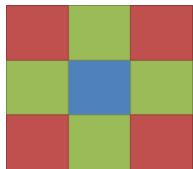


Bundeswettbewerb Informatik – Aufgabe 2: Ameisen

Team: „DER LAMABÄR“

Lösungsidee:

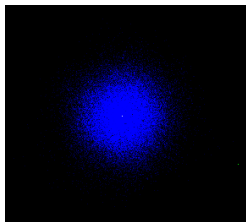
Es gilt ein Simulationsprogramm zu schreiben, dem Werte für die Position des Nestes, der Zahl der Ameisen, der Zahl der Futterquellen, der Menge Futter pro Futterquelle, sowie die Lebensdauer der Duftmarken per Nutzereingabe mitgegeben werden können.



Laut Aufgabe können Ameisen sich nur auf angrenzende Felder bewegen.

Angrenzende Felder sind lediglich die direkt benachbarten Felder (grün) mit einer Grenzfläche zum Ausgangsfeld (blau). Rote Felder sind nicht nur nicht angrenzend, sondern haben auch einen größeren Abstand zum Ausgangsfeld, sie sind ca. 1,41 LE anstatt 1 LE vom Ausgangsfeld entfernt.

Die ‚Welt‘ der Simulation hat gemäß Aufgabenstellung 500x500 Felder. Die Ameisen bewegen sich, solange sie noch kein Futter tragen, entweder auf das Feld mit der höchsten Duftkonzentration, welches vom Nest weg führt, oder, wenn kein benachbartes Feld eine erhöhte Konzentration aufweist, auf ein zufälliges Feld in ihrer Nachbarschaft. Anfangs hat, da noch keine Ameise Futter gefunden hat und alle Ameisen im Nest sind, kein Feld Duftspuren. Entsprechend bewegen sich alle



Verteilung von 50000
Ameisen ohne nahes
Futter

Ameisen in zufällige Richtungen. Dementsprechend ist zu erwarten, dass sich die Gesamtheit der Ameisen in Nestnähe befindet: Der „Schwerpunkt“ der Ameisen ist wahrscheinlich das Nest, die Wahrscheinlichkeit Ameisen in großer Entfernung vom Nest zu finden ist eher gering: Jede Ameise hat 4 Möglichkeiten sich zu bewegen. Die Wahrscheinlichkeit p ein bestimmtes Feld, beispielsweise das oberste zu betreten, beträgt daher $1/4$. Entsprechend ist die Wahrscheinlichkeit nach 100 Simulationsschritten eine Ameise 100 LE vom Nest entfernt zu finden gleich $1/4^{100} * 4 = 2,5 * 10^{-60}$, also praktisch 0. Mit zunehmender Zahl von Simulationsschritten wird es wahrscheinlicher, hier

```

type
  TFeld = record
    b_futterquelle:boolean;
    i_futter:integer;
    b_nest:boolean;
    i_ameisen:integer;
    i_duftstaerke:integer;
    i_duftdauer:integer;
  end;

  TAmeise = record
    b_trägtFutter:boolean;
    x:integer;
    y:integer;
  end;

```

eine Ameise anzutreffen. Es empfiehlt sich jedoch, eine ausreichend hohe Zahl von Ameisen zu simulieren und vor allem, deutlich mehr als 5 Futterquellen zu setzen, um lange Wartezeiten zu vermeiden.

Jedes Feld der simulierten Welt trägt bestimmte Informationen: Ist es eine Futterquelle? Wenn ja, wie viel Futter gibt es dort zurzeit? Oder ist es das Nest? Weiterhin wird gespeichert, wie viele Ameisen das Nest zurzeit beherbergt, wie stark der Duft ist und wie lange er noch anhält.

Die Ameisen wissen, ob sie Futter tragen und wo sie sind.

Die Welt wird von der Simulation als 500x500 Pixel Grafik dargestellt. Die Farbe der Pixel gibt Aufschluss über die Werte

des entsprechenden Feldes: Das Nest ist weiß. Futterquellen sind grün, die Intensität hängt von der relativen Futtermenge ab. Blaue Farbanteile stellen die Ameisen dar; rote hingegen die Anwesenheit von Duftstoffen.

