FUNDAMENTALS OF PROGRAMMING ASSIGNMENT

LARA AHMAD SALEM

OCTOBER 28, 2019

STUDENT ID: 19491787

CURTIN UNIVERSITY

Abstract

This report outlines an experiment that was conducted to simulate the lifecycle of a population brine that are constrained to a fish tank. This program was created with various predefined libraries and supporting functions that assist in creating brine shrimps that start from eggs up until they are adults and die from a randomly generated integer that is checked against a death probability. These methods were heavily created and based on probability, to create a simulation that was as close to a real-life population as possible. A greater understanding of brine shrimp and their life cycle was necessary to conduct a simulator that would try to display a shrimp community reproducing, living and dying. In my report I intend to clarify and justify my choices, methods and how my program runs and functions.

Background

The research paper provided for the assignment states that brine shrimps do not have an extensively long life-cycle. They start out as eggs and are inactive as long as they are dry. The moment they hit water they begin their life. It takes around approximately 24 to 36 hours before they begin hatching. Shrimps that are carried by an adult female Shrimp are guarded by their mother as eggs in a process known as Ovoviviparity in which the eggs leave their mothers when they are ready to hatch. Larvae shrimps will swim in the water and once the shrimps reach a mature age they are ready to mate. A male Adult shrimp will clasp onto the tail of a female shrimp when it mates, and the life cycle continues. As the shrimps are mating they tend to swim faster together.

Based on the information gathered from the Scientific Research Paper I was able to formulate the conditions and parametres required for this Assignment. This extensive knowledge allowed me to formulate a constructive and realistic representation of brine shrimp. It was decided that the shrimps were needed to start out as eggs that are then released into salt water. Once the Eggs reached the salt water it triggered their ability to begin their lifecycle and age. In the simulation program the eggs shuffle a small amount of distance to mimic that they are being dropped into the water. As the days pass the egg slowly hatches and transitions into a hatchling. The older a shrimp is the faster it progresses its speed and movement in the water. However, it was also identified in the scientific research report that the brine shrimp survival rate rapidly decreases as it ages. This is because the maximum natural age of a brine shrimp is around 4- 5 weeks, no more no less. The probability utilised was to consider other factors that may affect the survival of the shrimp, those include whether it has been perhaps eaten by a prey or if it strays into freshwater where it is unable to survive.

Furthermore, the information regarding collision and reproduction were taken into account when constructing the simulation program. In my program I had decided to check for collisions during each step change. In this collision detection method shrimps that were of an adult age and opposite gender were considered to be suitable for mating. Once both shrimps have mated a single egg is released into the salt water at the exact location the mating took place. This is parallel to reality in which any two adult brine shrimps that mate will reproduce eggs and release them into the water where they then hatch and continue their life stage.

My choice of customisable parameters stems from the fact that the size of a fish tank can have a massive effect on how frequently shrimps will mate. If a tank is small and the population of shrimps is greater than more shrimps will reproduce and that will increase the total population of shrimps. The population of the shrimp was taken into account as well. Users can manipulate the program to include as little or as much shrimps as possible provided that no negative numbers will be provided. In that case the program will invalidate everything and exit the program. Finally, I have also decided to incorporate the option for users to select how long they would like to step into the lifecycle of the shrimps. The longer the timestep the more possibility there is for users to witness the death of older shrimps and new foetus shrimps to be reproduced by existing mature adult shrimps.

Methodology

I decided to keep my methodology for my simulation as simple and as practical as possible. My Simulation program starts by identifying how the program will be run. Whether it be through default input, command line parametres or just regular file input. If a user chooses to adapt the parametres they can select a maximum size for the fish tank where the shrimp will reside. The starting population as well as the timestep are also modifiable if the user chooses to change them.

The main function also handles the error checking to ensure that no invalid parametres are passed that could crash the simulation. At the very beginning of the simulation I had decided that I wanted to randomly seed the simulation so that every output given is unique from the previous simulation. This ensures that no single simulation is ever the same which is exactly how it would occur in real life. Once everything has been validated and seeded the Simulation subsequently sends the relevant parametres to the Simulation function where all the new shrimp eggs will be populated depending on the Population number passed in. While it is looping each shrimp is allocated a randomly generated gender and a unique coordinate on the plot. The object Shrimps that were created are then appended to a list. The reason a list was created and not an array was because there is no limit to how much a list can carry, and a list does not require a maximum value to be defined.

