

4CCS1PPA Programming Practice and Applications

Coursework 3: Predator/Prey Simulation

21/03/03

Arctic Ecosystem Simulator

Ioana-Daria Vasile (20047633)

Alexandra-Maria Anastase (20022789)

1. Description

The “Arctic Ecosystem Simulation” shows the behaviour of animals living in the arctic region. The ecosystem consists of both land and marine animals and plants. The purpose of the simulation is to observe how the ecosystem would be impacted if penguins were to be brought from the Southern Hemisphere (Could some species disappear due to the large number of predators hunting them?, Will the penguins influence the size of the populations already existing there? etc.). In order to make the simulation as realistic as possible, the simulator keeps track of the time of day, the weather and diseases that may affect the ecosystem. Both carnivores and herbivores can die if they do not find their food source. For some animals it may be difficult to breed if they do not meet a partner of the opposite sex. This project focuses on the following species:

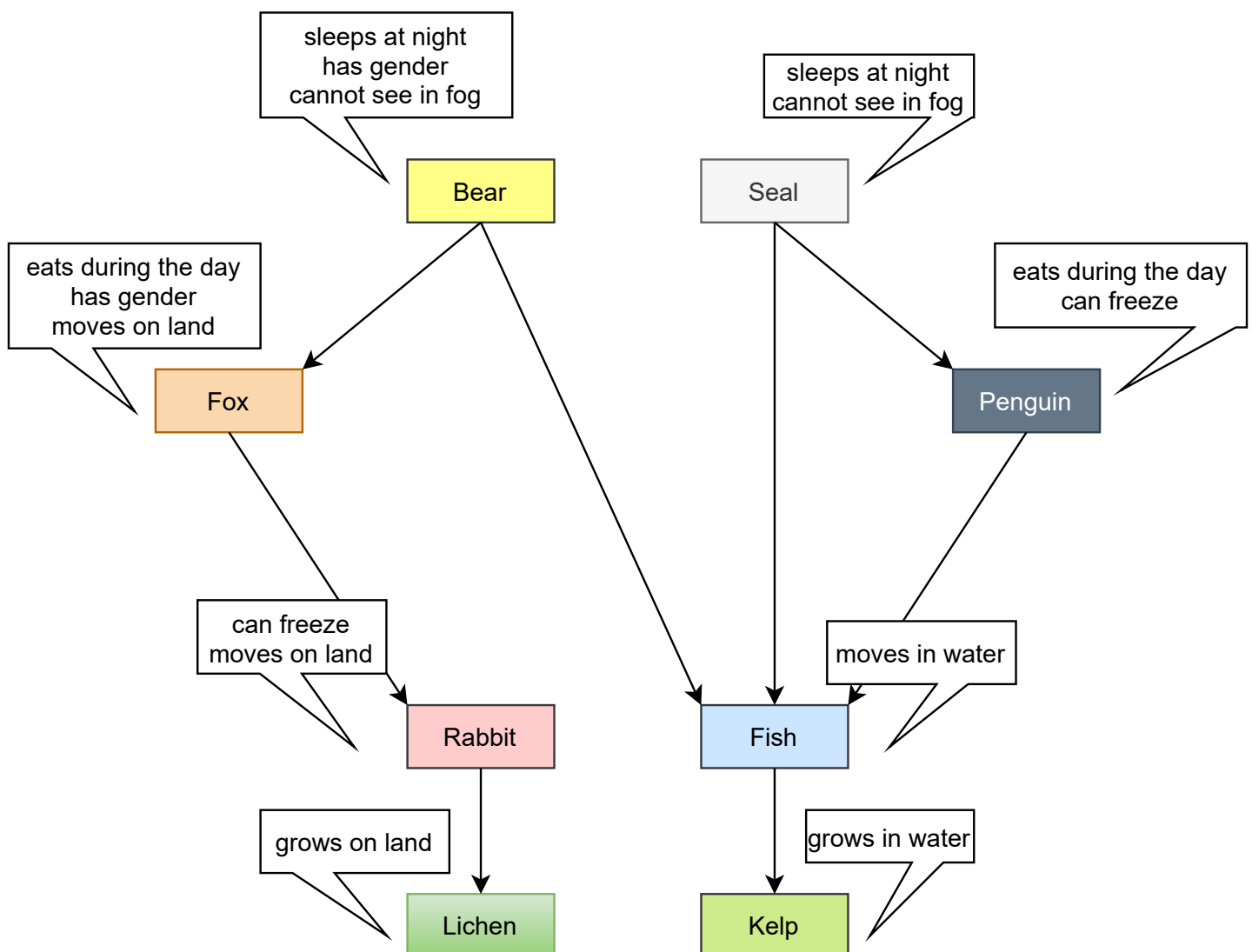
Animals:

