Computational Modeling of Social Behavior

Day 2

Spreading Infection

Paul Smaldino

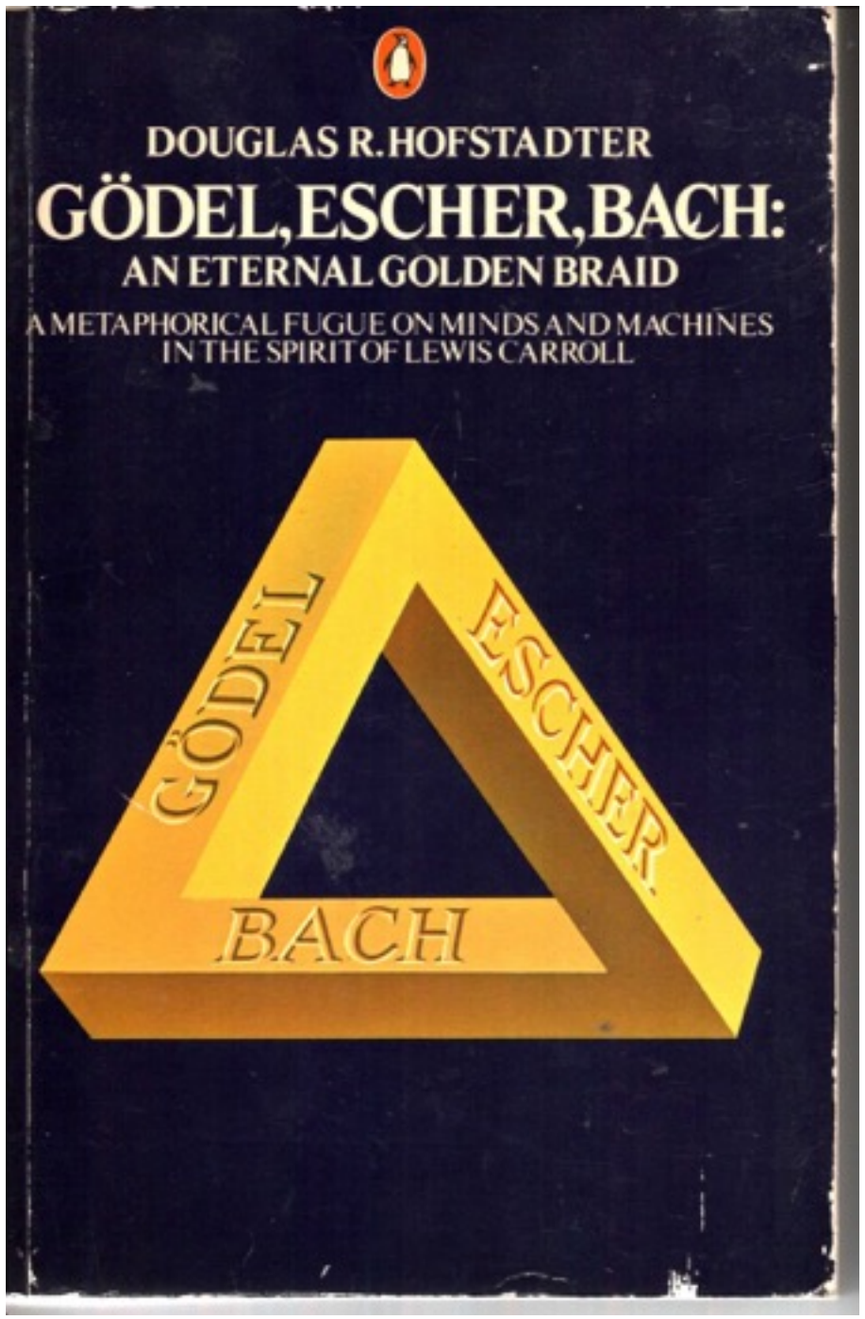
Outline of the day

- **Morning**
  - Review modeling challenges
  - Infection Models
- **Afternoon**
  - Modeling infections with collisions

If you wish to make an apple pie from scratch, you must first invent the universe.

Writing, planning, outlining

Hofstadter's Law: It always takes longer than you expect, even when you take into account Hofstadter's Law.



Separating Design and Implementation

The iterative process

Sometimes people ask: “How does your show get written?”
Like this: performance, feedback, revision

–Baba Brinkman, *The Rap Guide to Evolution*

Pseudocode

- A description of a computer programming algorithm that uses the structural conventions of programming languages, but omits detailed subroutines or language-specific syntax.

- Valuable for ***stepwise refinement***

- **Initialize:**
  - ???
- **At each tick:**
  - ???

Color Spreading

- **Initialize:**
  - All patches are initially white
  - Set center patch to blue

- **At each tick:**
  - Each patch checks its neighbors to see if any of them are blue. If so, it turns blue.
  - Stop if no patches are white

```
to setup
  clear-all
  ask patches [ set pcolor white] ;; set patches to white
  ask patch 0 0 [ set pcolor blue ] ;;select center patch blue
  reset-ticks
end

to go
  if not any? patches with [pcolor = white]
  [stop]
  ask patches [ spread-move ]
  tick
end

to spread-move
  if any? neighbors with [ pcolor = blue ]
  [set pcolor blue]
end
```

