

Programming Skills

Random Things that need to be mentioned:

Allocation of tasks:

Some initial code, reading methods, build, profiling?: Xiao

Some initial code, populate/ grid map, user terminal input: Colum *Testing areas?

Animal superclass, printing methods, unit tests: Sarah

Make sure we all have something for coding/ testing

Checks for negative population

Check cannot try to access a square not on the grid (shouldn't be possible in the task as defined here but a user could try to do that if they misuse the code.

Group Organization and Task Allocation

We met at least once a week to discuss and review where we were with the project and talk through any design or coding issues we had come across.

We chose to have a quite flexible and fluid allocation of tasks allocation. We are on three completely separate degree programs and have very different sets of courses and other coursework and commitments so we decided we should try to work around this as much as possible. This meant that we knew that some weeks certain people would be able to do less work than others and we allocated the work for that week accordingly but tried to make sure that that every person did a fairly equal share of the work across the entire time we worked on the project.

Testing Framework

We made sure that we were all involved in testing the code. Unit tests were created for the various classes we created to make sure that we were getting correct values for certain methods or that exceptions were being thrown as we wished. We tried to make sure we had unit tests for things we knew from the outset and occurred to us while writing/ testing the code that the user might try to do wrong that might make the code fail. We acknowledged that we would not find everything or even manage to think through all the possibilities in the beginning. We accepted that certain methods are best tested through running the entire code and viewing the results. While the methods may run as expected and we do not know the exact nature of the results it would have been difficult to create unit tests for these but we could look for certain trends and properties in the outputs to test that the program was working as we expected.

The Code

In our first draft of our code information about hares and pumas was integrated into the "Population" class. We found this rather undesirable in the long term as we felt that the methods in the "Population" class should be able to deal with any predator/ prey system that followed the same structure of equation. To this extent the "Animal" superclass was

created to hold the general properties of an animal: type (predator/ prey), birth rate, diffusion rate, mortality rate and predation rate. The superclass ensures that only an animal of type "predator" is assigned a predation rate. This was a subjective decision but the logic is essentially just that the predator is the animal that eats the prey and such the predation rate describes their action on the hare and such is a property associated with the predator. Prey can have a mortality rate [I want to change code to allow this] if a user wants to describe the model differently. Currently the Animal superclass can only deal with Animals that fall into one of the types "predator" or "prey" not an animal that is both. Perhaps the class should in general allow this although the user would then have to have a discretized differential equation that would could model the interaction and the additional complexity might well mean that they needed a completely need structure of code. [This is something additional I may modify the Animal class to allow this but make sure that the population class only takes animals of type, predator/ prey??]

The use of a superclass rather than an interface was to allow explicit coding of the methods all Animal objects can have. The set methods for the various rates ensure that the user cannot enter a negative rate (that throws an IllegalArgumentException and checks that prey do not have predation rates, as mentioned above). This meant that the classes for "Hare" and "Puma" that defined the properties of these particular animals were very bare, just some specific constructors only. A more complicated structure of an "Animal" superclass and then "prey" and "predator" subclasses were considered but this was deemed an extra level of complexity that wasn't needed for the problem and all the issues could be dealt with by the superclass for the type of problem we had.

The Printing Methods class contains a number of static methods to print different information from a population to a file. These are all static methods as all of them can be called without needing a "PrintingMethods" object to set them up but it made sense to give them a separate class and separate them from other parts of the code making those classes clear and readable. The "printingDensityFile" methods prints the average densities of at a given time for a predator (1st column) and a prey (2nd column) each separated to a file specified by the user. This allows to user to view the densities of the two animals side by side on the same graph and they can check that the two populations interact as expected. Average density was defined as being the number density in a square relative to the total animal population. This means that if the user manages to print a file where at a given time the densities of both populations are "1" there is a bug in the code. This is a nice simple way to check the simulations give the required results.

There is a method to convert an double value into an RGB integer value. This normalized the number by dividing by the maximum possible and multiplying by the maximum RGB value as specified by the user. In this project this means that the population of a given square is scaled by the total animal population and multiplied by 255 and adding 10, to ensure any square with some population is at least slightly colored. So a square with no population would have a value of (255,255,255) (white) and a square with the entire population would have a population of (0,0,0) (black).

There is a method to print the a density grid to a plain ppm file as per the manual specifications. This method will create a single color ppm file to view. Each animal is

viewed in a separate color to make it easier for the user to watch the individual animals and changes. The magic number 'P3' is on the first line, the next line is blank, next maximum RGB value, the next line contains the dimensions of the population grid. The rest of the file is a matrix (a double array in actual fact but can be regarded as a matrix) with the same number of rows as the population grid and three times the number of columns. As the RGB values are separated into three columns per one column of the grid. Each square is represented by one pixel.

The method to create the matrix is separate to the print method the matrices to represent the densities of hares and pumas can be printed separately and we can also print the the combined maps of their densities. The only trick to the addition is to make sure that adding 255 is the same as adding 0 (white added to white is still white).

Reading Methods

Reading in initial conditions can be done from an input text file or from the user entering them from command line prompts. The latter can be time consuming and if the user wants to quickly vary a value it is probably easiest to use a file and just change that when they want to run it again. This method only deals with setting up the initial conditions by setting proportions of hares and pumas across the landscape.

The user can also chose to run the simulation just with “ant run” when they do this they will be lead through a series of questions that allow them to change an individual values one and at time. They can also choose where they want the initial populations of hares and pumas to be very precisely by setting the population in an individual square. It would be nice to facilitate this in both functionalities which would be a further extension of the code. This is seen as more useful for the user to specify this level of detail here rather than in a data file when they might not know the layout of the map so well?