

# 34. BUNDESWETTBEWERB INFORMATIK

RUNDE 1  
01.09. - 30.11.2015

## Aufgabe 2

### Ameisenfutter

30. Oktober 2015

**Eingereicht von:** *Der Skript-Tim*

Tim Hollmann  
ich@tim-hollmann.de

**Verwaltungs-Nr.:** 34.00003

Ich versichere hiermit, die vorliegende Arbeit ohne unerlaubte fremde Hilfe entsprechend den Wettbewerbsregeln des Bundeswettbewerb Informatik angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben.

*Tim Hollmann*, den 30. Oktober 2015

## INHALTSVERZEICHNIS

<b>Inhaltsverzeichnis</b>	<b>2</b>
<b>1 Lösungsidee</b>	<b>3</b>
1.1 Interpretationen und Erweiterungen zur Aufgabenstellung . . . . .	3
1.1.1 Die Ameise . . . . .	3
1.1.2 Das Feld-Element . . . . .	3
1.1.3 Manipulation des Bewegungs-Zufalls . . . . .	3
1.1.4 Verlassen einer Pheromonspur, an deren Ende sich kein Futter be- findet . . . . .	5
1.1.5 Verdunstung der Pheromone . . . . .	5
1.1.6 Konzentration der Pheromone und Verhalten beim Treffen auf eine Pheromonspur . . . . .	5
1.2 Klassendiagramm . . . . .	6
1.3 Ameisenlogik . . . . .	6
1.4 Ablauf der Simulation . . . . .	7
<b>2 Umsetzung</b>	<b>7</b>
2.1 Konfiguration und Start der Simulation . . . . .	7
2.2 Spielrunde . . . . .	8
<b>3 Beispiele</b>	<b>9</b>
3.1 Beobachtungen . . . . .	9
<b>4 Wichtigster Quelltext</b>	<b>14</b>
<b>Abbildungsverzeichnis</b>	<b>24</b>

## 1 LÖSUNGSIDEE

Die Schwierigkeit bei dieser Aufgabe „beschränkte“ sich auf die Implementierung der Simulation in der geforderten Form.

### 1.1 INTERPRETATIONEN UND ERWEITERUNGEN ZUR AUFGABENSTELLUNG

#### 1.1.1 DIE AMEISE

Die Ameise besitzt folgende Attribute:

- X- und Y-Position (Koordinaten)
- Ladungszustand (beladen/nicht beladen)

und folgende Methoden:

- Zufällige Bewegung
- In der Umgebung nach Futter suchen und dieses ggf. aufnehmen
- In der Umgebung nach Pheromonen suchen und diesen ggf. folgen
- Zum Nest zurückkehren (wenn beladen)

#### 1.1.2 DAS FELD-ELEMENT

Ein einzelnes Feld besitzt folgende Eigenschaften:

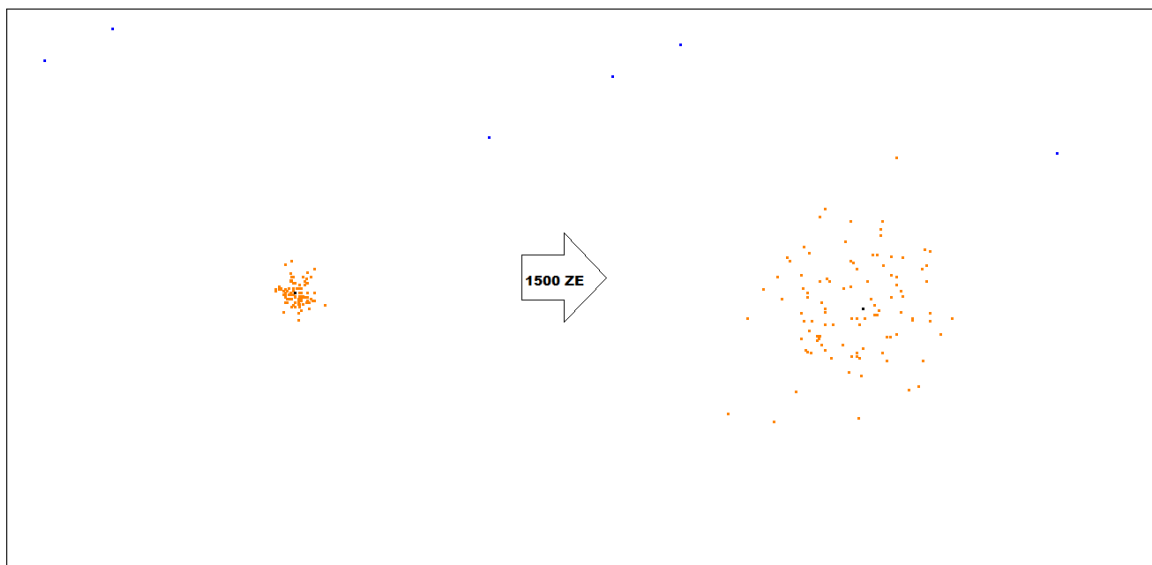
- Futterpunkte
- Duftpunkte

#### 1.1.3 MANIPULATION DES BEWEGUNGS-ZUFALLS

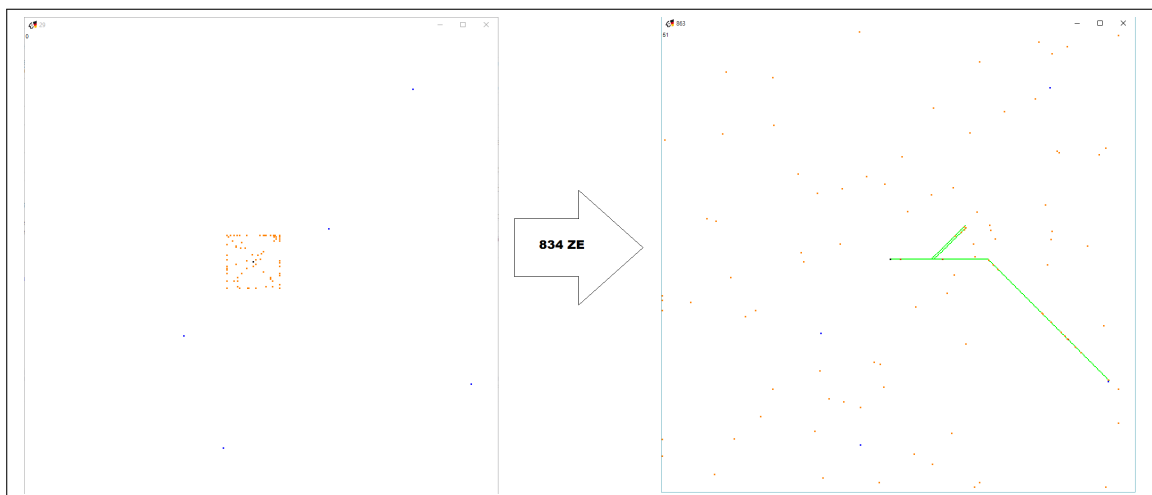
Findet die Ameise in ihrer Umgebung weder Futter noch eine Pheromonspur, so solle sie sich nach Aufgabenstellung „auf ein zufällig gewähltes“ Nachbarfeld bewegen; diese entstehende Richtung solle „ständig ändernd“ sein.

