



Agent-Based Modelling in NetLogo

David Hales

www.davidhales.com/abm-netlogo

NetLogo programs

- Last time we:
 - Typed some commands directly into netlogo
 - Created some buttons & sliders
- This time we will:
 - Write a simple program
 - Create some output plots

NetLogo Programs

- In general programs contain (in this order):
 - Any global declarations
 - A setup procedure definition
 - A go procedure definition
 - Any other procedure definitions
- The interface generally contains buttons to execute the setup and go procedures

[Open exercise 2 pdf – starting from the beginning](#)

A simple program

```
to setup
  clear-all
  create-turtles 100 [
    setxy random-pxcor random-pycor
    set heading random 360
  ]
  reset-ticks
end

to go
  ask turtles [
    forward 0.1
    if count turtles-here > 1 [
      right random 360
      forward 1
    ]
  ]
  tick
end
```

Tasks

- Add “setup” and “go” buttons to the interface that call the procedures. Make the “go” button continuous.
- Test the buttons work.
- Add a slider for a variable “popsize” ranging from 1 to 1000.
- Change the program so it creates “popsize” turtles rather than 100
- Test the slider works.
- Continue with exercise 2 (from 2.2) and complete the tasks

final code from ex. 2

```
globals [deaths births]
turtles-own [age]

to setup
  clear-all
  create-turtles popsize [
    setxy random-pxcor random-pycor
    set heading random 360
    set age 0
  ]
  reset-ticks
end

to-report prob [x]
  report (random-float 1 < x)
end
```

```
to go
  set deaths 0
  set births 0
  ask turtles [
    set age age + 1
    forward 0.1
    if count turtles-here > 1 [
      set deaths deaths + 1
      die]
  ]
  ask turtles [
    if prob birthprob [
      set births births + 1
      hatch 1 [
        set age 0
        set heading random 360
        forward 1]
      ]
  ]
  tick
end
```

final interface from ex. 2

