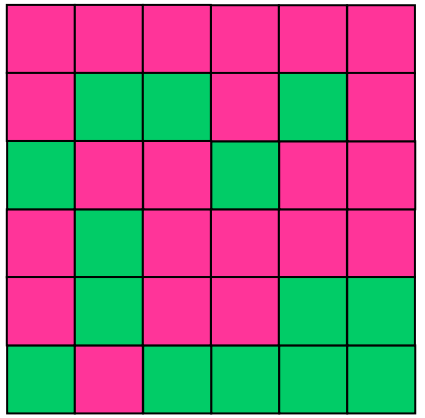
CARL VON OSSIETZKY universität OLDENBURG

Conways Game of Life

Ganz einfaches Beispiel



- Conways Game of Life
  - Raum aus quadratischen Zellen.
  - Jede Zelle kann lebendig oder tot sein.
  - Eine Zelle stirbt, wenn sie weniger als zwei oder mehr als drei lebendige Nachbarn besitzt
  - Eine Zelle wird lebendig, wenn sie genau drei lebendige Nachbarn besitzt.

Modell

- Zellen ~ Patches
- Zustand eines Patches: alive?: ja oder nein
- Verhaltensregeln:
  - Zähle die lebendigen Nachbarn
  - Falls drei Nachbarn lebendig sind, setzte alive? auf true
  - Falls nicht genau zwei oder drei Nachbarn lebendig sind, setzte alive? auf fasle

Modell in NetLogo

- Zellen ~ Patches
- → Initialisierung der Patches (später)
- Zustand eines Patches: lebendig? ja oder nein
  - Variable lebendig? Mit Wahrheitswerten true oder false
  - Patches-own [alive?]
  - Bedeutung: Jeder Patch besitzt eine Variable alive?
  - In NetLogo: Fragezeichen kennzeichnet Variablen, die Wahrheitswerte annimmt

Modell in NetLogo

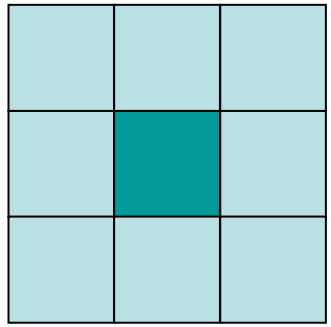
- Verhaltensregeln eines Agenten
  - Aufbau: werden einklammert durch

    to *nachrichtenname*
    Anweisungen
    end

    - Nachrichtenname ist ein beliebiger Name.
    - Eine Nachricht kann auch Parameter haben (später)
  - Wenn ein Agent eine solche Regel erhält, wertet er sie für sich aus.

Modell in NetLogo

- Anweisungen
  - Zähle deine lebendigen Nachbarn
    - Nachbarn eines Patch in Netlogo: neighbors
      - Maximal acht Nachbarn
    - Lebendige Nachbarn: neighbors with [alive?]
    - Zählen der Nachbarn: count
  - count neighbors with [alive?]



Modell in NetLogo

- Falls drei Nachbarn lebendig sind, *setze alive? auf true*
  - Setzen eines Wertes: set *variable wert*
    - set alive? false oder set alive? true
  - Fallunterscheidung in Netlogo
    - if bedingung [aktion] oder
    - if-else bedingung [true-aktion] [false-aktion]
  - if-else count neighbors with [alive?] = 3
    [set alive? true]
    [if count neighbors with [alive?] != 2
    [set alive? false]
    ]

Initialisierung des Feldes

- Angenommen x % der Zellen des Feldes sollen lebendig sein.
- Die Verteilung soll zufällig sein.
  - Zufallszahl zwischen 0 und n: random n
- if-else random $n < x$
  [set alive? true]
  [set alive? false]