Würde man dies konsequent implementieren, so würde eine Ameise bei jeder zufälligen Bewegung eine neue Richtung zufällig wählen, also kein übergeordnetes Ziel verfolgen. Dies führt dazu, dass die Ameisen wie eine Wolke um das Nest herum wabern und sich - auch über lange Zeit - nicht signifikant über das Spielfeld verbreiten. (Abb. 1).

Man könnte sich jetzt natürlich freuen ob der Tatsache, den Zufall richtig implementiert zu haben; andererseits wirkt dies der Zeit und Qualität der Simulation erheblich entgegen. Um dies zu optimieren habe ich die Zufällige Bewegung folgendermaßen implementiert: Eine Ameise sucht sich ein entferntes - zufälliges - Ziel, zu dem sie sich zugübergreifend hinbewegt. Bei jeder Bewegung besteht eine Wahrscheinlichkeit, ihr bisheriges Ziel aufzugeben und sich ein zufälliges neues Ziel zu suchen.



**Abbildung 1:** Ausbreitung der Ameisen bei konsequent senil-zufälliger Bewegung über 1500 Zeiteinheiten. Die Ausbreitung ist stark beschränkt.



**Abbildung 2:** Ausbreitung der Ameisen bei zugübergreifender Zielsetzung über lediglich 834 Zeiteinheiten. Die Ameisen sind über das gesamte Spielfeld verbreitet und haben sogar bereits mehrfach Futter ins Nest gebracht.

Dadurch ist eine deutliche Verbreitung erkennbar und erreicht, die die Zufälligkeit der Bewegung nicht beschränkt. (Abb. 2).

Dafür erhält die jede Ameise ein weiteres Attribut: `target_x` und `target_y`, die die Koordinaten des temporären Bewegungszieles beinhalten. Beim Beginn einer neuen Simulation hat dies den Effekt, dass sich die Ameisen - zunächst - beinahe Quadratisch in alle Richtungen ausbreiten.

#### 1.1.4 VERLASSEN EINER PHEROMONSPUR, AN DEREN ENDE SICH KEIN FUTTER BEFINDET

Wenn sich ein Transportweg etabliert, so bleibt er durch seine hohe Pheromonkonzentration auch dann bestehen, wenn die korrespondierende Futterquelle bereits vollständig abgebaut wurde (siehe Beobachtungen, **x**). Die Ameisen befinden sich dann also am Ende einer Pheromonspur, an der es kein Futter mehr gibt. Damit sie diese dann verlassen, speichert jede Ameise die letzten 5 Felder, denen sie aufgrund ihrer Pheromonkonzentration gefolgt ist und ignoriert diese bei der Wahl des zu folgenden Pheromonfeldes im nächsten Zug. Dadurch bleibt am Ende der Spur kein Pheromonfeld mehr übrig und sie verlässt die Spur mit einer zufälligen Bewegung.

Dazu erweitere ich die Ameisen um das Array-Attribut `last_pheromone_fields`.

#### 1.1.5 VERDUNSTUNG DER PHEROMONE

Laut Aufgabenstellung sollen die abgesonderten Pheromone nach einer gewissen Zeit verdunsten. Es hat sich bei der Implementierung allerdings als einfacher herausgestellt, - anstatt eine Art „Verfallsdatum“ für jeden Duftpunkt zu erstellen - bei jeder Zeiteinheit einen Duftpunkt abzuziehen und bei der Absonderung entsprechend um so viele Punkte zu erhöhen, wie viele Zeiteinheiten das Pheromon brauchen soll, um zu verdunsten. Dies hat keinen Einfluss auf den Simulationsablauf, würde aber streng gesehen nicht der Aufgabenstellung entsprechen.

#### 1.1.6 KONZENTRATION DER PHEROMONE UND VERHALTEN BEIM TREFFEN AUF EINE PHEROMONSPUR

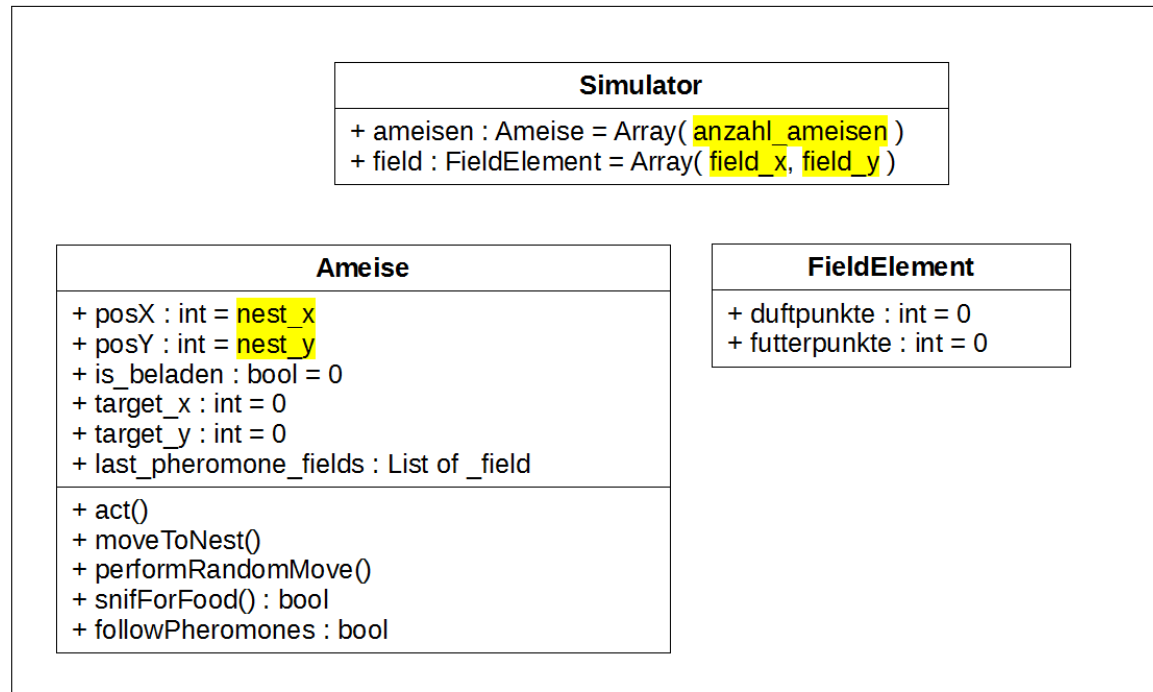
Die Aufgabenstellung verlangt, dass sich eine Ameise beim Treffen auf eine Pheromonspur auf dasjenige mit der „stärksten Konzentration weiter weg vom Nest“ bewegt. Auf meine Frage im EI-Community-Forum bezüglich des Verhältnisses zwischen Entfernung und Konzentration bei der Entscheidung, welchem Feld man eher folgen würde, erhielt ich bis jetzt noch keine Antwort. Ich teile die Interpretation von Pascal Otto<sup>1</sup>: Trifft eine Ameise auf ein Pheromon, hat zunächst die Entfernung zum Nest höhere Priorität für die Entscheidung. Entfernen sich zwei angrenzende Felder vom Nest (offenbar eine Verzweigung / andere Spur), dann werden zusätzlich deren Konzentrationen verglichen. Wie sich herausgestellt hat (Beobachtungen, **xxx**), ist das Beachten der Pheromonkonzentration

