

Participatory Simulations with Netlogo and HubNet

Aaron L Bramson

HubNet Participatory Models

- Allow humans to control agents and/or parameters.
- Support for computers, calculators, cell phones, PDAs.
- Remote data collection and model integration.
- Teach concepts on distributed action and emergence.
- Run social science experiments.
- Still limited and somewhat difficult to set up and use.

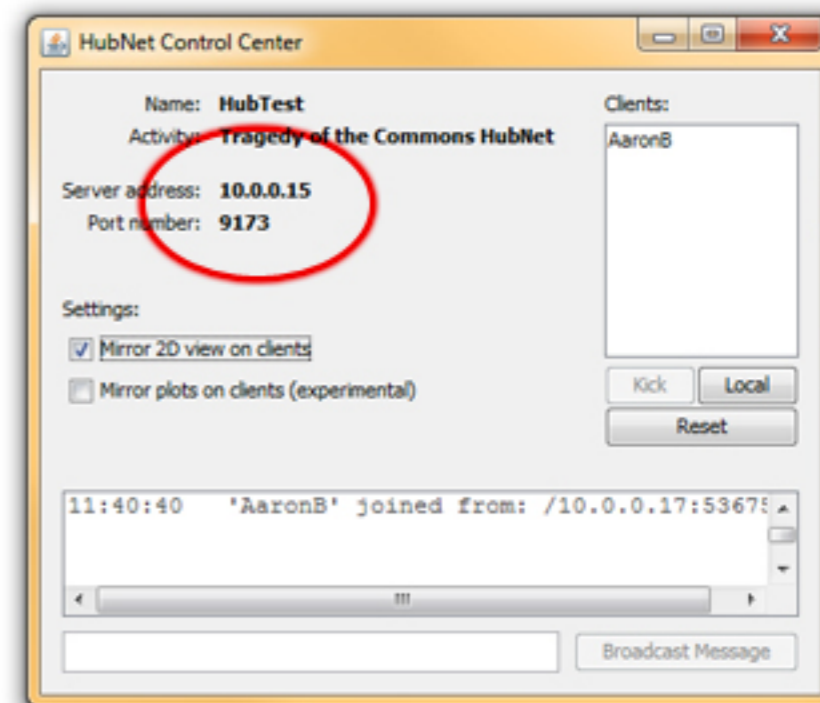


Running a HubNet Activity

- The teacher/experimenter runs a HubNet model from inside the normal Netlogo application.
- Also open the [HubNet Control Center](#) from the tools menu to get connection info.
- The model will have a [login](#) button to allow participants to join.
- Keep that running until everybody has joined, then unclick.
- Then adjust variables, click setup, then go, and run it like an ordinary Netlogo model.

Joining a HubNet Activity

- The teacher/experimenter opens the [HubNet Control Center](#) to show the connection info.
- Participants open the [HubNet Application](#), **not** ordinary Netlogo (but included).
- Accept any firewall warning.
- Login to the HubNet server using the connection information or click the appropriate item from the list provided.



OKAY, YOU'RE IN!

The screenshot shows a HubNet simulation interface for a 'Tragedy of the Commons' model. The interface includes a menu bar (File, Edit, Tools, Zoom, Tabs, Help) and tabs for 'Interface', 'Info', and 'Code'. Below the menu is a toolbar with 'Edit', 'Delete', and 'Add' buttons, a 'Button' dropdown menu, a 'normal speed' slider, a 'view updates' checkbox, and a 'Settings...' button. The main simulation area contains several components:

- Control Panel:** 'setup', 'login', and 'go' buttons. Sliders for 'init-num-goats/farmer' (5 goats), 'grazing-period' (34), 'grass-growth-rate' (0.5 oz/cycle), and 'cost/goat' (492 \$).
- Day Counter:** A box showing 'Day 109'.
- Graphs:** Four line graphs showing 'Milk Supply', 'Average Revenue', 'Grass Supply', and 'Goat Population' over 108 days. The 'Milk Supply' and 'Average Revenue' graphs show a peak around day 100. The 'Grass Supply' graph shows a sharp decline to zero by day 108. The 'Goat Population' graph shows a steady increase over time.
- Summary Table:** A table with four columns: 'Milk Supply' (2559), 'Grass Supply' (730), 'Avg-Revenue' (2559), and 'Goat Population' (193).
- 3D Viewport:** A 3D view showing a field of many small white goat icons on a green field. The viewport has a 'ticks: 3715' indicator and a '3D' button.
- Instructions:** A 'Quick Start Instructions' box and a 'Reset Instructions' box with 'PREV' and 'NEXT' navigation buttons.

The title 'Tragedy of the Commons' is overlaid in large, stylized orange and red text across the center of the simulation area.

Client View of the HubNet Model

HubNet: Tragedy of the Commons HubNet

Personal Variables:

My Goat Color

My Goat Population

Current Revenue

Total Assets

num-goats-to-buy Selecting a negative number here will eliminate some of your goats.

Goat Seller Says:
Everyone starts with 5 goats.