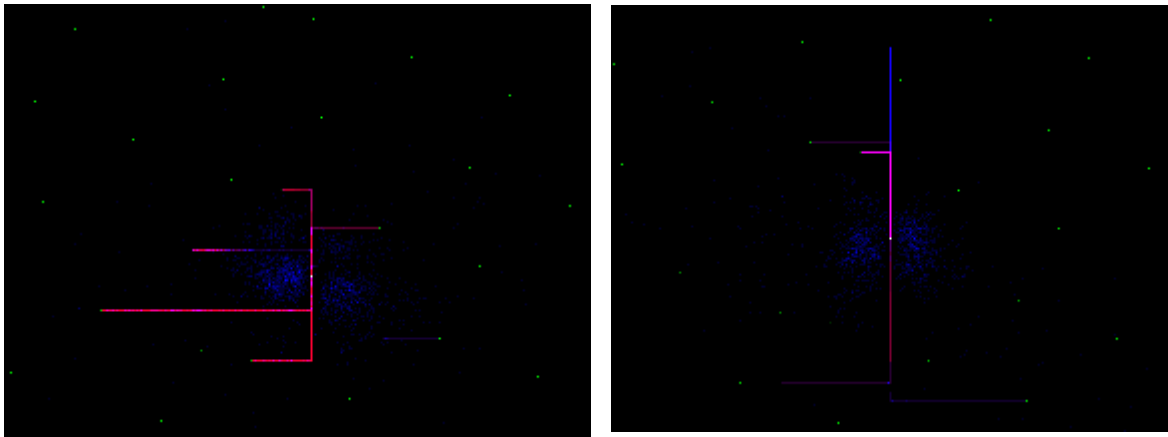
Da ein Pixel ziemlich klein ist, ist leider nicht allzu viel zu erkennen. Einzelne Ameisen lassen sich auch nur schwer ausmachen.

Umsetzung:

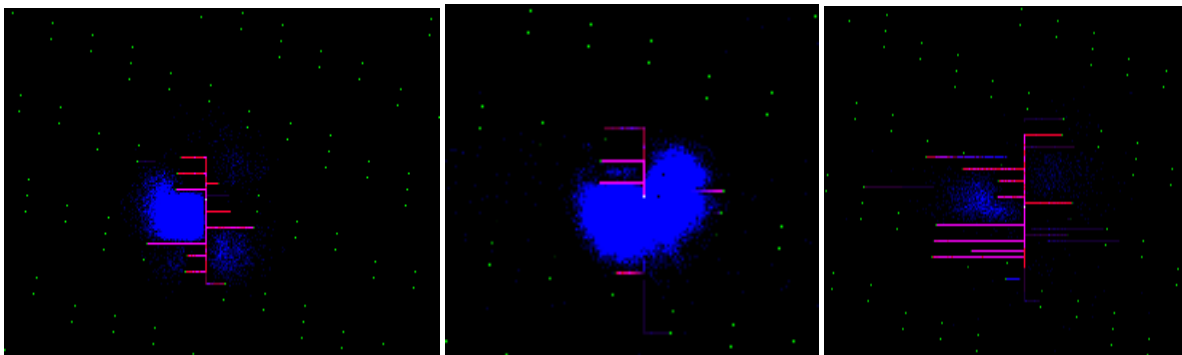
Die Simulation wurde mit Delphi in Pascal geschrieben. Wie bereits oben angedeutet, werden die wichtigen Daten in Records gespeichert. Das Programm verwaltet die verschiedenen Ameisen in einem Array[1..n] of Tameise, bzw. die 250000 Felder in einem Array[x,y] of TFeld.

Die Anfangseinstellungen des Nutzers werden global gespeichert.