Rainbow Spiral

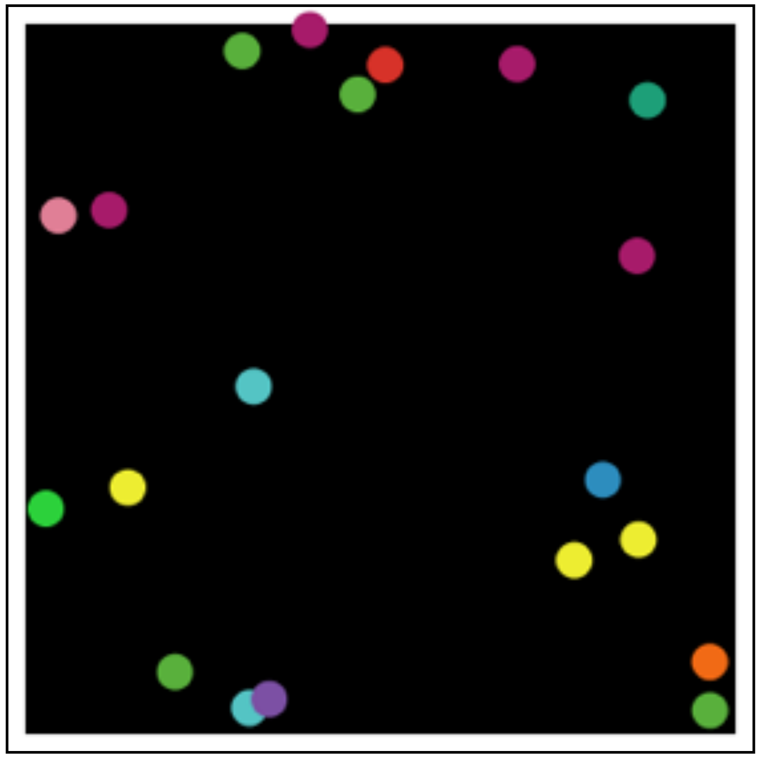
- **Initialize:**
  - All patches are initially white
  - Set center patch to random color
  - Place invisible agent on center patch, headed east.
  - s*tepLength* = 1
  - t*wostep* = 0

- **At each tick:**
  - For *x* from 1 to *stepLength*
    - Move forward 1
    - Set patch occupied by agent to random color
  - Turn left 90 degrees
  - *Twostep*++
  - If *Twostep = 2*
    - *Twostep* = 0
    - *StepLength*++
  - Continue until no patches are white.

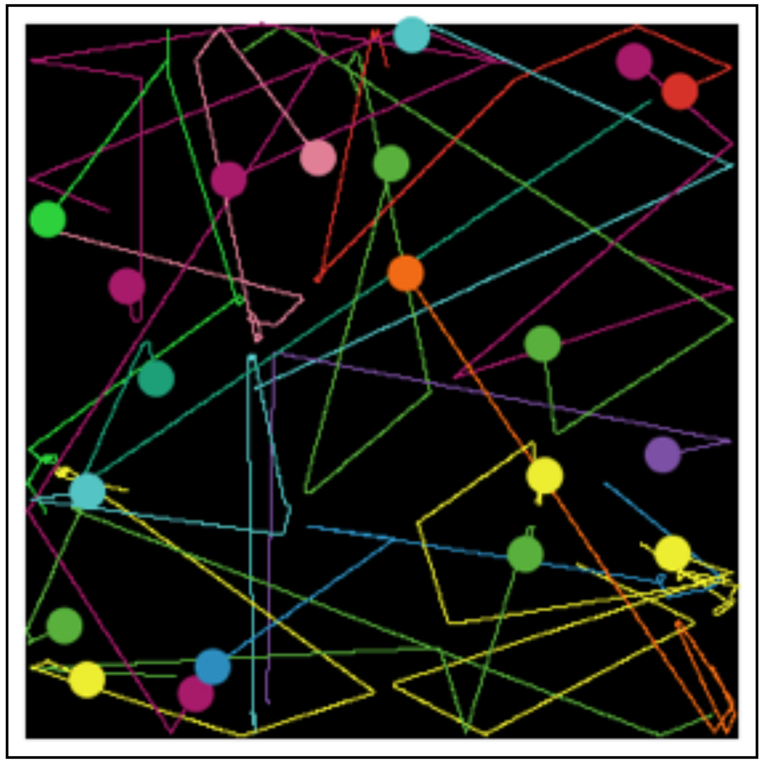
Collisions

- **Initialize:**
  - turtles are placed in random locations with random headings
- **At each tick:**
  - For each turtle:
    - If a step forward would place outside grid
      - turn to “bounce” off wall
    - If another turtle is within search-radius
      - set heading to random direction, and set other turtle’s heading to the opposite direction
    - move forward *step-length*

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;            SETUP                     ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to setup
  clear-all
  set-default-shape turtles "circle"
  draw-walls
  create-turtles n-turtles  ;; create some turtles
    [ randomize ]           ;; place them randomly
  ask turtles [set size 2]
  reset-ticks
end

; draws the boundaries (walls) of the space
to draw-walls
  ; draw left and right walls
  ask patches with [abs pxcor = max-pxcor]
    [ set pcolor white ]
  ; draw top and bottom walls
  ask patches with [abs pycor = max-pycor]
    [ set pcolor white ]
end

; set random location
to randomize
  setxy random-xcor random-ycor
  if pcolor = white        ; if it's on the wall...
    [ randomize ]         ; ...try again
end
```

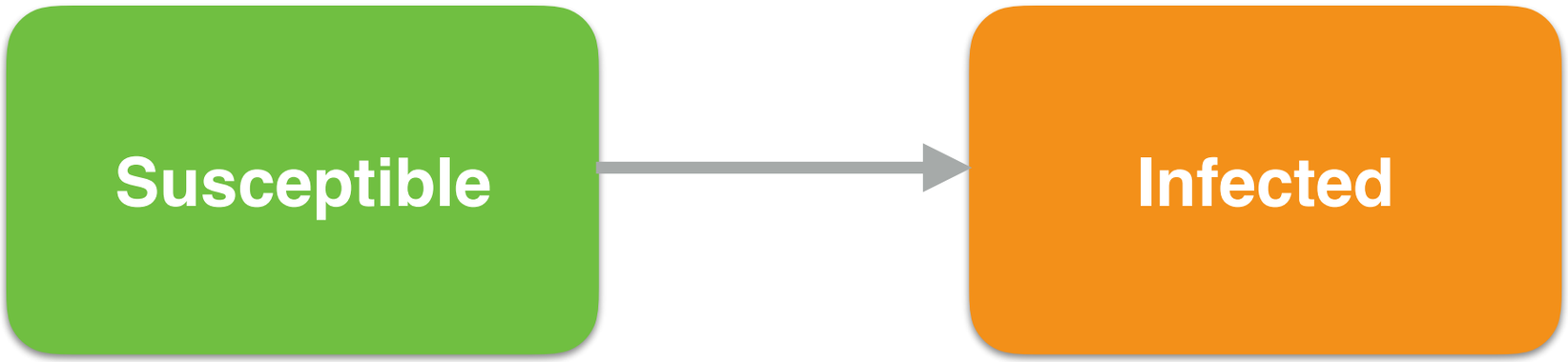


