Primitives

; comment

Random-normal – uses stochastic distribution

Hide-turtles

Distancexy – distance between turtle and patch

Downhill – moves turtle to neighbors patch for lowest value

Uphill – moves turtle to highest patch

Floor number – reports largest integer

Hatch – turtle creates another turtle with all characteristics

Tie – binds two turtles together. Movement of one turtle affects the other

Random-seed – set seeds of random number generator

Histogram – summarize given list or agent set

Stamp – leaves image of turtle

If – condition then command.

Ifelse – condition, then command, then another command

Die – murders turtle or link

Building an ABM

Chpts. 6, 4, & 5

Verifying/Debugging

Typos

Syntax errors

Runtime errors

Misunderstanding primitives

Logic errors

Formulation errors

Common Errors

Syntax errors

Setup structure and test one line at a time (works for submodels too)

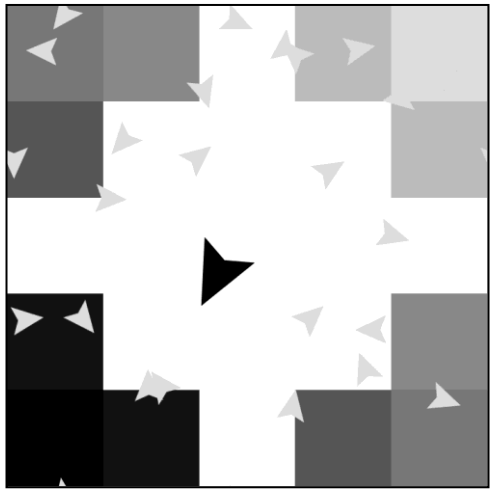
ifelse condition
[ ]
[ ]

ifelse condition
[x = 1]
[ ]

ifelse condition
[x = 1]
[x = 2]

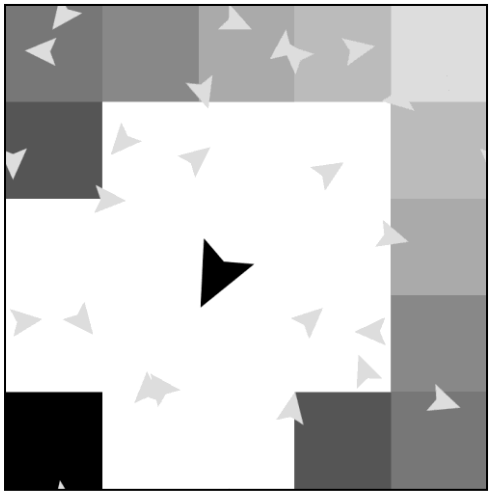
Misunderstanding primitives

Most common issue with Netlogo



Let neighbor-turtles turtles in-radius 2

Patch context



Let neighbor-turtles turtles-on patches in-radius 2

Turtle context

Identifying Errors

Implement in other software

```
to Astro

let lat 29.7

let declination (- asin ((sin(23.45)) * (cos(2 * pi * (day + 10) / 365 * rad-to-degree)))) ;sin and cosine assume degrees, while other programs assume radians.
set declination (declination * degree-to-rad) ;declination must be in radians
;print declination

;intermediate variables
 set sinld ((sin(lat)) * (sin(declination * rad-to-degree))) ;degrees
 set cosld cos(lat) * cos(declination * rad-to-degree) ;degrees
 let aob (sinld / cosld) ;degrees
 ;print word "aob: " aob


let temp1 asin(aob)              ;intermediate calculations used calculate daylength (arcsin of aob in degrees)
set temp1 (temp1 * degree-to-rad) ;converts degrees to radians


set daylength 12 * (1 + 2 * temp1 / pi) ;calculates daylength based on declination and latitude

;print word "daylength: " daylength


set dsinB 3600 * (daylength * sinld + 24 * cosld * sqrt(1 - aob * aob) / pi)


set dsinBE 3600 * (daylength *(sinld + 0.4 * (sinld * sinld + cosld * cosld * 0.5)) + 12 * cosld * (2 + 3 * 0.4 * sinld) * sqrt (1 - aob * aob) / pi)
;print word "dsinBE: " dsinBE
let sc 1370 * (1 + 0.033 * cos(2 * pi * day / 365 * rad-to-degree))
set dso sc * dsinB


;Verified with R and Matlab model 21 April 2015

;print sc
;print dso
```

	A	B
1	Day	258
2		
3	Declination	0.039335481
4		
5		
6	Latitude	30
7		
8	Intermediate variables	
9	sinld	0.019662669
10		
11	cosld	0.865355498
12		
13	aob	0.022722071
14		
15	daylength	12.17359877
16		
17	dsinB	24654.55574
18		
19	dsinBE	31505.60879
20		
21	sc	1357.892115
22		
23	dso	3.3478226842E+07
24		

Use print statements liberally

```
set C-alpha map [x -> x ^ (- alpha)] C
print word "C: " C
```

```
set WC-alpha (map * W-Vect C-alpha)
print word "WC: " WC-alpha
```

