Introduction to agent-based modelling for social scientists

Dr. J. Kasmire
UK Data Service

16 January 2020

UK Data Service

ABM for social scientists – webinar series

- **ABM: An Intro**
  - **– Jan 16, 2020**
- ABM: Adding Data
  - – Jan 30, 2020
- ABM: Experiments and Output
  - – Feb 13, 2020

Table of Contents

Top-down

- A problem can be understood as “top-down” if it occurs within a system that is ‘whole’, ‘well-understood’ and with central control or structure.
- The ‘cause’ can be isolated to constituent parts, defined interactions, or specific problem areas.
- Top-down problems suit classic scientific method, which focusses on isolation, prediction, repetition, etc.

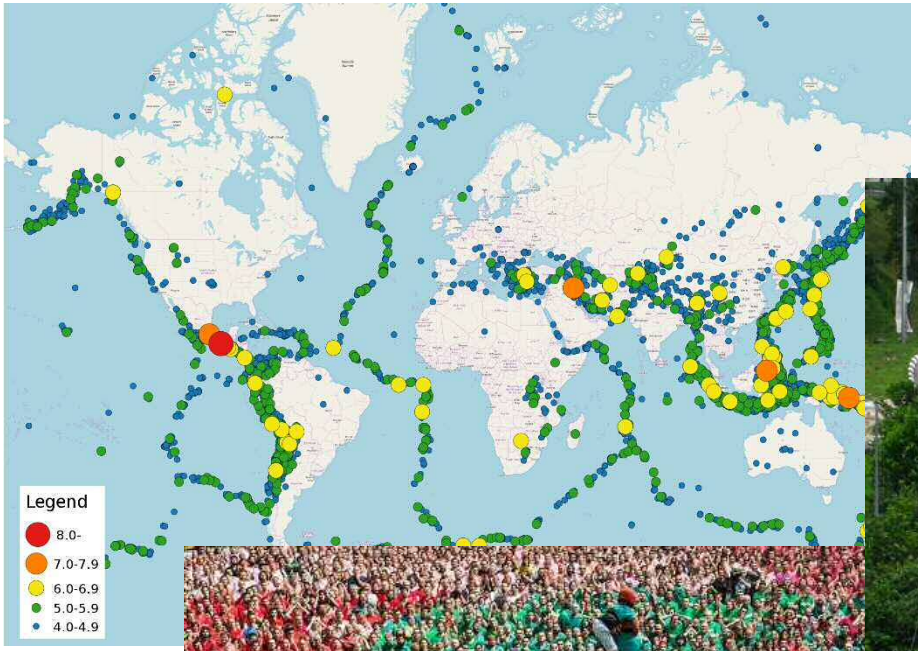
- Top-down examples

UK Data Service

• Bottom-up

- A problem is “bottom-up” if it happens within a poorly-understood, open system with vague parts and interactions and with partial or decentralised control.
- Not reasonable to assume reducibility, linearity, time/context-independence, or closed/simplicity.
- Understand the problem by recreating it though:
  - Deterministic low-level rules,
  - Finite parameters,
  - Varying (including extreme or counter-factual) conditions.
- Not easy to apply classic scientific method.

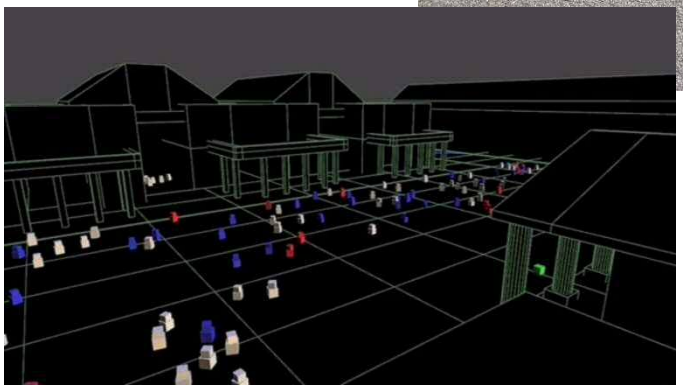
- Bottom-up examples



UK Data Service

• Agents – Three common views

- Artificial Intelligence –
  - Autonomous
  - Individual
  - Learners and/or problem solvers
- Multi-agent systems –
  - Distributed control
  - Specific problems
- Agent-based models –
  - Multiple
  - Interacting
  - Rule-based



- Agent-based models have:

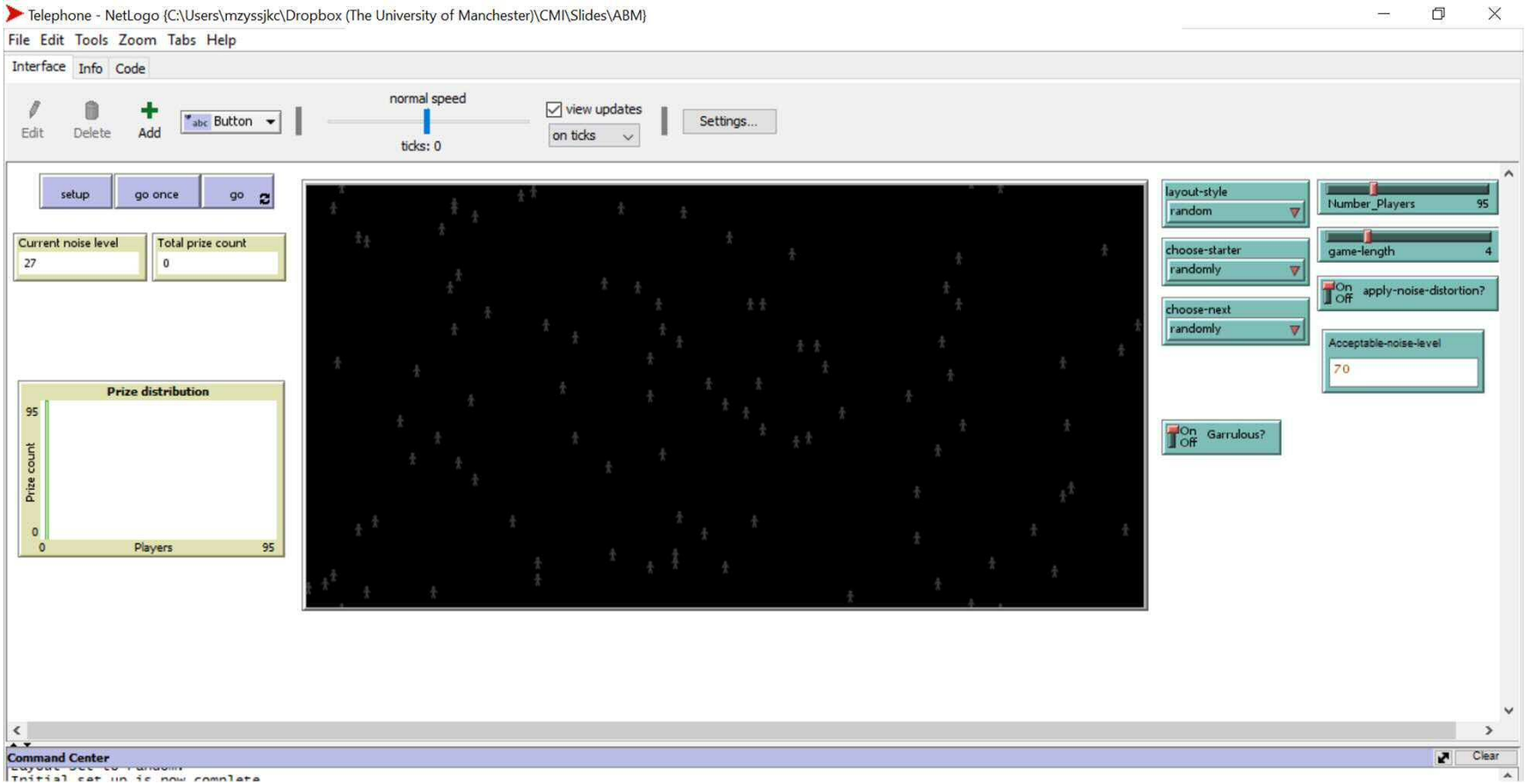
- A simulated world of varying richness, including time
- Objects and/or agents that have:
- States
- Rules

