

Your First Model

EES 4760/5760

Agent-Based & Individual-Based Computational Modeling

Jonathan Gilligan

Class #5: Tues. January 23 2018

Homework:

- In the mushroom hunt, were there always 80 red patches?
- Any questions about modified mushroom hunt model?
- Let's talk about ODD exercise.

Writing a model from an ODD

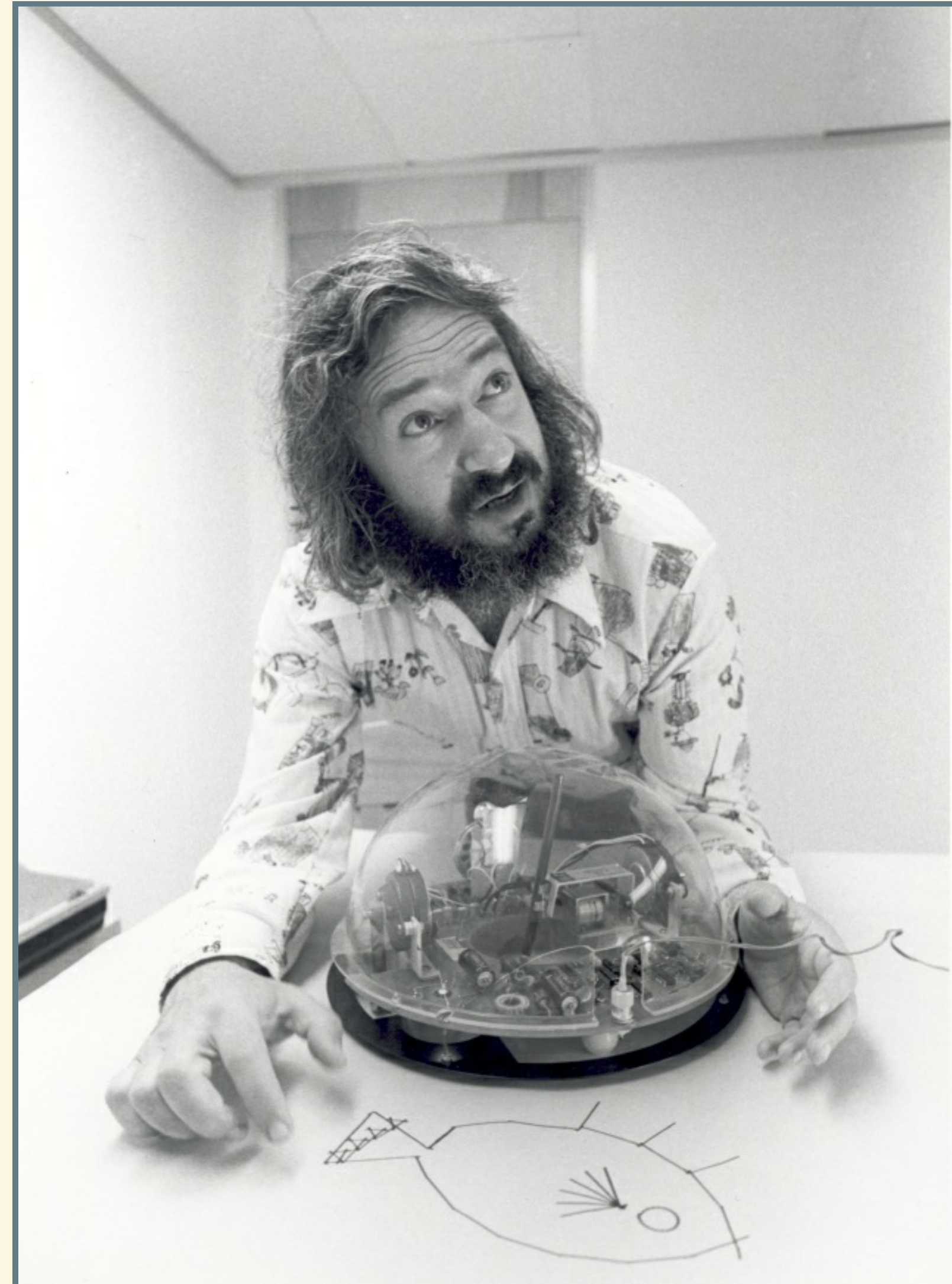
- Questions about writing a model from Butterfly ODD?
- Were there things the ODD was unclear about?

Digression on Turtles

Why “Turtles”?