Output

Create output files to analyze data or test procedures

Output procedure provided in class folder. (Output.nlogo)

Primitives to know

- *File-open*
- *File-close*
- *File-delete*
- *File-type*
- *File-print*

```
globals[output-file Var1 Var2 Var3 Var4]

to initialize-output-file
  ;Open output file
  ;First, delete file rather than appending to it
   set output-file (word "Results.csv")


  if(file-exists? output-file)
  [
    carefully
    [file-delete output-file]
    [print error-message]
  ]

  file-open output-file ;opens file for writing

  ;Write main heading and column headings
  file-type "Variable 1,"
  file-type "Variable 2,"
  file-type "Variable 3,"
  file-print "Variable 4,"

  file-close

end

to output-resultss

  file-open output-file

  file-type (word Var1 ",")
  file-type (word Var2 ",")
  file-type (word Var3 ",")
  file-print (word Var4 ",")

  file-close
end
```

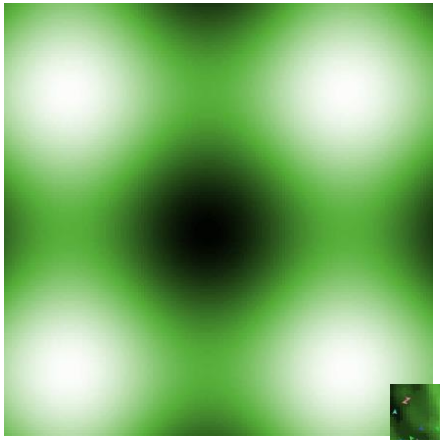
1

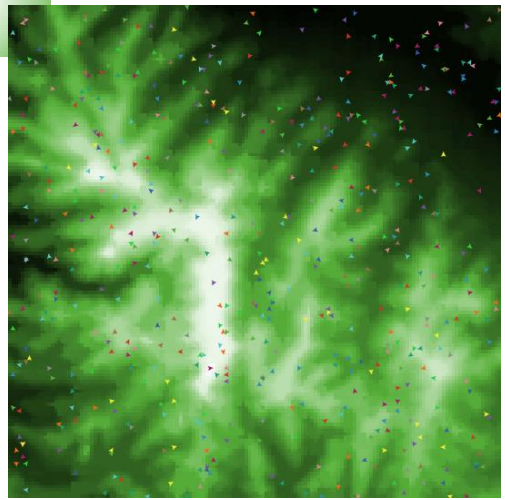
Building the model

What's the first step?

Virtual Corridors

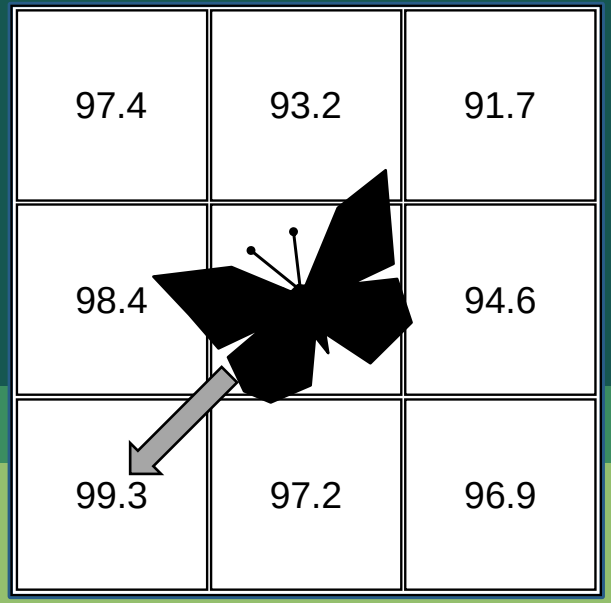
- Develop an ABM to test virtual corridors
- Recreate model of Pe’er et al 2005



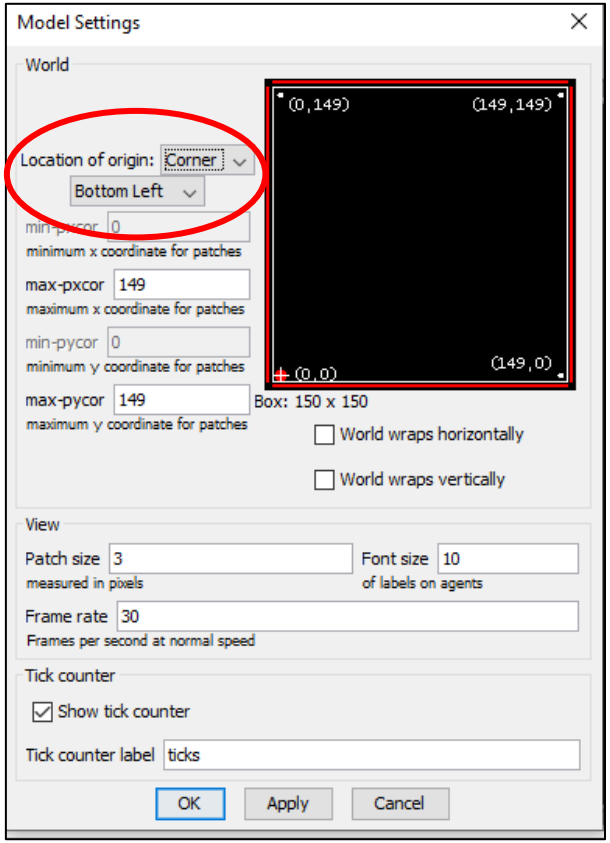


Model Objective

Determine if interactions between butterflies and landscape topography (hill topping) can create virtual corridors



