

A photograph of a turtle in a pond. The turtle is on the left, partially submerged in the water. The water is dark and reflects the surrounding greenery. Large, vibrant green leaves are prominent in the foreground and background, creating a lush, natural setting. The overall scene is bright and detailed, suggesting a high-quality digital simulation.

Computerbasierte Ökosystem Simulation

Regelwerk [version 1.0]

Lukas Henning (11b)
Kreisgymnasium Bad Krozingen
NwT - Halbjahresprojekt 2009

Computerbasierte Ökosystem-Simulation

Regelwerk v.1.0

Inhalt

1.0 Einführung	3
1.1 Funktionsumfang der Simulation	3
1.2 Grundlegende Regeln der Simulation	4
2.0 Die Parameter der Biofaktoren	4
<i>Umwelt</i>	<i>4</i>
<i>Topos/Destruenten</i>	<i>4</i>
<i>Produzenten</i>	<i>5</i>
<i>P-Konsumenten</i>	<i>5</i>
<i>S-Konsumenten</i>	<i>6</i>
3.0 Ablauf der Simulation und Algorithmen	7
<i>Produzenten</i>	<i>7</i>
<i>P-Konsumenten</i>	<i>8</i>
<i>S-Konsumenten</i>	<i>8</i>
<i>Tod</i>	<i>8</i>
<i>Fortpflanzung</i>	<i>9</i>
<i>Nahrungssuche</i>	<i>9</i>
4.0 Zur Computerumsetzung	10

1.0 Einführung

Dieses Simulations-System hat die Aufgabe, grundlegende Abläufe innerhalb eines Ökosystems zu simulieren und darzustellen. Es ermöglicht dem Benutzer, alle Parameter eines Ökosystems zu jedem beliebigen Zeitpunkt zu verändern und die Folgen seiner Eingriffe durch grafisches und textbasiertes Feedback schnell sehen und verstehen zu können.

Dabei geht die CÖS (Computerbasierte-Ökosystem-Simulation) ausschließlich auf die grundlegendsten Vorgänge ein und vereinfacht sie in einem maximal einfachen Simulationsmodell:

Die Zeit wird in feste Abschnitte, Runden geteilt; man unterscheidet ausschließlich zwischen 5 Ökofaktoren (Umweltbedingungen, Topos/Destruenten, Produzenten, Primär-Konsumenten, Sekundär-Konsumenten) und das Biotop ist in gleiche, quadratische Felder unterteilt.

Abseits dieser 3 festen Grundprinzipien sind noch diverse andere Vereinfachungen eingebaut worden, um die Entwicklung für eine einzelne Person mit begrenzter Zeit überhaupt möglich zu machen; so brauchen Lebewesen in der Simulation bspw. nicht extra einen komplexen Paarungsvorgang zu durchlaufen, um sich fortzupflanzen.

Durch die Fokussierung auf die einfachsten Grundprinzipien soll in der CÖS auch der Energiekreislauf besonders gut für den Nutzer durchschaubar sein.

Dieses Regelwerk ist dabei (noch) nicht auf das Computerprogramm bezogen, das basierend auf diesen Regeln entwickelt werden wird, aber noch einer mehr auf die programmiererische Umsetzung basierte Planung bedarf.

Theoretisch könnte man mit den zur Zeit aufgestellten Simulationsregeln also schon eine (ausschließlich) per Hand ausgerechnete und geführte Simulation laufen lassen, nur wäre das sehr aufwändig, ermüdend und fehleranfällig.

Auf die Computer-Umsetzung gehe ich am Ende dieses Dokuments kurz ein.