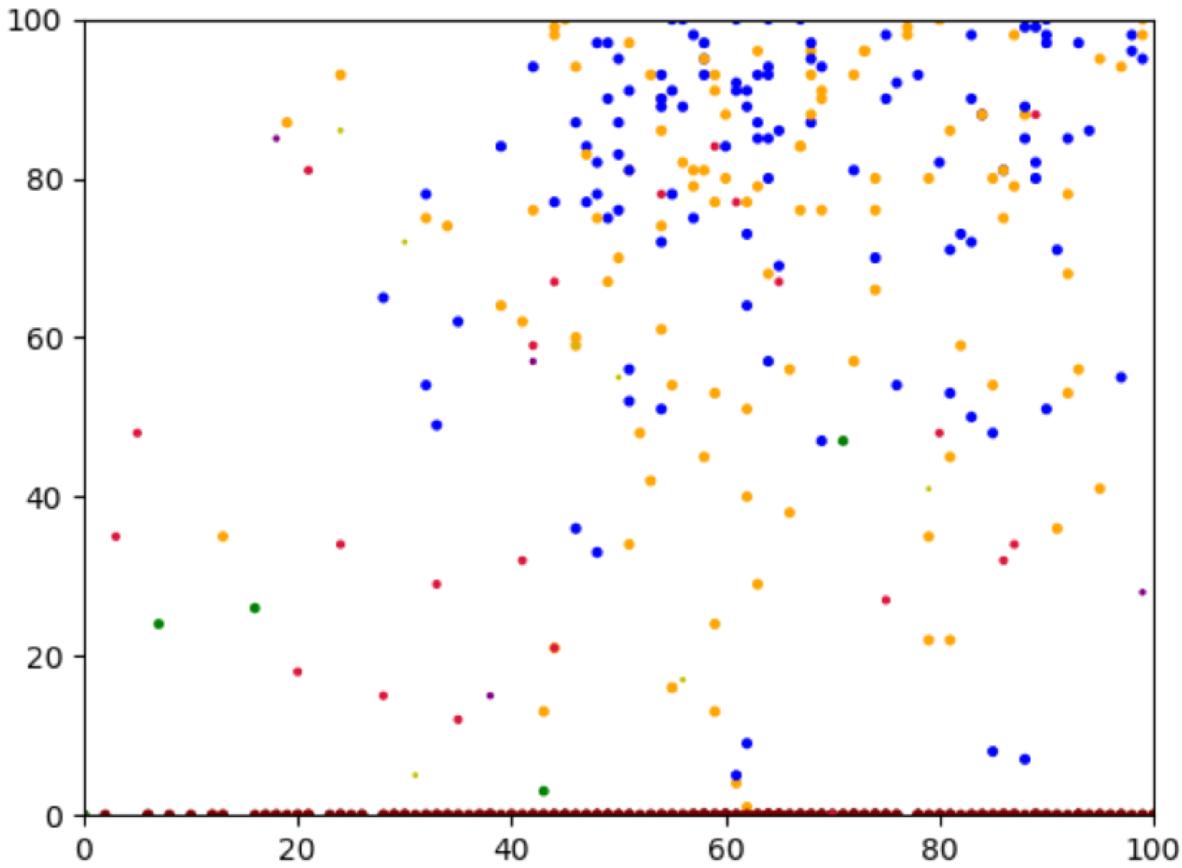
Provided that the users have selected a timestep the shrimps will loop through each time step and their positions will change depending on their state. I chose to vary the speed of the shrimps depending on their age. I made this decision so I could represent the realistic movement of Shrimps during each stage of their life cycle. Although the movements are not exact, they do give out a pretty clear indication of the differences that arise between the movements of a hatchling and matured adult. When the step change function is called a random number from the predefined random library in python is returned and checked against the death probability of the current shrimp. If a shrimp is an adult it has a greater chance of dying by many internal and external factors.

In order to maintain the boundary of the scatter plot I have created a separate function known as the boundary check. The boundary check is called after each step change to guarantee that no shrimp escapes or leaves the fish tank. The way this method works is by checking to see whether the current X and Y Coordinates of the shrimp exceed the maximum value that was set to create the fish tank. If one of the coordinates exceed the maximum size of the fish tank then the coordinates are modified so that they are within the fish tank.

After the boundary checking of the shrimps the execution of the next code for each of the time step continues. The next factor we need to consider would be whether any shrimps collide with each other. For this I had decided that a separate function would need to be created in order to identify if any shrimps in the list have intersected with the current shrimp. In the check collide function a single shrimp iterates through the entire shrimp list and checks each shrimps coordinates against its own. If a shrimp shares the same coordinates and is not the same shrimp then a collision is detected. Once a collision is detected then the function will execute a particular section of code to check and see whether these two shrimps are adults and are the opposite gender.

Once the collision detection method finishes up it returns a string and if that string returns "adults_and_can_mate" a new brine shrimp object is created. The object shrimp will be set to the default state of egg and will age with every passing time step.

