Abstract

This abstract is mainly to explain a project I have implemented, which simulates various life forms on the planet PaSciRo. In abstract I will introduce three lifeforms of interest, each with their own behaviour and interaction rules, there are complex interactions between their populations.

Discipline of Computing , Curtin University

By simulating life on the planet PaSciRo, which is inhabited by three lifeforms of interest, each with their own behaviour and interaction rules. In the experiment, feature selection is import , I choose species's normal speed, accelerated speed for interaction, color/shape for showing, sizes for Individuality, adulthood, juvenile. varying the input parameters(different population of all species and steps of all stage to determin how long the experiment runs), to see how they impact the overall simulation.

My research findings indicate that: at each stage, there is a lot of randomness in the survival, reproduction and death of individuals in each group(, but from a longer-term perspective, the results of each time are still regular and feasible from a statistical point of view. For example, different ethnic groups directly run multiple times. The result shows that the population ratio is close to the initial value we set. Change the paramter of populations and steps of whole expeirement, Statistical probability of being discovered.

Due to time is limited, rough assumptions are used in many places(like the range we assume PaSciRoTiger catch PaSciRoSheep), and when there is more time in the future, various assumptions can be further refined.

Background

This pasciro Life game project: I have learned simulations, object-orientation, data Visualisation(matplotlib etc.), data processing(especially numpy and etc). In this assignment, I am making use of all those knowledge to extend a given simulation to provide more functionality, complexity and allow automation by given turns. By Simulating how the individual attributes of different populations affect the survival state of the final population on the planet (total number, number of new generations, location distribution)

features of this pasciro Life game project:
I will be simulating life on the planet PaSciRo, which is inhabited by three lifeforms of interest, each with their own behaviour and interaction rules:
PaSciRoTiger
PaSciRoSheep
PaSciRoBird
PaSciRoTiger is the alien predator (bigest size, yellow polygon) runs faster, has fewer numbers, is larger, and has faster average speed and acceleration spped;
there are more alien sheep **PaSciRoSheep** (blue, medium size，Cross shape), compare to predators Speed and acceleration speed, they are a bit lower than that of predators (but there is still a little chance of escape;
triangles are alien birds **PaSciRoBird**(red , small, triangle), the number is the most, the speed is the least, and the probability of predators eating them is lower.

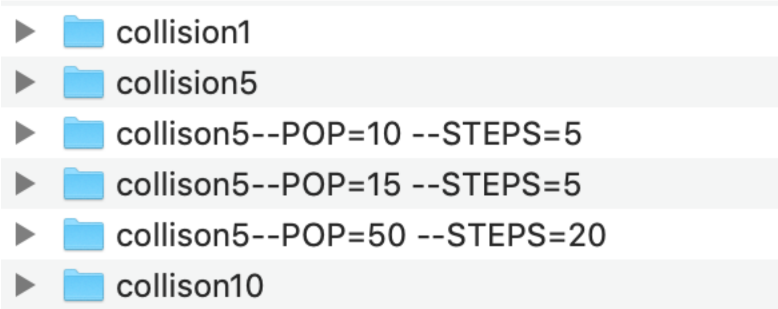
Each ethnic group is different, for example, whether different life forms move in different ways, **PaSciRoTiger** usually fight alone, PaSciRoSheep group and PaSciRoBird generally move in the same direction unless they meet direction change event with a certain probability: When encountering a **PaSciRoTiger**, both PaSciRoSheep group and PaSciRoBird start the acceleration function, and escape at a maximum speed faster than usual.

Different types of encounters or collisions，in bad luck situation：the predator will die and be removed from the map. When the same type is in collisions, the pregnancy attribute of the population is set, and after a random time, the population will produce a certain number of new generation.The speed and acceleration speed of the predator **PaSciRoTiger**, the dynamic change between the speed and acceleration of the predator’s **PaSciRoSheep** will affect the number of their new generation, and final count at end stage of game.

The inital number of total population and number of stage（the length of time the experiment was conducted）also alter the result.

