

Gegeben ist das folgende NetLogo-Programm.

```

patches-own [ grass ]

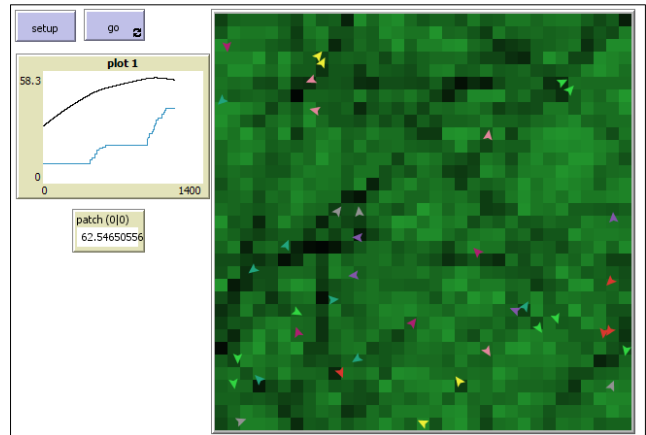
to setup
  clear-all
  ask patches [
    set grass 10 + random 30 ]
  color-patches
  reset-ticks
end

to go
  grow-grass
  color-patches
  tick
end

to grow-grass
  ask patches [
    let max-grow 0.1
    if grass > 40 [ set max-grow 0.05 ]
    if grass > 70 [ set max-grow 0.02 ]
    set grass min (list 100 (grass + random-float max-grow))
  ]
end

to color-patches
  ask patches [ set pcolor 60 + grass / 20 ]
end

```



## Hinweise

- Der Hintergrund ist in Quadrate (patches) eingeteilt. Anders als Turtles können die sich aber nicht bewegen und auch ihre Form nicht ändern.
- Mit patches-own bekommt jedes dieser Quadrate eine zusätzliche Eigenschaft namens grass.
- Das Farbsystem findet man mit den Suchbegriffen NetLogo und Color.

## Aufgaben

1. Lies das Programm und beschreibe das Verhalten.
2. Gib das Programm ein, erzeuge die Buttons für setup und go und prüfe die Vorhersage. Formuliere die Aufgabe von Zeile 21 des Programms umgangssprachlich.
3. Erzeuge einen plot (ein Diagramm). Ändere die Farbe des vorhandenen Pens auf blau und ergänze einen grünen Pen mit `plot sum [ grass ] of patches / 900`. Ermittle, was dargestellt wird, und finde die Stelle im Programm, die für die Form der blauen Linie verantwortlich ist.
4. Erzeuge einen Monitor und trage `[ pcolor ] of patch 0 0` als Reporter ein. Was wird angezeigt?

**Erweitere das Programm um Turtles, die auf der Wiese herumlaufen und Gras „fressen“.**

- Die setup-Prozedur soll 10 Turtles erzeugen, die auf zufällige Positionen verteilt sind. Der einfachste Weg dazu ist der Befehl `setxy random-xcor random-ycor`.
- Alle Turtles sollen eine zusätzliche Eigenschaft namens `food` bekommen. Den entsprechenden Befehl sollte man erraten können. Turtles sollten nach dem setup einen zufälligen `food`-Wert zwischen 50 und 60 haben.
- Ergänze eine Prozedur `move-turtles`. Sie lässt alle Turtles ihre Bewegungsrichtung ein wenig verändern und schickt sie um 0,1 Patch vorwärts. Weil das Energie kostet, verliert außerdem jede Turtle eine bestimmte `food`-Menge, z.B. 0,5 Einheiten. Die Prozedur muss natürlich in `go` aufgerufen werden.
- Turtles mit negativem `food`-Wert sind offenbar verhungert. Lass sie in `move-turtles` mit dem Befehl `die` sterben.
- Schreibe eine Prozedur `eat-grass`. In dieser Prozedur soll jede Turtle eine Menge an Gras fressen. Dadurch steigt ihr `food`-Wert, und der `grass`-Wert des Patches wird entsprechend kleiner. Natürlich kann so eine kleine Turtle nur eine bestimmte Menge pro Runde fressen (z.B. 1,5 Einheiten) – wenn das Patch genug Gras enthält. Wenn nicht, wird es leergefüttert.
- Turtles mit mehr als 100 `food`-Einheiten haben genug Energie, um sich zu vermehren. Dazu wird zunächst der `food`-Wert halbiert und dann mit dem Befehl `hatch 1` ein neues Turtle erzeugt, das automatisch die andere Hälfte des Futters bekommt.

**Zusätzliche Anregungen**

- Man könnte sich auch einen Schieberegler vorstellen, mit dem der Anwender die Anfangszahl der Turtles einstellen kann.
- Auch die Menge an verlorenem (`food-needed`) und maximal fressbaren (`max-eat`) Futter kann mit Schiebereglern gesteuert werden.
- Finde heraus, wie man im Diagramm nur die Anzahl der Turtles mit einer bestimmten Farbe oder `food`-Menge darstellen kann. Stichwort: `with`
- Untersuche, wie sich die Population entwickelt:
  - Gibt es einen stabilen Endzustand, also eine einigermaßen konstante Turtleanzahl?
  - Vergleiche den Verlauf der Gras- und der Turtlekurve im Diagramm.
  - Überleben alle Farben? Wenn nicht: Was haben die ausgestorbenen Farben „falsch“ gemacht?
  - Finde Werte für Verbrauch und Fressmenge, bei denen die Wiese schön grün bleibt und trotzdem keine Einsamkeit herrscht.