COMP1005 Assignment Swamp Of Life

Mitchell Pontague 19126924

Curtin University

Overview

The swamp of Life application is designed to simulate the ecosystem of a swamp compromised of 3 different creatures these being Ducks which spawn as eggs and then progress to their adult forms as the simulation progresses, Newts and also shrimps.

The features of this application are movement of creatures (note ducks that are still in their egg state are stationary) this movement is either target towards their designated food source with Ducks 🡪Newts🡪Shrimps🡪Food spawn being the food chain this movement occurs when the distance between the locations of the two objects satisfies the requirements. Otherwise the creature will randomly move to a lesser extent.

COMP1005 Assignment Swamp Of Life

# User Guide

The user can run the application from terminal by navigating to the directory containing the application then entering ‘python3 swampLife.py’ the user will then be prompted to enter in values to indicate the starting values of the three creature objects for the initialization of the simulation.

## Traceability Matrix

|  |  |  |
| --- | --- | --- |
| Feature | Code | Test |
| 1 Animal Object Behaviour |  |  |
| 1.1 Animal Objects (Duck, newt,Shrimps) have position | Class Animal subclass Duck, Newt and Shrimp self.xpos and self.ypos in \_\_init\_\_ function | [Passed] Created all animal object types with assigned position |
| 1.2 Animal Objects (Duck, newt,Shrimps) have age | Class Animal subclass Duck, Newt and Shrimp self.age in \_\_init\_\_ function | [Passed] Created all animal object types with assigned age |
| 1.3 Animal Objects (Duck, newt,Shrimps) have speed | Class Animal subclasses Duck, Newt and Shrimp self.speed in \_\_init\_\_ function | [Passed] Created all animal object types with assigned speed |
| 1.4 Animal Objects (Ducks, Newt and Shrimps) can reproduce | Class Animal reproduction ()  If object age is greater then 5 new Animal created | [Passed] created multiple new Animal objects next to spawner object |
| 1.5 Animal Objects (Ducks, Newts and Shrimps) death | Class Animal death Check ()  Once Animal’s age is greater then 6 chance for death occurs determined by random number generated with age\*2 added to increase change of death as age increases with death guaranteed once age = 10 | [Passed] Animal objects died and where removed from the simulation on death |
| 1.6 Animal Objects (All) corpse to become new food source | Not Implemented  Make new food object at position of animal on death | [Not Implemented] |
| 1.7 Animal Objects(All) have hunger counter | Not implemented | [Not Implemented] |
| 2. Movement |  |  |
| 2.1 Hunting Movement is affected by speed | Animal Class Hunt()  For loop will loop by the speed \*2 times | [Passed] Animal Objects hunting move twice the distance |
| 2.2 Moore Neighbourhoods Used | Movement Class hunt()  Makes the object change the x or y value or both to reduce distance to the target allowing the Animal to move in either of the 8 tiles around them | [Partially Passed] Animal Objects moved using Moore Neighbourhoods when hunting other Animals |
| 2.3 Animal Objects random movement | Movement Class movement()  Objects will move randomly using randomly generated value to change the x and y position | [Partially Passed] Allows for random movement but doesn’t use Moore Neighbourhoods |
| 2.4 Movement is affected by age | Not Implemented | [Not implemented] |
| 2.5 Movement is affected by terrain | Not Implemented | [Not Implemented] |
| 3 Interaction |  |  |
| 3.1 Animal Objects(Hunter) can eat other Animal Objects(prey) when 1 tile away or on same tile | swampLife.py pursue() Will calc distance between the two objects if its 1 or 0 will call the remover function | [Passed] Animals are removed when eaten by other animals next to it |
| 3.2 Animal Object(Shrimp) eats food source when 1 or 0 tiles away | Same as above | [Passed] Same as above |
| 4. Boundaries and terrain |  |  |
| 4.1 Animals and food objects stay within the plot boundaries | swampLife.py moveCrt()  uses a while loop to repeat random movement until a value within the boundary is set | [Passed] Animals all stay within the plot boundaries |
| 4.2 Different terrain is used | Not implemented | [Not Implemented] |
| 5 Visualisation |  |  |
| 5.1 Animals and food are displayed on the plot | swampLife.py plotDucks(),plotNewts(),plotShrimps() and plotFood()  plot all objects on the plot as different coloured dots | [Passed] All objects are displayed on the plot |
| 5.2 Log of events and statistics | Partially Implemented various print statement through the swampLife.py file log events mainly surrounding the hunting of creatures and a print out of all animals at the start of each timestep with their position occurs | [Partially Implemented] |