When all the shrimps have been iterated through in the loop they are then sent out to be plotted using matplotlib. I decided for my shrimp simulation program to incorporate different colours to signify the different states of my brine shrimp. When the brine shrimp are at their younger stagers in life they have different colours that are represented in the Scatter Plot. For example, when my brine shrimp are in the juvenile stage they plots are represented as green points. When they reach the adult stage, if a shrimp is a female then it is represented in an orange colour, whereas if they are males then I chose to represent the shrimps as a blue point. The reason I chose to use different colours was so that it was visible for users simulating the program to identify the shrimps in their different life stages. The image below represents the simulation with over 500 shrimps in a 100 by 100 size fish tank, when the shrimps die they drop down to the fish tank and change into a maroon colour plot.



To give my assignment more depth I had decided to give my Users the option to Save the string representation of the object in a text file so that the shrimps formed in this randomly unbiased simulation can investigate in depth the gender, states and coordinates of each shrimp during the last time step of the program. This functionality has improved my assignment by making the data generated persistent. The following output below was generated from the previous plot figure and has been outputted to a file.

```
adult @ [58, 81] Gender female
adult @ [34, -6] Gender female
dead @ [12, 0] Gender female
adult @ [62, 91] Gender male
adult @ [58, 45] Gender female
adult @ [68, 88] Gender female
adult @ [46, 87] Gender male
dead @ [26, 0] Gender female
adult @ [65, 69] Gender male
adult @ [46, 60] Gender female
dead @ [65, 0] Gender female
adult @ [72, 57] Gender female
adult @ [53, 42] Gender female
dead @ [33, 0] Gender male
dead @ [67, 0] Gender female
dead @ [44, 0] Gender female
adult @ [90, 51] Gender male
adult @ [91, 36] Gender female
dead @ [20, 0] Gender male
dead @ [59, 0] Gender male
juvenile @ [43, 3] Gender male
juvenile @ [28, 28] Gender female
dead @ [72, 0] Gender male
juvenile @ [16, 26] Gender male
juvenile @ [7, 24] Gender male
juvenile @ [71, 47] Gender male
dead @ [20, 0] Gender female
dead @ [51, 0] Gender male
larvae @ [44, 21] Gender male
larvae @ [33, 29] Gender female
larvae @ [54, 78] Gender female
larvae @ [28, 15] Gender female
larvae @ [35, 12] Gender female
larvae @ [70, 0] Gender male
larvae @ [86, 32] Gender male
larvae @ [59, 84] Gender female
larvae @ [44, 67] Gender female
larvae @ [3, 35] Gender female
larvae @ [41, 32] Gender male
larvae @ [87, 34] Gender male
larvae @ [20, 18] Gender female
larvae @ [24, 34] Gender female
larvae @ [65, 67] Gender male
dead @ [74, 0] Gender female
larvae @ [89, 88] Gender female
larvae @ [80, 48] Gender male
hatchling @ [99, 28] Gender female
hatchling @ [38, 15] Gender male
hatchling @ [42, 57] Gender female
hatchling @ [18, 85] Gender male
dead @ [51, 0] Gender male
egg @ [50, 55] Gender female
egg @ [24, 86] Gender female
egg @ [79, 41] Gender male
egg @ [46, 57] Gender male
egg @ [31, 5] Gender female
egg @ [56, 17] Gender male
egg @ [30, 72] Gender female
egg @ [82, 100] Gender female
```

Finally, once the simulation function completes and the output is saved to a file the program exits gracefully. During the creation of this Program my main goal was to keep the simulation straight forward and concise. This meant that a lot of planning went into the fact that no redundant functions were created, that the functions remained as generic and reusable as possible so that the files remained readable. This is reflected in the several validation submodules that were constructed to ensure all imports were authorized.