1.1 Funktionsumfang der Simulation

- Rundenbasierte Simulation
- 5 Ökofaktoren (Umweltbedingungen, Topos, Produzenten, P-Konsumenten, S-Konsumenten), die sich aufeinander auswirken
- Simulation eines Topos, unterteilt in „Planquadrante“ mit 5 Höhenstufen
- Simulation von Photosynthese, Nahrungsaufnahme und Verarbeitung (Nahrungs- und Energiekreislauf innerhalb der Population(en))
- Simulation von Fortpflanzung
- Simulation von (einfacher) Intelligenz und Jagdverhalten
- Stark vereinfachte Simulation des Selektionsprinzips (die Eigenschaften von Lebewesen können sich bei ihrer Entstehung aufgrund von (in der Simulation zufallsbedingter) „genetischer Mutation“ mit den Generationen leicht verändern; die Auswirkungen der genetischen Veränderungen auf den Erfolg einer Spezies macht die Simulation dann deutlich)

1.2 Grundlegende Regeln der Simulation

- Die Simulation läuft in Runden ab
- Das „Spielfeld“ ist in gleiche Quadrate unterteilt, der Spielfeldrand darf nicht überschritten werden
- Auf einem Quadrat kann sich gleichzeitig nur ein Organismus befinden, ein Organismus darf also weder auf einem bereits besetzten Feld entstehen, noch sich dorthin bewegen.
- In jeder Runde müssen die festgelegten Algorithmen (je nach Organismotyp) für jeden Organismus in der Simulation ausnahmslos durchschritten werden, dabei werden die Organismen in der Reihenfolge „Produzenten -> P-Konsumenten -> S-Konsumenten“ einzeln durchgegangen.
- Für jeden entstehenden Organismus muss ein entsprechender Eintrag nach Vorgabe der „Parameterlisten“ in der Simulationsdatenbank angefertigt werden. Stirbt hingegen ein Organismus ab, muss sein Eintrag gelöscht werden.

2.0 Die Parameter der Biofaktoren

Die Simulation behält stets die Parameter der Umwelt, der einzelnen Planquadrate (des Topos) und der einzelnen (lebenden) Lebewesen in Listen gespeichert um bei der Berechnung verschiedener Prozesse zwischen den Runden auf sie zugreifen zu können.

Dabei gibt es statische und variable (grün gekennzeichnet) Parameter.

Die Listen werden stets mit folgenden Parametern geführt:

1. Umwelt

- Sonneneinstrahlung (0-? E) (in Energieeinheit: E)
- Wasserstand (steht indirekt auch für Niederschlag/Verdunstung) (1-5)

2. Topos/Destruenten

- Höhenstufe des Planquadrats (1-5)
- Destruentenpopulation (0-6) (steht indirekt für Nährstoffgehalt des Bodenabschnitts)
- Position X
- Position Y

3. Produzenten

- Art (beliebiger Name, wird bei der Erschaffung vom Nutzer festgelegt)
- **Generation (Zahlenwert)** (steigt mit jeder Fortpflanzung innerhalb der Art)
- Mutationsfreudigkeit (1 - 99%) (bestimmt bei der Fortpflanzung, mit welcher Wahrscheinlichkeit Mutationen auftreten können)
- Lebenserwartung (0-99) (in Runden)
- Energiebedarf (10-50 E)
- Energiebedarf (zur Einleitung einer Fortpflanzung) (10-50 E)
- Nährstoffbedarf (0-6) (Anzahl Destruenten im Boden)
- Fortpflanzung (1-4) (in Anzahl an Nachkommen)
- Leben unter/über Wasser (0; 1; 2) (0: nur unter Wasser lebensfähig; 1: nur über Wasser lebensfähig; 2: überall lebensfähig)
- Kompensationspunkt (0-99 E) (Sonneneinstrahlung)
- Lichtsättigung (0-99 E) (Sonneneinstrahlung)
- Photosynthese-Effektivitätsfaktor (1 - 99 %)
- **Energiestand (in E)**
- **Alter (in Runden)**
- **Position X**
- **Position Y**

