Charles Boy de la Tour - Zakariae El Asri

Prey - Predator Model

Summary

A simple ecological model, consisting of three agent types: wolves, sheep, and grass. The wolves and the sheep wander around the grid at random. Wolves and sheep both expend energy moving around, and replenish it by eating. Sheep eat grass, and wolves eat sheep if they end up on the same grid cell.

If wolves and sheep have enough energy, they reproduce, creating a new wolf or sheep (in this simplified model, only one parent is needed for reproduction). The grass on each cell regrows at a constant rate. If any wolves and sheep run out of energy, they die.

The model is tests and demonstrates several Mesa concepts and features:

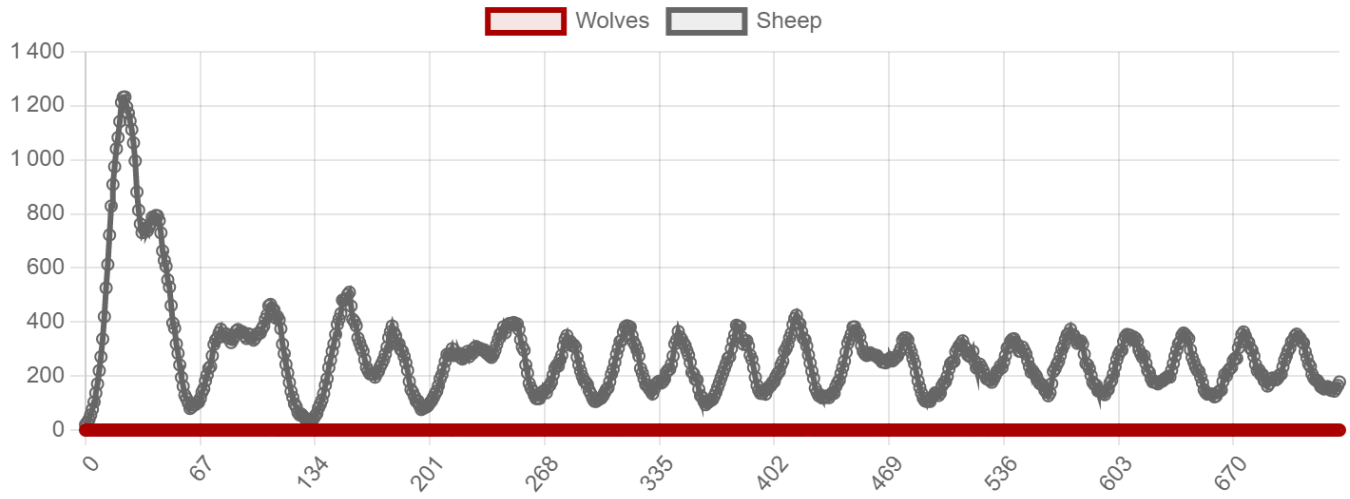
- MultiGrid
- Multiple agent types (wolves, sheep, grass)
- Overlay arbitrary text (wolf's energy) on agent's shapes while drawing on CanvasGrid
- Agents inheriting a behavior (random movement) from an abstract parent
- Writing a model composed of multiple files.
- Dynamically adding and removing agents from the schedule

Implementation

Results

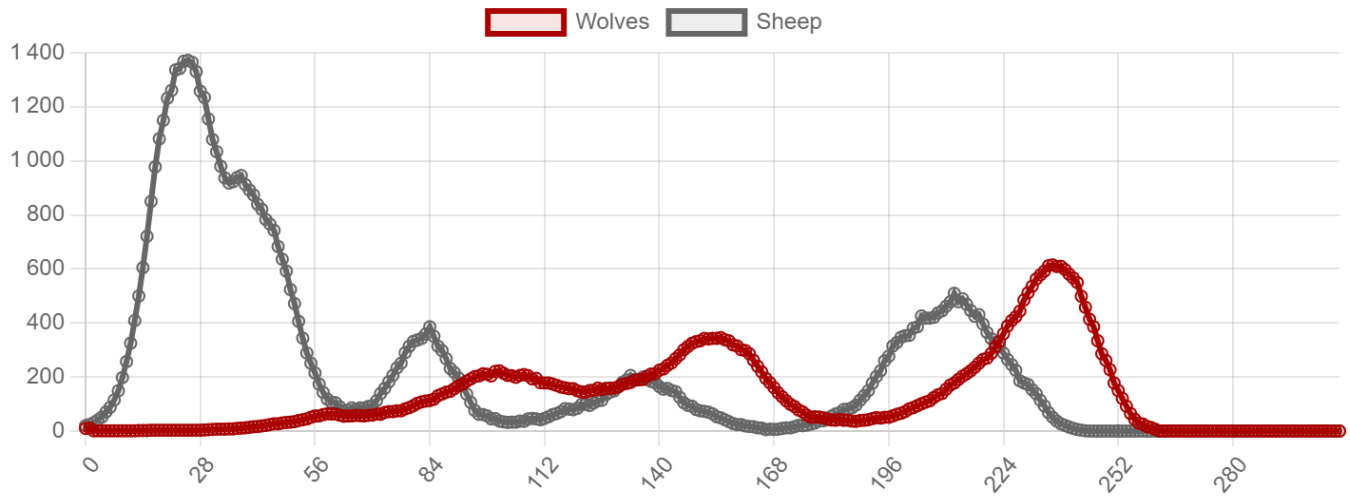
We experimented 3 different scenarios: Equilibrium with no wolf, wolf domination and overall equilibrium.