Methodology

different collision range we used to experiement as following:
range =1
range=5
range=10
results save in save_png folders as following:



different parameter we experiement as following:
python3 pasciro_life_simulation.py
python3 pasciro_life_simulation.py --POP=50 —STEPS=20
python3 pasciro_life_simulation.py --POP=10 --STEPS=20
python3 pasciro_life_simulation.py --POP=10 --STEPS=5

python3 pasciro_life_simulation.py —STEPS=5

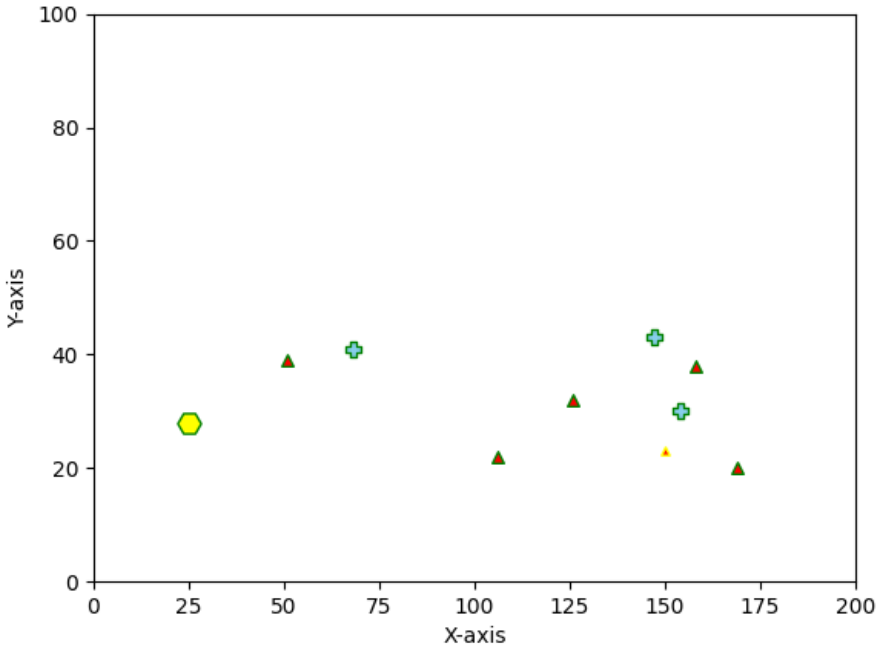
output of the programm should as following:

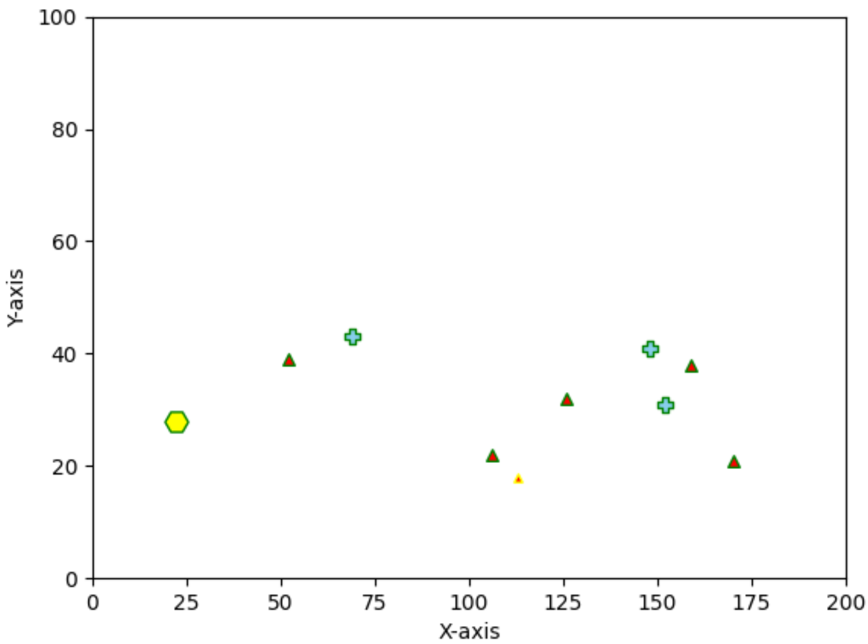
```
parameter: 20 5
PASCIROTIGER        : normal_speed 8       Colour:  yellow \size:  120
PASCIROSHEEP         : normal_speed 3      Colour:  skyblue \size:  60
PASCIROBIRD         : normal_speed 1       Colour:  red \size:  30
[186  71 153] [64 77 51]
[ 14  71 148 102  73 114 128] [20 72 47 13 20 68 12]
[163  22  99 185  70  50  21  16 128 179] [14 57 77 32 42 11 10 11 66 75]
---------- turn 0 ----------
reproduction ->->->->->->->->->->->
---------- turn 0 ----------
run faster, in dangerous in distance: 13.892443989449804 71 72  in  [ 14  71 148 102  73 114 128]
[20 72 47 13 20 68 12]
run faster, in dangerous in distance: 5.830951894845301 148 47  in  [ 14  71 148 102  73 114 128]
[20 72 47 13 20 68 12]
reproduction ->->->->->->->->->->->
---------- turn 0 ----------
run faster, in dangerous in distance: 17.69180601295413 179 75  in  [163  22  99 185  70  50  21
16 128 179] [14 57 77 32 42 11 10 11 66 75]
reproduction ->->->->->->->->->->->
---------- turn 1 ----------
reproduction ->->->->->->->->->->->
---------- turn 1 ----------
run faster, in dangerous in distance: 3.1622776601683795 80 85  in  [ 28  80 160 116  85 127 141]
[29 85 59 24 29 80 25]
Collision ####################
removed out of map, death!
run faster, in dangerous in distance: 11.045361017187261 160 59  in  [ 28 500 160 116  85 127
141] [ 29 500  59  24  29  80  25]
boundary warning! PaSciRoSheep [ 37 509 172 125  99 136 150] [ 40 510  71  33  42  94  38]
```

