Exercises Part 3
Predator/Prey Modelling

Hans Georg Schaathun

5–26 February 2015

Project 1 *Implement a predator-prey simulator with visualisation for an eco-system with species of your choice, including food for the prey species (e.g. grass). It is suggested that you base the project on Project 4 (wolf/sheep/grass) of Shiflet & Shiflet, page 512 (see attachment), varying the exact rules and parameters used. Use the similator to predict (estimate) future population levels. Work in a group of three (or two if numbers don't fit).*

The project consists of three phases, one week each, with an oral presentation of current results at the end of each week. The three phases have different focus:

1. Modelling
2. Implementation
3. Analysis

1 Modelling

Note that the modelling stage gives you a lot of options. Modelling is always the exercise of selecting key features to be captured and less essential detail to be ommitted. In your group, you need to discuss what species to model and which features and properties to prioritise.

1.1 Session 1

Exercise 1.1 *Choose the species for prey and predator in your simulation, or use wolf, sheep, and grass as in Shiflet & Shiflet. Look up a couple of facts about the species, relevant to the simulations. Examples include fertility rates, life span, etc.*

We are not aiming for a fully realistic simulation model, as a sufficient understanding of biology would be way out of scope for this module. The objective is to get your minds at least somewhat in to the real world problem. Assemble at least a handful of relevant facts.

Exercise 1.2 *Discuss (briefly) the properties of agents. List all properties which are common for all agents (prey and predator alike).*

Exercise 1.3 *Discuss the behaviour of prey agents. Formulate behaviour rules which can feasibly be implemented. Prioritise the list of behaviour rules in four sections, labelled* must have, should have, can have, *and* not in this project.

Exercise 1.4 *Discuss the behaviour of predator agents. Formulate behaviour rules and prioritise them as in the previous exercise.*

1.2 Session 3

Exercise 1.5 *Make a high-level agent-based predator-prey model for the problem scenario, including the following items:*

- *Landscape design (grid size, boundary conditions)*
- *Enumerate the different types of agents, with brief description.*
- *Food model for prey. (Is the food (grass) an agent or not?)*
- *Time scale. What is the length of the time step in the model?*

Document any reasons you find, both in favour of and against the choices you make. It is alright to make some of the choices arbitrarily, but you must document your lack of certainty. Choices may be revisited later if necessary.

Exercise 1.6 *Make an object-oriented software model for your simulator, suitable for implementation in Java or in your favourite object-oriented language. Obviously, the software model must closely reflect the domain model that you have described above.*

1.3 Presentation 11 February

In the mandatory presentation on Tueseday 11 February, we want to learn about your agent-based model, i.e. the domain model rather than the soft-ware model.

You should start with an overview of your model, i.e.

1. what eco-system do you model?
2. time scale, landscape model

You should focus particularly on how you model the individual agents.

3. Which behavioural characteristics do you plan to implement for the prey?
4. Which behavioural characteristics do you plan to implement for the predator?
5. What features of the real animals have been abstracted away for the sake of simplicity?

2 Implementation

Hopefully, you start the second week with a good model of the eco-system with predator and prey from last week, as well as an object-oriented software model for the simulator (Exercise 1.6). Now it is time to implement it.

Remember that Barnes and Kölling have a similar project in their textbook and they have also published sample code. Feel free to use their examples. If you cannot find the sample code, please ask the lecturer. The Barnes and Kölling example is a very good starting point but does not suffice for the current project. The agent behaviour is too simple, and as a minimum you need to introduce fodder for the prey and add appropriate behavioural rules.

It is suggested that you use Java. You are allowed to use other languages if you please, but you may then not be able to benefit from the example from Barnes and Kölling.

2.1 Session 1

In the first session we will focus on adapting the fox and rabbit example of Barnes and Kölling to whichever eco-system you chose to model.

