

Project Title: EcoSysSim

Team Members: Gavin Morrison and Jonathan Wu

Repo link: <https://github.com/jonathancsci/eco-sys-sim>

Work done:

As of right now, we have implemented the grid environment and its basic structure, animals objects that will evaluate their surroundings and move throughout the environment, full implementations for various animal actions which can be executed but aren't included in the decision logic yet, and the interface for the population graph which we will use to plot animal populations in the final product.

Work breakdown:

Jonathan: High level application architecture, implementation of Ecosystem, Environment, Plot with related tests, set up a typer cli app.

Gavin: Basic implementation of the animals classes and their behavior logic and full implementation of actions.

Changes/Issues: Added typer as dependency to create a cli app. There were some issues with Gavin being unfamiliar with Python imports, but Jonathan was able to help.

Patterns:

- Facade: Ecosystem is a Facade encapsulating the details of the simulation
- Observer: The matplotlib plotting is handled by an observer that looks at the Ecosystem from the outside
- Composite: Grids contain a list of their neighbors, creating an implicit interconnected graph structure
- Command: the animals make decisions and return their intended Actions
- Strategy: the Actions are also strategies, only having an abstract execute method implemented by the concrete Action classes

Test Coverage report:

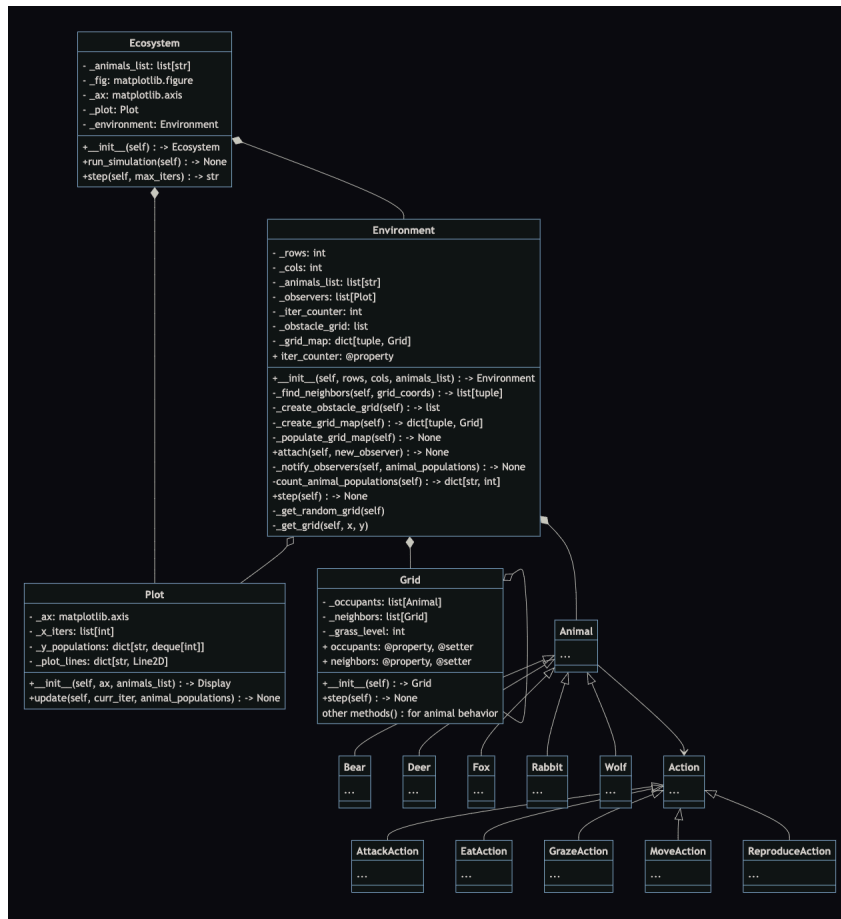
```

----- coverage: platform linux, python 3.12.2-final-0 -----
Name                                                                    Stmts   Miss  Cover
-----
/workspaces/python/eco-sys-sim/eco_sys_sim/__init__.py                  0      0   100%
/workspaces/python/eco-sys-sim/eco_sys_sim/actions/__init__.py          0      0   100%
/workspaces/python/eco-sys-sim/eco_sys_sim/actions/action.py            19      3    84%
/workspaces/python/eco-sys-sim/eco_sys_sim/actions/attack_action.py     13      0   100%
/workspaces/python/eco-sys-sim/eco_sys_sim/actions/eat_action.py         16      0   100%
/workspaces/python/eco-sys-sim/eco_sys_sim/actions/graze_action.py       13      0   100%
/workspaces/python/eco-sys-sim/eco_sys_sim/actions/move_action.py        16      0   100%
/workspaces/python/eco-sys-sim/eco_sys_sim/actions/reproduce_action.py     20      0   100%
/workspaces/python/eco-sys-sim/eco_sys_sim/animals/__init__.py          0      0   100%
/workspaces/python/eco-sys-sim/eco_sys_sim/animals/animal.py            54      2    96%
/workspaces/python/eco-sys-sim/eco_sys_sim/animals/bear.py               4      0   100%
/workspaces/python/eco-sys-sim/eco_sys_sim/animals/deer.py               4      0   100%
/workspaces/python/eco-sys-sim/eco_sys_sim/animals/fox.py               12      6    50%
/workspaces/python/eco-sys-sim/eco_sys_sim/animals/rabbit.py            14      4    71%
/workspaces/python/eco-sys-sim/eco_sys_sim/animals/wolf.py               4      0   100%
/workspaces/python/eco-sys-sim/eco_sys_sim/environment.py                76     10    87%
/workspaces/python/eco-sys-sim/eco_sys_sim/grid.py                       32      1    97%
/workspaces/python/eco-sys-sim/eco_sys_sim/plot.py                       20      0   100%
__init__.py                                                               0      0   100%
test_actions.py                                                            67      0   100%
test_animals.py                                                            38      0   100%
test_environment.py                                                        49      1    98%
test_grid.py                                                                9      0   100%
test_plot.py                                                                12      0   100%
-----
TOTAL                                                                    492     27    95%

===== 17 passed in 3.32s =====
vscode => /workspaces/python/eco-sys-sim/tests (main) $ 

```

UML:



BDD:

When CLI app is executed

Then I should be prompted to enter simulation settings

Given CLI app is executed

When I successfully enter in simulation settings

Then I should see a plot appear on the screen.

Given I set grid size to 1x3

And I set obstacles to 0

And I add a Fox to 0,0

And I add a Rabbit to 0,1

When I step

Then Rabbit should be in 0,2

And Fox should be in 0,1

Given I set grid size to 2x2

And I add 4 Fox

And I add 8 Rabbit

When a Rabbit dies

Then Fox population should be 3

And Rabbit population should be 7

Plan for next iteration:

- Complete the CLI application.
- Add feature to change the probability that a grid is an obstacle
- Implement full animal behaviors
- Add graphical display of environment and animals if time allows

Video link: https://youtu.be/4qta-k4Z1_E