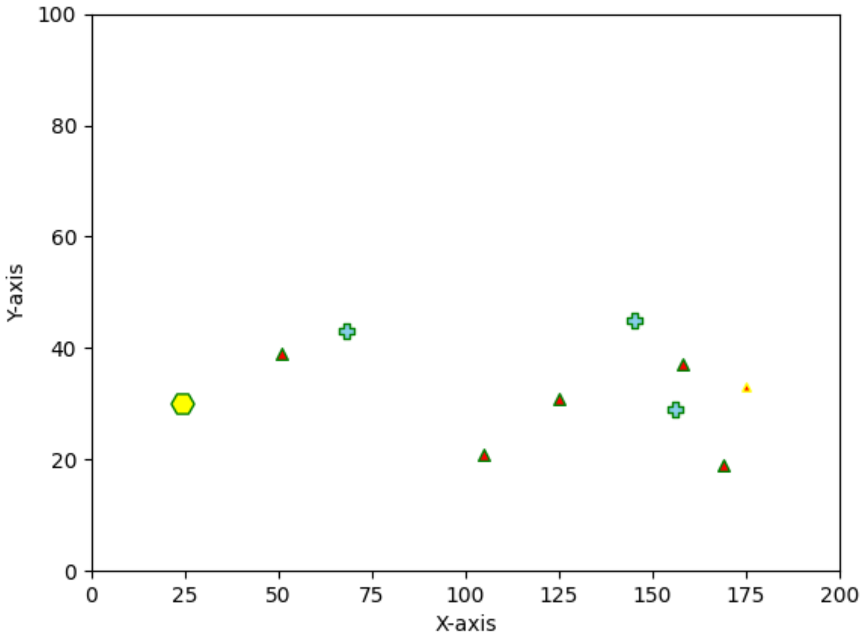
```
(array([], dtype=int64),) (array([1]),) (array([], dtype=int64),) (array([1]),)
x_step, y_step [ 9  0 12  9 14  9  9] [11  0 12  9 13 14 13]
reproduction ->->->->->->->->->->->
---------- turn 1 ----------
run faster, in dangerous in distance: 18.384776310850235 100 79  in  [164  23 100 186  71  52  22
18 129 180] [16 58 79 34 43 12 11 12 67 77]
run faster, in dangerous in distance: 8.54400374531753 180 77  in  [164  23 100 186  71  52  22
18 129 180] [16 58 79 34 43 12 11 12 67 77]
reproduction ->->->->->->->->->->->
---------- turn 2 ----------
reproduction ->->->->->->->->->->->
---------- turn 2 ----------
run faster, in dangerous in distance: 14.035668847618199 172 71  in  [ 37 500 172 125  99 136
150] [ 40 500  71  33  42  94  38]
run faster, in dangerous in distance: 12.36931687685298 150 38  in  [ 37 500 172 125  99 136
150] [ 40 500  71  33  42  94  38]
boundary warning! PaSciRoSheep [ 48 514 183 137 111 149 159] [ 54 512  82  46  54 103  49]
(array([], dtype=int64),) (array([1]),) (array([], dtype=int64),) (array([1, 5]),)
x_step, y_step [11  0 11 12 12 13  9] [14  0 11 13 12  0 11]
reproduction ->->->->->->->->->->->
---------- turn 2 ----------
run faster, in dangerous in distance: 9.219544457292887 184 81  in  [167  27 104 189  75  56  26
21 132 184] [19 62 83 37 46 15 14 15 70 81]
reproduction ->->->->->->->->->->->
---------- turn 3 ----------
reproduction ->->->->->->->->->->->
---------- turn 3 ----------
run faster, in dangerous in distance: 15.0 183 82  in  [ 48 500 183 137 111 149 159] [ 54 500  82
46  54  94  49]
run faster, in dangerous in distance: 10.295630140987 137 46  in  [ 48 500 183 137 111 149 159] [
54 500  82  46  54  94  49]
run faster, in dangerous in distance: 13.152946437965905 159 49  in  [ 48 500 183 137 111 149
159] [ 54 500  82  46  54  94  49]
boundary warning! PaSciRoSheep [ 66 518 198 155 128 165 175] [ 73 515 101  66  73 114  69]
(array([], dtype=int64),) (array([1]),) (array([], dtype=int64),) (array([1, 2, 5]),)
x_step, y_step [18  0 15 18 17 16 16] [19  0  0 20 19  0 20]
reproduction ->->->->->->->->->->->
---------- turn 3 ----------
run faster, in dangerous in distance: 15.297058540778355 186 82  in  [168  28 106 190  77  57  27
22 133 186] [21 63 84 39 47 17 15 16 72 82]
reproduction ->->->->->->->->->->->
---------- turn 4 ----------
reproduction ->->->->->->->->->->->
---------- turn 4 ----------
run faster, in dangerous in distance: 14.142135623730951 66 73  in  [ 66 500 198 155 128 165
175] [ 73 500  82  66  73  94  69]
run faster, in dangerous in distance: 10.44030650891055 175 69  in  [ 66 500 198 155 128 165
175] [ 73 500  82  66  73  94  69]
boundary warning! PaSciRoSheep [ 79 513 208 164 138 178 186] [ 82 509  94  76  82 105  82]
(array([], dtype=int64),) (array([1, 2]),) (array([], dtype=int64),) (array([1, 5]),)
x_step, y_step [13  0  0  9 10 13 11] [ 9  0 12 10  9  0 13]
reproduction ->->->->->->->->->->->
---------- turn 4 ----------
reproduction ->->->->->->->->->->->
```

