

## 案例 1—生物智能活动模拟设计与实现

一、问题描述 .....	1
1.1 基本要求 .....	1
1.2 蚂蚁觅食行为分析 .....	1
1.2.1 信息素 ( pheromone ) 的重要作用 .....	1
1.2.2 蚂蚁觅食规则 .....	1
二、Console 下的 C++绘图 .....	2
2.1 基本概念 .....	2
2.2 绘图过程 .....	2
2.3 常用绘图函数 .....	3
2.4 完整示例 .....	3
三、觅食模拟程序的设计 .....	4
3.1 类结构 .....	4
3.2 各类主要功能描述 .....	5
3.2.1 world 类 .....	5
3.2.2 Ant 类 .....	6
3.2.3 Home 和 Food 类 .....	6
3.3 主程序流程 .....	6
3.4 程序运行主界面 .....	7

## 一、问题描述

### 1.1 基本要求

模拟  $n$  只蚂蚁在  $M \times N$  的区域内寻找食物的过程。要求在  $M \times N$  的区域内，可随机设置蚁巢和食物地点，在对蚂蚁觅食规律掌握的基础上，编写程序采用动画形式展示蚁群的觅食过程。

### 1.2 蚂蚁觅食行为分析

#### 1.2.1 信息素 ( pheromone ) 的重要作用

蚂蚁之间并没有直接的关系，但是每只蚂蚁都和环境发生交互，而通过信息素这个纽带，实际上把各个蚂蚁之间关联起来了。比如，当一只蚂蚁找到了食物，它并没有直接告诉其它蚂蚁这儿有食物，而是向环境播撒信息素，当其它的蚂蚁经过它附近的时候，就会感觉到信息素的存在，进而根据信息素的指引找到了食物。

#### 1.2.2 蚂蚁觅食规则

##### 1、范围

蚂蚁观察到的范围是一个方格世界，蚂蚁有一个参数为视距半径（一般是 3），那么它能观察到的范围就是  $3 \times 3$  个方格世界，并且能移动的距离也在这个范围之内。

##### 2、环境

蚂蚁所在的环境有障碍物，有别的蚂蚁，还有信息素，信息素有两种，一种是找到食物后，蚂蚁洒下的食物的信息素，一种是找到巢后，蚂蚁洒下的巢的信息素。每个蚂蚁都仅仅能感知它范围内的环境信息。环境以一定的速率让信息素消失。

##### 3、觅食规则与找巢规则

在每只蚂蚁能感知的范围内寻找是否有食物，如果有就直接过去。否则看是否有信息素，并且比较在能感知的范围内哪一点的信息素最多，这样，它就朝信息素多的地方走，并且每只蚂蚁多会以小概率犯错误，从而并不是往信息素最多的点移动。

蚂蚁找巢的规则和上面一样，只不过它对巢的信息素做出反应，而对食物信息素没反应。

##### 4、移动规则

每只蚂蚁都朝向信息素最多的方向移，并且，当周围没有信息素指引的时候，蚂蚁会按照自己原来运动的方向惯性的运动下去，同时在运动的方向有一个随机的小的扰动。为了防止蚂蚁原地转圈，它会记住最近刚走过了哪些点，如果发现要走的下一点已经在最近走过了，它就会尽量避开。

### 5、避障规则：

如果蚂蚁要移动的方向有障碍物挡住，它会随机的选择另一个方向，并且有信息素指引的话，它会按照觅食规则或者找巢规则进行动作。

### 6、播撒信息素规则：

每只蚂蚁在刚找到食物或者窝的时候散发的信息素最多，并随着它走远的距离，播撒的信息素越来越少。

## 二. Console 下的 C++绘图

### 2.1 基本概念

#### 1. GDI

Windows 向用户提供了 Graphics Device Interface (GDI, 图形设备接口) 这个与设备无关的编程接口

#### 2. DC

Device Context (DC): 设备场境是一个定义了图形对象集,这些对象的属性以及影响输出的图形模式的结构.