```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;           GO                         ;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
to go
  ask turtles [
    ifelse leave-trace?  ;; pen up or down?
      [ pen-down ]
      [ pen-up ]
    bounce
    fd move-speed
  ]
  tick
end

to bounce
  ; Hitting left or right wall? Reflect heading around x axis
  if abs [pxcor] of patch-ahead 0.4 = max-pxcor
  [ set heading (- heading) ] ;
  ; Hitting top or bottom wall? Reflect heading around y axis
  if abs [pycor] of patch-ahead 0.4 = max-pycor
  [ set heading (180 - heading) ]

  ; check if any other turtles are close enough to hit.
  if collisions? and count turtles in-radius 1.5 > 1
    [
      let new-heading (heading + random 90)
      set heading new-heading
      fd 0.1
      ask other turtles in-radius 2 [
        set heading new-heading + 180
        fd 0.1
      ]
    ]
end
```



Contagion

Simple diffusion model



N: population size
I: # of infected
τ: transmissibility
c: contact rate

Probability a meeting leads to infection

$$\tau \frac{I}{N} \left(\frac{N-I}{N} \right)$$

Pr(meeting —> infection)

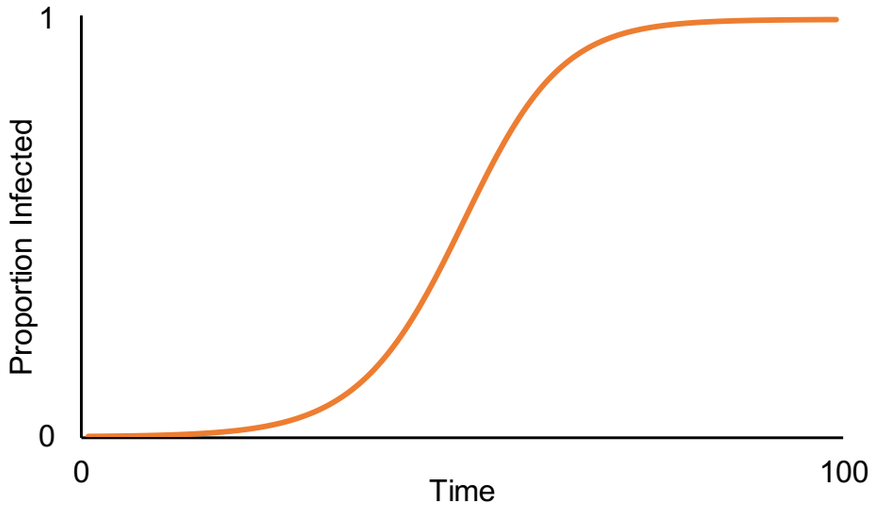
Pr(susceptible meets infected)

Trajectory of infected population

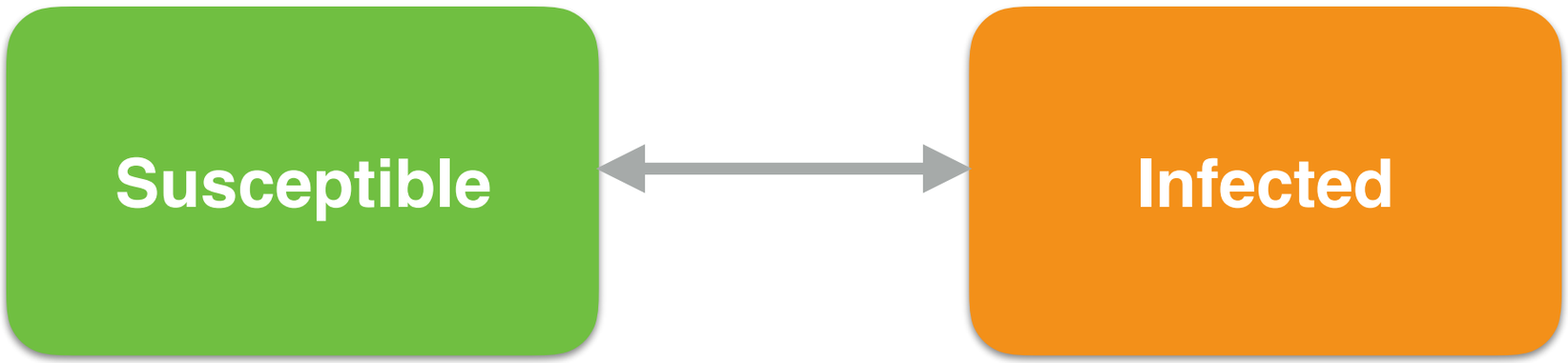
$$I_{t+1} = I_t + N c \tau \frac{I_t}{N} \left(\frac{N-I_t}{N} \right)$$

# of meetings

$$I_{t+1} = I_t + c \tau I_t \left(1 - \frac{I_t}{N} \right)$$



SIS Model



$$I_{t+1} = I_t + c\tau I_t\left(1 - \frac{I_t}{N}\right) - \gamma I_t$$

new infections

recovered

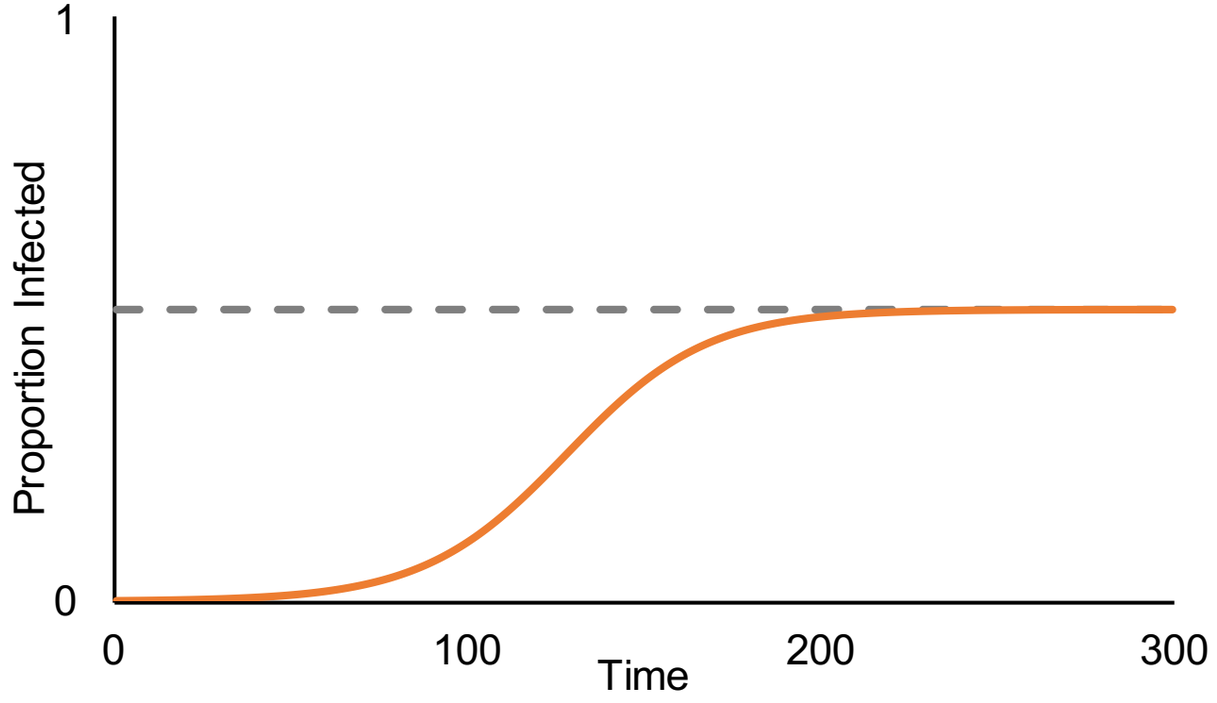
N: population size
I: # of infected
τ: transmissibility
c: contact rate
γ: recovery rate

Equilibrium infection rate

$$c\tau \cancel{I}\left(1 - \frac{I}{N}\right) = \gamma \cancel{I}$$

$$1 - \frac{I}{N} = \frac{\gamma}{c\tau}$$

$$\frac{I}{N} = 1 - \frac{\gamma}{c\tau}$$



Will the infection spread?

N: population size
I: # of infected
τ: transmissibility
c: contact rate
γ: recovery rate

$$I_{t+1} = I_t + c\tau I_t \left(1 - \frac{I_t}{N}\right) - \gamma I_t$$

close to 1 when infection is new

$$I_{t+1} \approx I_t + I_t(c\tau - \gamma)$$

Infection spreads if: $c\tau > \gamma$

$$R_0 = \frac{c\tau}{\gamma} > 1$$

"basic reproduction number"

