Emergence

EES 4760/5760

Agent-Based and Individual-Based Computational Modeling

Jonathan Gilligan

Class #9: Wednesday, September 17 2025

Team Projects

Team Projects

- Team Project assignments are posted in the course web site, on the “Projects” page.
- For next Wednesday (Sept. 24): In addition to the assigned reading, read the ODD of the model you will work on for your team project. You will spend some time in class working with your partner(s) to start turning the ODD into a working NetLogo model.

Emergence

Emergence

- Download and open the “modified flocking model” from https://ees4760.jonathangilligan.org/models/class_09/modified_flocking.nlogo
- It’s easiest if you right-click on the link and choose “Save As” and save the model in a folder on your computer.

Flocking Model

Flocking Model

- Play with the model.
  - Adjust the parameters and see how they change the flocking behavior

About the Flocking Model

- Invented by Craig Reynolds in 1987
  - Reynolds, C. (1987). “Flocks, herds and schools: A distributed behavioral model”. *Proc. 14th annual conference on Computer graphics and interactive techniques*. Association for Computing Machinery. pp. 25???34. doi:10.1145/37401.37406
  - Used to generate flocks of bats and swarms of penguins for *Batman Returns* (1992)

Batman Returns: Bat Flock

Batman Returns: Penguin Army

Overview

- Entities:
  - Birds: state-variables `flockmates`, `nearest-neighbor`
- Process:
  - Each bird identifies its `flockmates`
  - Each bird adjusts its direction
  - Each bird moves forward one patch

Design Concepts

- **Emergence:**
  - Large flocks emerge from each bird acting independently, looking only at nearby birds.
- **Adaptation:**
  - If the `nearest-neighbor` is too close, the bird `separates` by turning away from it.
  - Otherwise, the bird:
    1. `aligns`: turns toward its `flockmates`
    2. `coheres`: turns slightly toward the direction the rest of its `flockmates` are flying.
- **Sensing:**
  - The bird can only see a certain distance (`vision`)
- **Interaction:**
  - Each bird interacts with its `flockmates`

Submodels

- `find-flockmates`:
  - `flockmates` are all birds within `vision` distance (“`vision range`” flock type)
  - Alternate interactions:
    - `flockmates` are the 6 nearest birds, regardless of distance (“`six closest`”)
    - `flockmates` is the one nearest bird, regardless of distance (“`closest`”)
- `separate`: Turn away from `nearest-neighbor` by up to `max-separate-turn` degrees
- `align`: Turn toward center of `flockmates` by up to `max-align-turn` degrees
- `cohere`: Turn toward average direction `flockmates` are flying, by up to `max-cohere-turn` degrees

Observations:

- How to measure flock formation?

```
count turtles with [any? flockmates]
mean [count flockmates] of turtles
mean [min [distance myself] of other turtles] of turtles
standard-deviation [heading] of turtles
```

Digression: Selecting Turtles

- Selection primitives:
  - Selecting agent-sets
    - `n-of`, `min-n-of`, `max-n-of`, `other`,
    - `turtles-on`, `turtles-at`, `turtles-here`, `at-points`
    - `in-radius`, `in-cone`,
    - `with`, `with-min`, `with-max`
  - Selecting individual turtles
    - `one-of`, `min-one-of`, `max-one-of`
    - (may return `nobody`)
  - Look at `Agentset` category in NetLogo dictionary
- Be careful:
  - Some primitives expect agent-sets
  - Others expect individual turtles or patches.

Practice Selecting Turtles

- Turn 5 random turtles red:

```
ask n-of 5 turtles [ set color red ]
```

- Now for each of those turtles:
  - Select all the turtles within a radius of 5
    - and turn them green

```
ask n-of 5 turtles with [color = red] [
  ask other turtles in-radius 5 [
    set color green
    ]
  ]
```

- Now ask each green turtle to calculate the distance to the closest red turtle

```
ask turtles with [color = green] [
  show min [distance myself] of turtles with [color = red]
  ]
```

- Now get the average over all the green turtles of the distance to the closest red turtle

```
show mean [
  min [distance myself] of turtles with [color = red]
  ] of turtles with [color = green]
```