利用 GDI 接口,用户把各种图形设备都看成窗口,你只要获得这个"窗口"的设备场境句柄,就能在这个窗口上自由的作图

### 2.2 绘图过程

#### 1. 找到程序运行窗口的句柄

```
HWND hWnd= FindWindow("ConsoleWindowClass ", NULL);
```

#### 2. 通过窗口句柄得到该窗口的设备场境句柄

```
HDC hDC=GetDC(hWnd);
```

#### 3. 设置画笔和画刷

画笔即线条属性，包括颜色，线型，大小等

画刷即填充属性，包括颜色，填充方式等

##### ( 1 ) 变量申明

```
HPEN hPen,hOldPen; //画笔
```

```
HBRUSH hBrush,hOldBrush; //画刷
```

##### ( 2 ) 创建资源

```
hPen=CreatePen(PS_SOLID,2,0x00ff00);
```

```
hBrush=CreateSolidBrush(RGB(255,0,0));
```

##### ( 3 ) 把资源引入设备场境

```
OldPen=SelectObject(hDC,hPen);
```

```
hOldBrush=(HBRUSH)SelectObject(hDC,hBrush);
```

#### ( 4 ) 开始绘图

- MoveToEx(hDC,20,50,NULL); //设置画线起点
- LineTo(hDC,520,550); //画到终点
- Ellipse(hDC,80,50,200,100); //绘制椭圆

#### ( 5 ) 释放资源

- SelectObject(hDC,hOldPen);
- DeleteObject(hPen);
- SelectObject(hDC,hOldBrush);
- DeleteObject(hBrush);
- ReleaseDC(hWnd, hDC);

### 2.3 常用绘图函数

- MoveToEx, LineTo ,
- Rectangle , Ellipse , Arc , Polygon , PolylineTo
- TextOut , SetTextColor , SetBkColor , SetBkMode
- SetPixel
- DrawText

### 2.4 完整示例

```
#include <windows.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char* argv[]) {
    //找到程序运行窗口的句柄
    HWND hWnd=FindWindow( "ConsoleWindowClass ", NULL);
    HDC hDC=GetDC(hWnd);
    HPEN hPen,hOldPen; //画笔
    HBRUSH hBrush,hOldBrush; //画刷
    int i=0;
    //用画点的办法画一根线,最后一个参数是颜色 ( 32 位 )
    for(i<500;++i)
        SetPixel(hDC,10+i,10+i,0x0000ff);
    //生成绿色画笔
```

```

    hPen=CreatePen(PS_SOLID, 2, 0x00ff00);
    //把画笔引入设备场境
    hOldPen=(HPEN)SelectObject(hDC , hPen);

    hBrush=CreateSolidBrush(RGB(255, 0, 0));
    hOldBrush=(HBRUSH)SelectObject(hDC , hBrush);
    MoveToEx(hDC,20,50,NULL); //设置画线起点
    LineTo(hDC,520,550);    //画到终点

    Arc(hDC,100,100,300,300,350,500,350,500);//画圆
    Ellipse(hDC,80,50,200,100);
    SetTextColor(hDC,RGB(0,0,255));

    TextOut(hDC,100,100,"ZENG",1);
    //释放资源
    SelectObject(hDC , hOldPen);
    DeleteObject(hPen);
    SelectObject(hDC , hOldBrush);
    DeleteObject(hBrush);
    ReleaseDC(hWnd,hDC);
    //下面是对比，表明它确实是控制台程序
    printf("hello console");
    system("pause");
    return 0;
}