- Example – a game of ‘Telephone’

• Agent-based models: The World

- Representing most anything (stock market, city, room, etc.)
- Proceeds through time in discrete ‘time steps’
- Has representations of potentially interesting factors (current selling price, weather, capacity, etc.)
- Is unique for each agent as it contains everything *else* (including the other agents)
- Has states composed of their features and contents, which may affect/are affected by agents, objects, itself, random factors, etc. according to the rules

• The World - Telephone Game

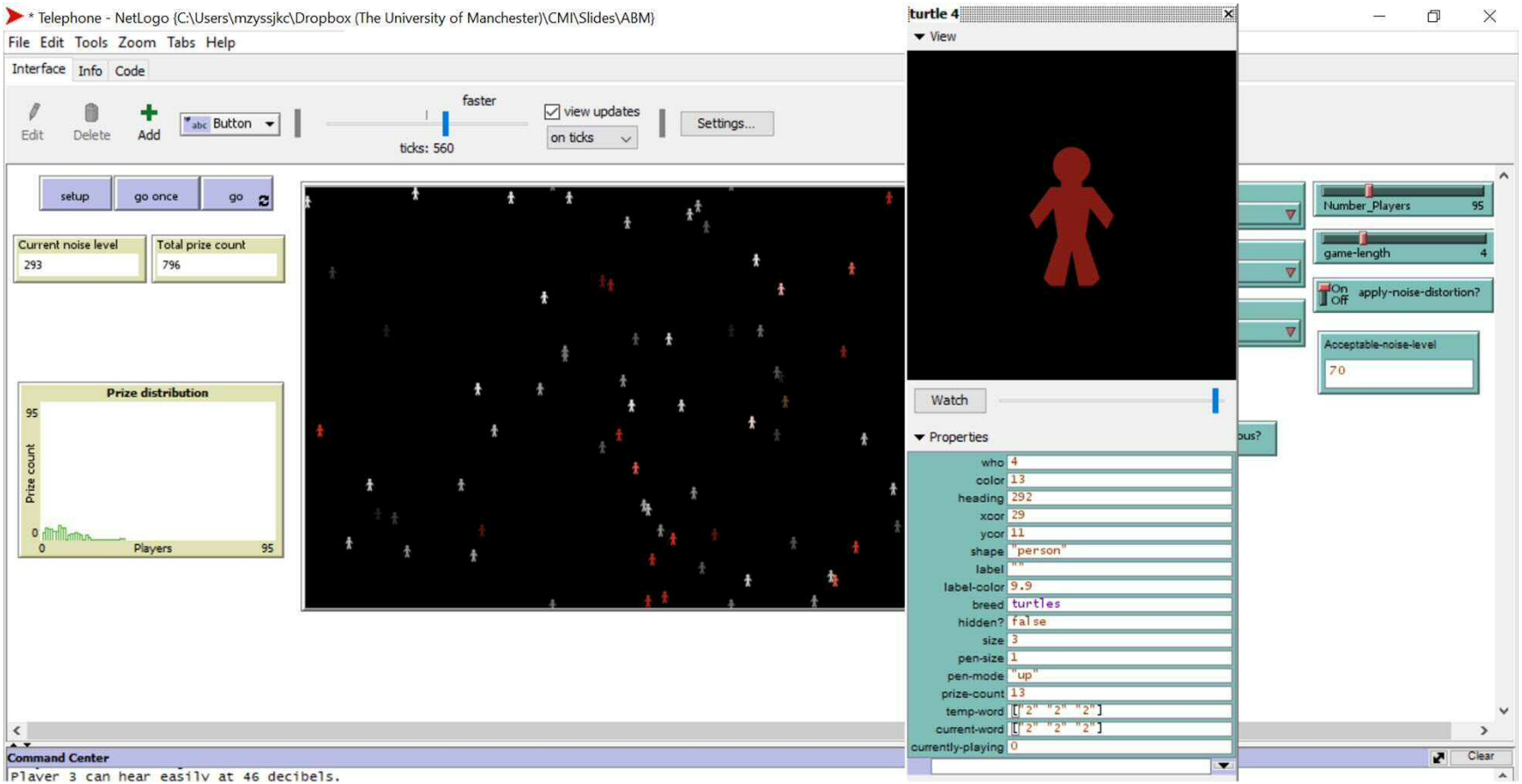


UK Data Service

• Agent-based models: The Agents

- Representing most anything (people, cars, ocean waves, etc.)
- Have representations of potentially interesting factors (money, age, colour, speed, etc.) that can be dynamic
- Are unique and behave uniquely
- Have states that affect/are affected by their unique world according to the rules
- Make decisions based on world, states, rules, other agents, etc.