Versenden einer Nachricht an…

- Agenten können zu Agentenmengen (Agentsets) zusammengefasst werden
- turtles ist die Menge aller Turtles
- patches ist die Menge aller Patches
  - patches with [alive? = false]
    ist die Menge aller „toten“ Patches
- Eine Agentenmenge kann gebeten werden etwas zu tun:
  - ask patches [*Anweisungen*]

Nachricht zur Initialisierung (setUp)

```
;; Erzeuge ca. 30% lebendige Zellen
to setUp
    ask patches [              ;; Jeder Patch soll:
        if-else random 100  < 30
          ;; Zufallszahl zwischen 0 und 99  auf „< 30“ testen
          [ set alive? true]    ;; Zelle lebendig
          [ set alive? false]   ;; Zelle tot
    ]
end
```

CARL VON OSSIETZKY universität OLDENBURG

Nachricht zur Berechnung des neuen Zustands

```
to go
    Ask patches [
        if-else count neighbors with [alive?] = 3
            [set alive?true]
            [ if count neighbors with [alive?]  != 2
                [set alive? false]
             ]
    ]
end
```

Verbesserte Version

```
to go
    ask patches [
        let n count neighbors with [alive?]  ;; temporäre Variable
        if-else n = 3
            [set alive? true]
            [ if n  != 2
                [set alive? false]
             ]
    ]
end
```

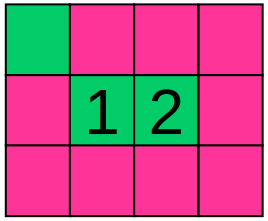
Noch bessere Version

```
patches-own[ alive? numNeighbors]
to go
      ask patches [
        set numNeighbors
            count neighbors with [alive?]
      ]
      ask patches[
            if-else numNeighbors = 3
               [cellBirth]
               [ if numNeighbors != 2
                        [cellDeath]
               ]
      ]
end
```

```
To cellBirth
      set alive? True
End
To cellDeath
     set alive? False
End
to setUp
       ask patches [
            if-else random 100  < 30
             [ cellBirth] [ cellDeath]
       ]
end
```

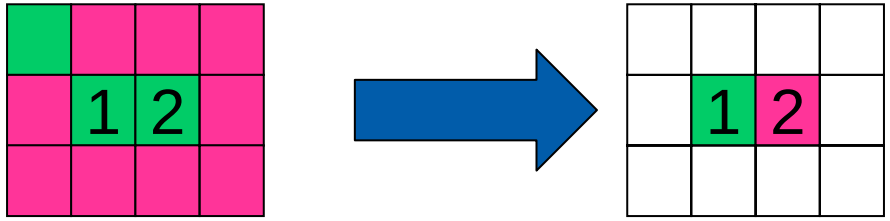
Warum ist die Version besser?

- Ask patches [aktion]:
  - Die Patches werden in zufälliger Reihenfolge aktiviert, um die Aktion auszuführen.
- Problem: Welches Ergebnis erhält man
  - Wenn Zelle 1 zuerst aktiviert wird,
  - Wenn Zelle 2 zuerst aktiviert wird.

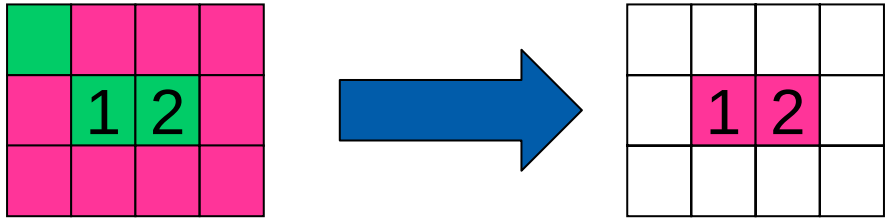


Unterschiedliche Ergebisse bei unterschiedlicher Reihenfolge

- Zelle1 vor Zelle 2



- Zelle 2 vor Zelle 1



Daher

- Erst für jeden Patch die Anzahl seiner Nachbarn feststellen
- Dann für jeden Patch den Zustandsübergang ausführen