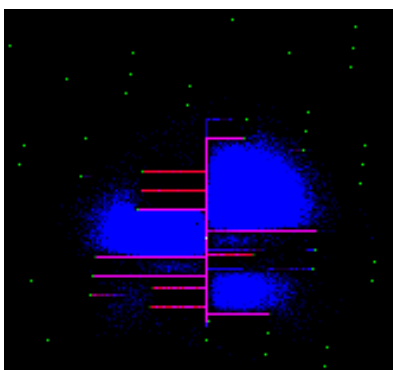
Beispiele:



5000 Ameisen suchen Futter bei einer Futtermenge von 50000/Futterquelle und einer Verdunstung von 100. Der blaue ‚Schwarm‘ in Nestnähe zeigt die Ameisen, die umher irren; rot deutet auf Duftstoffe hin. Rote und blaue Farbanteile zusammen sind ‚Ameisenstraßen‘, die aktiv genutzt werden.



Eine Million Ameisen suchen nach Futter. Jede der 500 Futterquelle enthält 1.000.000 Futtereinheiten. Duftspuren verdunsten nach 50 Simulationsschritten. Die vielen Ameisen sind logischerweise besser darin, auch sehr viele Straßen zu unterhalten. Auch ist der Schwarm (siehe Abb. 2) gut dafür geeignet, Futterquellen aufzuspüren.



Bei 10 Millionen Ameisen zeigt sich besonders gut die „Sogwirkung“ der Ameisenstraßen. In der Nähe von Duftspuren werden die herumirrenden Ameisen weniger, da sie, sobald sie

auf die Duftspur treffen, dem Duft Richtung Futter folgen.

Quellcode (gekürzt, vollständige Version in Main.pas):

```

procedure TForm2.StartClick(Sender: TObject);    //Hauptprozedur
var i,j,l:integer;
begin
    randomize;
    offset:=0;
    //[...] Übertrage Eingaben aus InputFelder
    //initisalisiere Welt
    initialisiereWelt;
    for l := 1 to schrittzahl do begin
        for i := 0 to ameisenzahl-1 do bewegeAmeise(i); //Simulation der
    einzelnen Ameisen
        updateAmeisenpos;           //die Welt entsprechend anpassen
        application.ProcessMessages; //Damit Programm ansprechbar bleibt.
        schritte.Caption:=inttostr(l); //Anzahl der Schritte anzeigen
        zeichneWelt;           //Aktuelle Positionen anzeigen.
    end;
end;

procedure TForm2.updateAmeisenPos;
var x,y,i:integer;
begin
    for x := 1 to 500 do
        for y := 1 to 500 do begin
            welt[x,y].i_ameisen:=0; //Ameisen-Reset
            if welt[x,y].i_duftdauer>0 then
                welt[x,y].i_duftdauer:=welt[x,y].i_duftdauer-1 else
                welt[x,y].i_duftstaerke:=0; //Duft verdunstet
            end;
            for i := 0 to ameisenzahl-1 do begin //Für jede Ameise
                welt[ameise[i].x, amese[i].y].i_ameisen:=welt[ameise[i].x,
                amese[i].y].i_ameisen+1; //Welt updaten (Anzahl der Ameisen +1)
                if amese[i].b_trägtFutter=true then begin //Wenn die Ameise Futter
                trägt, Duft hinzufügen
                    welt[ameise[i].x, amese[i].y].i_duftstaerke:=welt[ameise[i].x,
                    amese[i].y].i_duftstaerke+1;
                    welt[ameise[i].x, amese[i].y].i_duftdauer:=verdunstung;
                end;
            end;
        end;
end;

procedure TForm2.bewegeAmeise(a:integer);
var
    x,y,i:integer; //X,Y für die aktuelle Position
    n,o,s,w:integer; //Duftwert im Norden, Osten, ...
    done:string; //s.u.
    value:integer; //Maximaler Duftwert in Nachbarschaft
    r:integer; //Random
begin
    //Hat die Ameise bereits Futter?
    if Ameise[a].b_trägtFutter=true then begin
        //Gehe zum Nest!
        if amese[a].x<>nestx then begin //Gehe solange nach Ost/West, bis du
        die XPosition des Nestes erreicht hast.
            if amese[a].x<nestx then amese[a].x:=ameise[a].x+1 else
            amese[a].x:=ameise[a].x-1;

