Assignment 3 - Predator Prey Simulation Report

Authors

Laksh Mahajan (K25006772)
Vivash Pitroda (K25085419)

Simulation Description

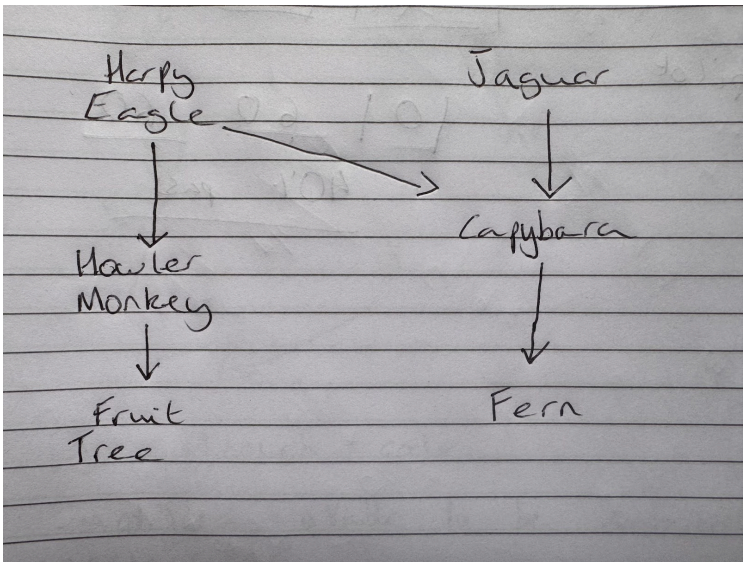
Our simulation represents a rainforest habitat, and the following six acting species: Jaguar, Harpy Eagle, Howler Monkey, Capybara, Fern, Fruit Tree.

During development, significant effort was put into designing the codebase according to core OOP principles, with the goals of clean design, scalability, and elimination of repeated code.

Inheritance is used throughout the class hierarchy. Animal and Plant both inherit from the abstract base class SimulationEntity, which holds the state and behaviour shared by every entity in the simulation: alive/dead status, grid location, and age. This means neither Animal nor Plant need to re-declare these fields. For the predator species specifically, a three-level hierarchy was used: Jaguar and HarpyEagle both extend ApexPredator, which itself extends Animal. All constants and behaviour shared between the two predators (maximum age, breeding age, mate search radius, hunger behaviour) are declared once in ApexPredator and inherited by both, so adding a third apex predator in the future would require no changes to existing code.

Polymorphism is leveraged so that the simulation loop in Simulator never needs to know which species it is processing. field.getAnimals() returns a list of Animal references; calling act() on each dispatches to the correct species behaviour at runtime. Similarly, field.getPlants() returns Plant references and each plant species handles its own spreading logic transparently.

The image below depicts the food web we implemented:



Simulation species description

Fern (producer)

Grows on the forest floor. Spreads spores into adjacent free cells once it is mature (age ≥ 10). Spreading probability is weather-dependent and is not restricted to daylight. Dies of old age at step 80.

Fruit Tree (producer)

Grows in the canopy. Disperses seeds only during daylight hours - seeds require sunlight to germinate. Matures at age 8, lives up to step 150.

Capybara (herbivore)

Feeds on ferns. Only hunts when its food level drops below half the value of a Fern, so it does not feed continuously when well-fed. Breeds only when an opposite-gender Capybara is within a search radius of 10 cells. Active only during daylight, shelters during storms. Initial population is 1% infected.

Howler Monkey (herbivore)

Feeds on FruitTrees. Same hunger-threshold behaviour as the Capybara - only eats when hungry. Requires a mate within radius 10. Active during daylight only, shelters in storms. Initial population is 1% infected.

Jaguar (apex predator)

Hunts Capybaras exclusively. Unlike herbivores, always hunt every step regardless of food level. Requires a mate within a wide search radius of 20 cells, reflecting the large territories real jaguars hold. Lives up to age 150, breeds slowly (probability 0.10, litter size 1).

Harpy Eagle (apex predator)

Primary prey is HowlerMonkey (always hunts if adjacent). Hunts Capybara opportunistically only when its food level drops below half the prey food value - this models the real behaviour of switching to alternative prey under prey shortage. Competes with Jaguar for Capybara when monkey populations crash. Same lifespan and mate-search radius as the Jaguar.

Core tasks implemented

1. ### At least five different kinds of acting species

   There are 6 different acting species as illustrated above.

2. ### At least two predators should compete for the same food source

   Jaguars and Harpy Eagles compete for Capybaras.

3. Keep track of the time of day

   There are 4 day states: dusk day, dawn, night. Different behaviours are exhibited according to the different times of day, for example animals do not move and fruit trees do not disperse spores at night.

4. Distinguish male and female individuals

   Gender is assigned at birth in the Animal constructor randomly. giveBirth() performs a mate search using hasOppositeGenderNeighbour() within a given search radius. It is an intentional design choice that mates don't necessarily need to be adjacent to the animal before they can breed. If no opposite gender mate is found then no offspring can be produced.

Extension tasks implemented

5. Add plants

   We added plants that behave as their own species, and operate on their own layer, so that apex predators can walk over plants, without therefore being blocked unnecessarily, while herbivorous species consume the plants once their cells come into contact. We successfully implemented growth mechanics by requiring a certain number of steps to pass (maturity age) before they can be eaten. Plants react to time of day and weather successfully, via use of the plantGrowthFactor() method which reacts to each weather condition, and isDaylight() gates spreadOffspring() for fruit trees, so that they can only disperse spores at night.

6. Add weather

   We added the following weather conditions: sunny, rainy, foggy, windy. One is randomly picked during the day from this group, and at night time from another group excluding sunny, so that weather remains realistic. Weather influences breeding, hunting, and movement mechanics, via ...SuccessFactor() and the allowsMovement() methods in Simulation Context. For example, during storms, the predators cannot hunt or move as they seek shelter, and breeding is also not possible in storms. Plant growth rates are also directly influenced by each weather condition.

7. Add disease

   Disease was implemented such that during the population process 1% of all herbivores are infected - predators never start infected but can catch the disease during the simulation. There are two ways in which disease is spread: proximity, where any adjacent animal to an infected animal has a 20% of contracting the disease in each step, and predation, where when a predator eats an infected prey, it immediately becomes infected too. On each step there is a 0.3% chance per step of death, but infection lasts for 50 steps so this compounds to 14% chance of death per infection. If an animal is infected, its chance of successfully breeding is 40% its normal rate. If an animal survives

infection it is immune for the next 30 steps. Note immune and infected animals are identified in the simulation by yellow and red markers on cells, respectively.

Known bugs and limitations

1. ### Processing order bias

   Animals are stored in an ArrayList and processed in insertion order every step. Animals added to the field first (during populate()) always act before animals added later. This gives early-inserted animals a systematic advantage - they always get the first pick of adjacent free cells for movement, hunting, and breeding. This is a fundamental limitation of the sequential update model.

2. ### One-step viability lag

   When the last Fern is eaten mid-step, isViable() is not checked until the top of the next step. The simulation runs one extra step with Capybaras alive but foodless before terminating.

3. ### Weather lacks persistence

   Each day's weather is drawn completely independently. A stormy day is equally likely to be followed by a sunny day as another stormy day. Real weather has momentum - a storm is more likely to persist than to instantly flip to sunshine. Could use Markov Chain approach to fix.