4. P-Konsumenten

- Art (beliebiger Name, wird bei der Erschaffung vom Nutzer festgelegt)
- **Generation (Zahlenwert)** (steigt mit jeder Fortpflanzung innerhalb der Art)
- Mutationsfreudigkeit (1 - 99%) (bestimmt bei der Fortpflanzung, mit welcher Wahrscheinlichkeit Mutationen auftreten können)
- Lebenserwartung (0-99) (in Runden)
- Energiebedarf (10-50 E)
- Energiebedarf (zur Einleitung einer Fortpflanzung) (10-50 E)
- Fortpflanzung (1-3) (in Anzahl an Nachkommen)
- Leben unter/über Wasser (0; 1; 2) (0: nur unter Wasser lebensfähig; 1: nur über Wasser lebensfähig; 2: überall lebensfähig)
- Nahrungsverarbeitung-Effektivitätsfaktor (1 - 99 %)
- **Energiestand (in E)**
- **Alter (in Runden)**
- **Position X**
- **Position Y**

5. S-Konsumenten

- Art (beliebiger Name, wird bei der Erschaffung vom Nutzer festgelegt)
- **Generation (Zahlenwert)** (steigt mit jeder Fortpflanzung innerhalb der Art)
- Mutationsfreudigkeit (1 - 99%) (bestimmt bei der Fortpflanzung, mit welcher Wahrscheinlichkeit Mutationen auftreten können)
- Lebenserwartung (0-99) (in Runden)
- Energiebedarf (10-50 E)
- Energiebedarf (zur Einleitung einer Fortpflanzung) (10-50 E)
- Fortpflanzung (1-2) (in Anzahl an Nachkommen)
- Leben unter/über Wasser (0; 1; 2) (0: nur unter Wasser lebensfähig; 1: nur über Wasser lebensfähig; 2: überall lebensfähig)
- Jagdeignung (1-3) (in Aktionen pro Runde)
- Nahrungsverarbeitung-Effektivitätsfaktor (1 - 99 %)
- **Energiestand (in E)**
- **Alter (in Runden)**
- **Position X**
- **Position Y**

3.0 Ablauf der Simulation und Algorithmen

Mit der Berechnung jeder neuen Runde wird ein festes Schema durchgegangen, das sich stets wiederholt: Erst werden die Algorithmen aller Produzenten nacheinander durchgegangen, dann die der P-Konsumenten, dann die der S-Konsumenten. Erst wenn alle Algorithmen durchgegangen worden sind, ist die Runde beendet.

Die Algorithmen sehen wie folgt aus:

1. Produzenten

- Alter > Lebenserwartung
-> Tod (siehe Algorithmus nr. 4 „Tod“)
- Energiestand - Energiebedarf = Energiestand
- Energiestand < Energiebedarf
-> Tod
- Lichteinstrahlung > Lichtättigung
-> Tod
- Sonneneinstrahlung < Kompensationspunkt
-> Tod
- Wasserstand - Höhe des unterliegenden Planquadrats = (lokale Variable) Überflutung
- „Überflutung“ ≥ 0 und Lebensfähigkeit unter Wasser = 1
-> Tod
- „Überflutung“ < 0 negativ und Lebensfähigkeit unter Wasser = 0
-> Tod
- Destruentenpopulation des unterliegenden Planquadrats < Nährstoffbedarf
-> Tod
- Energiestand > Energiebedarf für Fortpfl. - Energiebedarf
-> Fortpflanzung (siehe Algorithmus nr. 5 „Fortpflanzung“)
- Energiestand + Sonneneinstrahlung * PhotosyntheseEffektivität = Energiestand
- Alter + 1 = Alter

