Animal Ecosystem Simulation

Karl Preisner, Michael Esposito, Joseph Crandall

Continuous Algorithms

May 9, 2017

Abstract

Our project is an ecosystem simulator. We have been working create an accurate model of animal populations over time, one in which the sinusoidal nature these populations is visible. Our simulator has various forest animals and plants, which are all subsets of an overarching Organism interface. So far, the included organisms are:

- Plants
- Mice
- Rabbits
- Foxes
- Bears

The organisms share a lot of methods, variables, and ways of interacting with other parts of the program. These methods are outlined in Organism.java. Various organisms are created in AnimalSimulator.java. This is instantiated from a GUI interface by running animalSimulatorGUI.

The simulator runs one time interval at a time. During each time interval, it updates the organisms' (1) health, (2) reproduction, (3) organism state, (4) organism movement, and (5) population statistics, respectively.

Lit Review (Papers Vs Animal Sim)

- Conway's Game of Life
- https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life
    - Similarities
        - Zero-player game, the evolution is determined by its initial state
        - Animal Sims initial state is randomly determined
    - Differences
        - Animal Sim has reproductions, just not as an individual animal dies, with reproduction happening randomly. At specific reproduction rates all animals in simulation die. Conway's Game of life includes reproduction and takes into account proximity.
- A comparison of whole-community and ecosystem approaches (biomass size distributions, food web analysis, network analysis, simulation models) to study the structure, function and regulation of pelagic food webs
- (look in papers folder, Ursula Gaedke)
- https://drive.google.com/drive/folders/0B1_sjt-7I9qFT0NzNDBDRnFzaEU
    - Similarities
        - Gaedke uses Biomass size distributions to provide a structural and energetic food web analysis based only on measurements of abundance and body size

        - Animal sim utilizes animal health (body size) and a random distribution of stationary and mobile food sources
    - Differences
        - Gaedke takes into account Biomass flows between living and non-living compartments, and provides
        - In animal sim, the biomass of dead organism does no persist once the animal has died.
- Vortex: A Computer Simulation Model for Population Viability Analysis
- (look in papers folder, Robert C. Lacy)
- https://drive.google.com/drive/folders/0B1_sjt-7I9qFT0NzNDBDRnFzaEU
    - Similarities
        - Although the traditional methods of wildlife ecology can reveal such deterministic trends, random fluctuations that increase as populations become smaller can lead to extinction even of populations that have, on average positive population growth when below carrying capacity.
        - Although animal sim does not produce predictions on carry capacity, experimental data when graphed shows different species values that persist for longer periods of time suggesting certain equilibrium values that the system can persist in for a period.
    - Differences
        - Vortex models population processes as discrete, sequential events, with probabilistic outcomes.
        - In animal sim all moves are precomputed prior to execution, however outcomes are determined based on proximity within the simulation.