---

<sup>1</sup><http://www.einstieg-informatik.de/community/forums/topic/386/frage-zur-auf-logik-der-aufgabe-nr-2-ameinse>

eher hinderlich als hilfreich. Da es aber von der Aufgabenstellung so explizit gefordert ist, habe ich die Beachtung der Pheromonkonzentration beim Folgen einer Pheromonspur als deaktivierbar implementiert.

## 1.2 KLASSENDIAGRAMM



**Abbildung 3:** Klassendiagramm des Simulators, der Ameise und des Feldelementes.

(Abb.3). Alle bei der Konfiguration der Simulation eingestellten Werte sind hier gelb hinterlegt (Spielfeldgröße, Ameisenanzahl usw.).

## 1.3 AMEISENLOGIK

Durch die Aufgabenstellung ergibt sich die folgende Logik für die Ameise: (1.3).

```
1  ' Beladen?  
2  If is_beladen Then  
3      ' Im Nest?  
4      If posX = nest_x And posY = nest_y Then  
5          is_beladen = False ' Futter ablegen  
6      Else  
7          moveToNest()  
8      End If  
9  Else  
10     ' Futter oder Pheromone in der Nähe?  
11     If Not sniffForFood() And Not followPheromones() Then  
12         ' Sonst zufällige Bewegung ausführen  
13         performRandomMove()  
14     End If  
15 End If
```

#### 1.4 ABLAUF DER SIMULATION

Die Simulation ist - ähnlich wie im Turniersystem - in Spielrunden aufgeteilt. In jeder Spielrunde („Zeiteinheit“) verdunsten zuerst die Pheromone und dann handeln die Ameisen entsprechend ihrer Logik.

Die entsprechende Methode (`Simulator::tick()`) wird periodisch von einem Timer aufgerufen.

## 2 UMSETZUNG

Die Umsetzung erfolgte in Visual Basic mit Zielframework .NET 4.5.2.

### Lautzeitumgebung

Sie benötigen das .NET Framework in der Version 4.5.2

#### 2.1 KONFIGURATION UND START DER SIMULATION

Vor den Beginn der Simulation ist das Konfigurationsformular (Abb. 4) gestellt, in dem der Benutzer die Simulation nach seinen Wünschen anpassen kann. Einstellbar sind unter anderem

- Größe des Spielfeldes (Breite und Höhe)
- **Position des Nests** (Koordinaten)
- Farben der Ameisen, des Nests, der Pheromone und des Futters
- **Anzahl und Kapazität der Futterquellen**
- **Anzahl der Ameisen**

- **Verdunstungszeit der Pheromone**
- Bewegungsmanipulation (Wahrscheinlichkeit der Richtungsänderung und Suchradius des temporären Ziels)
- Die Geschwindigkeit der Simulation
- Darstellung und Vergrößerung der Felder und Elemente

(Die von der Aufgabenstellung geforderten einstellbaren Werte sind Fett gedruckt).

**Abbildung 4:** Simulations-Konfigurationsformular

Die eingegebenen Daten werden dann über ein Modul auf eine neue Instanz der Simulations-Form übergeben, welche als Zeichenfläche dient. Zudem startet ein Timer auf der Simulations-Form, der in dem Intervall tickt, das man auf der Konfigurations-Form eingegeben hat. Die Größe der Simulations-Form passt sich automatisch der eingestellten Spielfeld-Größe in Verbindung mit der proportionalen Vergrößerung an. (Wird sie dabei größer als der Bildschirm, kann man dies durch eine kleinere Vergrößerung beheben.)

## 2.2 SPIELRUNDE

In jeder Spielrunde verdunsten zunächst die Pheromone; diejenigen Felder, deren Duftpunkt-Anzahl  $> 0$  ist, werden dekrementiert;



```
1 ' Pheromone verdunsten
2 For Each f As FieldElement In field
3     If f.duftpunkte > 0 Then
4         f.duftpunkte -= 1
5     End If
6 Next
```

Daraufhin handeln die Ameisen;

```
1 ' Ameisen-Handlungen
2 For Each a As Ameise In ameisen
3     a.act()
4 Next
```