Rewrite equilibrium frequency:

$$\frac{I}{N} = 1 - \frac{1}{R_0}$$

Vaccinate a proportion V of the population

$$\Delta I = \underbrace{c\tau\left(1 - \frac{I}{N}\right)}\underbrace{(1 - V)}I - \gamma I$$

Pr(encounter a susceptible and transmit)

Pr(the susceptible isn't vaccinated)

N: population size
I: # of infected
τ: transmissibility
c: contact rate
γ: recovery rate

When infection is new

$$\Delta I \approx I\left(c\tau(1 - V) - \gamma\right)$$

Infection spreads if:

$$\frac{c\tau}{\gamma}(1 - V) > 1$$

$$(1 - V)R_0 = r_0$$

effective basic reproduction number

$$(1-V)R_0 = r_0$$

$$(1-V)R_0 \leq 1$$

$$R_0 - R_0 V \leq 1$$

$$R_0 - 1 \leq R_0 V$$

$$1 - \frac{1}{R_0} \leq V$$

threshold for herd immunity

V* = 14/15 ≈ .93

V* = 4/5 = .80

V* = 2/3 ≈ .67

Estimated R_0:

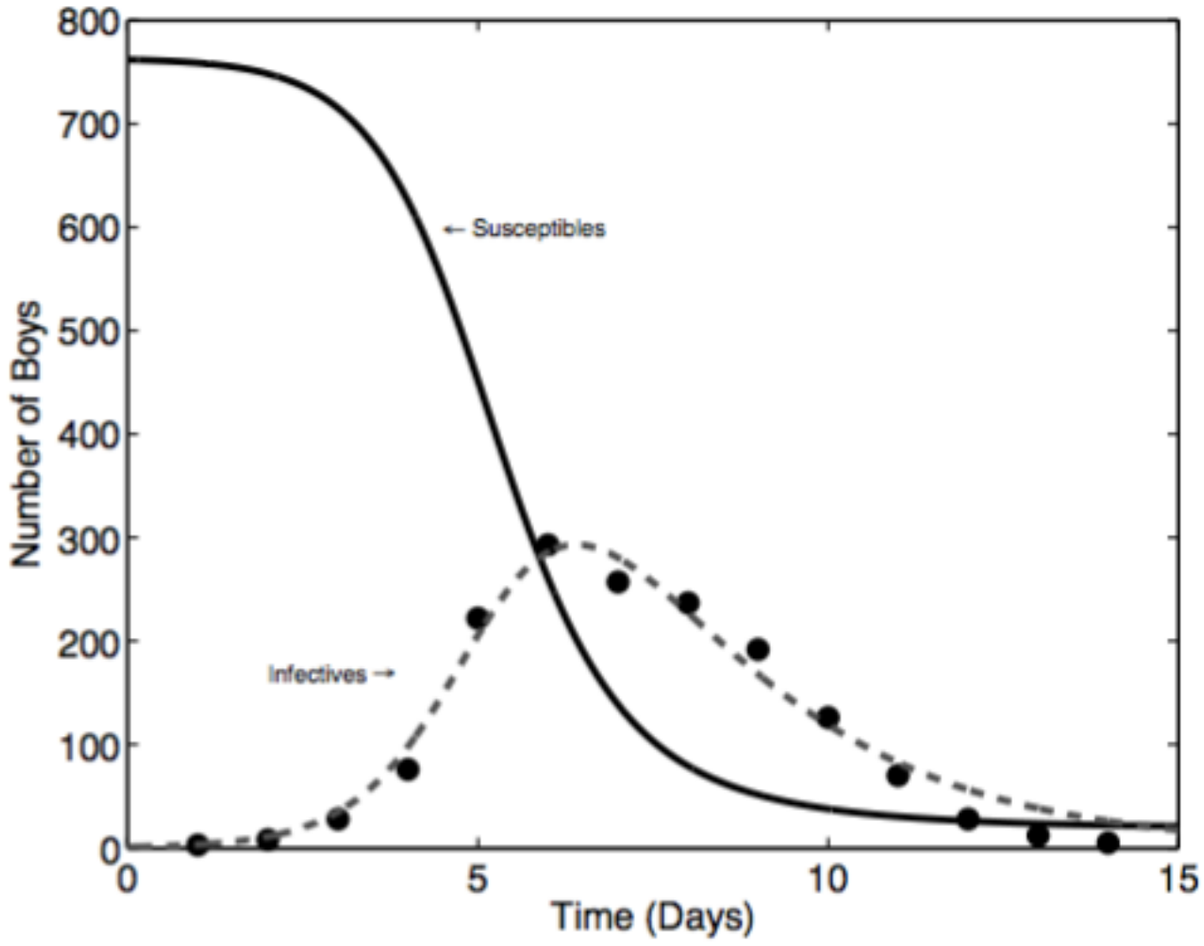
Measles ≈ 15

Mumps ≈ 5

Influenza ≈ 3

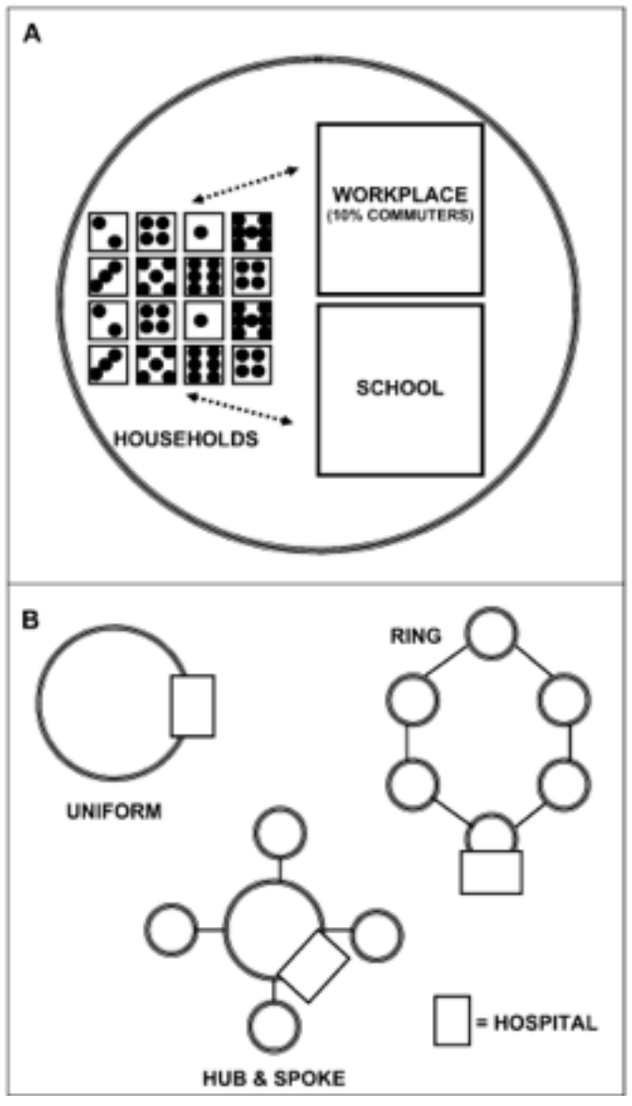
Do these models reflect reality?

Flu outbreak
English boarding school
1978

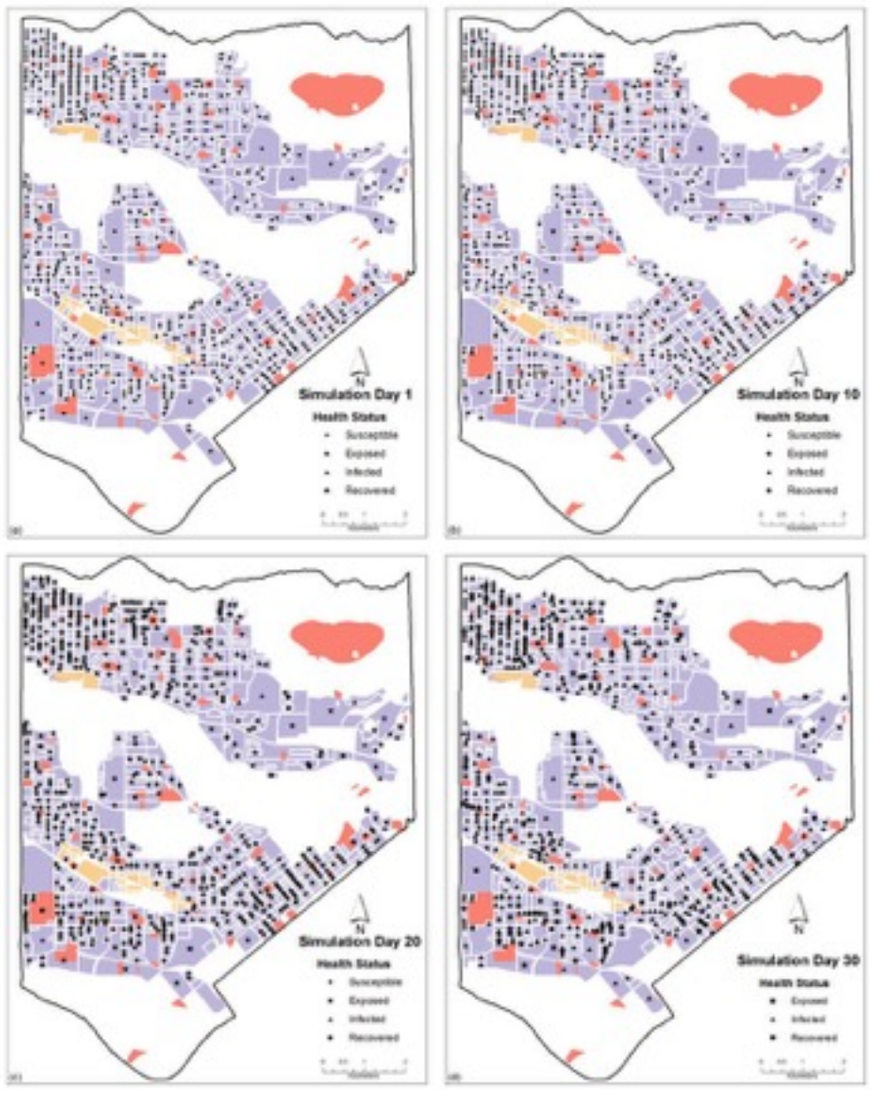


Why use ABMs?

- Nano-sheep problem
- Allows for greater complexity