• The Agents – Telephone Game

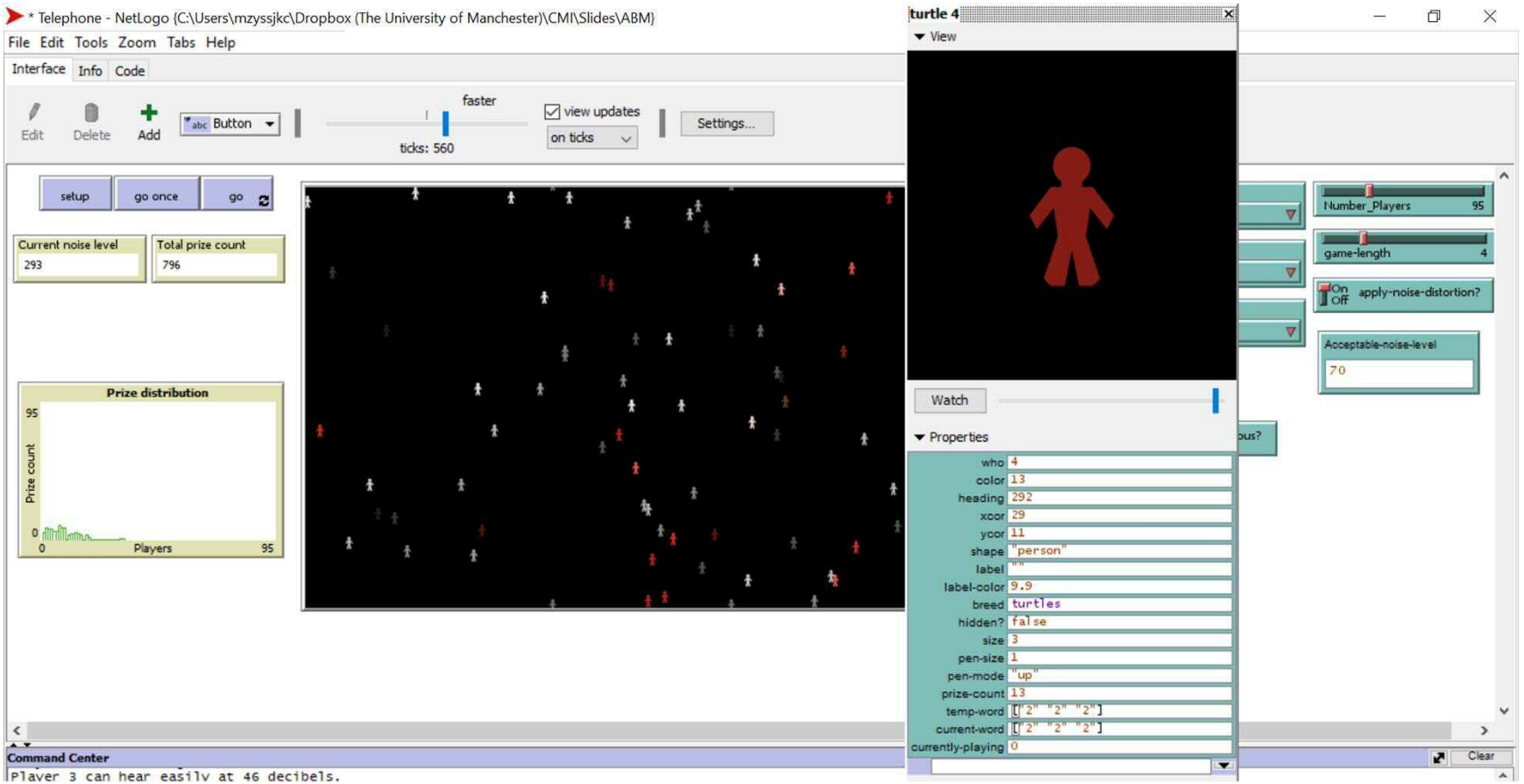


UK Data Service

• Agent-based models: States

- Sum of current features
- Can be dynamic or static
- Can be inputs, outputs, etc.
- Used to make decisions