und das Spielfeld wird gezeichnet

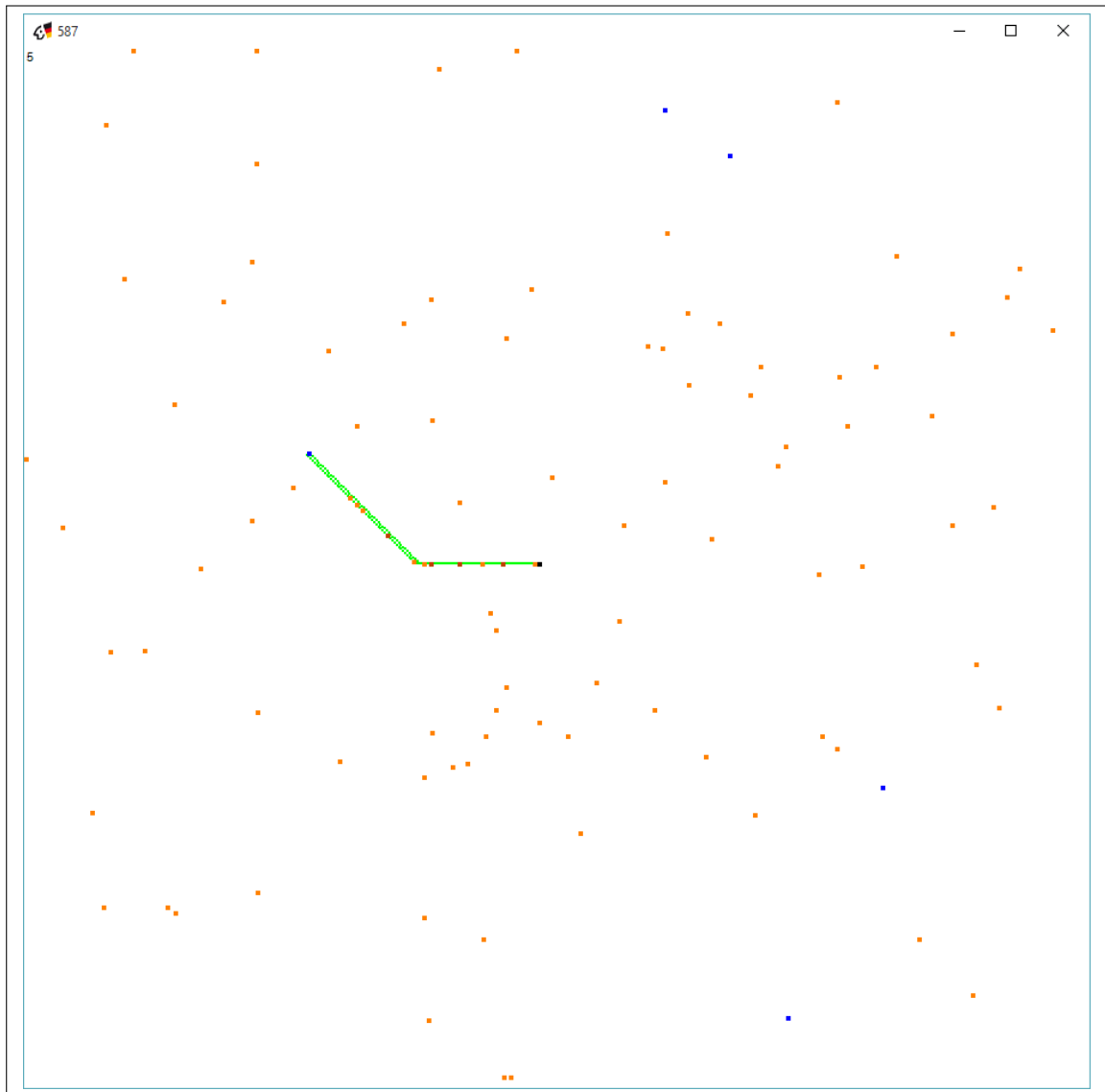
```
1 drawField()
```

Dabei werden Ameisen, Pheromome, Futterstellen und Nest als Quadrat eingestellter Größe dargestellt. (Abb. 5)

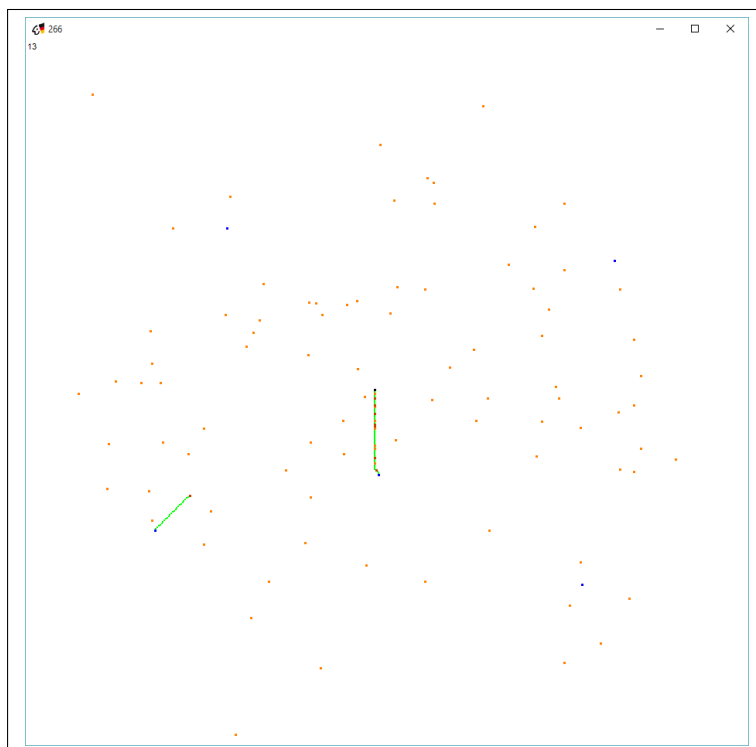
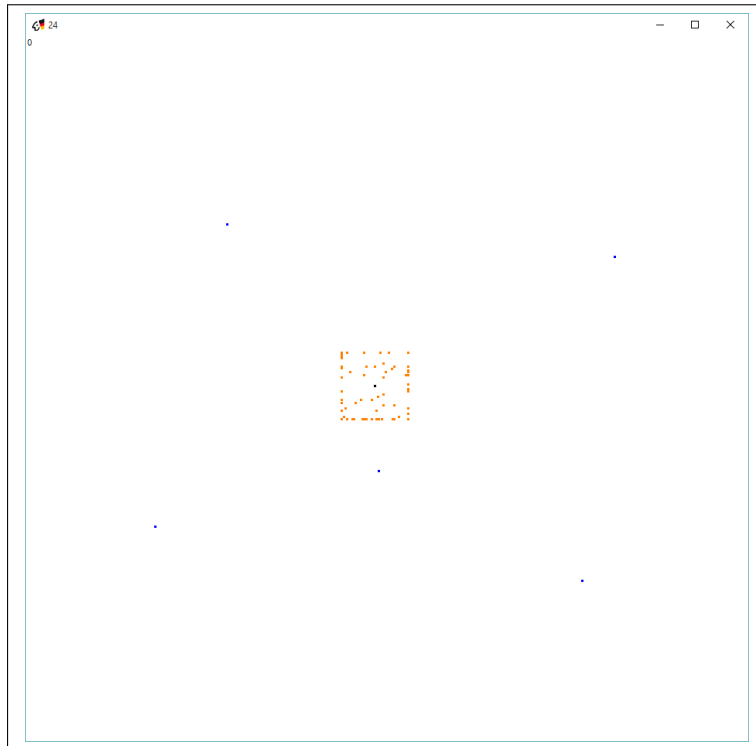
### 3 BEISPIELE