- Builds intuition



Burke et al. (2006) Acad. Emer. Med..



Perez & Dragicevic (2009) Int J. Health Geo.

Relevance to social science?

Complex Contagions and the Weakness of Long Ties[1]

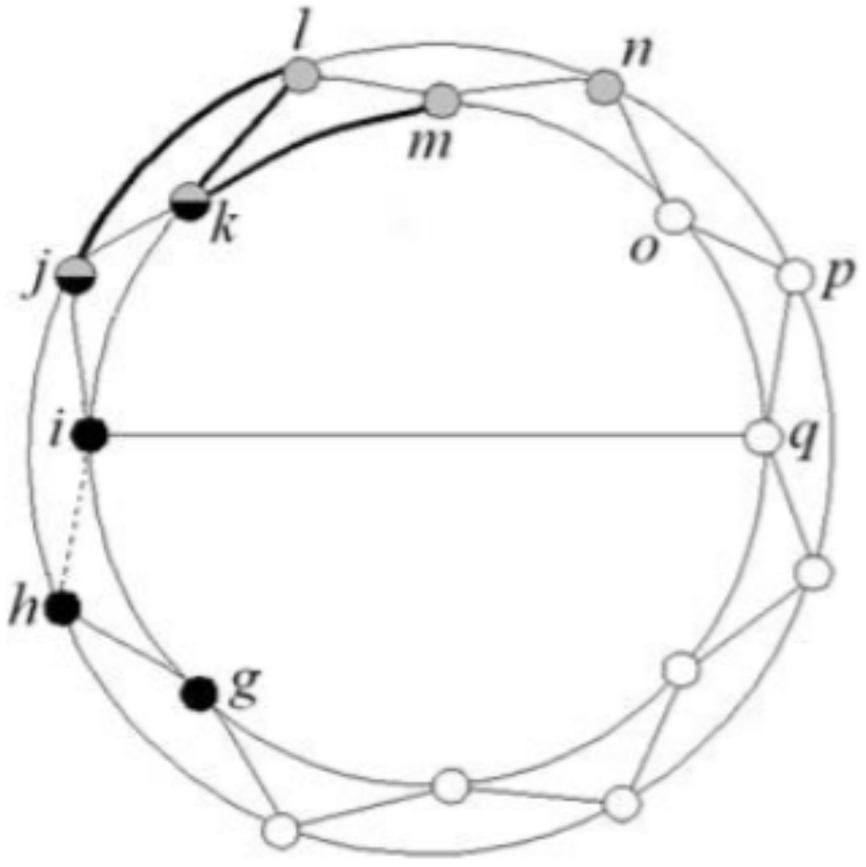
Damon Centola
Harvard University

Michael Macy
Cornell University

The strength of weak ties is that they tend to be long—they connect socially distant locations, allowing information to diffuse rapidly. The authors test whether this "strength of weak ties" generalizes from simple to complex contagions. Complex contagions require social affirmation from multiple sources. Examples include the spread of high-risk social movements, avant garde fashions, and unproven technologies. Results show that as adoption thresholds increase, long ties can impede diffusion. Complex contagions depend primarily on the width of the bridges across a network, not just their length. Wide bridges are a characteristic feature of many spatial networks, which may account in part for the widely observed tendency for social movements to diffuse spatially.

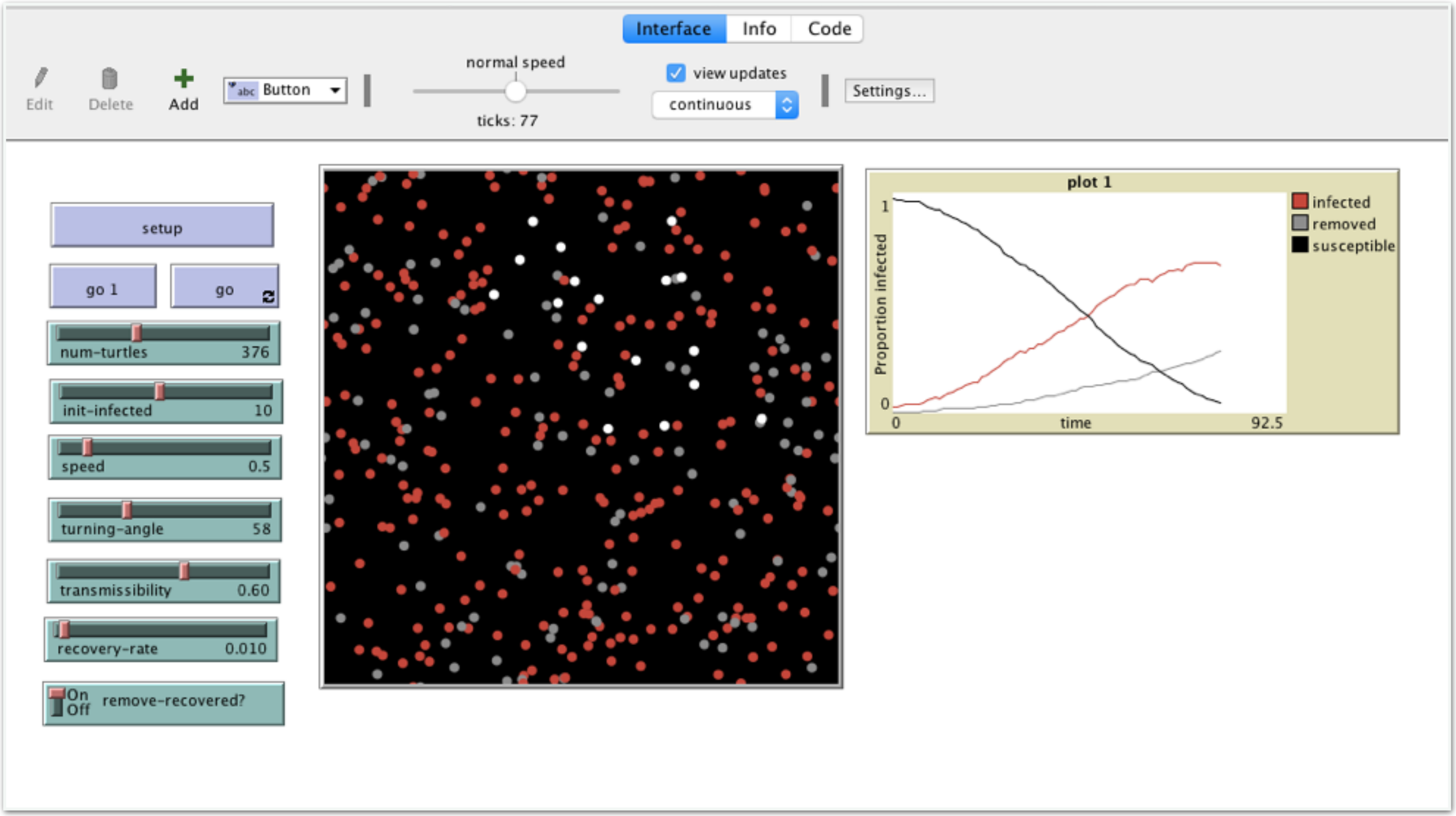
Most collective behaviors spread through social contact. From the emergence of social norms (Centola, Willer, and Macy 2005), to the adoption of technological innovations (Coleman, Katz, and Menzel 1966), to the growth of social movements (Marwell and Oliver 1993; Gould 1991, 1993; Zhao 1998; Chwe 1999), social networks are the pathways along which these "social contagions" propagate. Studies of diffusion dynamics have demonstrated that the structure (or topology) of a social network can have

[1] The authors wish to express their gratitude to the National Science Foundation for support of this research through Human and Social Dynamics grant SES-0432917 and through an IGERT fellowship for the first author. We also thank the Robert Wood Johnson Foundation for support of the first author. We thank Duncan Watts, Steve Strogatz, Jon Kleinberg, Vlad Barash, Chris Cameron, and Stephen Moseley for useful comments and suggestions. Direct correspondence to Damon Centola, Institute for Quantitative Social Science, 1730 Cambridge Street, Harvard University, Cambridge



Models as scaffolds

Some questions for a simple model