```

### 三、觅食模拟程序的设计

#### 3.1 类结构

通过对程序任务的分析，可将程序主体结构分解为以下四个类组成：

World：该类用来模拟模拟所在的周围世界

Ant：该类模拟蚂蚁个体

Food：该类模拟蚂蚁所要寻找的食物目标

Home：该类模拟整个蚁群的蚁巢

这四个类之间的关系如图 1 所示。

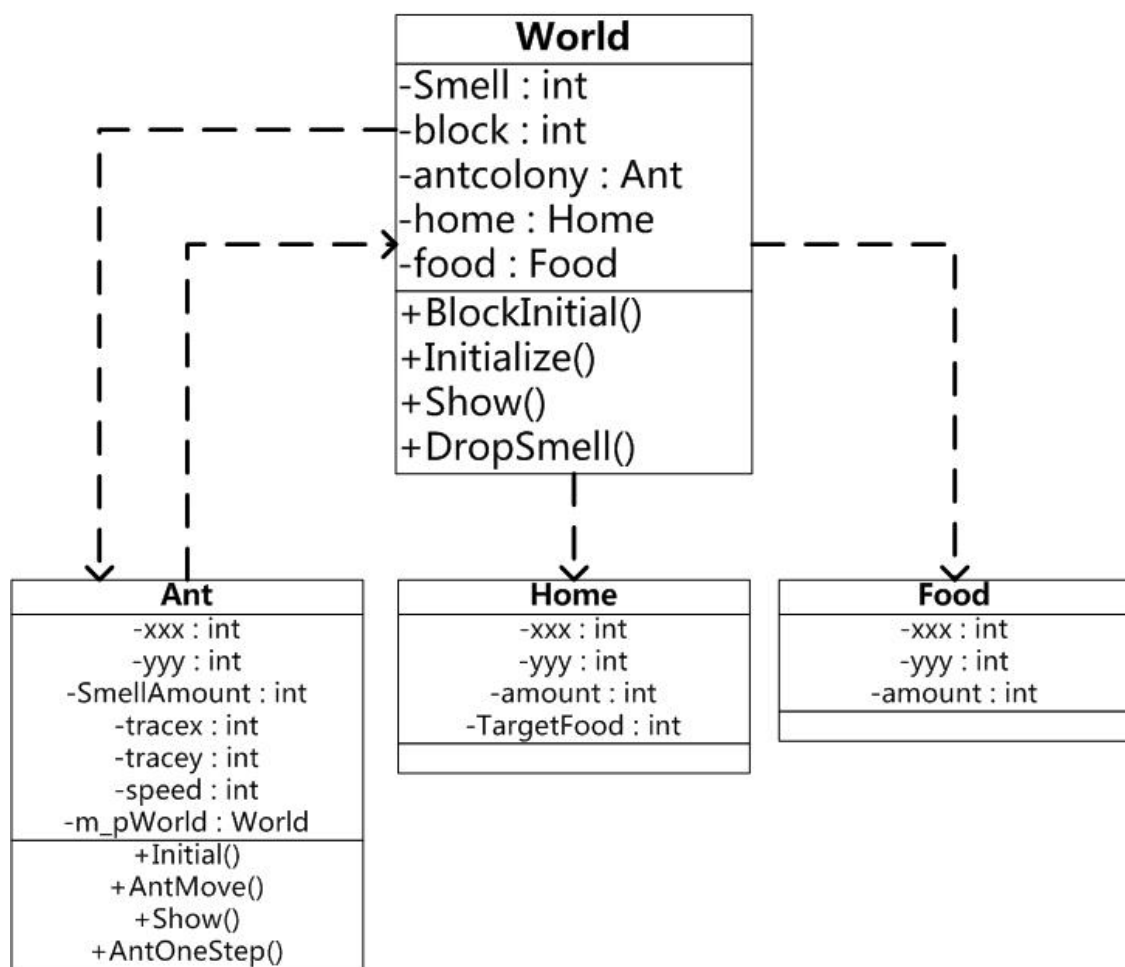


图 1 觅食模拟程序类结构图

## 3.2 各类主要功能描述

### 3.2.1 world 类

该类用来模拟蚁群所处的外部世界，其包含的属性有：

( 1 )  $M \times N$  的方格，每格指示是否为障碍以及每个格中蚂蚁信息素的量（环境每个位置有两种信息素），分别用二维数组和三维数组表示

```

int* block = new int[(width+1)*(height+1)];
int* Smell = new int[2*(width+1)*(height+1)]; //三维数组
  
```

( 2 ) 蚂蚁群、蚁巢与食物

```

Home home;
Food food;
Ant* antcolony = new Ant[MAX_ANT];
  
```

### (3) 其它

方格的宽与高，食物的总量，画图参数等

除了属性，该类还有很多动作（方法），比如初始化，显示，时间模拟等等。

#### 3.2.2 Ant 类

该类用来模拟蚂蚁个体，包括位置，移动方向，速度，是否有食物，信息素量，路径记忆等。

蚂蚁最重要的动作为移动，其流程为：

1. 擦除原位置显示
2. 计算蚂蚁下一个位置
  - 2.1. 找出下一个可行方向
  - 2.2. 移动到新位置
  - 2.3. 如果有食物  
如果到家，放下食物，重新初始化
  - 2.4 否则，如果在食物位置  
食物信息素初始化  
如果在食物位置，搬食物返回，路径记忆初始化，信息素初始化
  - 2.5 否则，如果到家，巢信息素初始化
3. 显示蚂蚁
4. 更新路径记忆
5. 撒播信息素

#### 3.2.3 Home 和 Food 类

分别表示食物和巢，包括位置和数量信息等。

### 3.3 主程序流程

主程序流程步骤如下：

1. 环境初始化
2. 环境显示
3. 环境更新
4. 计时模拟
5. 等待按键判断
6. 重复 2-5，直至按键退出或者觅食完成

### 3.4 程序运行主界面

程序运行后的主界面如图 2 所示。



图 2 程序运行后的主界面