Results

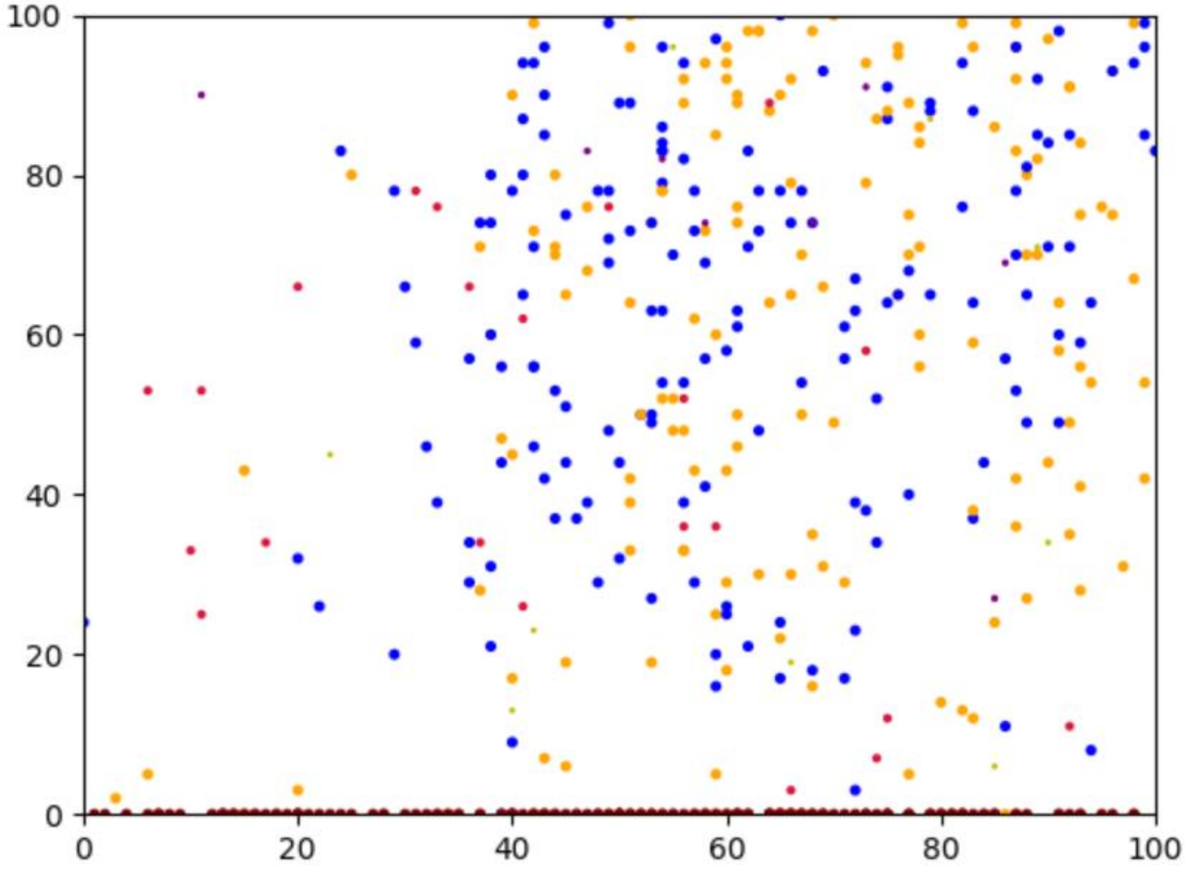
This section explores the results generated from a simulation. I will attempt to run the simulation with custom values and display their results in this report to help reinforce and assist in strengthening an understanding of how this simulation works.

To run this program, I will Attempt to run the Simulation with the following command Line Parametres:

```
python3.7 "c:/Curtin-Uni/2019-Sem 2/Fundamentals of Programming/Assignment New Version/Assignment New 2/shrimpSimBase.py" 100 100 500 20
```

In this simulation I have ran the program with a fish tank size of 100 by 100. The population for the shrimps starts at 500 and the timestep will loop through the shrimps 20 times.

During Timestep 10 the shrimp plot is displayed showing the adults females in orange, the males in blue and a wide range of other shrimps that are younger and in different colours depending on their state.



While the plots are being displayed, during each timestep a terminal output of the string representation of the Shrimps during each step change.

```
hatchling @ [3, 43] Gender female
hatchling @ [29, 70] Gender female
hatchling @ [42, 59] Gender male
hatchling @ [17, 51] Gender male
hatchling @ [12, 57] Gender male
hatchling @ [17, 21] Gender female
hatchling @ [51, 36] Gender female
egg @ [84, 76] Gender male
egg @ [80, 30] Gender male
egg @ [61, 82] Gender male
```

Once all the plots have been displayed, the simulation sends a message to the user asking if the user would like to save the final stage of the shrimps in an output file or to end the program.

```
Press (1) to Output results to a File
(2) to Return
```

In this example I have chosen to write out to a file called fileOutput.txt, once I input the name of the file the program terminates. To view the files contents, I Just need to look into the current directory with the name of the file I just inputted.

```
Please enter the name of the file you would like to save the Shrimps to...
fileOutput.txt
```

Conclusion

This report attempts to exemplify the reasons for the creation of particular functions and the reasoning behind the structure of the assignment. With more planning and thought other methods could have been implemented to create a more efficient and interactive simulation. For example, the simulation program could have included images to represent the shrimps instead of plots to clearly represent which stage the shrimps are currently at. This simulation was heavily influenced by the assistance of the scientific research paper. However, I did have the chance to make my program more interactive in other ways such as including file Input and Output to drive the simulation. With further investigation a more realistic approach could have been considered when two brine shrimps reproduced, I would have instead created a random number of shrimps to mimic the real-life reproduction stage. All in all, this Assignment proved to be exciting and engaging and definitely taught me a lot about the life -stages that shrimp go through.