### Showcase.

#### Introduction.

The Swamp of Life feature include the three animal objects Ducks, Newts and Shrimp which have speed, age, state and reproduction features. These are represented on a plot which are displayed as different colored dots to show their type along with the food Object that is also displayed upon the plot. A terrain is generated which has land and water used as the background of the plot. The simulation allows for the user to enter the desired starting values for Ducks, Newts and Shrimps it also allows for the duration to be modified as well as the boundaries for the plot this is done through the input prompts that are displayed at the start when the program is first started. The results are saved with screenshots being taken at each time step by the application.

##### **Results**.

***Simulation 1. The Standard***

Parameters:

* Ducks: 5
* Newts: 10
* Shrimp: 10
* Time: 15
* Boundaries: 1000x1000

These values are entered directly into the terminal at runtime

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

In this simulation the recommend standard parameters are used this results in a simulation where consumption amongst the creatures doesn’t occur in the early stages allowing for the first reproduction cycle to occur. Steady numbers are held throughout the simulation with no creatures being wiped out. One would notice that the duck population didn’t drop from it peak of ten even after the tenth timestep with the guaranteed death of the original five ducks but the population remained the same due to the second batch of ducks reproducing on the same time step as the death of the original batch this occur due to the reproduction cycle being five time steps and the death cycle being ten time steps.

***Simulation 2. The Big One***

Parameters:

* Ducks: 25
* Newts: 40
* Shrimp: 45
* Time: 20
* Boundaries: 2000x2000

These values are entered directly into the terminal at runtime

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

From this simulation the greater amount of creature resulted in a much larger amount of reproduction and hunting with the ducks who have no natural predators peaking at a population of 50. The Newts peaked at 80 even with the high amount of ducks being present they spend half of their life cycle stationary in their egg state allow for the newts to build up in numbers. The shrimps peaked at 90 then spending the remainder of the simulation hovering around the 70s range.

***Simulation 3. Crowded***

Parameters:

* Ducks: 15
* Newts: 30
* Shrimp: 40
* Time: 15
* Boundaries: 250x250

These values are entered directly into the terminal at runtime

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

This simulation had a look reducing the boundaries while starting with a high number of creatures. Noticeable events in this simulation is the Shrimp lining up along the same X value on step 5, this is due to the limited food spawns on the map making them all advance towards the same spot and this can be seen again on the 9th time step. The population of the shrimp starts to decline significantly in the last couple of time steps due to the small boundaries resulting in the shrimp being hunted down.

#### Conclusion and Future Work

Overall majority of the intended features have been implemented to some extent with the main aspects which are absent being the implementation of a hunger system and the terrain only serving as a backdrop and not having any real impact on the Animals ability to move or other such features. Some features that could be reworked would be the way the Animal classes pursue their targets with many ending up on the same position as each other when hunting the same Animal and then all executing the same moves along as another prey is in range resulting in many overlapping with each other. A logging system could also be implemented to better record the results of a simulation. Food sources need to also be balance to have more random spawns be available beside the first initial spawn and then relying on the natural passing of other creatures to replenish it.

*Sources*

COMP1005 Practical 3 character.py and test3.py

Post for terrain <https://piazza.com/class/l5z3muv3ulcxm/post/216>

COMP1005 Assignment Sample Code

Abs() function https://www.geeksforgeeks.org/abs-in-python/