• States – Telephone Game

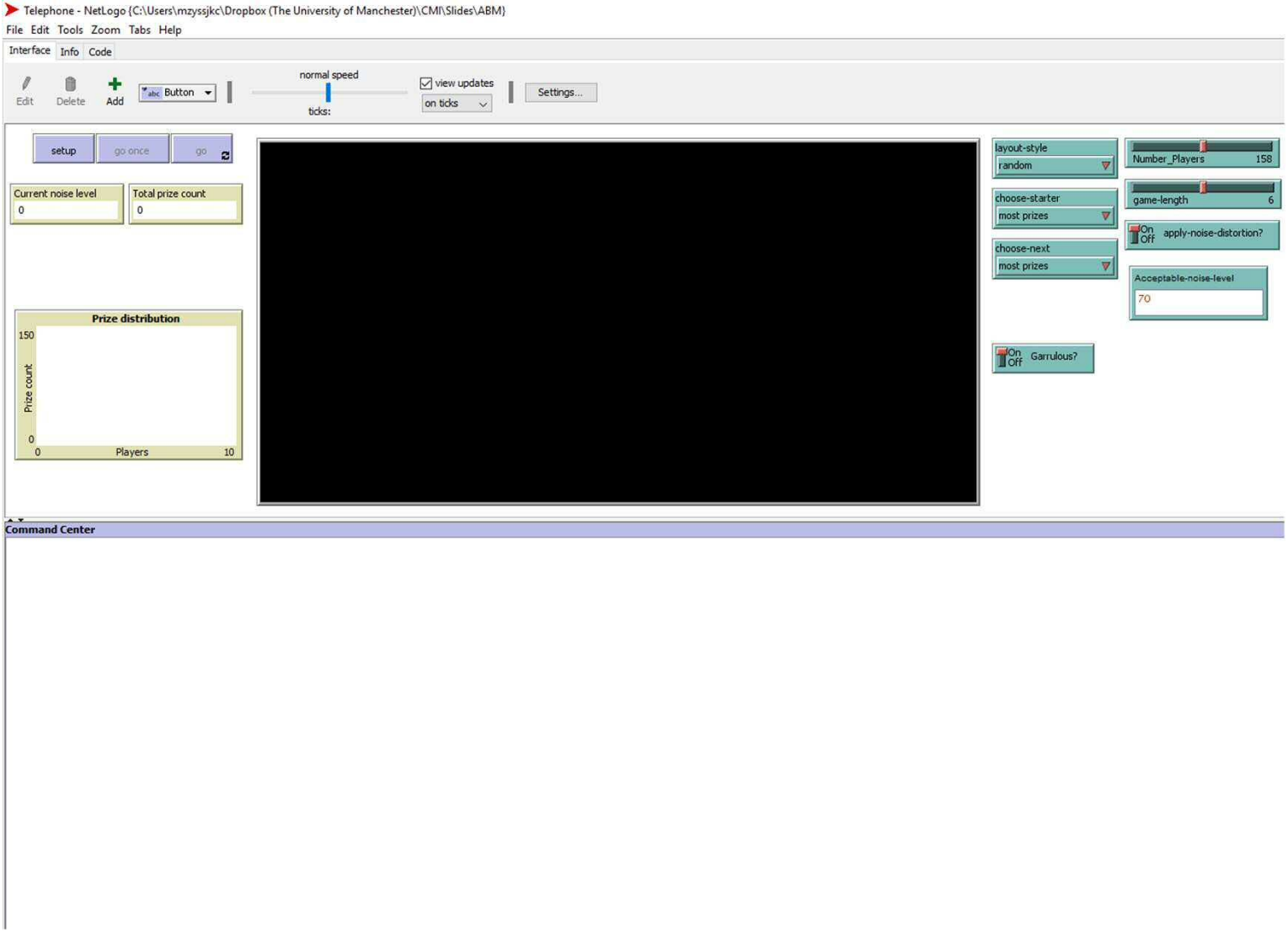


• Agent-based models: Rules

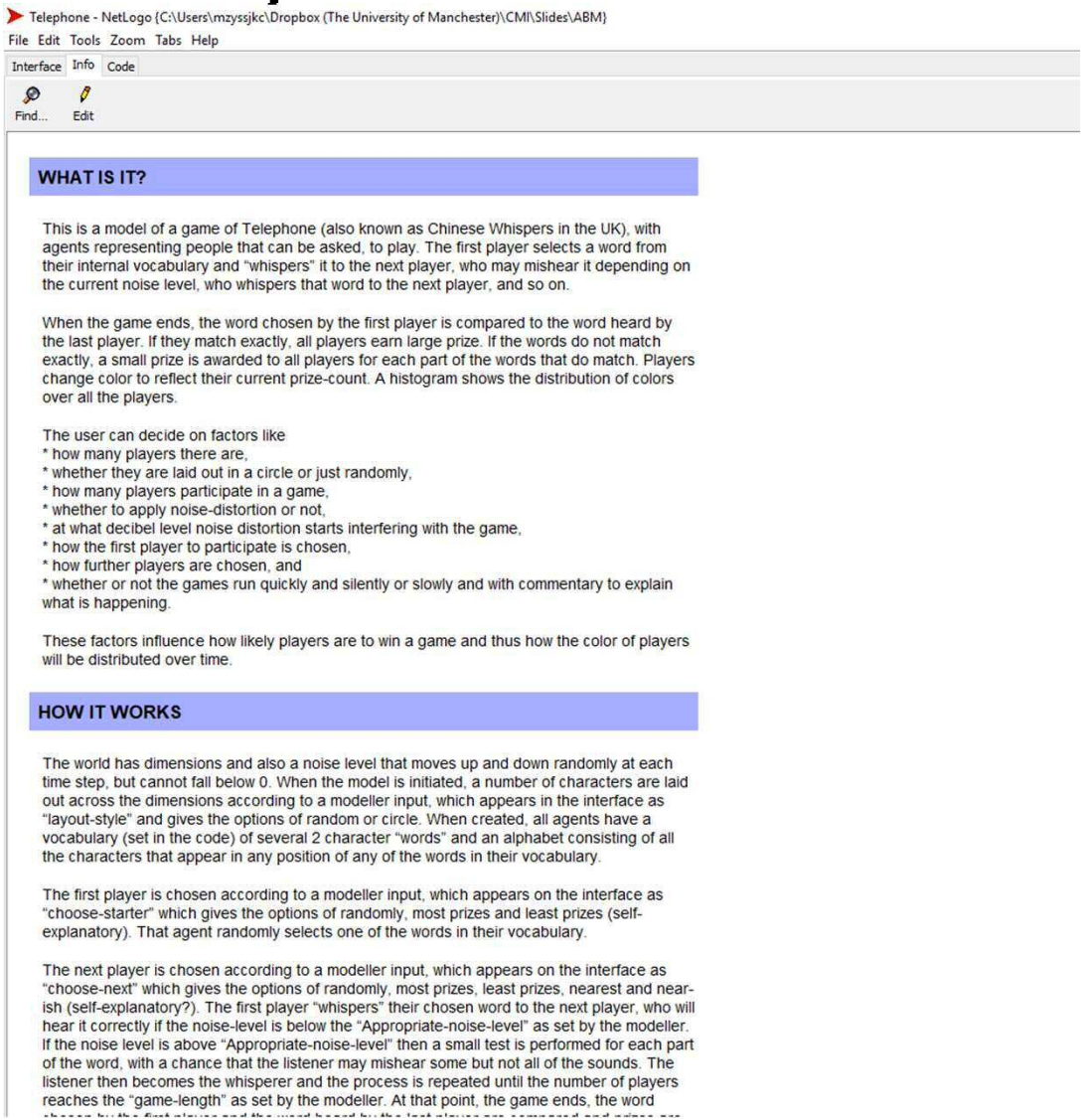
- Govern how states change or decisions are made
- Use model generated info, modeller choice, randomness
- Can be simple or complex (if-then, while, for each, etc.)

```
to go
  set noise-level noise-level + random 10 - random 10
  if noise-level < 0 [set noise-level 0]
  ifelse length current-turtle = 0
  [start-new-game]
  [ifelse length current-turtle = game-length
    [wait .5
     end-game]
    [play-game]]
  tick
end

to start-new-game
  ask turtles [set currently-playing 0]
  if choose-starter = "randomly" [ask-random]
  if choose-starter = "most prizes" [ask-most]
  if choose-starter = "least prizes" [ask-least]
end

to ask-random
  ask one-of turtles with [currently-playing = 0 ]
  [  if Garrulous? [print "Random player asked to play..."]
   participate]
end

to ask-most
  ifelse mean [prize-count] of turtles != 0 and any? turtles with [currently-playing = 0 and prize-count > mean [prize-count] of turtles]
  [ask one-of turtles with [currently-playing = 0 and prize-count > mean [prize-count] of turtles] [participate]]
    [ask-random]
end
```


• Telephone Game – The interface tab



• Telephone Game – The info tab



UK Data Service

• Telephone Game – The code tab

Telephone - NetLogo {C:\Users\mzyssjkc\Dropbox (The University of Manchester)\CMI\Slides\ABM}