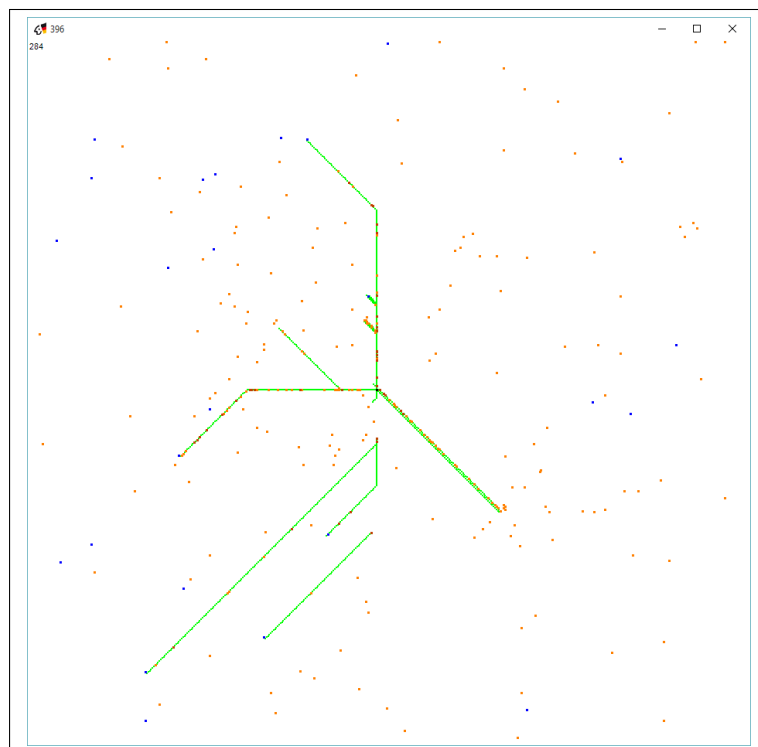
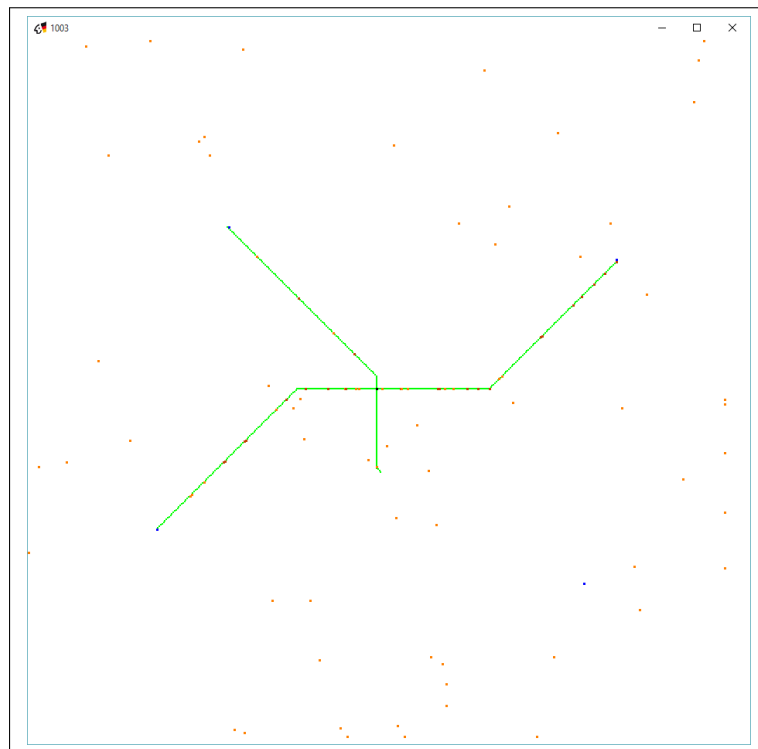
#### 3.1 BEOBACHTUNGEN

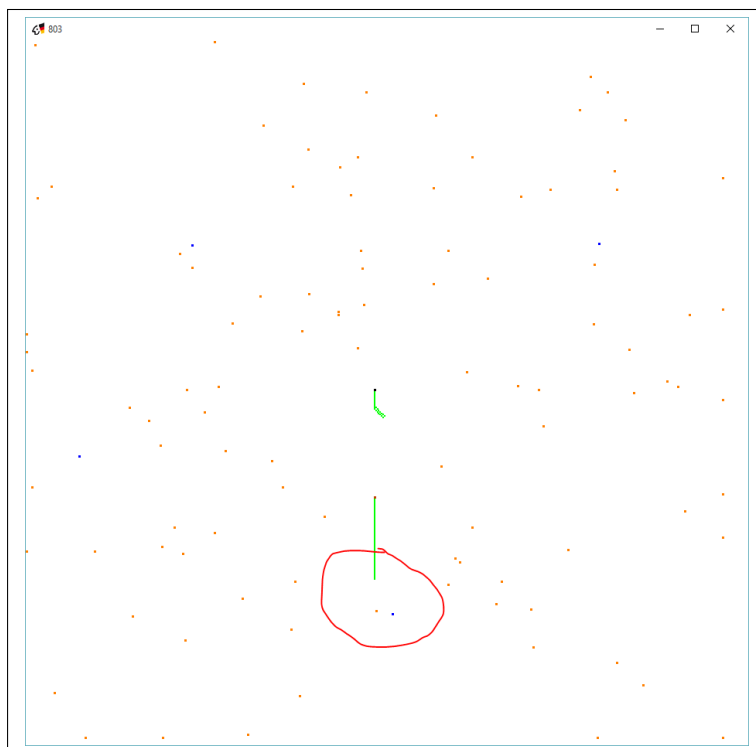
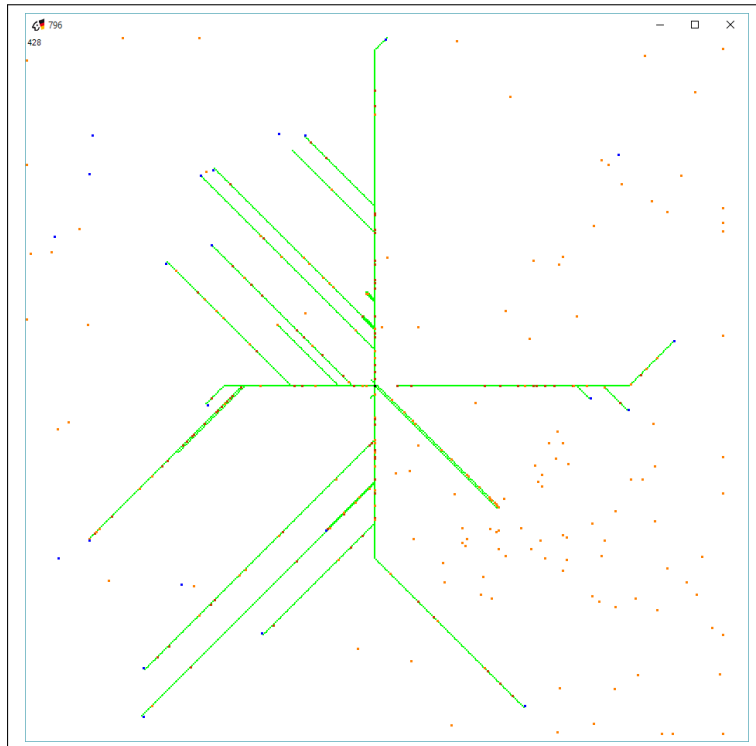
1. Bei kurzen Verdunstungszeiten können Transportketten von einer Futterquelle zu einem Nest nur dann aufrecht erhalten werden, wenn innerhalb der kurzen Verdunstungszeit es eine zweite Ameise schafft, der Spur bis zur Futterquelle zu folgen und ihrerseits Pheromone abzusondern
2. Bei langen oder auch schon gehäuft bei mittellangen Verdunstungszeiten ( > 100 ZE) bleiben Transportketten auch über den Zeitpunkt hinaus bestehen, bei dem die korrespondierende Futterquelle abgebaut worden ist. Die Pheromone locken die Ameisen in eine potenziell falsche Richtung.
3. Beachtet die Ameise bei ihrer Verfolgung der Pheromone deren Konzentration, verfolgt sie nie eine „frische“ mit geringerer Konzentration. Und bis der etablierte Transportweg abgebaut worden ist, vergehen in der Startkonfiguration sogar  $50 * 1 * \text{Weglänge} * \text{Verdunstungszeit}$  Zeiteinheiten. Deshalb halte ich die Betrachtung der Konzentration als nicht sinnvoll.
4. Bei wenigen Ameisen: insgesamt längere Simulationszeit, da Futterquellen schwerer gefunden werden und falls gefunden, ein Transportweg zum Nest nur sehr schlecht aufrecht erhalten werden kann, sofern die Verdunstungszeit der Pheromone nicht erheblich hoch ist.
5. Bei sehr vielen Ameisen kann in sehr kurzer Zeit sehr viel und weit entferntes Futter ins Nest gebracht werden.