2. P-Konsumenten

- Alter > Lebenserwartung
-> Tod (siehe Algorithmus nr. 4 „Tod“)
- Energiestand - Energiebedarf = Energiestand
- Energiestand < Energiebedarf
-> Tod
- Wasserstand - Höhe des unterliegenden Planquadrats = (lokale Variable) Überflutung
- „Überflutung“ ≥ 0 und Lebensfähigkeit unter Wasser = 1
-> Tod
- „Überflutung“ < 0 negativ und Lebensfähigkeit unter Wasser = 0
-> Tod
- Energiestand > Energiebedarf für Fortpfl. - Energiebedarf
-> Fortpflanzung (siehe Algorithmus nr. 5 „Fortpflanzung“)
- Nahrungssuche (siehe Algorithmus nr. 6 „Tod“)
- Energiestand gefressenes Organismus * Nahrungsverarbeitungseffektivität + Energiestand = Energiestand
- Alter + 1 = Alter

3. S-Konsumenten

- Alter > Lebenserwartung
-> Tod (siehe Algorithmus nr. 4 „Tod“)
- Energiestand - Energiebedarf = Energiestand
- Energiestand < Energiebedarf
-> Tod
- Wasserstand - Höhe des unterliegenden Planquadrats = (lokale Variable) Überflutung
- „Überflutung“ ≥ 0 und Lebensfähigkeit unter Wasser = 1
-> Tod
- „Überflutung“ < 0 negativ und Lebensfähigkeit unter Wasser = 0
-> Tod
- Energiestand > Energiebedarf für Fortpfl. - Energiebedarf
-> Fortpflanzung (siehe Algorithmus nr. 5 „Fortpflanzung“)
- Nahrungssuche (siehe Algorithmus nr. 6 „Tod“)
- Nahrungssuche ... falls Energiestand des gefressenen Organismus = 0; Jagdeignung = 2
- Nahrungssuche ... falls Energiestand des gefressenen Organismus = 0; Jagdeignung = 3
- Energiestand gefressenen Organismus * Nahrungsverarbeitungseffektivität + Energiestand = Energiestand
- Alter + 1 = Alter

4. Tod

- Wenn abgestorbener Organismus ein Produzent ist,
-> Destruentenpopulation des unterliegenden Planquadrats + 1
- Wenn abgestorbener Organismus ein P- oder S-Konsument ist,
-> Destruentenpopulation des unterliegenden Planquadrats + 2
- Lösche Eintrag des Organismus in der Datenbank

5. Fortpflanzung

- Energiestand - (Energiebedarf(fortpfl.) - Energiebedarf) = Energiebedarf
- Erstelle (je nach Fortpflanzungsgeschwindigkeit) 1-4 neue Einträge für die entstehenden Organismen in der Datenbank
- Kopiere die Parameterliste des Ursprungsorganismus für jeden entstehenden Organismus
- Gehe die einzelnen statischen Parameterwerte innerhalb der neuen Listen nacheinander (ausgenommen „Art“, „Generation“, „Position X“, „Position Y“) durch und ändere sie mit der Wahrscheinlichkeit von [„Mutationsrate“/25] um ± 1 . Ob die Änderung positiv oder negativ ist, hängt von einem 1:1-Zufall ab. Achte dabei auf die Werte-Maxima und -Minima der jeweiligen Parameter. (das bedeutet insgesamt auch, dass die Nachfahren eines Organismus innerhalb einer Generation dieselbe Eigenschaftskombination haben)
- Erhöhe den „Generation“-Wert aller neuen Einträge um +1.
- Verändere die räumliche Position der neuen Organismen um $x+1$ oder $y+1$ oder $x-1$ oder $y-1$. Wenn auf diesen Feldern zu diesem Zeitpunkt bereits ein anderer Organismus stehen sollte oder dort die Grenze des Spielfelds liegt, so lösche die Einträge der Organismen einfach sofort aus der Datenbank.