```
turtles-own [
  infected?
  removed?
]


to setup
  clear-all
  setup-turtles
  setup-infected
  reset-ticks
end

to setup-turtles
  create-turtles num-turtles [
    set shape "circle"
    set color white
    set infected? false
    set removed? false
    setxy random-xcor random-ycor
  ]
end

to setup-infected
  if init-infected > num-turtles
  [set init-infected 1]
  ask n-of init-infected turtles [
    set infected? true
    set color red
  ]
end
```

```
to go
```

```
end
```

```
to go
  if (count turtles with [infected?]) = num-turtles or
     (count turtles with [infected?]) = 0
  [ stop ]
```

```
end
```

```
to go
  if (count turtles with [infected?]) = num-turtles or
     (count turtles with [infected?]) = 0
  [ stop ]

  ask turtles [ ;; S -> I




  ]
```

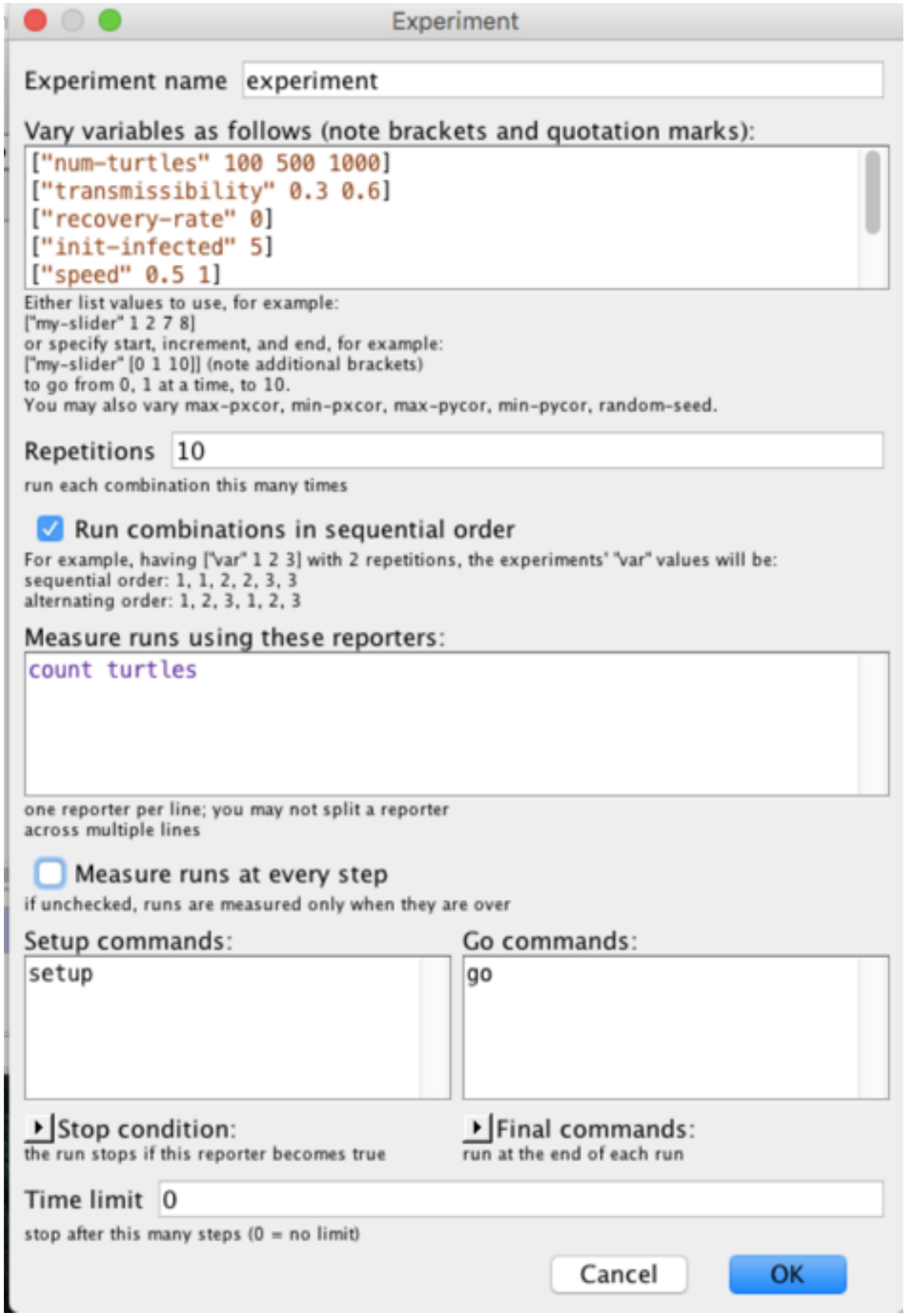
```
end
```

```
to go
  if (count turtles with [infected?]) = num-turtles or
     (count turtles with [infected?]) = 0
  [ stop ]

  ask turtles [ ;; S -> I
    if not removed? and
       (any? turtles with [color = red] in-radius 1) and
       (random-float 1 < transmissibility)
    [set infected? true]
  ]

  ask turtles with [infected? and color = red] ;; I -> S or R
  [


  ]



end
```

```
to go
  if (count turtles with [infected?]) = num-turtles or
     (count turtles with [infected?]) = 0
  [ stop ]

  ask turtles [ ;; S -> I
    if not removed? and
       (any? turtles with [color = red] in-radius 1) and
       (random-float 1 < transmissibility)
    [set infected? true]
  ]

  ask turtles with [infected? and color = red] ;; I -> S or R
  [
    if random-float 1 < recovery-rate [
      set infected? false
      ifelse remove-recovered?
        [set color gray
          set removed? true
        ]
        [ set color white ]
    ]
  ]

  ask turtles [ ;;move
    if infected? [set color red]
    left random turning-angle
    right random turning-angle
    fd speed
  ]

  tick
end
```

Batch Runs

BehaviorSpace



Statistical tests for simulation data

- Statistical tests are important for comparing model results to empirical results.
- However, statistical tests are **rarely** appropriate for identifying an effect within the model itself.
  - A statistical test creates a model of the data generation process
  - However, you already have such a model! The solution is to run enough simulations to obtain limiting distributions.

- Remainder of today:
  - Modeling Challenge #2
- Tomorrow:
  - Modeling cooperation and social evolution