**Abbildung 5:** Simulations-Form. Die Objekte werden in eingestellter Farbe und Größe als Quadrate dargestellt.







6. Nestposition: die zentrale Lage des Nestes in der Aufgabenstellung verkürzt den Weg vom Nest zu den Futterquellen im Durchschnitt. Wird das Nest aus dem Zentrum entfernt, vergrößert sich der Abstand zu einigen Futterquellen, die Ameisen brauchen längere Zeit, Futter zu finden und es wird schwieriger, auf diese Distanz eine Transportkette aufrecht erhalten zu können.
7. Mit zunehmender Feldgröße wird es unmöglich, eine Transportkette aufrecht zu erhalten. Die Zeit, Futter zufällig zu finden, steigt zudem stark an.

## 4 WICHTIGSTER QUELLTEXT

```

1 Public Class Form1
2
3     Public locked As Boolean = False
4     Public ende As Boolean = False
5
6     Private Sub Form1_Activated(sender As Object, e As EventArgs) ←
7         Handles Me.Activated
8
9         ' Fenstergröße anpassen
10        Me.Size = New System.Drawing.Size(field_display_size * field_x ←
11            + manipulation_window, field_display_size * field_y + ←
12            manipulation_window)
13        ' Timer starten
14        Timer1.Enabled = True
15
16    End Sub
17
18    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles ←
19        Me.Load
20        eingabe.ShowDialog()
21
22        ' Feld mit neuen Zuständen füllen
23        For x As Integer = 1 To field.GetLength(0)
24            For y As Integer = 1 To field.GetLength(1)
25                field(x - 1, y - 1) = New FieldElement
26            Next
27        Next
28
29        ' Ameisen erschaffen
30        For a As Integer = 0 To ameisen.Count - 1
31            ameisen(a) = New Ameise
32        Next
33
34        ' Futter platzieren
35        For i As Integer = 1 To futterquellen_anzahl
36            field(zufall.Next(0, field_x), zufall.Next(0, ←
37                field_y)).futterpunkte += futterquellen_kapazitaet
38        Next

```

```
35     ' Timer-Intervall anpassen
36     Timer1.Interval = timer_interval
37
38 End Sub
39
40 Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles ←
41     Timer1.Tick
42     If locked Or ende Then Return
43     locked = True
44
45     ' Simulation zu Ende?
46     If nest_futter_counter = futterquellen_anzahl * ←
47         futterquellen_kapazitaet Then
48             ende = True
49
50             MsgBox("Die Simulation ist beendet; alle " & ←
51                 (futterquellen_anzahl * futterquellen_kapazitaet) & " ←
52                 Futterpunkte wurden ins Nest gebracht." & vbCrLf & ←
53                 "Es wurden " & Me.Text & " Zeiteinheiten benötigt.")
54
55 End If
56
57 ' Ameisen-Handlungen
58 For Each a As Ameise In ameisen
59     a.act()
60 Next
61
62 ' Pheromone verdunsten
63 For Each f As FieldElement In field
64     If f.duftpunkte > 0 Then
65         f.duftpunkte -= 1
66     End If
67 Next
68
69 ' Zeichnen
70 drawField()
71
72 ' Frames
73 Me.Text += 1
74
75 locked = False
76 End Sub
77
78 ' Spielfeld zeichnen
79 Private Sub drawField()
80
81     Dim formGraphics As System.Drawing.Graphics
82     formGraphics = Me.CreateGraphics()
83
84     ' Spielfeld löschen
85     formGraphics.FillRectangle(stift_radieren, New Rectangle(0, 0, ←
86         field_display_size * field_x + manipulation_window, ←
87         field_display_size * field_y + manipulation_window))
88
89 End Sub
```

```

81 ' Pheromone zeichnen
82 For x As Integer = 0 To field.GetLength(0) - 1
83     For y As Integer = 0 To field.GetLength(1) - 1
84         If field(x, y).duftpunkte > 0 Then
85             formGraphics.FillRectangle(stift_pheromone, New Rectangle(x * field_display_size, y * field_display_size, field_display_size, field_display_size))
86         End If
87     Next
88 Next
89
90 ' Ameisen zeichnen
91 For Each a As Ameise In ameisen
92     Dim brush As System.Drawing.SolidBrush
93
94     If a.is_beladen Then
95         ' Beladene Ameise
96         brush = stift_ameise_beladen
97     Else
98         ' Normale Ameise
99         brush = stift_ameise_normal
100    End If
101    formGraphics.FillRectangle(brush, New Rectangle(a.posX * field_display_size, a.posY * field_display_size, zoom_verzerrend * field_display_size, zoom_verzerrend * field_display_size))
102 Next
103
104 ' Nest zeichnen
105 formGraphics.FillRectangle(stift_nest, New Rectangle(nest_x * field_display_size, nest_y * field_display_size, zoom_verzerrend * field_display_size, zoom_verzerrend * field_display_size))
106
107 ' Futterpunkte zeichnen
108 For x As Integer = 0 To field.GetLength(0) - 1
109     For y As Integer = 0 To field.GetLength(1) - 1
110         If field(x, y).futterpunkte > 0 Then
111             formGraphics.FillRectangle(stift_futter, New Rectangle(x * field_display_size, y * field_display_size, zoom_verzerrend * field_display_size, zoom_verzerrend * field_display_size))
112         End If
113     Next
114 Next
115
116 ' Ausgabe von Informationen
117 Dim drawBrush As New SolidBrush(Color.Black)
118 formGraphics.DrawString(nest_futter_counter.ToString, New Font("Arial", 8), drawBrush, New Rectangle(0, 0, 8 * 10, 20 * 8))