File Edit Tools Zoom Tabs Help

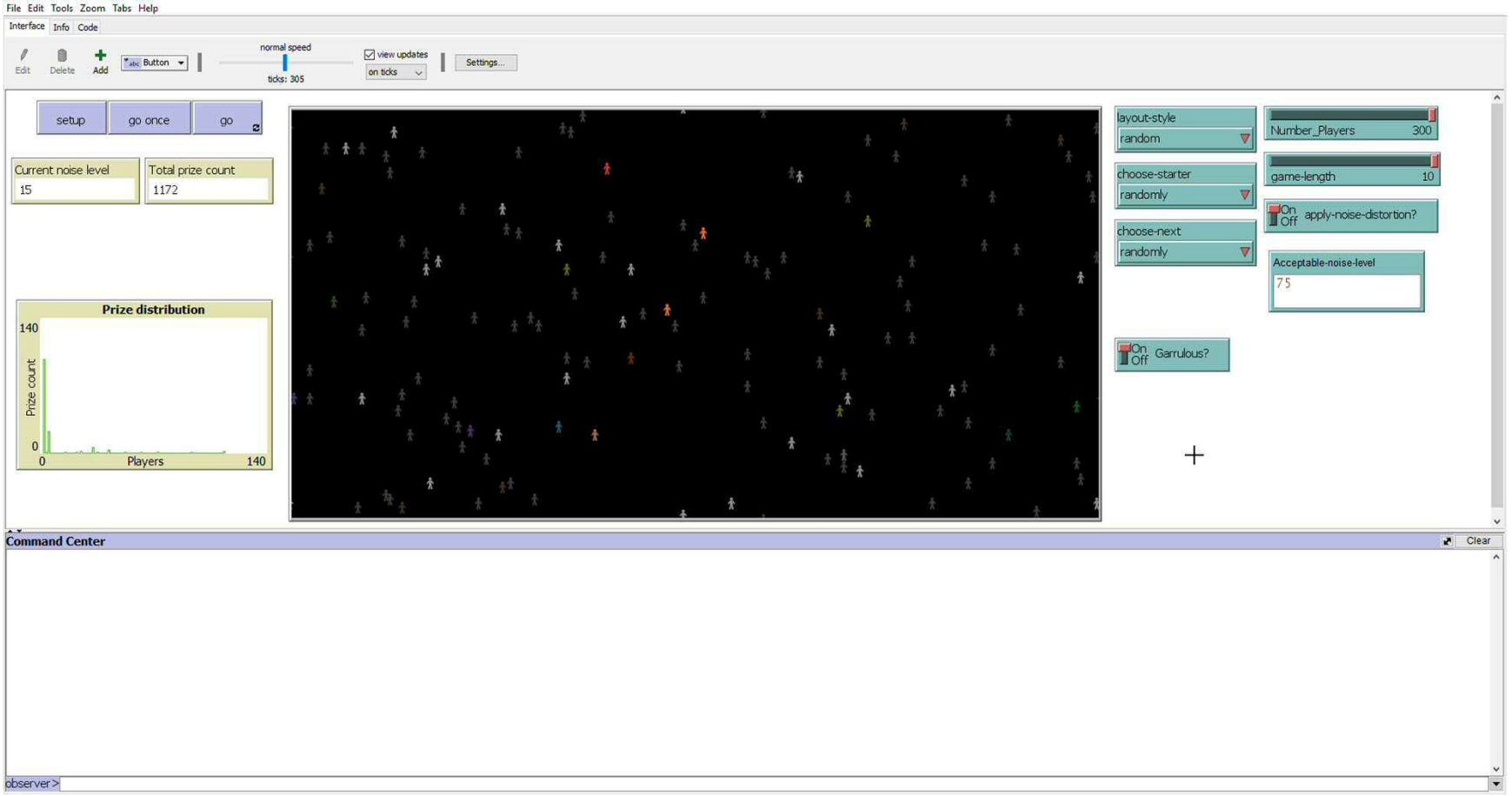
Interface Info Code

Find... Check Procedures Indent automatically

```
globals [ noise-level
 current-turtle
 vocabulary
 alphabet
]

turtles-own [
  prize-count
  temp-word
  current-word
  currently-playing
]

to setup
  clear-all
  ifelse Garrulous?
    [resize-world -100 100 -50 50]
    [resize-world -100 100 -100 100]
  set current-turtle []
  set alphabet ["a" "b" "C" "c" "d" "e" "f" "g" "o" "1" "2" "3" "4" "5"]
  set vocabulary [["a" "a" "a"] ["b" "b" "b"] ["C" "c" "c"] ["1" "1" "1"] ["2" "2" "2"] ["d" "3" "d"] ["f" "o" "o"] ["g" "e" "e"] ["g" "4" "5"] ["l" "o" "l"] ]
  set-default-shape turtles "person"
  ask patches with [ pxcor < -11 and pxcor > 11 ] [ set pcolor white]
  create-turtles Number_Players [
    set size 3                  ;; be easier to see
    set color 2
    __set-line-thickness 2
    set currently-playing 0
    set prize-count 0
    set current-word [""]]
  establish-layout
  set noise-level random 100
  reset-ticks
  if Garrulous? [print "Initial set up is now complete."]
end

to establish-layout
  ifelse layout-style = "random" [ask turtles [move-to one-of patches]]
      [layout-circle turtles 50]
  if Garrulous? [print (word "Layout set to " layout-style ".")]
end

to go
  set noise-level noise-level + random 10 - random 10
  if noise-level < 0 [set noise-level 0]
  ifelse length current-turtle = 0
  [start-new-game]
  [ifelse length current-turtle = game-length
    [wait .5
     end-game]
    [play-game]]
  tick
end

to start-new-game
  ask turtles [set currently-playing 0]
```

