*i; //矩阵中每一行有三个字节

for (j=0;j<24;j++) // 每一行有24位

{

if ((mask[j%8]&mat[pos+j/8])!=NULL)

putpixel(x+j,y,color);

}

y++;

}

x+=part;

s+=2;

}

x=x0;y0+=flag+10;

}

break;

}

case 32 :

{

char mat[128]; //32\*32的汉字需要128个字节的数组来存储

int y0=y;

int x0=x;

hzk\_p = fopen("HZK\\HZK32S","rb");

if(hzk\_p==NULL)

{

settextjustify(LEFT\_TEXT,TOP\_TEXT); //左部对齐，顶部对齐

settextstyle(GOTHIC\_FONT,HORIZ\_DIR,3); //黑体笔划输出，水平输出，24\*24点阵

outtextxy(10,10,"Can't open hzk32 file!Press any key to quit...");

getch();

exit(1);

}

while(\*s!=NULL)

{

while (x<640-flag && (\*s!=NULL))

{

y=y0;

quma=s[0]-0xa0; //求出区码

weima=s[1]-0xa0; //求出位码

offset=(94\*(quma-1)+(weima-1))\*128L;

fseek(hzk\_p,offset,SEEK\_SET);

fread (mat,128,1,hzk\_p);

for(i=0;i<32;i++)

{

pos=4\*i; //32\*32矩阵中有每一行有两外字节

for(j=0;j<32;j++)

{

if((mask[j%8]&mat[pos+j/8])!=NULL)

{

putpixel(x+j,y,color);

}

}

y++;

}

//以上是一个汉字显示完

x+=part; //给x 一个偏移量part

s+=2; //汉字里存放的是内码，2个字节，所以要加2

}

x=x0;y0+=flag+10; //一行汉字显示完后，给y一个偏移量

}

break;

}

case 48:

{

char mat[288]; //48\*48的汉字需要288个字节的数组来存储

int y0=y;

int x0=x;

hzk\_p = fopen("HZK\\Hzk48k","rb");

if(hzk\_p==NULL)

{

settextjustify(LEFT\_TEXT,TOP\_TEXT); //左部对齐，顶部对齐

settextstyle(GOTHIC\_FONT,HORIZ\_DIR,3); //黑体笔划输出，水平输出，24\*24点阵

outtextxy(10,10,"Can't open hzk48 file!Press any key to quit...");

getch();

exit(1);

}

while(\*s!=NULL)

{

while (x<640-flag && (\*s!=NULL))

{

y=y0;

quma=s[0]-0xa0; //求出区码

weima=s[1]-0xa0; //求出位码

offset=(94\*(quma-1)+(weima-1))\*288L; //求出要显示的汉字在字库文件中的偏移

fseek(hzk\_p,offset,SEEK\_SET); //重定位文件指针

fread (mat,288,1,hzk\_p); //读出该汉字的具体点阵数据,1为要读入的项数

for(i=0;i<48;i++)

{

pos=6\*i;

for(j=0;j<48;j++) //一行一行地扫描，将位上为了1的点显示出来

{

if((mask[j%8]&mat[pos+j/8])!=NULL) //j%8只能在0—8之间循环，j/8在0，1之间循环

{

putpixel(x+j,y,color);

}

}

y++;

}

//以上是一个汉字显示完

x+=part; //给x 一个偏移量part

s+=2; //汉字里存放的是内码，2个字节，所以要加2

}

x=x0;y0+=flag+10; //一行汉字显示完后，给y一个偏移量

}

break;

}

default:

break;

}

fclose(hzk\_p);

}

**44.hz.h**

#ifndef \_\_HZ\_H\_\_

#define \_\_HZ\_H\_\_

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

#endif

# 第八部分 个人总结

## 组员总结

在这个数月的C语言课程设计中，我们经历了挑战，也获得了成长。这是我第一次接触如此复杂的系统模拟和数据处理，这次课程设计对我们来说是一次全新的挑战。作为对C语言的初次应用，我面临着诸多未知的困难和障碍。然而，正是在这个过程中，我不断突破自我，不断学习成长

作为一个刚刚接触C语言的初学者，我一开始对课程设计的要求感到有些迷茫和困惑。然而，在不断学习下，我逐渐理解了C语言的基本语法和编程思维。通过阅读和分析往届的代码，我学到了很多关于界面设计和图形化操作的技巧，也明白了整体设计对于一个项目的重要性。我们共同克服了许多困难，最终完成了一个令我们满意的作品。