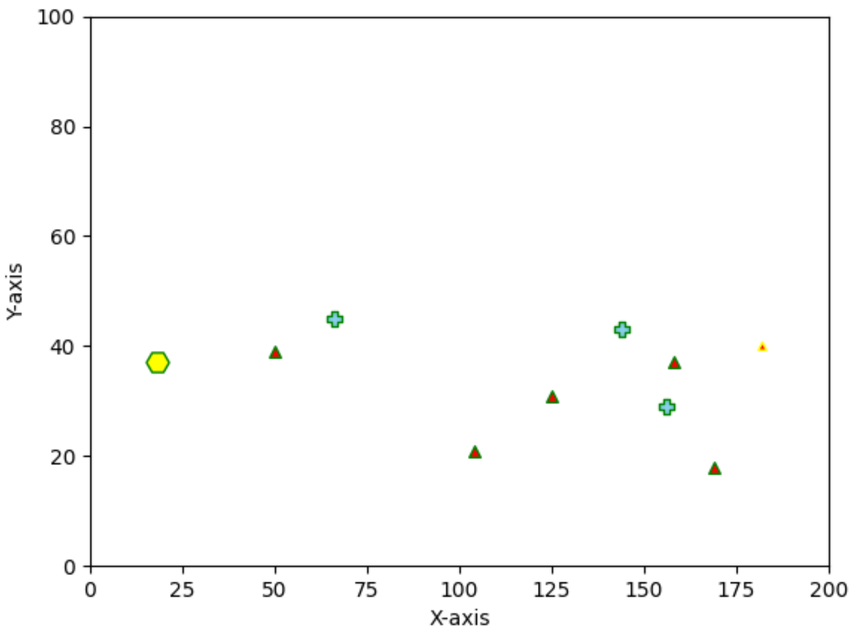
Results

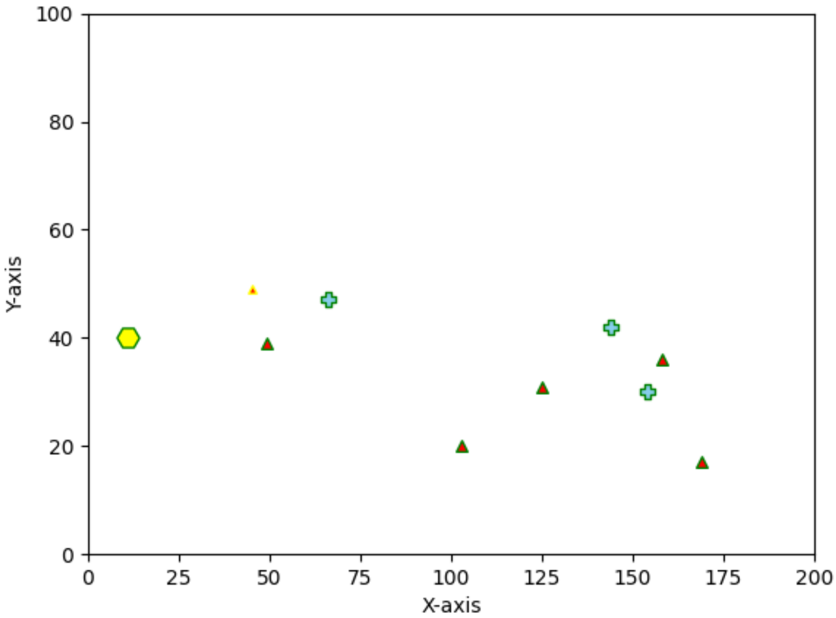
python3 pasciro_life_simulation.py --POP=10 —STEPS=5
The results of your simulations:






100
80
60
40
20
0
Y-axis
0
25
50
75
100
125
150
175
200
X-axis


100
80
60
40
20
0
Y-axis
0
25
50
75
100
125
150
175
200
X-axis



100
80
60
40
20
0
0
25
50
75
100
125
150
175
200
X-axis
Y-axis

Conclusion and Future Work: Give conclusions and what further investigations could follow.

If the total number of all populations (which is controlled by terminal POP parameters) is relatively small, then the space is open, and regardless of the degree of natural carrying capacity of planet pasciro, it is beneficial to PaSciRoSheep and PaSciRoBird, because none of them is eaten.

Increase the total number of population of 3 species, the planet becomes more crowded, and the success rate of predators greatly increases. You can see compare result of:
(results save in save_png folders as following)
collison5--POP=10 —STEPS=5
collison5--POP=15 --STEPS=5

The latter one, more PaSciRoSheep and PaSciRoBird are caputured by PaSciRoTiger.

The total length of the experiment (which is controlled by terminal STEPS parameters)will also greatly change the results. If the predator is given long enough time, and the predator PaSciRoTiger can be patrolled and hunted without considering the fatigue of the predator and lack of energy supply, the time is long enough to always increase the number of successful predators.
You can see compare result of:
(results save in save_png folders as following)
collison5--POP=15 —STEPS=5
collison5--POP=50 --STEPS=20

Future work
We can add fatigue and rest attributes to PaSciRoTiger/PaSciRoSheep/PaSciRoBird, and death and birth can also consider adding age and gender, so that the simulation is more abundant. The distance between the young generation and adult individuals can also be considered: the longest predation distance should be subject to the positive correlation with the maximum distance of the young generation.

Reference material
https://matplotlib.org/3.3.2/api/_as_gen/matplotlib.pyplot.scatter.html
https://matplotlib.org/3.3.2/api/markers_api.html#module-matplotlib.markers
https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life