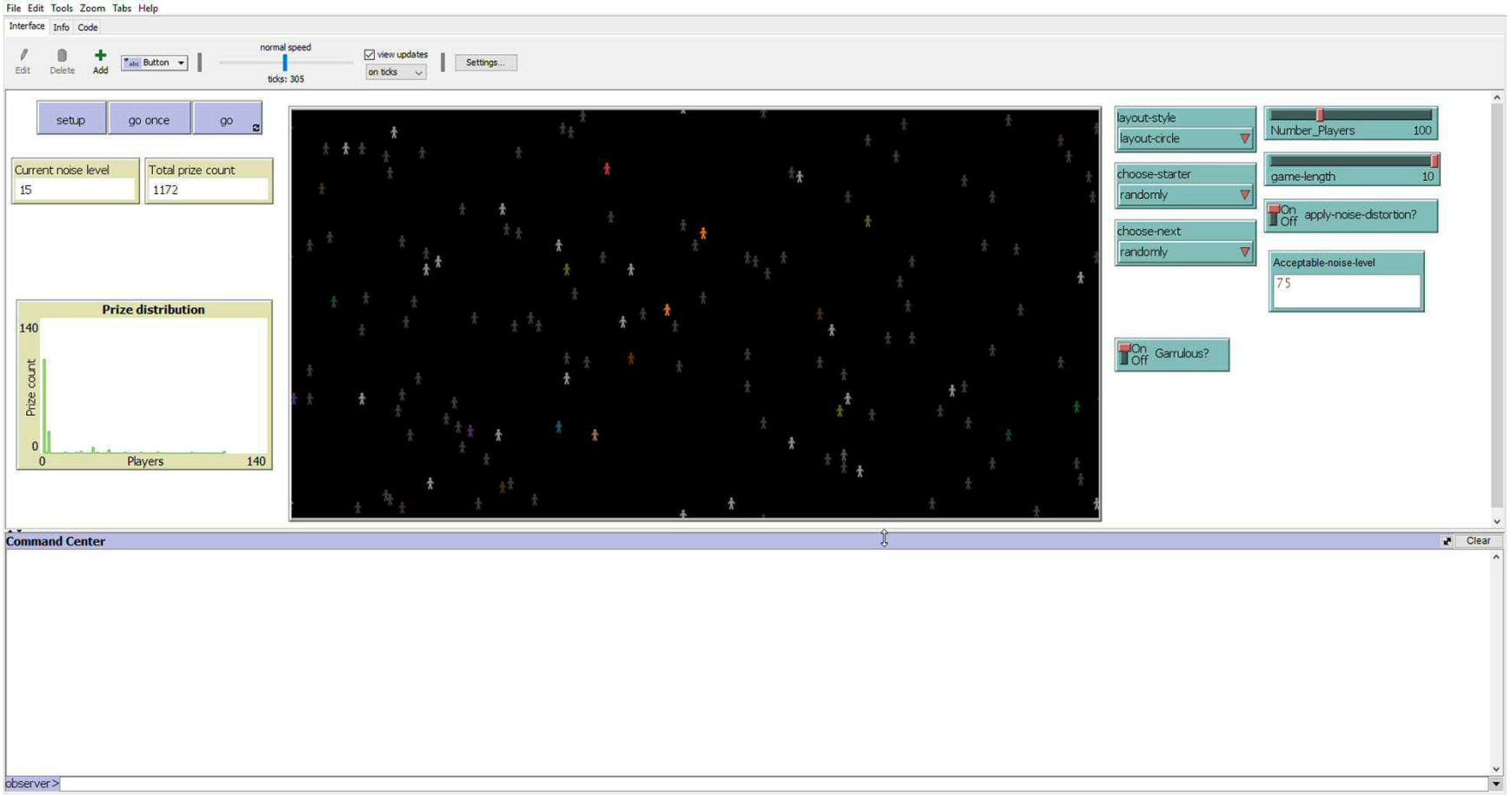
UK Data Service

• Telephone Game – Modeller choices



UK Data Service

• Telephone Game – Sliders

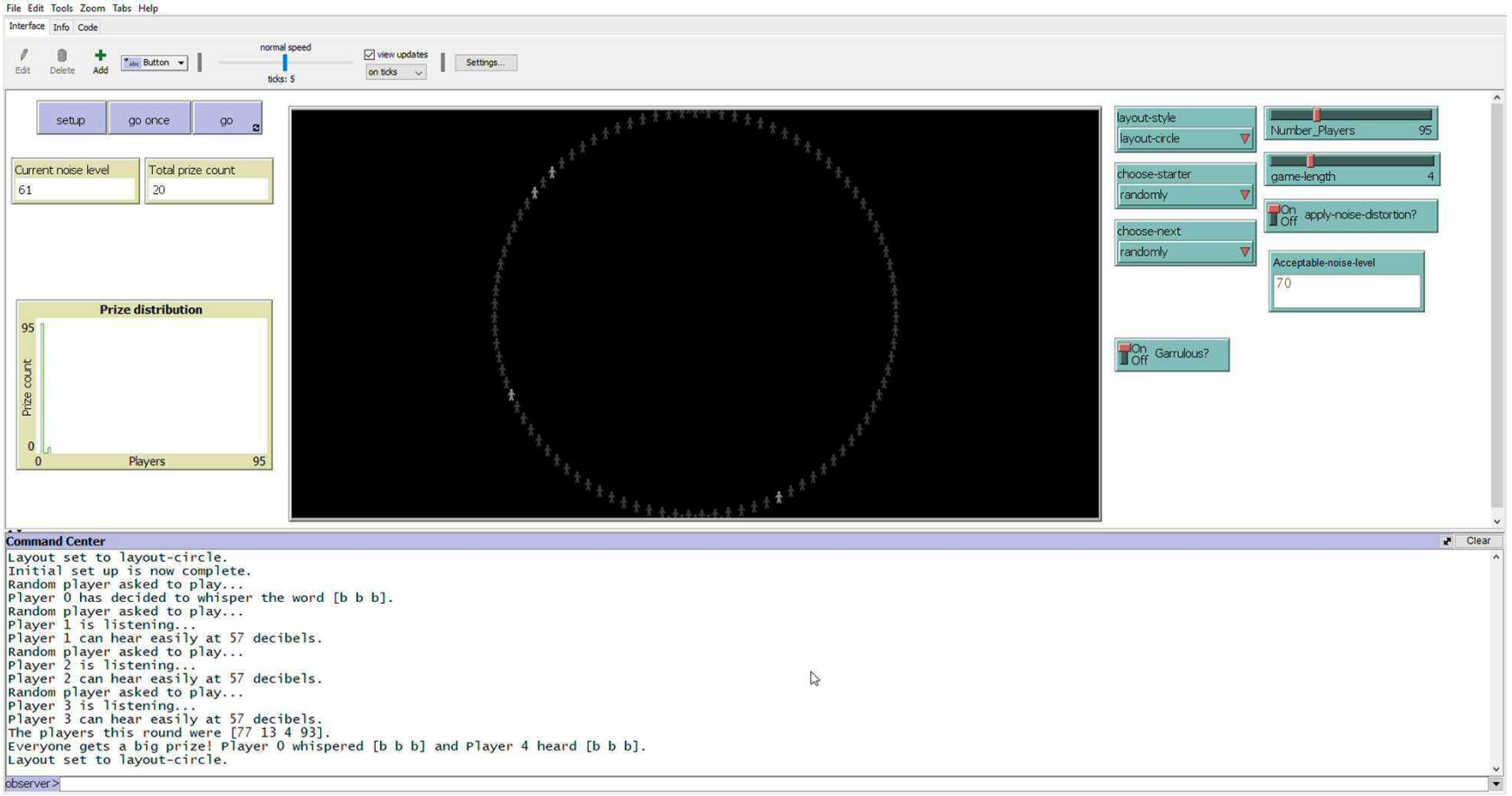


UK Data Service

• Telephone Game – Setup and Go

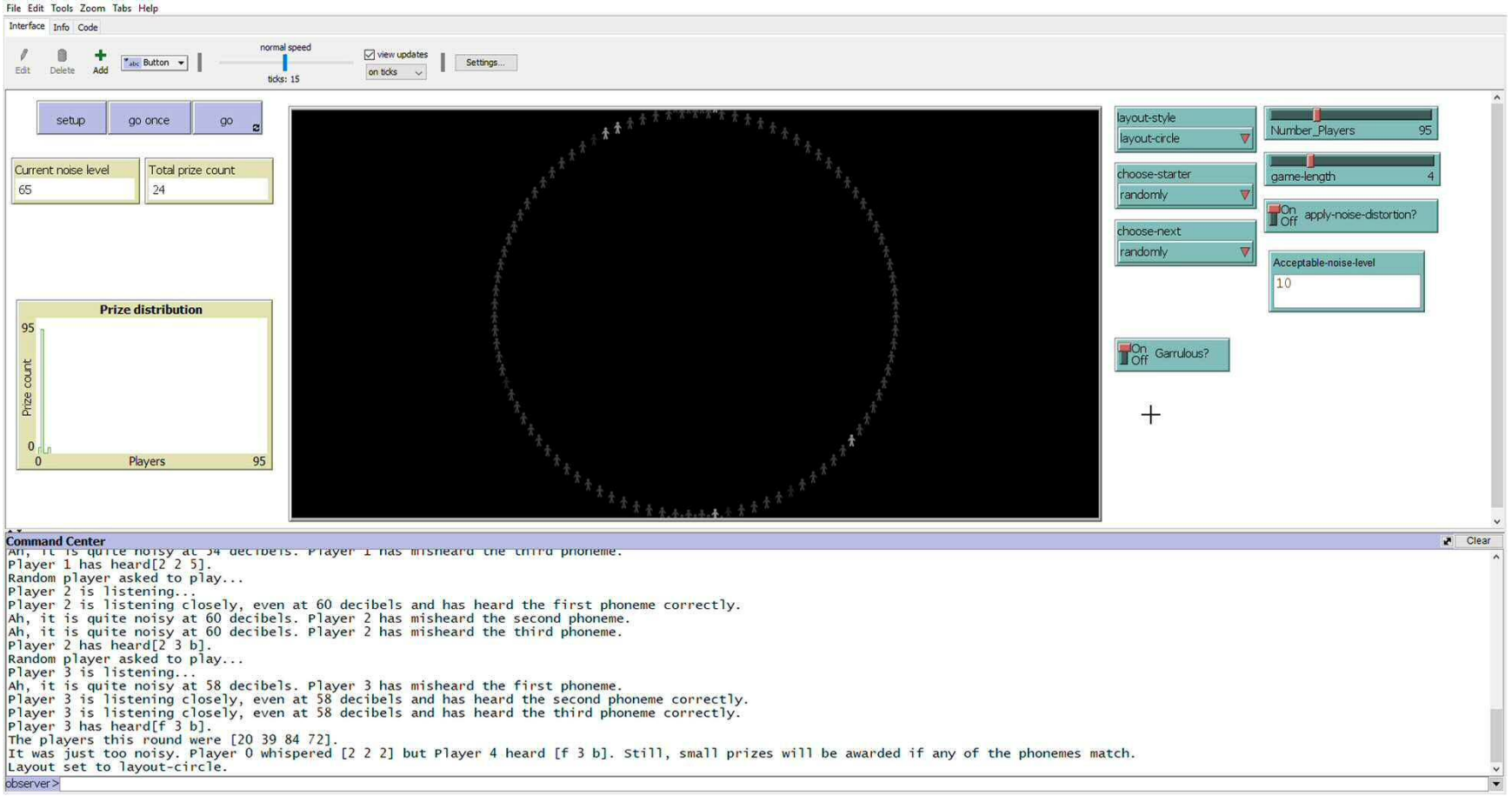
UK Data Service

• Telephone Game – Action

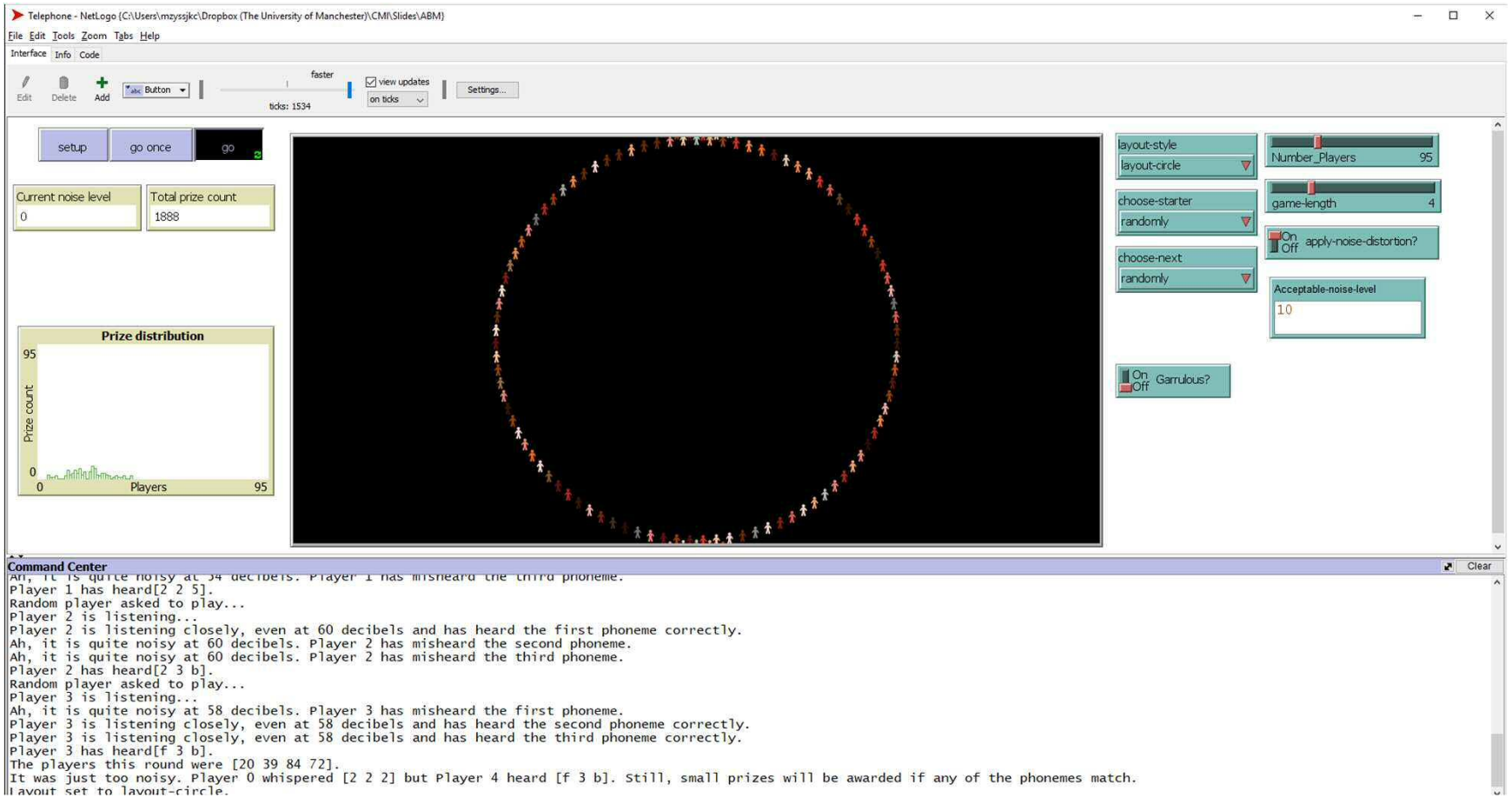


UK Data Service

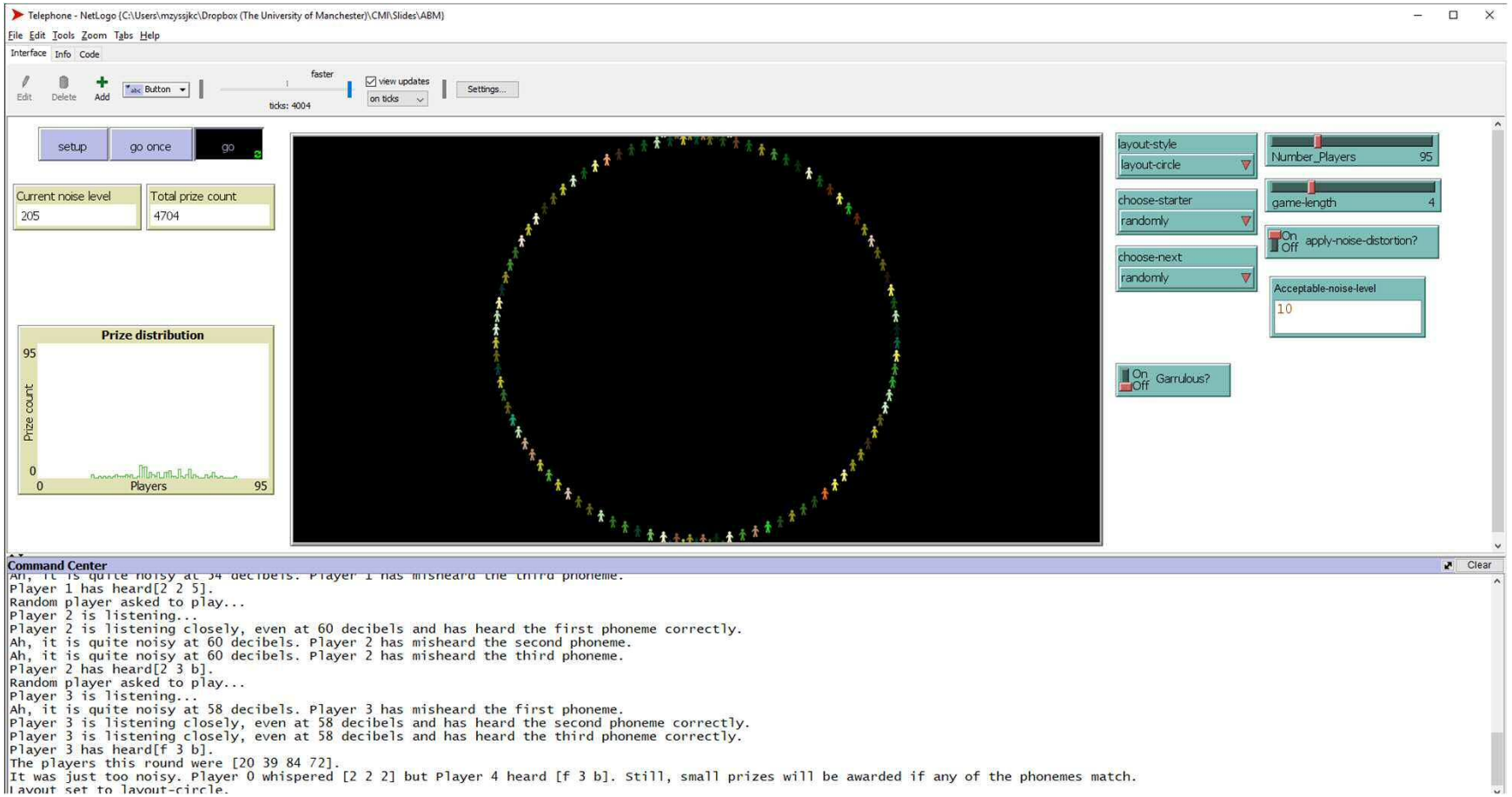
• Telephone Game – Go forever



- Telephone Game – After 1500 time steps



- Telephone Game – After 4000 time steps



• Ok… But why an ABM?

- Some features are not intuitive.
- Try to imagine the histogram after changing:
  - Choose-starter?
  - Choose-next?
  - Number of players?
  - Layout-style?
  - Vocabulary size?
  - Noise-distortion and/or Acceptable-noise-level?
  - Other possibilities?

• Pros of ABM

- No need to rely on intuition for complex behaviours
- Formalise mental models so can be inspected by others