Building the model



1) Set up world (150 x 150 patches)

2) Define state variables **(check code)**

globals [ ]
Patches-own [ ]
Turtles-own [ ]

Check

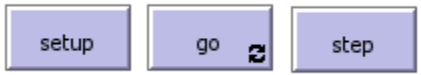
3) Create patch attribute *elevation* **(check code)**

Patches-own [Elevation]

Check

USE COMMENTS (;)

4) Set up setup and go procedures **(check code & SAVE)**




```
To setup

ca   ;clears all variables
Ask patches
[
;do something
]


Reset-ticks
End
```

What is the code for *go* and *step*?

Assignment

- Complete the model in chapter 4.
  - Create 500 turtles and have them move uphill based on a random number function.
  - Run for 1000 ticks
  - Note general trends

Netlogo primitives to learn

- Ask
- Move-to
- Uphill
- Neighbors
- One-of
- Setxy
- random
- Pxcor, pycor
- Create-turtles (crt)
- Clear-all (ca)
- Reset-ticks
- Stop
- to, end

2

Onwards to Science!

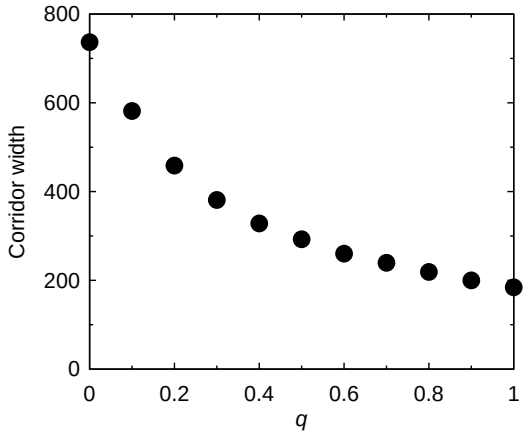
Let’s start with the first set of slides

Scientific models

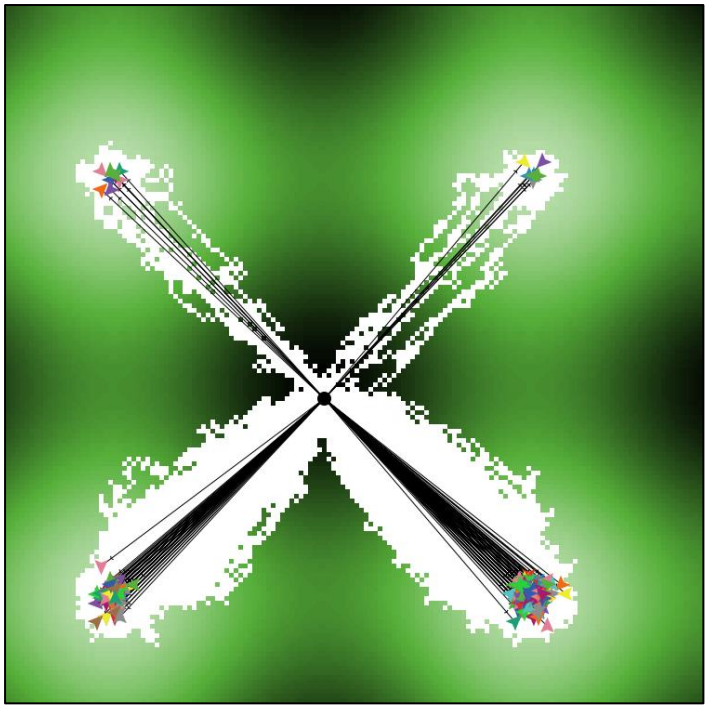
- Move past visualizations into modeling for science
- What are some ways we could make this model more realistic?

Working with output

- Defining the patterns of expected behavior helps identify output



Science up the model



- Calculate the average width of corridors used by butterflies
- Plot corridor width on the interface at the end and over time
- Export the plots and graph 5 different scenarios
- Import real landscape, scale the color, run the model again

Netlogo primitives to learn

- Patch-here
- True/false
- Count
- Mean
- Let/set
- With
- Scale-color
- Export
- Min/max
- To-report
- File-open, file-close, file-at-end
- While
- Not
- Next-X, Y, variable

Developing your own model

It's kinda hard

Go slow

Use your conceptual model as a template

Build one component first, get it to work, then move on.

Add interactions piecemeal

The dictionary and examples will be helpful

Don't get too frustrated. We're here to help.

Power of psuedocode

If you get stuck, create a bulleted list of what you want the model to do

Psuedocode: plain language description of code blocks