```



```

119         formGraphics.Dispose()
120
121     End Sub
122
123     ' Simulation bei Klick auf Simulator-Fenster pausieren
124     Private Sub Form1_Click(sender As Object, e As EventArgs) Handles ←
125         Me.Click
126         Timer1.Enabled = Not Timer1.Enabled
127     End Sub
128 End Class

```

Listing 1:  2\src\Form1.vb - Form1.vb (Simulator-Form)

```

1  Module Module1
2
3      Public field_x As Integer
4      Public field_y As Integer
5
6      Public nest_x As Integer
7      Public nest_y As Integer
8
9      Public futterquellen_anzahl As Integer
10     Public futterquellen_kapazitaet As Integer
11
12     Public anzahl_ameisen As Integer
13     Public ameisen_movement_target_range As Integer
14     Public ameisen_wahrscheinlichkeit_richtungsaenderung As Double
15     Public pheromoneAnzahl As Integer
16
17     Public field_display_size As Double
18     Public zoom_verzerrend As Double
19
20     Public timer_interval As Integer
21
22     ' Farben
23     Public farbe_ameise_normal As System.Drawing.Color
24     Public farbe_ameise_beladen As System.Drawing.Color
25     Public farbe_nest As System.Drawing.Color
26     Public farbe_pheromone As System.Drawing.Color
27     Public farbe_futter As System.Drawing.Color
28
29     ' Stifte
30     Public stift_ameise_normal As System.Drawing.SolidBrush
31     Public stift_ameise_beladen As System.Drawing.SolidBrush
32     Public stift_nest As System.Drawing.SolidBrush
33     Public stift_pheromone As System.Drawing.SolidBrush
34     Public stift_futter As System.Drawing.SolidBrush
35     Public stift_radieren As System.Drawing.SolidBrush
36
37     ' Laufzeitvariablen
38     Public ameisen() As Ameise
39     Public field(,) As FieldElement

```

```
40
41 Public nest_futter_counter As Integer = 0
42 Public zufall As New Random
43
44 Public manipulation_window As Double = 50
45
46 Public beachte_konzentration As Boolean = True
47
48 End Module
49
50 Public Class Ameise
51
52 Public posX As Integer = nest_x
53 Public posY As Integer = nest_y
54
55 Public is_beladen As Boolean
56
57 Public target_x As Integer = 0
58 Public target_y As Integer = 0
59
60 Public last_pheromone_fields As New List(Of _field)
61
62 Public Sub act()
63     ' Ameisen-Logik
64
65     If is_beladen Then
66         If posX = nest_x And posY = nest_y Then
67             is_beladen = False
68             nest_futter_counter += 1
69             last_pheromone_fields.Clear()
70         Else
71             moveToNest()
72             last_pheromone_fields.Clear()
73         End If
74     Else
75         If Not sniffForFood() And Not followPheromones() Then
76             performRandomMove()
77         End If
78     End If
79
80 End Sub
81
82 ' Bewegung zum Nest zurück (wenn beladen)
83 Public Sub moveToNest()
84     ' Pheromonpunkte absetzen und zum Nest bewegen
85     field(posX, posY).duftpunkte += pheromoneAnzahl
86
87     If posX > nest_x And posX > 0 Then
88         posX -= 1
89     ElseIf posX < nest_x
90         posX += 1
91     End If
92
```

```
93     If posY > nest_y And posY > 0 Then
94         posY -= 1
95     ElseIf posY < nest_y
96         posY += 1
97     End If
98
99 End Sub
100
101 ' Zufällige Bewegung ausführen
102 Public Sub performRandomMove()
103
104     ' Temporäres Ziel
105     If (target_x = 0 And target_y = 0) Or (posX = target_x And ←
        posY = target_y) Or zufall.Next(1, 101) / 100 < ←
        ameisen_wahrscheinlichkeit_richtungsänderung Then
106         ' Neues temporäres Ziel
107         target_x = posX + ←
            zufall.Next(-ameisen_movement_target_range, ←
            ameisen_movement_target_range + 1)
108         target_y = posY + ←
            zufall.Next(-ameisen_movement_target_range, ←
            ameisen_movement_target_range + 1)
109     End If
110
111     ' Ziel verfolgen
112     If posX < target_x Then
113         If posX < field.GetLength(0) - 1 Then
114             posX += 1
115         End If
116     ElseIf posX > target_x
117         If posX > 0 Then
118             posX -= 1
119         End If
120     End If
121
122     If posY < target_y Then
123         If posY < field.GetLength(1) - 1 Then
124             posY += 1
125         End If
126     ElseIf posY > target_y
127         If posY > 0 Then
128             posY -= 1
129         End If
130     End If
131 End Sub
132
133 ' Umliegende Felder auf Futter untersuchen
134 Public Function sniffForFood()
135
136     ' Links
137     If posX > 0 Then
138
139         If posY > 0 Then
```

```
140         ' Links Oben: posX-1|posY-1
141         If field(posX - 1, posY - 1).futterpunkte > 0 Then
142             field(posX - 1, posY - 1).futterpunkte -= 1
143             is_beladen = True
144             Return True
145         End If
146     End If
147
148     ' Links Mitte: posX-1|posY
149     If field(posX - 1, posY).futterpunkte > 0 Then
150         field(posX - 1, posY).futterpunkte -= 1
151         is_beladen = True
152         Return True
153     End If
154
155     If posY < field_y Then
156         ' Links unten: posX-1|posY+1
157         If field(posX - 1, posY + 1).futterpunkte > 0 Then
158             field(posX - 1, posY + 1).futterpunkte -= 1
159             is_beladen = True
160             Return True
161         End If
162     End If
163
164 End If
165
166 ' Mitte
167 If posY > 0 Then
168     ' Mitte Oben
169     If field(posX, posY - 1).futterpunkte > 0 Then
170         field(posX, posY - 1).futterpunkte -= 1
171         is_beladen = True
172         Return True
173     End If
174 End If
175
176 If posY < field_y Then
177     ' Mitte unten
178     If field(posX, posY + 1).futterpunkte > 0 Then
179         field(posX, posY + 1).futterpunkte -= 1
180         is_beladen = True
181         Return True
182     End If
183 End If
184
185 ' Rechts
186 If posX < field_x Then
187     If posY > 0 Then
188         ' Rechts oben
189         If field(posX + 1, posY - 1).futterpunkte > 0 Then
190             field(posX + 1, posY - 1).futterpunkte -= 1
191             is_beladen = True
192             Return True
```

```

193         End If
194     End If
195
196     ' Rechts mitte
197     If field(posX + 1, posY).futterpunkte > 0 Then
198         field(posX + 1, posY).futterpunkte -= 1
199         is_beladen = True
200         Return True
201     End If
202
203     ' Rechts unten
204     If posY < field_y Then
205         If field(posX + 1, posY + 1).futterpunkte > 0 Then
206             field(posX + 1, posY + 1).futterpunkte -= 1
207             is_beladen = True
208             Return True
209         End If
210     End If
211
212     End If
213
214     Return False
215 End Function
216
217 ' Umliegende Felder auf Pheromone untersuchen und evtl. verfolgen
218 Public Function followPheromones()
219
220     Dim moeglichkeiten As New List(Of _field)
221     Dim entfernung_current_field As Double = <←
222     Math.Abs(Math.Sqrt((nest_x - (posX)) ^ 2 + (nest_y - <←
223     (posY)) ^ 2))
224
225     Dim konzentration_max As Integer = 0
226
227     For dx As Integer = -1 To 1
228         For dy As Integer = -1 To 1
229             ' Nicht aktuelles Feld
230             If Not (dx = 0 And dy = 0) Then
231                 ' Feld erlaubt?
232                 If posX + dx > 0 And posX + dx < field_x - 1 And <←
233                 posY + dy > 0 And posY + dy < field_y - 1 Then
234
235                     ' Nicht in letzten 5 besuchten Pheromonfeldern
236                     Dim f As Boolean = False
237                     For Each a In last_pheromone_fields
238                         If a.dx = posX + dx And a.dy = posY + dy Then
239                             f = True
240                         End If
241                     Next
242
243                     ' Mindestens ein Duftpunkt
244                     If field(posX + dx, posY + dy).duftpunkte > 0 <←
245                     And Not f Then


