Überblick über Netlogo

Netlogo ist eine multiagentenbasierte Programmierumgebung zur Modellierung und Simulation der zeitlich/räumlichen Entwicklung ökologischer oder sozialer Systeme. Sie wurde durch Uri Wilenski im Jahr 1999 entwickelt und wird seitdem am Center for Connected Learning and Computer-Based Modeling der Northern Western University Universität weiter entwickelt.

- **NetLogo-Welt:**
  - Ist zweidimensional
  - ist durch ein Gitter in quadratische Zellen (Patches aufgeteilt)
  - wird durch einen Observer beobachtet

- **Arten von Agenten**
  - **Patch** ist eine Zelle des Raumes
  - **Turtle** ist ein beweglicher Agent
  - **Link** ist eine Verbindung zwischen zwei Turtles.
  - *Observer: Beobachter der "Welt", kann auf alle Agenten zugreifen*
  - Der Modellierer kann auch spezielle Arten von Agentenarten als Unterart (breed) von Turtles, Patches oder Links definieren.

- **Agentenmengen**
  - Agenten mit (teilweise) gleichen Eigenschaften können zu Agentenmengen zusammengefasst warden (agent-set).
    Turtles ist die Agentenmenge aller turtle-Agenten, Patches ist die Agentenmenge aller patch-Agenten
  - Agenten derselben Agentensorte haben dieselben **Attribute**, z.B. hat jede Turtle das Attribut *color*: jeder einzelne Agent hat aber seinen eigenen Wert für color, der sich während der Simulation auch verändern kann. Color ist also eine **Variable**.
  - Agenten haben **vordefinierte Eigenschaften** (Farbe, Position, …),
    sie können zusätzliche vom Modellierer aber auch zusätzlich **selbstdefinierte Eigenschaften** bekommen.
  - Man kann Agentenmengen anhand von Eigenschaften selbst zusammenfassen:
    turtles with [color = green]

    Agenten können nicht von außen verändert werden: Wenn Agenten aktiv werden sollen, muss man die Menge der entsprechende Agenten bitten (**ask**), es selbst zu tun, z.B.

    - ask turtles [*aktionen*]
    - ask patches [*aktionen*]
    - ask turtles with [color = yellow] [*aktionen*]

- **Eigenschaften, Attribute, Variablen:**
  - Agenten haben **vordefinierte Eigenschaften** (Farbe, Position, …),
    sie können zusätzliche vom Modellierer aber auch zusätzlich **selbstdefinierte Eigenschaften** bekommen.
  - Hilfswerte, die ein Agent zwischendurch berechnet können durch
    let *varName wert* definiert und initialisiert werden.
  - Ein Agenet kann seine Attribute während der Simulation ändern:
    set *attributname neuerWert* , z.B. **set color red**
  - Für manche vordefinierte Attribute gibt es spezielle set-Befehle, z.B. setzen der Koordinaten einer Turtle: **setxy 7 15**
    anstelle von: set xcor 7
    set ycor 15
  - Variablen, die die Wahrheitswerte true oder false annehmen, enden mit einem „?"

- ## Aktionen: Kommandos und Befehle:

  - **Command**: Eine Aktion, die ein Agent ausführt
  - **Reporter**: Führt eine Berechnung durch und gibt ein Ergebnis zurück
  - **Primitives**: Vordefinierte Commands und Reporters
  - **Procedures**: Benutzerdefinierte Commands und Reporters:
    - Definition eines Commands durch: **To *prozedurname aktionen* end**
    - Definition eines Reporters durch:
      To-report *prozedurname aktionen* report *wert* end

- ## Turtles:

  - sind bewegliche Agenten, haben also eine veränderliche Position (xcor, ycor), die nicht unbedingt ganze Zahlen sein müssen
  - sie haben ein Attribut heading für die Richtung, in die sie schauen, ein Farbattribut color und können durch unterschiedliche Shapes dargestellt werden
  - sie werden über ihr who-Attribut identifiziert: z.B. turtle 13
  - die Anzahl der Turtles in einer NetLogo-Weltkann variieren, d.h. Turtles können auch während einer Simulation entstehen (create-Turtles) oder verschwinden (die)
  - Turtles können Fragen an die Patches ihrer Umgebung stellen.

  - Einige vordefinierte Variablen einer Turtle:
    - color : Farbe
    - heading : Richtung
    - hidden? : Versteckt
    - shape size: Form und Größe
    - who : Identifikator
    - xcor ycor : Position der Turtle
    - pen-mode pen-size: Zeichnet die Turtle ihre Bewegung mit einem Stift mit?

  - Einige wichtige Turtle-Commands und -Reporters:
    - Definition von Eigenschaften turtles-own
    - Bewegen der Turtle: back (bk), forward (fd), move-to, right (rt), left (lt)
    - Position : random-xcor random-ycor setxy turtles-at turtles-here turtles-on
    - Erzeugen und Löschen von Turtles: create-turtles (crt), hatch, die, sprout
    - Lokalisieren von Patches: patch-ahead patch-at patch-at-heading-and-distance patch-here patch-left-and-ahead patch-right-and-ahead
    - Berechnung von Werten: distance distancexy
    - Identifikation von Agenten: myself nobody self
    - Richtungsbestimmung: towards towardsxy
    - Darstellung im Interface: pen-down (pd) pen-erase (pe) pen-up (pu), set-default-shape, show-turtle (st), hide-turtle
    - Und viele weitere

- **Patches**
  - Sind Zellen des Raumes
  - Sie werden durch Ihre ganzzahligen Koordinaten (pxcor, pycor)
  - identifiziert, z.B. Patch 7 15 ist der Patch an den Koordinaten pxcor = 7 pycor = 15
  - haben ein Farbattribut pcolor.
  - Patches kennen die Turtles, die sich auf ihnen befinden, und können Fragen an diese richten.
  - Einige wichtige Patche-Attribute
    - Farbe des Patches: pcolor
    - Koordinaten des Patches: pxcor pycor
  - Wichtige Patch-bezogene Commands und -Reporters:
    - Definition von Eigenschaften patches-own
    - Erzeugen von Turtles: sprout sprout-<*breeds*>
    - Position : random-pxcor random-pycor
    - Initialisieren von Patches: create-turtles (crt), hatch, die, sprout
    - Lokalisieren von Patches: patch-at patch-ahead patch-at-heading-and-distance patch-here patch-left-and-ahead patch-right-and-ahead patch-set
    - Nachbarschaft eines Patches: distance distancexy neighbors neighbors4
    - Identifikation von Agenten: myself no-patch self is-patch?
    - Richtungsbestimmung: towards towardsxy
    - Darstellung im Interface: clear-patches (cp)
    - Und viele weitere

- **Agent-set**
  - Einige wichtige Commands und –Reporters für Agent-Sets:
  - Anfrage an Agent-Set richten: ask
  - Test, ob eine Bedingung für alle oder für irgendeinen Agenten des Agent-sets gilt: all? any?
  - Agentenmenge ohne den fragenden Agenten: other
  - Angabe von Bedingungen für Agent-Sets: with with-max with-min
  - Test, ob Agentenmenge: is-agentset? is-patch-set? is-turtle-set?
  - Leere Agentenmengen: no-patches no-turtles
  - Wähle bezüglich eines Attributs aus einer Agentenmenge eine Teilmenge max-n-of max-one-of min-n-of min-one-of n-of
  - Definition von Bereichen im Raum: in-cone in-radius