Figure 1 shows one of our turtles—so named in honor of a famous species of cybernetic animal made by Grey Walter, an English neurophysiologist. Grey Walter's turtle had life-like behavior patterns built into its wiring diagram. Ours have no behavior except the ability to obey a few simple commands from a computer to which they are attached by a wire that plugs into a control-box that connects to a telephone line that speaks to the computer, which thinks it is talking to a teletype so that no special system programming is necessary to make the computer talk to the turtle. (If you'd like to make a fancier turtle, you might use a radio link. But we'd like turtles to be cheap enough for every kid to play with one.)



Enhancing the Butterfly model

- Download butterfly model from https://ees4760.jgilligan.org/models/class_05/butterfly_model_class_5.nlogo
- Put a slider for q
- Add patches-own variable to indicate whether it was visited.

```
patches-own
[
  elevation
  visited? ; question mark means it's a true/false variable
]

to setup
[
  ...
  ask patches [
    set visited? false
    ...
  ]
  ...
]
```

Enhancing the Butterfly model

- Put a slider for q
- Add patches-own variable to indicate whether it was visited.
- Add turtles-own variable to remember the patch where it started.
- Increase the number of butterflies to 50.
- Stop butterfly from moving if it's at the top of a hill.
 - How can you tell whether it's on the top?

Enhancing the Butterfly model

- Write a reporter for corridor width

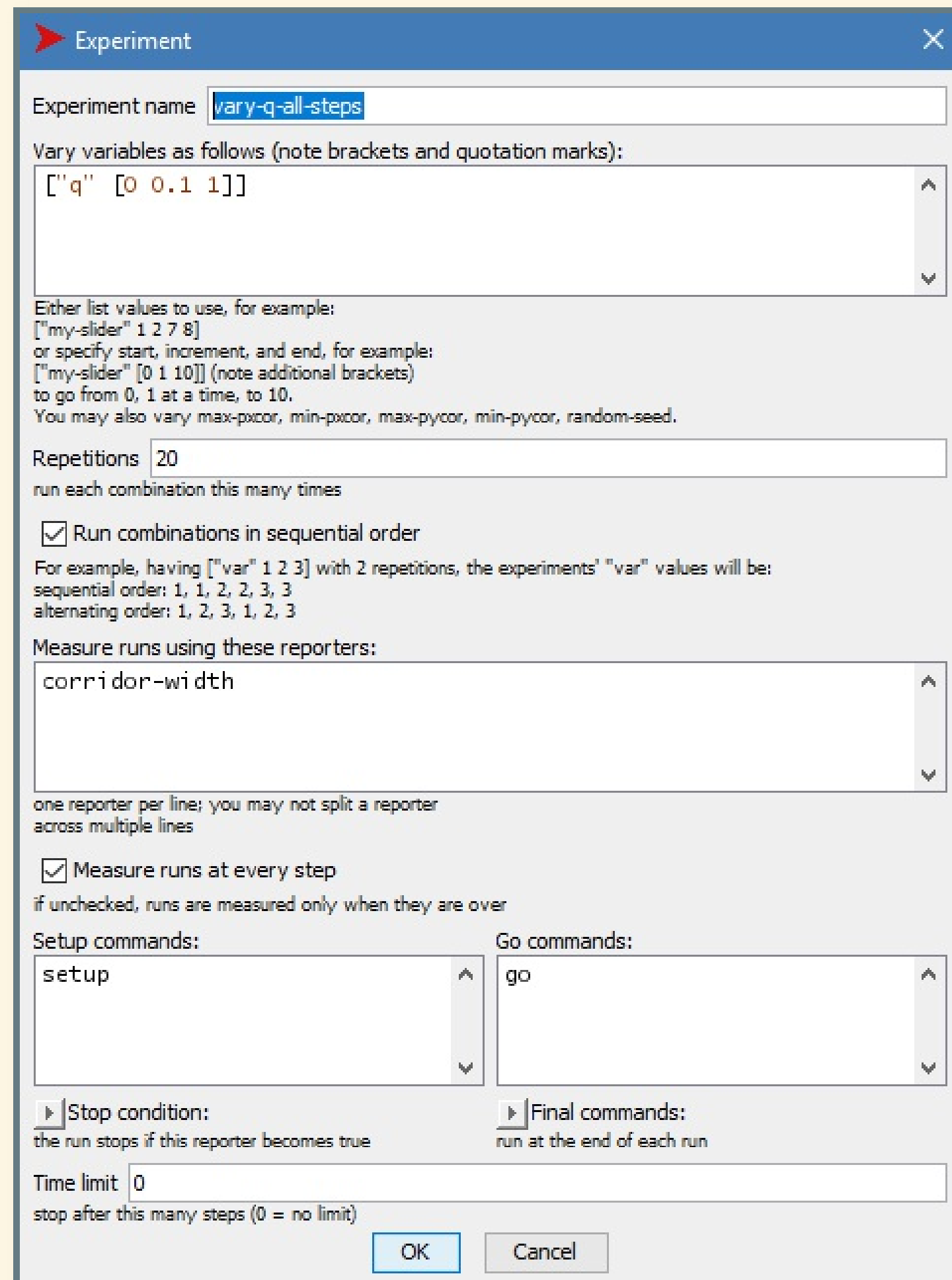
$$\text{Corridor width} = \frac{\# \text{ patches visited}}{\text{distance from start}}$$