在整个课程设计的过程中，我们不仅学会了如何应用所学的知识，更重要的是，我们获得了成长和成就感。每一次解决一个bug，优化一段代码，我们都感到无比的满足和自豪。尤其是在最后的验收阶段，当我们展示出我们的成果时，我感到了一种前所未有的成就感和自信。

我们也从中汲取了许多宝贵的教训。比如，在整个开发过程中，我们经常会遇到各种各样的问题和挑战，有时甚至会陷入困境。但正是在这些困境中，我们学会了如何坚持不懈地解决问题，如何相互合作，如何克服困难。这些经历让我们变得更加坚强和成熟。

最后，我想说的是，这次课程设计不仅让我们学会了如何应用C语言进行软件开发，更重要的是，它教会了我们如何面对挑战，如何不断学习和成长。在未来的学习和工作中，我会继续努力，不断提升自己的能力，为实现更多的梦想而努力奋斗。

总的来说，这次C语言课程设计是一次充满挑战和收获的经历。通过与队友的合作和努力，我们不仅完成了一个令人满意的作品，更重要的是，我们获得了成长和成就感。我相信，这次经历将成为我们人生道路上宝贵的财富，激励我们不断向前，追求更高的目标和更好的未来。

## 组员总结

这次C语言课程设计是我目前参与设计的，对我的各项能力考察最为全面和彻底的程序课题。作为对课程设计的前期准备，我提前搜集和了解了相关的各类C语言程序与技术，当然最重要的依然是在编写程序的过程中发现或意识到相关问题，并及时补充学习相应知识技能，在纠正错误的过程中提升自己的能力。  对工程思维，功能实现，以及追求复杂精简程序结构而非简单繁杂程序结构的课设内容核心的理解，是我在实践设计和与之同步的学习过程中逐步意识到的，因此对于自己参与编写的程序，我认为将精力花费在对图像界面的吹毛求疵是不大值当的，因此对于程序的实现，我和队友达成的共识是着重于对所要实现的程序功能，采取尽量实用的功能结构。  这次C课设，对于程序的整个编写过程，我通过实践巩固和进一步深化了对既有知识和技能的理解，也在编写过程中为克服相应问题，具有明确目的性的学习了实用性的技能和知识，除却基本的，明确的在编写过程对某些知识的提升，从整体和系统性上来说，对工程思维和结构思维的明确把握，以及对实践的重要性的理解对于我而言也同样主要。具体到在本次C课设的获得的设计技巧，主要有以下几点:

 一: 程序整体思维与特定功能实现的平衡，假如从程序的宏观实现上思考，界面函数间程序的跳转，程序的顺序的编排很大程度上会受直觉的限制，而在具体的程序实现上，哪些数据需要被传递，哪些函数会在哪些界面被运用，都可能因为之前对函数宏观实现的武断决定而受到限制，所以，应当在做整体设计时，为适度的修改保留空间，尽量做到宏观与微观的平衡。

 二: 实现目标功能的函数本身比功能实现本身更加重要，假如编写者充分了解自己函数实现的过程，而此时函数实现出现问题，或者因为需求的变更而需要对函数进行修改，编写者可以更轻松地变更代码。

三: 实际是检验真理的唯一途径。相较于单纯空对空地学习知识，通过实践所验证或获取的知识无疑更加深刻而且透彻，对于C课设课题的实现而言，通过在编写代码的过程中发现问题以及自己的不足之处无疑对自己更具指导价值。

这次C课设相关程序的编写使我受益匪浅，对我的相关技能和思维模式有极大的提升。

# 第十部分 时间安排与代码分工

## 一 时间安排

**七，进度分析**

|  |  |
| --- | --- |
| **周数** | **完成的任务** |
| **第一周** | **需求分析与任务分工** |
| **第二周** | **程序总体框架流程确定** |
| **第三周** | **大农田以及植物绘制** |
| **第四周** | **大农田生长检测，结果查看，农药配置功能编写** |
| **第五周** | **小农田及植物绘制** |
| **第六周** | **小农田生长检测，结果查看，农药配置功能编写** |
| **第七周** | **大小农田喷洒农药功能。优化界面。** |
| **第八周** | **合并代码并进行维护,优化代码机构，功能完善。** |
| **第九周** | **完成终期报告并做好验收准备** |

## 代码分工

：主要负责大农田和小农田的界面绘制。生长检测功能，结果查看功能。植物绘制，患病虫害功能。

总有效代码约 2358行。

：主要大小农田的农药配置功能。最优路径规划功能。无人机喷洒农药并且避障碍物功能。

总有效代码约 2135 行。

最后，两人一起编译、调试、Debug并撰写分析报告。