

Introduction to Agent-Based Modeling for Social Scientists

Day 6

A Practical Guide to Building Models

By: Aaron Bramson

INTRODUCTION TO AGENT-BASED MODELING

A Practical Guide to Building Models in Netlogo

Creating the Hatfield and McCoy Model



A Practical Guide to Building Models in Netlogo

Creating the Hatfield and McCoy Model

What Are We Modeling?

- Can Inter-Group Marriage End Inter-Group Conflict?
- Two Feuding Factions (the Hatfield and McCoy Families)
- Rules for Marriage, Fighting, Birth, and ...?
- Measure(s) of Conflict Level and Tolerance
- Visualization(s) of the System Dynamics

Modeling Considerations

- Keep it Simple (after all it's just an exercise)
- Not Really Spatially Explicit
- Don't Worry about Family Structure
- Forget the Effects of Economy
- No Other Kinds of Agents (e.g. no Police or Villagers)

A Practical Guide to Building Models in Netlogo

Creating the Hatfield and McCoy Model

Start from the Beginning

- Open a New (Blank) Netlogo Application
- Open ANOTHER Netlogo Application (to look at other code)
- Open a Browser to the Netlogo User Manual
- Save the New Model as 'Hatfields and McCoys.nlogo'
- Delete Text from the Information Tab

First Step: Create the Bare Skeleton of a Model

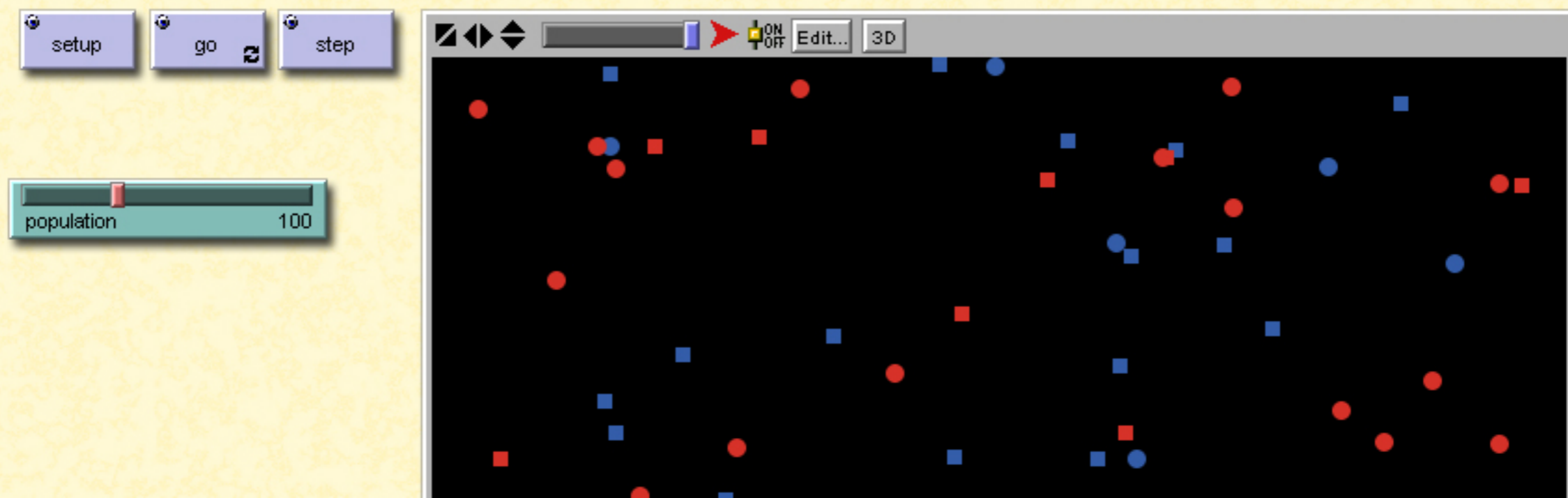
- Create a **population** slider (up to 300)
- Create Buttons for **setup**, **step**, and **go**
- Create Simple Methods for **setup** and **go**
- Create a **global variable**, **ticker**, and **monitor**, for **time**
- Set **patch-size** to **10** and **screen-size-x** and **y** to **30**

A Practical Guide to Building Models in Netlogo

Creating the Hatfield and McCoy Model

Creating Agents for the Model

- Create Three Breeds: **hatfields**, **mccoys**, & **tolerants**
- Make Hatfields and McCoys each **Half** the Population
- Set Hatfields Blue and McCoys Red (and Tolerants Yellow)
- Create a **turtles-own** variable for **alignment** (0-9)
- Give Agents Sex and Set Shapes by Sex
- Set Random Initial Locations for each Agent (**Hatch**)



A Practical Guide to Building Models in Netlogo

Creating the Hatfield and McCoy Model

Agent Behavior: With Whom Do Agents Interact?