No Wolf Equilibrium



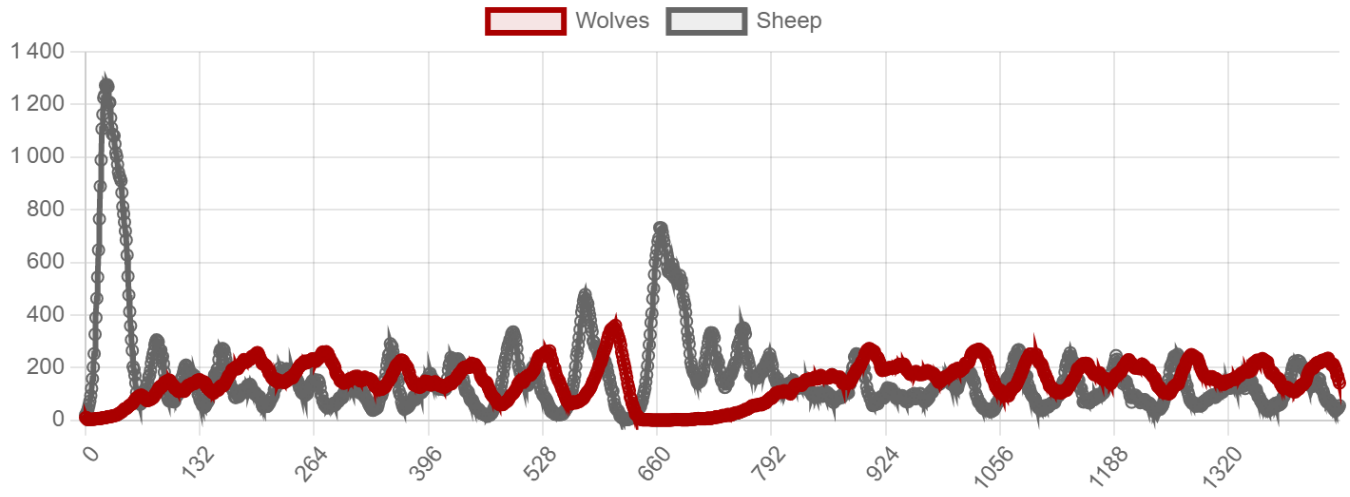
Grass Regrowth Time	**65**
Initial Sheep Population	20
Sheep Reproduction Rate	0.3
Initial Wolf Population	0
Wolf Reproduction Rate	0.06
Wolf Gain From Food	17
Sheep Gain From Food	6
Initial Energy of Sheep	30
Initial Energy of Wolf	2

Wold domination



Grass Regrowth Time	**65**
Initial Sheep Population	20
Sheep Reproduction Rate	0.3
Initial Wolf Population	10
Wolf Reproduction Rate	0.08
Wolf Gain From Food	17
Sheep Gain From Food	6
Initial Energy of Sheep	30
Initial Energy of Wolf	2

Overall equilibrium



Grass Regrowth Time	**65**

Grass Regrowth Time	65
Initial Sheep Population	20
Sheep Reproduction Rate	0.3
Initial Wolf Population	10
Wolf Reproduction Rate	0.06
Wolf Gain From Food	17
Sheep Gain From Food	6
Initial Energy of Sheep	30
Initial Energy of Wolf	2

Installation

To install the dependencies use pip and the requirements.txt in this directory. e.g.

```
$ pip install -r requirements.txt
```

How to Run

To run the model interactively, run `mesa runserver` in this directory. e.g.

```
$ mesa runserver
```

Then open your browser to http://127.0.0.1:8521/ and press Reset, then Run.

Files

- `prey_predator/random_walker.py`: This defines the `RandomWalker` agent, which implements the behavior of moving randomly across a grid, one cell at a time. Both the Wolf and Sheep agents will inherit from it.

- `prey_predator/agents.py`: Defines the Wolf, Sheep, and GrassPatch agent classes.
- `prey_predator/schedule.py`: Defines a custom variant on the RandomActivation scheduler, where all agents of one class are activated (in random order) before the next class goes -- e.g. all the wolves go, then all the sheep, then all the grass.
- `prey_predator/model.py`: Defines the Prey-Predator model itself
- `prey_predator/server.py`: Sets up the interactive visualization server
- `run.py`: Launches a model visualization server.

Further Reading

This model is closely based on the NetLogo Wolf-Sheep Predation Model:

Wilensky, U. (1997). NetLogo Wolf Sheep Predation model. http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

See also the Lotka–Volterra equations for an example of a classic differential-equation model with similar dynamics.