```

```

    end else begin
        if ameise[a].y < nesty then ameise[a].y := ameise[a].y + 1 else begin
//Gehe solange nach Nord/Süd, bis du die YPosition des Nestes und damit das
Nest erreicht hast.
            if ameise[a].y > nesty then ameise[a].y := ameise[a].y - 1 else
ameise[a].b_trägtFutter := false; //Wenn auf Nest gibt die Ameise ihr Futter
ab.
        end;
    end;
end else begin //Wenn die Ameise kein Futter hat:
//Suche Futter!
//Suche nach benachbarten Feldern mit Duft
//Duftstärke der Felder nach Himmelsrichtung
w:=0; //Initialisieren: Alle Felder haben standartmäßig keinen Duft.
o:=0;
n:=0;
s:=0;
x:=Ameise[a].x; //Speichere die Position als x und y -- ist einfacher zu
schreiben...
y:=Ameise[a].y;
if x>1 then w:=welt[x-1,y].i_duftstaerke+1; //Werte aus Welt lesen, 1
addieren, sofern das Feld existiert. Nicht existente Felder haben daher den
Duftwert 0 und werden nicht gewählt.
if x<499 then o:=welt[x+1,y].i_duftstaerke+1;
if y>1 then n:=welt[x,y-1].i_duftstaerke+1;
if y<500 then s:=welt[x,y+1].i_duftstaerke+1;
//Was ist die höchste Konzentration auf den Nachbarfeldern? -
Speichern als 'Value'
value:=max(s,Max(n,Max(w,o)));
i:=0;
repeat
    done:='-'; //Ameise hat sich noch nicht bewegt.
    i:=i+1; //Dies ist der i-te Aufruf
    if value>1 then begin //Wenn der maximale Duftwert größer 1 ist,
d.h. wenn min. 1 benachbartes Feld duftet.
        if (value=w) and (done='-') then begin //Wenn der maximale
Duftwert dem Duftwert des westlichen Feldes entspricht.
            if x-1-nestx<x-nestx then begin
//Move west
Ameise[a].x:=Ameise[a].x-1; //Position verändern
done:='w'; //Done auf 'w' setzen (entspricht true),
als kennzeichnung, dass die Ameise sich bereits bewegt hat.
            end else begin
                value:=max(s,max(n,o));
            end;
        end;
    end;
    if (value=o) and (done='-') then begin //s.o.
        if x+1-nestx>x-nestx then begin
//Move east
Ameise[a].x:=Ameise[a].x+1;
done:='o';
        end else begin
            value:=max(s,max(n,w));
        end;
    end;
    if (value=n) and (done='-') then begin //S.o.
        if y+1-nesty>y-nesty then begin
//Move north
Ameise[a].y:=Ameise[a].y-1;
done:='n';
        end else begin
            value:=max(s,max(o,w));