- Put an **observer** on the interface
- Define a reporter:

```
to-report corridor-width  
  let wid ... ; calculate corridor width  
  report wid  
end
```


Running Experiments: BehaviorSpace

- Vary any parameter that has a control on the model's interface
- Writes output to .csv spreadsheet file (table output is the most useful).



The screenshot shows the 'Experiment' dialog box in BehaviorSpace. It contains the following fields and options:

- Experiment name:** A text field containing 'vary-q-all-steps'.
- Vary variables as follows (note brackets and quotation marks):** A text area containing `["q" [0 0.1 1]]`.
- Either list values to use, for example:** `["my-slider" 1 2 7 8]`
or specify start, increment, and end, for example: `["my-slider" [0 1 10]]` (note additional brackets)
to go from 0, 1 at a time, to 10.
You may also vary max-pxcor, min-pxcor, max-pycoor, min-pycoor, random-seed.
- Repetitions:** A text field containing '20'. Below it, the text 'run each combination this many times' is displayed.
- ☒ **Run combinations in sequential order**
For example, having `["var" 1 2 3]` with 2 repetitions, the experiments' "var" values will be:
sequential order: 1, 1, 2, 2, 3, 3
alternating order: 1, 2, 3, 1, 2, 3
- Measure runs using these reporters:** A text area containing 'corridor-width'.
- one reporter per line; you may not split a reporter across multiple lines
- ☒ **Measure runs at every step**
if unchecked, runs are measured only when they are over
- Setup commands:** A text area containing 'setup'.
- Go commands:** A text area containing 'go'.
- ☐ **Stop condition:**
the run stops if this reporter becomes true
- ☐ **Final commands:**
run at the end of each run
- Time limit:** A text field containing '0'. Below it, the text 'stop after this many steps (0 = no limit)' is displayed.
- Buttons:** 'OK' and 'Cancel' at the bottom.

Output from BehaviorSpace Experiments

- Note: Data written in spreadsheet might be out of order.

```
"BehaviorSpace results (NetLogo 5.3.1)"  
"jg_butterfly_1.nlogo"  
"vary-q"  
"01/25/2016 23:08:47:963 -0600"  
"min-pxcor", "max-pxcor", "min-pycor", "max-pycor"  
"0", "149", "0", "149"  
"[run number]", "q", "[step]", "corridor-width"  
"4", "0", "999", "424.71585264477375"  
"3", "0", "999", "407.8948972331853"  
"2", "0", "999", "402.16008464319225"  
"1", "0", "999", "413.09183879201066"  
"5", "0", "999", "380.4175502215263"  
"6", "0", "999", "408.25117143183326"  
"7", "0", "999", "431.37461560574894"  
"8", "0", "999", "408.38259535508286"  
"9", "0", "999", "421.7254402334981 "
```

Analyzing Behaviorspace Output

- Behaviorspace output format is annoying
 - Each line is some tick of some run
 - How to organize, and average over runs?
- analyzeBehaviorspace app
 - Works with Behaviorspace **table** output format.
 - <https://analyze-behaviorspace.jgilligan.org>
 - Or install on your own computer using R
 - Instructions at <https://github.com/jonathan-g/analyzeBehaviorspace>
 - After installing:

```
library(analyzeBehaviorspace)
launch_abs()
```

Emergence

- A tricky concept.
- *Growing Artificial Societies*: “stable macroscopic patterns arising from the local interaction of agents.”
- Epstein ten years later: “I have always been uncomfortable with the vagueness and occasional mysticism surrounding this word.”
- Epstein now prefers to talk about “*Generative Social Science*”