6. Nahrungssuche

- Wenn eine Nahrungsquelle (bei P-Konsumenten: Produzenten; bei S-Konsumenten: P-Konsumenten) auf einem Feld mit $X\pm 1$ oder $Y\pm 1$ gefunden wird, dann:
 - Energiestand des Opfers \rightarrow Energiestand des gefressenen Organismus (Variable)
 - Lösche Eintrag des Opfers
 - Springe zum Ende dieses Algorithmus
- Wenn Feld auf $X+1$ begehbar (nicht außerhalb des Spielfeldrands, nicht unter Wasser wenn Lebensfähigkeit unter Wasser = 1, nicht über Wasser wenn Lebensfähigkeit über Wasser = 0, nicht schon von einem anderen Organismus besetzt)
 - Bewege Organismus um 1 in X-Richtung
 - Springe zum Ende dieses Algorithmus
- Wenn Feld auf $X-1$ begehbar (nicht außerhalb des Spielfeldrands, nicht unter Wasser wenn Lebensfähigkeit unter Wasser = 1, nicht über Wasser wenn Lebensfähigkeit über Wasser = 0, nicht schon von einem anderen Organismus besetzt)
 - Bewege Organismus um -1 in X-Richtung
 - Springe zum Ende dieses Algorithmus
- Wenn Feld auf $Y+1$ begehbar (nicht außerhalb des Spielfeldrands, nicht unter Wasser wenn Lebensfähigkeit unter Wasser = 1, nicht über Wasser wenn Lebensfähigkeit über Wasser = 0, nicht schon von einem anderen Organismus besetzt)
 - Bewege Organismus um 1 in Y-Richtung
 - Springe zum Ende dieses Algorithmus
- Wenn Feld auf $Y-1$ begehbar (nicht außerhalb des Spielfeldrands, nicht unter Wasser wenn Lebensfähigkeit unter Wasser = 1, nicht über Wasser wenn Lebensfähigkeit über Wasser = 0, nicht schon von einem anderen Organismus besetzt)
 - Bewege Organismus um -1 in Y-Richtung
 - Springe zum Ende dieses Algorithmus

4.0 Zur Computerumsetzung

Das Simulationssystem ist (wie man unschwer erkennt) dafür gemacht, von einem Computer gerechnet zu werden, damit auch über viele Runden hinweg ohne viel Aufwand fehlerfreie Simulationen durchgeführt werden können.

Die Programmierung werde ich höchstwahrscheinlich mit der relativ simplen Programmiersprache „Blitz Basic“ vornehmen. Dabei werde ich verstärkt von der bereits bestehenden Ordnerstruktur Windows XPs Gebrauch machen, das Programm also die Einträge als Textdateien in einem Ordnersystem erstellen, löschen, in ihnen schreiben und aus ihnen lesen lassen.

Das eigentliche Programm wird dann für jede neue Berechnungsrunde erneut die Daten aus den Eintrags-Dateien einlesen, so kann ich mir die aufwändige Entwicklung einer guten Benutzerumgebung im Programm sparen, indem ich dem Nutzer erlaube, Parameter einfach per Hand mit Hilfe eines Texteditors zu ändern.

Im Programmfenster wird dann nur (in einer vereinfachten und abstrahierten Text-Grafik, in der „P“s beispielsweise für Produzenten stehen) grafisch und mit einer zusätzlichen Werte-Kolonne (die bspw. die Parameter-Werte von mit dem Cursor angewählten Teilen der Simulation anzeigt) der derzeitige Status der Simulation grafisch veranschaulicht und somit für den Benutzer leichter zugänglich gemacht.

Diese Nutzung der bestehenden Ordner- und Dateisysteme hat noch weitere Vorteile, so kann z.B. eine Simulation zu jedem Zeitpunkt als eine „Backup“-Kopie in ihrem Zustand „eingefroren“ abgespeichert werden und später weitergeführt werden, so lassen sich einfach mehrere Szenarios parallel abspielen.

Eine zusätzliche Funktionalität, die ich in das Computerprogramm einbauen möchte ist eine Statistikfunktion, die Werte wie z.B. den Gesamtenergieumsatz pro Runde für jede Runde speichert und später über ein Microsoft Excel -ähnliches Programm grafisch darstellbar macht.