Exercise 2.1 *Pick up the example code from Barnes and Kölling and check that it compiles and runs. Even if it does not simulate 'your' species, it is still your first running prototype simulator.*

Exercise 2.2 *Review the OO model from Exercise 1.6 and the prioritised features from Exercises 1.2 to 1.4. Discuss what amendments you need to make to the prototype simulator to implement* your *model.*

1. *Write the amendments down in order of priority.*
2. *Allocate one amendment (task) to each person to do first.*
3. *Whenever one amendment is made, check that the prototype still runs.*

As tasks (amendments) are complete, allocate new ones to keep busy.

2.2 Session 2

In the second session we will focus on adding functionality to your simulator, although you may also have to add features from the eco-system model to make the simulation more realistic.

Your simulator must provide the following features:

1. The simulator must be able to produce a log file recording the population size of each species for every time step, for the purpose of statistics. (Comma-separated values is a suitable format; easy to read and easy to parse.)

2. Make some (not all) of the model features configurable, so that you can experiment with variations of the model (e.g. starting population size, grid size, etc.).

3. Visualise the location of individual agents at each time step.

Exercise 2.3 *Review your runnning prototype and discuss how to implement the features above.*

1. *Divide the work into tasks, in order of priority.*

2. *Allocate one amendment (task) to each person to do first.*

3. *Whenever one amendment is made, check that the prototype still runs.*

Continue allocating and completing tasks until all the necessary *features are implemented.*

Exercise 2.4 *Evaluate your simulator. Which features have been implemented to satisfaction? What remains to be desired?*

1. *Write down all features that you would like to add as potential tasks.*

2. *Prioritise the tasks, and order them accordingly.*

3. *Split the list into three sections, of (1) tasks which* must be complete *before the presentation, (2) tasks which you* hope to complete*, and (3) the rest.*

4. *Take one task each and start implementing new features.*

5. *Whenever one amendment is made, check that the prototype still runs and discuss whether it is satisfactory.*

Continue allocating and completing tasks until you are satisfied.

2.3 Presentation 18 February

The core of your presentation should be a demonstration of your simulator.

1. Show how you operate the simulator.

2. Show that you can display and harvest useful information from the simulator, incl. statistical data such as population size.

3 Analysis

4 Session 8

Exercise 4.1 *Run the simulator with a range of different parameters. Playing around with the parameters, select* three *different scenarioes showing the variety of possible outcomes:*

1. Both species dies out.

2. One species dies out while the other survives.

3. Both species survive.

In the following exercises, you are asked to plot data from your simulator. A good and straight-forward approach to do this is the following:

1. Your simulator dumps the statistics to a file using comma-separated values (CSV).

2. Load the CSV file into Matlab or another appropriate tool.

3. Plot the results using the `plot()` function in matlab or similar.

Instead of Matlab, it is possible to use a spreadsheet tool (libreoffice) or python, or a number of other tools.

Exercise 4.2 *Using the log data from your simulator, plot the population size for each species as a function of time. Present one plot for each of the three scenarios from Exercise 4.1.*

Exercise 4.3 *Consider the scenario with co-existing predator and prey. Run the simulator a couple of times and plot population sizes versus time for each run. Discuss the results.*

We are interested to know what the expected population size is at a given time in the future. Because the populations are oscilating, the size at a single point in time is not meaningful. It is in contrast very interesting to compare the two populations, using the correlation coefficient.

Exercise 4.4 *Consider the scenario with co-existing predator and prey. Calculate the correlation coefficient of the two population sizes. If the population sizes at time step i are X_i and Y_i, calculate the correlation coefficient between*

1. X_i *and* Y_i

2. X_i *and* Y_{i+1}

3. X_i *and* Y_{i-1}

4. X_i *and* Y_{i+2}

5. X_i *and* Y_{i-2}

Feel free to try other offsets between the two signals. Discuss the results in your group.

5 Session 9

Let's explore some other meaningful characteristics, such as

- Expected population size at a troff.
- Expected population size at a peak.
- Number of births within an interval.
- Average age of animals dying within an interval.