```

```

242
243         Dim entfernung_temp As Double = <←
                Math.Abs(Math.Sqrt((nest_x - (posX + <←
                dx)) ^ 2 + (nest_y - (posY + dy)) ^ 2))
244         Dim konzentration As Integer = field(posX <←
                + dx, posY + dy).duftpunkte / <←
                pheromoneAnzahl
245
246         ' Bedingung: Muss vom Nest wegführen
247         If entfernung_temp > <←
                entfernung_current_field Then
248             moeglichkeiten.Add(New _field(dx, dy, <←
                konzentration))
249
250             If konzentration > konzentration_max Then
251                 konzentration_max = konzentration
252             End If
253
254         End If
255     End If
256
257     End If
258 End If
259 Next
260 Next
261
262 ' Dasjenige wegführende Feld mit der höchsten Konzentration <←
    ermitteln (sofern dies aktiviert ist)
263
264 Dim moeglichkeiten_final As New List(Of _field)
265
266 If beachte_konzentration Then
267     For i As Integer = 0 To moeglichkeiten.Count - 1
268         If moeglichkeiten(i).duftpunkte >= konzentration_max Then
269             moeglichkeiten_final.Add(moeglichkeiten(i))
270         End If
271     Next
272 Else
273     ' Pheromonkonzentration nicht beachten; Möglichkeiten <←
        bleiben unberührt
274     moeglichkeiten_final = moeglichkeiten
275 End If
276
277 ' Eine zufällige Entscheidung wählen
278 If moeglichkeiten_final.Count > 0 Then
279     Dim satzRandom As _field = <←
        moeglichkeiten_final(zufall.Next(0, <←
        moeglichkeiten_final.Count))
280
281     ' Bewegen
282     posX += satzRandom.dx
283     posY += satzRandom.dy
284
285     If last_pheromone_fields.Count > 4 Then

```



```
285         last_pheromone_fields.RemoveAt(0)
286     End If
287
288     last_pheromone_fields.Add(New _field(posX, posY, 0))
289
290     Return True
291 End If
292
293 Return False
294 End Function
295
296 Public Class _field
297     Public dx As Integer
298     Public dy As Integer
299     Public duftpunkte As Integer
300
301     Public Sub New(ByVal x As Integer, ByVal y As Integer, ByVal k ←
302         As Integer)
303         dx = x
304         dy = y
305         duftpunkte = k
306     End Sub
307 End Class
308
309 End Class
310
311 Public Class FieldElement
312
313     Public duftpunkte As Integer = 0
314     Public futterpunkte As Integer = 0
315
316 End Class
```

**Listing 2:**  2>src>Module1.vb - Module1.vb (Globale Klassen und Konfiguration)

## ABBILDUNGSVERZEICHNIS

1	Ausbreitung der Ameisen bei konsequent senil-zufälliger Bewegung über 1500 Zeiteinheiten. Die Ausbreitung ist stark beschränkt. . . . .	4
2	Ausbreitung der Ameisen bei zugübergreifender Zielsetzung über lediglich 834 Zeiteinheiten. Die Ameisen sind über das gesamte Spielfeld verbreitet und haben sogar bereits mehrfach Futter ins Nest gebracht. . . . .	4
3	Klassendiagramm des Simulators, der Ameise und des Feldelementes. . . .	6
4	Simulations-Konfigurationsformular . . . . .	8
5	Simulations-Form. Die Objekte werden in eingestellter Farbe und Größe als Quadrate dargestellt. . . . .	10

## LISTINGS

1	 2▸src▸Form1.vb - Form1.vb (Simulator-Form) . . . . .	14
2	 2▸src▸Module1.vb - Module1.vb (Globale Klassen und Konfiguration) . . . .	17