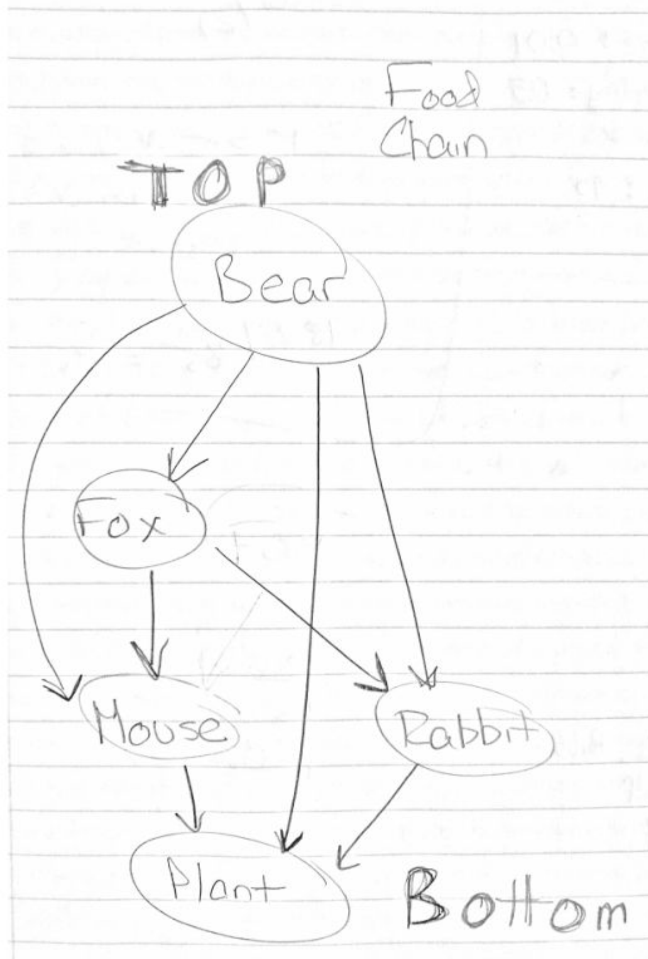
Health

Health is what keeps the organisms alive. Each organism type has a defined maxHealth, a defined amount of health that it loses per time interval, and a defined amount that it consumes during each time interval. Once an organism's health goes below a certain health threshold, it begins looking for food. Each organism has a defined list of predators and prey. Naturally, predators consume health points from prey, and prey loses health points from a predator.

If a predator is hungry and within sight distance of prey, it will begin to hunt this prey. Once prey is within eating distance, the prey loses health points, and gives its health points to the predator. All organisms also lose health gradually over time. Once an organism's total health falls below zero, due to either time or consumption by predators, it dies and is no longer a part of the simulation.

Predator/Prey

Each organism has a list of predators and prey. Organisms try to avoid encounters with predators and will run away if a predator is within their sight distance. If a predator sees prey within its sight distance, it will chase the nearest prey. The animal populations are monitored over time and outputted in a graph. The current predator/prey relationships are displayed in the following graph. A downward arrow represents predator at the beginning of the arrow, and prey at the end.



A user can see how animal populations are affected by changing some of the initial organism counts. Starting the simulation with an imbalance of animals can create interesting outcomes. For example, starting very few mice leads to the quick demise of all animal populations.

Organism State
Organisms can switch between 5 states:

0. idle
1. eating
2. beingEaten
3. hunting
4. escaping

Idle - organism performs a random walk
Eating - organism is currently eating prey
beingEaten - organism is losing health to a predator and cannot move
Hunting - organism's health is below specified hunger threshold. Organism will chase any prey that enters its sight radius.
Escaping - Takes precedence over any other states. Regardless of health level, the organism will attempt to run away from predators. Plants can't move, their escape method does nothing.

Organism Movement
The concept of "speed" for each organism is accomplished by giving organisms a different number of pixels that can be traveled in one time unit. We tried to keep the speeds of each animal accurate relative to one another. This means that a fast animal like a rabbit can potentially escape when being hunted. However, predators like a fox can capture faster predators if they team up and manage to trap prey.

Reproduction
The simulator also gives the organisms reproductive capabilities. We tried to keep the reproductive capabilities relatively accurate as well. This means that a mouse or rabbit reproduce much more frequently than bears and foxes. For every time unit, each organism has a very small chance of spawning offspring. If this boolean method is true, the number of

offspring produced is decided with a Gaussian distribution. These distributions have the average number of offspring produced at once at their centers. These numbers were discovered after brief research online for each organism.

This reproduction allows the simulator to model the rise and decline of individual animal populations. With the outlined capabilities, our simulator can quickly model the occurrences in a small forest ecosystem.

Population Statistics
When each simulation is finished, the animal populations are displayed to the screen. Here, the user can see the change in animal populations over time.

Scalability
We put a lot of effort into designing the simulator for scalability. This means that with minimal effort, new organisms can be added to the simulation. Performance is another issue we didn’t have time to tackle, but it is understandable due to the large number of organisms contained in one ArrayList.