Experiments

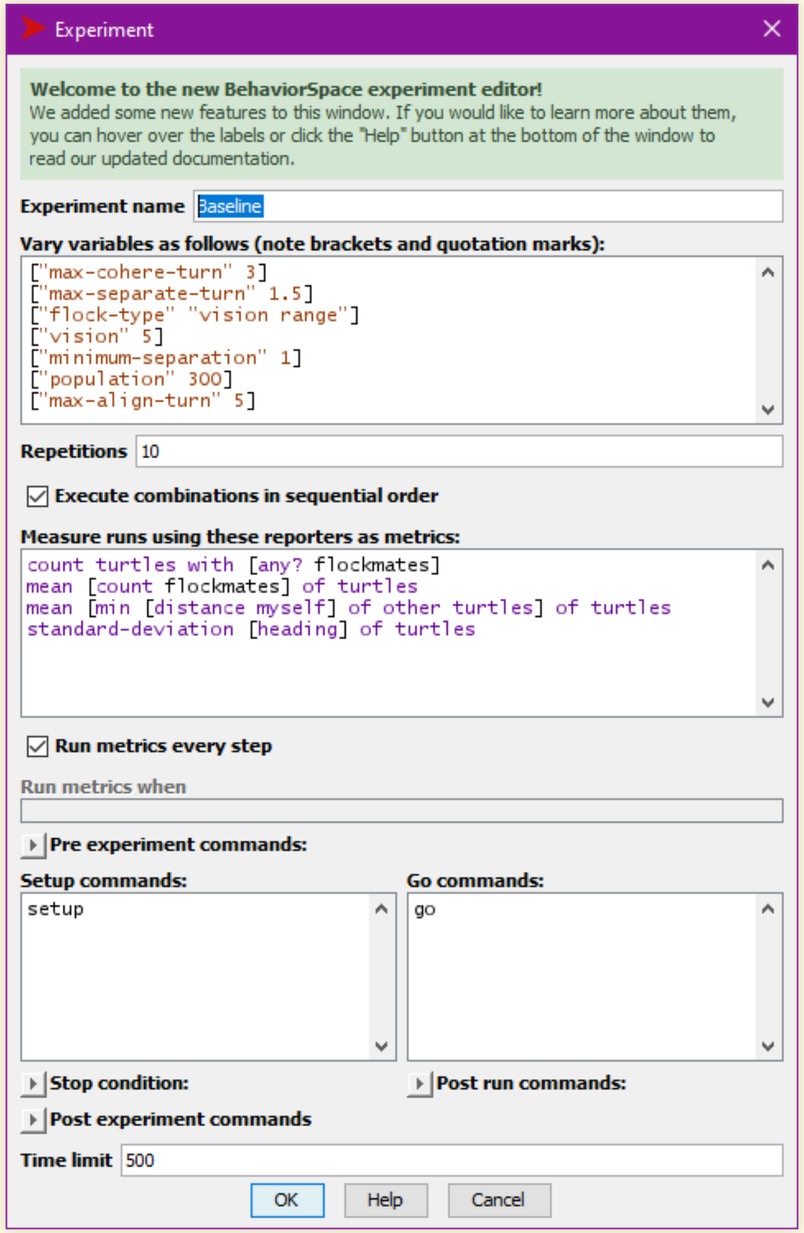
Experiments

- Open the Behaviorspace experiment called “**Baseline**”
  - Measures of flocking:

```
count turtles with [any? flockmates]
mean [count flockmates] of turtles
mean [min [distance myself] of other turtles] of turtles
standard-deviation [heading] of turtles
```

  - Change one parameter (`vision`, `max-cohere-turn`, or `max-align-turn`) and see how it affects flocking.

Experiments



Experiments

- Open the Behaviorspace experiment called “**Baseline**”
  - Measures of flocking:

```
count turtles with [any? flockmates]
mean [count flockmates] of turtles
mean [min [distance myself] of other turtles] of turtles
standard-deviation [heading] of turtles
```

  - Change one parameter (`vision`, `max-cohere-turn`, or `max-align-turn`) and see how it affects flocking.
  - Run the experiment: remember to turn off `Update view` and `Update plots and monitors`
- Next, duplicate “**Baseline**” and call it “**Flock Type**”
  - vary that parameter while also varying the `flock-type`
- Next, duplicate “**Baseline**” and call it “**Multiple**”
  - vary more than one parameter (e.g., `vision` and `max-cohere-turn` or `max-align-turn`)
- Use the `analyze_behaviorspace` app at https://ees4760.jonathangilligan.org/analyze_behaviorspace to graph the output from your BehaviorSpace experiments.