```

```

        end;
    end;
    if (value=s) and (done='-') then begin //s.o.
        if y-1-nesty<y-nesty then begin
            //Move south
            Ameise[a].y:=Ameise[a].y+1;
            done:='s';
        end else begin
            value:=max(n,max(o,w));
        end;
    end;
end;
until ((done<>'-' ) OR (i>1)); //Bricht ab, wenn passendes Feld
gefunden worden ist (done<>'-' ) oder wenn er zweifach erfolglos war
(Möglicherweise war der höchste Wert 1)
if value=1 then begin //Value=1 heißt, dass kein Feld Duft hat, bzw.
nur die Felder, die näher am Nest sind, als das aktuelle Feld.
//Zufällig ein Feld auswählen.
randomize;
r:=random(4+1); //Zahl kann 1,2,3,4 sein, 1 heißt West, 2 Ost,
...
if r=1 then Ameise[a].x:=Ameise[a].x-1;
if r=2 then Ameise[a].x:=Ameise[a].x+1;
if r=3 then Ameise[a].y:=Ameise[a].y-1;
if r=4 then Ameise[a].y:=Ameise[a].y+1;
end;
//Zur Sicherheit: Wenn die Position außerhalb der Welt ist: Reset. Wird
eigentlich nicht benötigt.
if ameise[a].x>500 then ameise[a].x:=500;
if ameise[a].x<1 then ameise[a].x:=1;
if ameise[a].y>500 then ameise[a].y:=500;
if ameise[a].y<1 then ameise[a].y:=1;
//Sammle Futter auf, wenn die Ameise eine Futterquelle betritt, sie
noch kein Futter trägt, und die Futterquelle noch über Futter verfügt.
if (welt[ameise[a].x, ameise[a].y].b_futterquelle=true) and
(ameise[a].b_trägtFutter=false) and (welt[ameise[a].x,
ameise[a].y].i_futter>0) then begin
    welt[ameise[a].x, ameise[a].y].i_futter:=welt[ameise[a].x,
ameise[a].y].i_futter-1; //Futtermenge um 1 reduzieren.
    ameise[a].b_trägtFutter:=true; //Ameise trägt Futter
end;
end;
end;
end;

procedure TForm2.zeichneWelt;
var
x,y,i:integer;
duft:integer;
futter:integer;
a_ameisen:integer;
begin
    image1.canvas.Brush.Color:=clblack;
    image1.Canvas.Rectangle(0,0,500,500);
    Form2.doubleBuffered:=true;
    for x := 1 to 500 do begin
        for y := 1 to 500 do begin
            //Kennzeichne duft, ameisen dann futter/nest
            if welt[x,y].b_nest=true then
image1.canvas.Pixels[x,y+offset]:=clwhite
            else begin
                if welt[x,y].b_futterquelle=true then begin
                    futter:=255-(futterm-welt[x,y].i_futter);

```

```

        if welt[x,y].i_futter>0 then
    imagel.canvas.pixels[x,y+offset]:=RGB(0,futter,0);
        end else begin
            if (welt[x,y].i_ameisen>0) or (welt[x,y].i_duftdauer>0) then
begin
            if welt[x,y].i_ameisen>0 then
a_ameisen:=welt[x,y].i_ameisen*50;
            if a_ameisen>255 then a_ameisen:=255;
            duft:=welt[x,y].i_duftstaerke*20;
            if duft>255 then duft:=255;
            imagel.canvas.Pixels[x,y+offset]:=RGB(duft,0,a_ameisen);
                end;
            end;
        end;
        end;
        end;
        end;

end;

procedure TForm2.initialisiereWelt;
var
i:integer;
rx,ry:integer;
x,y:integer;
begin
    //Ameisen
    for i := 0 to Ameisenzahl-1 do begin
        Ameise[i].x:=nestx;
        Ameise[i].y:=nesty;
    end;
    //Nest
    welt[nestx,nesty].b_nest:=true;
    //Futterquellen
    i:=0;
    repeat
        randomize;
        rx:=random(500)+1;
        ry:=random(500)+1; //Zufällige x,y Koordinaten, aber nicht am Rand
        if (welt[rx,ry].b_nest=false) and (welt[rx,ry].b_futterquelle=false)
then begin //Wenn noch kein Nest/keine Futterquelle
            welt[rx,ry].b_futterquelle:=true;
            welt[rx,ry].i_futter:=futterm;
            i:=i+1; //Eine Futterquelle mehr hinzugefügt
        end;
    until (i>=futterq); //bis alle Futterquellen hinzugefügt wurden
    //Färbe Welt schwarz
    for x := 1 to 500 do begin
        for y := 1 to 500 do begin
            canvas.Pixels[x,y+offset]:=clblack;
        end;
    end;
    //Markiere das Nest
    canvas.Pixels[nestx,nesty+100]:=clwhite;
    zeichnewelt;
end;

```