- Force abstract concepts to be represented concretely
- Relatively cheap and easy way to test the ‘untestable’
- Potentially very fast
- Can show range of possible futures
- Can run using real data, training data, random data, etc.
- Can be run again and again (with or without changes)
- Can be very intuitive for non-specialists to understand

• Cons of ABM

- Mental models (and ABM based on them) are hard to be critical about
- Concrete representations may not adequately capture abstract concepts
- Speed lends itself to being used for ‘prediction’
- Possible futures are hard to turn into risk profiles
- Can give false sense of security and truthiness
- Computational methods may motivate distrust
- Easy to misunderstand

• Platforms, languages, etc.

- Free, open-source, relatively small download size, tutorials and info available on web
  - Mason – focus on easy to learn, not most well recognised
  - Repast – more powerful, steeper learning curve
  - NetLogo – built in visualisation, not most powerful
  - EMLab-Agentspring – modular, not most well recognised
  - Object-oriented software (e.g. Python) – extremely powerful, not targeted so harder to learn/find answers
- https://en.wikipedia.org/wiki/Comparison_of_agent-based_modeling_software
- https://en.wikipedia.org/wiki/Agent-based_model
- https://www.comses.net/codebases/c32072dd-cd7c-4547-beba-e93a3f508c5f/releases/1.0.0/

Summary

- Bottom-up: non-intuitive large-scale effects of many small acts
- Agents: heterogeneous actors that observe, decide, act
- ABM: bottom-up simulation with easy testing
- Pros: can be inspected, cheap, easy, fast, can do ‘impossible’ experiments
- Cons: inevitably simplify complex concepts, often misused for prediction, easy to misunderstand

Questions

Dr. J. Kasmire

julia.kasmire@manchester.ac.uk
@JKasmireComplex

ukdataservice.ac.uk/help/

Subscribe to UK Data Service news at
https://www.jiscmail.ac.uk

@UKDataService