- Random Agents - Physical Location Irrelevant?
- Agents on the Same Patch (using larger patches)?
- Agents within a Radius (one or all agents nearby)?
- Agents in Front of the Active Agent (using heading)?
- ...I recommend **one-of turtles in radius radius** (# from slider).

Agent Behavior: Actions Depend on Types

- Create Conditionals for All 21 Types of Agents (nested **ifelse**)
- Use Systematic Properties to Limit Necessary Rules
- Still Need Nesting, Consider Optimal Nesting Order
- Write a Separate Method for Each Behavior
- Take Baby Steps: Minimize Change between Runnable Versions

A Practical Guide to Building Models in Netlogo

Creating the Hatfield and McCoy Model

Improving Code: Condensing and Refining

```
to setup2
```

```
ca
```

```
create-custom-turtles population
```

```
setxy random-xcor random-ycor
```

```
ifelse random 100 > 50
```

```
  [ set breed hatfields
```

```
    set color blue
```

```
    set alignment 0.0 ]
```

```
  [ set breed mccoys
```

```
    set color red
```

```
    set alignment 9.0 ]
```

```
set sex random 2
```

```
set age random 10
```

```
ifelse sex = 0
```

```
  [set shape "circle"]
```

```
  [set shape "square"]
```

```
set label sex
```

```
]
```

```
end
```

• Creates Turtles All at Once

• Set Ratio by Making This a Parameter

• Breed Properties Set Here

• All-Turtle Properties Set Here

• Conditional Properties Set with `ifelse`

• Always Check Your Work

A Practical Guide to Building Models in Netlogo

Agent Behavior: What Are the Effects of Interaction?



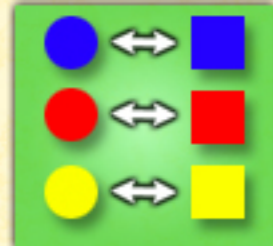
Both Alignments -1 (min 0)



Both Alignments +1 (max 9)



Average Alignments



Birth (and Alignment Adjustment)



Fight! (Probability of Death)



Intergroup Marriage:
Birth and Average Alignments



What to Do?

Average Alignments, Birth, Fight

A Practical Guide to Building Models in Netlogo

Creating the Hatfield and McCoy Model

Improving Code: Condensing and Refining

```
to go2
```

```
ask turtles [
```

```
  rt random 90 - 45
```

```
  fd 1
```

```
  without-interruption [
```

```
    let agent1 self
```

```
    if any? turtles in-radius radius [
```

```
      let agent2 one-of turtles in-radius radius
```

```
      ;;rule for making babies
```

```
      if sex-of agent1 = 0 and sex-of agent2 = 1 [
```

```
        let baby-alignment find-average-alignment agent1 agent2
```

```
        make-baby baby-alignment
```

```
      ]
```

```
      ;;rule for fighting behavior
```

```
      if (sex-of agent1 = sex-of agent2) [
```

```
        ifelse (breed-of agent1 = breed-of agent2)
```

```
          [ reinforce-alignment agent1 agent2 ]
```

```
          [ fight agent1 agent2 ]
```

```
      ]
```

```
    ]
```

```
  ]
```

```
]
```

```
;;to prevent run-away models that freeze your computer
```

```
if count turtles > 500 [stop]
```

```
end
```

- Simulates Simultaneous Updating
- Temporarily Store the Activated Agent
- Eliminate “nobody” and “agent/agentset” errors
- Can't Use != for Babies. Why?
- Method/Function takes inputs
- Nested Conditions to Minimize Code
- Model/Measure Interaction

A Practical Guide to Building Models in Netlogo

Creating the Hatfield and McCoy Model

Improving Code: Condensing and Refining

```
;;rule for making babies
if sex-of agent1 = 0 and sex-of agent2 = 1 [
  let baby-alignment find-average-alignment agent1 agent2
  make-baby baby-alignment
]
;;rule for fighting behavior
if (sex-of agent1 = sex-of agent2) [
  to-report find-average-alignment [agent1 agent2]
    let average-alignment ((alignment-of agent1 + alignment-of agent2) / 2 )
    report average-alignment
  end
end
```

```
to make-baby [new-alignment]
  hatch 1 [
    set alignment new-alignment
    set sex random 2
    ifelse sex = 0
      [set shape "circle"]
      [set shape "square"]
    set label sex
  ]
end
```

- Call Outside Functions to Simplify Code and Improve Reuse
- Can Perform Action or Calculation
- Can Take Multiple Inputs of Different Kinds