- **Fox (orange)** - Polar foxes only hunt during the day and eat rabbits. They can move and reproduce at any time. In order to reproduce, a female fox searches for a male partner in neighbouring locations. Foxes can also carry diseases which could kill them. They can only be eaten by bears. Foxes only move on land and do not go in water.
- **Bear (yellow)** - Bears eat fish and foxes and cannot be eaten by any other animal. Bears are active only during the day (they do not hunt, move or reproduce during night time). In order to reproduce, a female bear searches for a male partner in neighbouring locations. They can also carry diseases which could kill them. Bears are also affected by the weather: they cannot see the prey in the fog. Bears can move on land and can go in water.
- **Penguin (black)** - Penguins hunt only during the day and eat fish. They can move and reproduce at any time. We assumed that penguins may reproduce every time two animals are in adjacent locations. Penguins can be eaten by seals and are affected by the weather: during a snowstorm, they can freeze and become more hungry if there is no other animal around to keep them warm. Penguins can move on land and can go in water.
- **Seal (grey)** - Seals eat fish and penguins and cannot be eaten by any other animal. Seals are active only during the day (they do not hunt, move or reproduce during night time). We assumed that seals may reproduce every time two animals are in adjacent locations. Seals are also affected by the weather: they cannot see the prey in the fog. Seals can carry diseases which could kill them. Seals can move on land and can go in water.
- **Rabbit (pink)** - Polar rabbits eat lichens and can be eaten by foxes. They can eat, move and reproduce at any time of the day. We assumed that rabbits may reproduce every time two animals are in adjacent locations. Rabbits can carry diseases which could kill them. They are affected by the weather: during a snowstorm, they can freeze and become more hungry if there is no other animal around to keep them warm. Rabbits only move on land and do not go in water.

- Fish (blue) - Fish eat kelps and can be eaten by penguins, seals and bears. They can eat, move and reproduce at any time of the day. We assumed that fish may reproduce every time two animals are in adjacent locations. Fish can only move in water (they do not go on land).

Plants:

- Kelp (cyan) - Kelps only grow in water and are eaten by fish. They grow at a given rate, reproduce and cannot move.
- Lichen (green) - Lichens only grow on land and are eaten by rabbits. They grow at a given rate, reproduce and cannot move. Lichens are affected by the weather: during snowstorms they stop growing.



2. Challenge Tasks

1. Plant Simulation

The simulation currently contains two species of plants: kelp and lichen. All plants grow at a given rate (specific to each species) and do not age (only die if they are eaten). This behaviour is implemented in the abstract class `Plant`. Plants can reproduce if they have reached a certain age. Each species is represented as a subclass of `Plant`. The implementation for plant reproduction is specific to each subclass. The `Actor` class has been created as a superclass of `Animal` and `Plant` in order to avoid code duplication in the `Simulator` class where we iterate through the animals and plants and call the `act()` method. The `act()` method (which creates the behaviour of each plant and animal species) overrides the one found in the `Actor` class.

2. Weather Simulation

The simulation has three states of weather: normal weather(when the animals and plants are not affected in any way), snowstorm and fog. Either the snowstorm or the fog can occur multiple times during the simulation with a given probability. Each phenomenon can happen for a given maximum number of steps. Class `Weather` is used to determine the state of the weather. The `WeatherSimulator` class randomly generates different states for the weather and their duration. The `Simulator` class creates a `WeatherSimulator` object which is passed as a parameter to all actors. This way they are aware of the current weather and can act appropriately.

3. Disease Simulation

Animals can get infected with a disease. The disease can appear during the simulation (the disease has a status which determines whether the simulation is in a "state" of disease) and infect some animals. Infected animals may contaminate some of the animals around them. The disease has a spreading probability with which an animal can contaminate others. While they are sick, their food levels decrease faster. All animals are sick for the same number of steps from the moment they got contaminated, unless they die. A wave of disease ends when there are no more sick animals in the simulation, but other waves may appear later. Sick animals are stored in a `HashSet`.

4. Land Creation

The ecosystem is made up of two habitats: land and water. Some animals can live only in one of them, others can move freely in both. Plants also have specific habitats in which they can grow. To achieve this, each location must be either land or water. The `TerrainGenerator` class creates all locations in the field and their type and stores them in a `List`. The terrain is made after a pattern and will look similar in all simulations. We set as water all location which are above the main and secondary diagonal. Each time animals have to change their location, they check if it is possible to move (eg. a fox will never go in water).

3. Code limitations

- The *giveBirth()* method is present in all animal subclasses with very similar code because each young animal uses the subtype in its creation.
- The PopulationGenerator class uses repetitive code when creating the actors.
- Weather and terrain could have been implemented better, without using booleans, allowing for a greater variety of types.