The interval (in 3 and 4) should obviously cover a full period (a peak and a troff on the curve), and ideally and integer number of such periods. An odd fraction of a period could skew the distribution.

In the following exercise, you may have to rely on manual inspection to gather observations for the analysis, meaning that your sample may be small (10–20 observations). That is a good case for Student's t-distribution.

Exercise 5.1 *Choose at least two variables to estimate in the scenario where both predator and prey survive. Assume that it is normally distributed, and use Student's t-distribution to estimate it.*

Hint 1 *In order to assess the period length, you have to identify the peaks and the troffs on the curve. The following matlab function may be useful:*

1. *http://blogs.mathworks.com/pick/2008/05/09/finding-local-extrema/*

Other variables one might want to estimate are the period length (time between subsequent peaks), probability of a species becoming extint, probability of a rabbit being eaten as opposed to dying of other causes, etc.

Exercise 5.2 *Discuss in your group a number of numerical variables you might want to estimate, and the challenges involved in doing it.*

5.1 Presentation 25 February

In the presentation we are first and foremost interested in *your* interpretation of the simulation results from your *own* group. Feel free to choose observations that you find surprising and/or illuminating and show them. Hopefully you have reviewed a number of scenarios and chosen the best of them to share.

Make sure that you include at least some application of statistical methods, including use of the correlation coefficient.

Note that the algorithm avoids collisions by having at most four animals in each category moving in different directions.

b. Graph the population densities of predators and prey versus time.

c. Graph the number of predators versus the number of prey.

4. a. Develop a simulation with visualization involving wolves, sheep, and grass on a grid with periodic boundary conditions. A cell is empty or contains one of the following items: a male wolf, a female wolf, a female wolf with cub, a male sheep, a female sheep, a female sheep with lamb, or grass. Associated with each animal is an integer **food ration,** or amount of stored energy from food, up to some maximum value. Assume a population density for each item. The rules are as follows (He et al. 2003):

- A sheep moves into a neighboring empty site, preferring one with grass.
- A lamb leaves its mother and moves into a neighboring empty site. At random this new sheep is a male or female, and its food ration is the same as that of the mother.
- A wolf moves into a neighboring empty site.
- A cub leaves its mother and moves into a neighboring empty site. At random this new wolf is a male or female, and its food ration is the same as that of the mother.
- If its ration of food is less than the maximum, a sheep eats neighboring grass and increases its ration to the maximum amount.
- If a female sheep has at least a designated amount of food ration (such as 2), is of reproduction age (such as 8), and has a male sheep of reproduction age as a neighbor, she becomes a female sheep with lamb.
- If its ration of food is less than the maximum (such as 3), a wolf eats a neighboring sheep and increases its ration to the maximum amount.
- If a female wolf has at least a certain amount of food ration (such as 2), is of reproduction age (such as 8), and has a male wolf of reproduction age as a neighbor, she becomes a female wolf with cub.
- An independent baby matures in a certain number of time steps, such as 8.
- An animal's food ration decreases by 1 at each time step.
- An animal dies when its food ration becomes 0.
- Grass grows in a certain number of time steps, such as 4.

Avoid collisions as in the text of Module 11.3 on "Movement of Ants" or as in Project 20 of that module. Initialize the grid at random with certain densities of each item and with random food rations and ages for each animal. Run the simulation a number of times obtaining situations in which the sheep, wolves, and grass coexist with oscillating densities; in which the sheep become extinct; and in which all animals die.

b. Graph the population densities of sheep, wolves, and grass versus time.

c. Adjust the program to run the simulation a number of times, computing and storing the average number of sheep, wolves, and grass at each time step. Plot these averages versus time. Discuss the results.

5. a. Develop a simulation with visualization involving mobile predators and stationary prey. For example, algae that grow on rocks in inter-tidal areas are a favorite food of some snails. Use periodic boundary conditions, and