System Variables:

Cost per Goat

Grass Amt

Milk Amt

Day

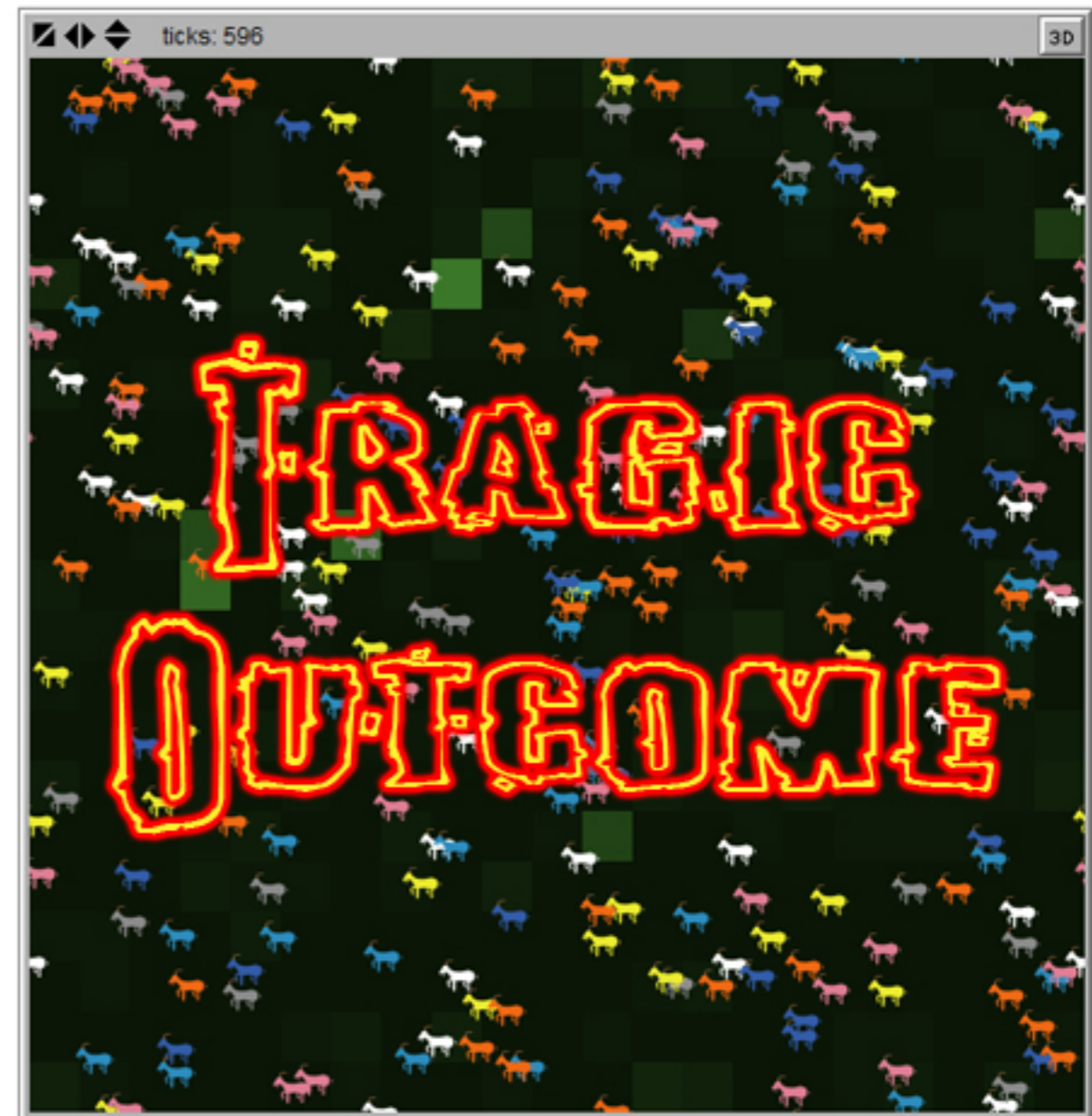
User name: Local 2

Server: 10.0.0.15 Port: 9173

Playing the Tragedy of the Commons

- Each player gets an initial number of goats.
- The server controls the global variables: cost per goat, grazing period, and the grass growth rate.
- Players control how many goats to buy or sell per period.
- Each goat eats grass, makes milk, is milked by you, and then milk is converted into assests to buy more goats with.
- Lots of hidden variables like max food, max grass, bite size, movement, etc.

Playing the Tragedy of the Commons



ReRunning a HubNet Activity

- If using the same participants, the teacher/experimenter just clicks **setup** to reset the model, then **go**.
- If you want to change the participants then on the model you can click **login** to let more in, to remove individuals click **kick** in the **HubNet Control Center**, to completely change the groups click **Reset** to kick everybody off.
- Adjust variables, click setup, then go, and run it like an ordinary Netlogo model.

Building/Editing a HubNet Activity

- Write code in Netlogo as usual, with special commands for client input and player actions.
- Use the [HubNet Client Editor](#) to add buttons, sliders, etc. to the user window...they have **tags** instead of commands/variables.
- The main code listens for commands from clients, each has an associated **ID** and the **tag**...which can then run the commands.
- The documentation includes the [HubNet Guide](#), the [Participatory Simulation Project](#) page, and the [HubNet Authoring Guide](#).

Let's Add Exploding Goats!

- Add a button to the client interface for “explode-goat”.
- Add a listener to the model for that command.
- Add code to make goats explode.
- Do exploding goats make cooperation easier to achieve?

