Dario Mammana

Object oriented programming homework 6

Question 1b

Why is grass not an inhabitant? Please explain in written form

If grass was treated as an inhabitant, then the sheep could never eat it since each cell can only have one inhabitant. Grass therefore needs to be treated as a background LifeForm that cannot move. If it moved it would also cause problems when running into the animals, is therefore makes more logical sense not to allow grass to move. The cell where grass is grown can therefore still be occupied by an animal inhabitant.

Question 1c

Why is it costly to let grass grow randomly on any cell? Can you make alternative suggestions? Please answer in written form.

Let grass randomly grow on any cell is costly because we need a double loop that checks all cells, and we call random for every single one of them. An alternative could be to sample just a number of cells and call random only on those randomly selected cells.

Question 1e

Should reproduce and eat be methods of the Sheep or of the Lifeform?
Please explain in written form.

They should be methods of the Sheep since grass, a lifeform, does not eat and reproduce. The methods are called in Lifeform but are empty, we define them according to the specific lifeform or animal. If the created a class Animal, then the methods eat and reproduce could be defined there, assuming there are not differences on how wolves and sheep eat and reproduce.

Question 1g

When and where are the methods move, eat, reproduce and die called?

In which order are they called and why?

Should this happen in Lifeform or in Wolf and Sheep?

What about grass? Please explain in written form.

Dario Mammana

At the setup of the simulation, the order is as follows:

1. Grass grows
2. Sheep are randomly placed in free cells and added to the lifeforms list
3. Wolves are randomly placed in free cells and added to the lifeforms list.

At every step Simulation.update() calls;

1. grass to grow
2. life forms to act
3. grass to be consumed

When lifeforms act, it happens in this order:

1. 1 is added to age
2. Energy from moving is lost
3. Move to target cell
4. Eat
    a. If Sheep, eat grass in target cell
    b. If wolf, eat sheep in target cell
5. Reproduce if energy is enough
6. Die is energy is 0 or if the maximum age is reached

Right now all life forms' actions are called directly in Sheep and Wolf. Which might not be the most efficient design, since many characteristics are very similar. In the next step/task, it will therefore be necessary to implement abstract classes to solve this efficiency issue.

Grass is treated as a different type of lifeform as it does not really inherit the methods from Lifeform.

Question 3

Explain how the abc module is helpful for Lifeform. Where else could the simulation use inheritance?

The abc module is helpful because it makes it impossible to create classes that inherit from lifeform if I do not define all the required methods (act, move, eat, reproduce, die), this makes the whole code more consistent (made me realize some mistake I made in task f).

Moreover, inheritance can be used to create a middle class between LifeForm and Sheep and Wolf. This class Animal could even specify some more features the two animals have in common so they could inherit the same things.