INF200 – BioSim Project

Modelling the Ecosystem of Rossumøya

By Sindre Elias Hinderaker & Mathias Kristiansen

Project structure

Data structure

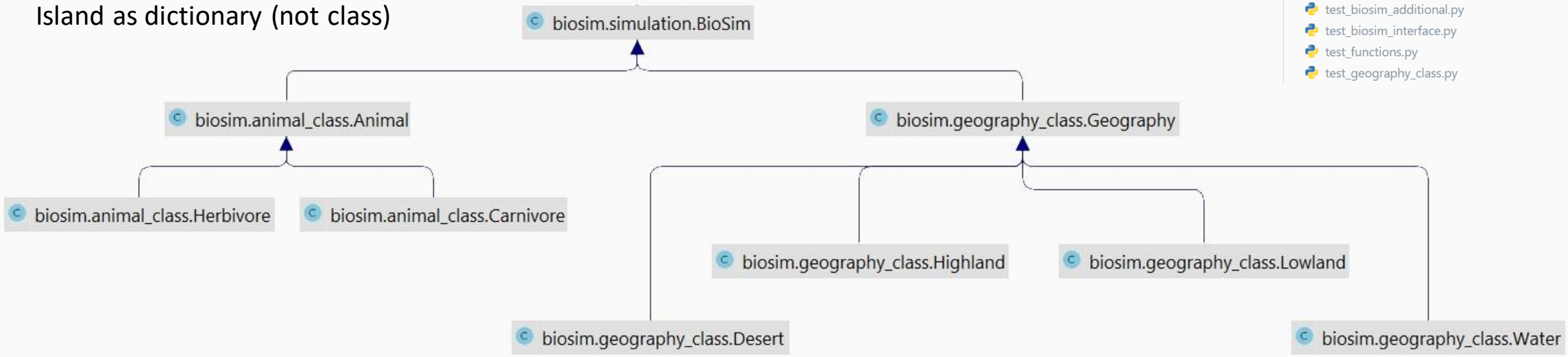
- Alternatives: Class, List, Dict, Set, (Array/Matrix)

Advantages of class-based structure:

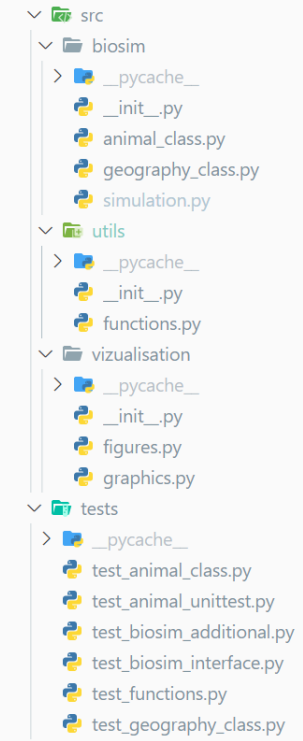
- Inheritance
- 'State' - management

Island as dictionary (not class)

/src/biosim - Class hierarchy



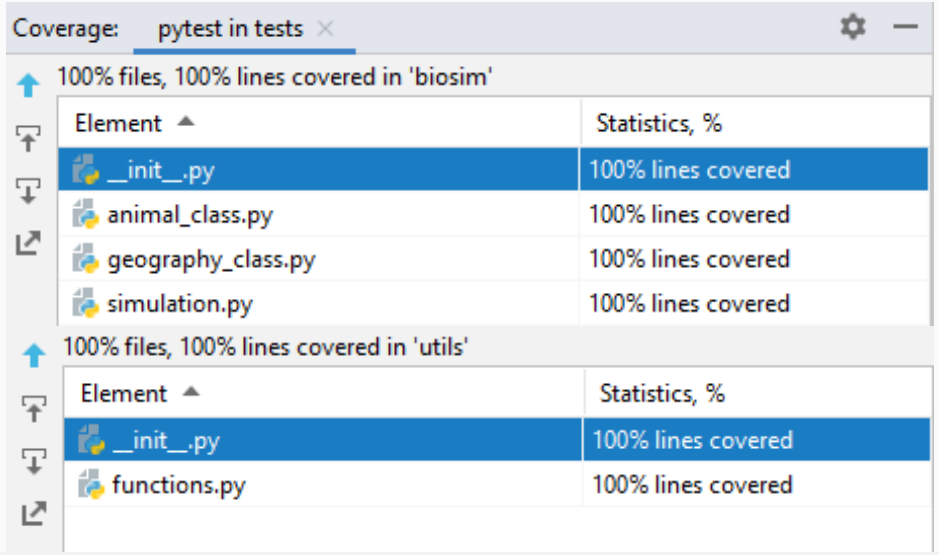
Folder structure

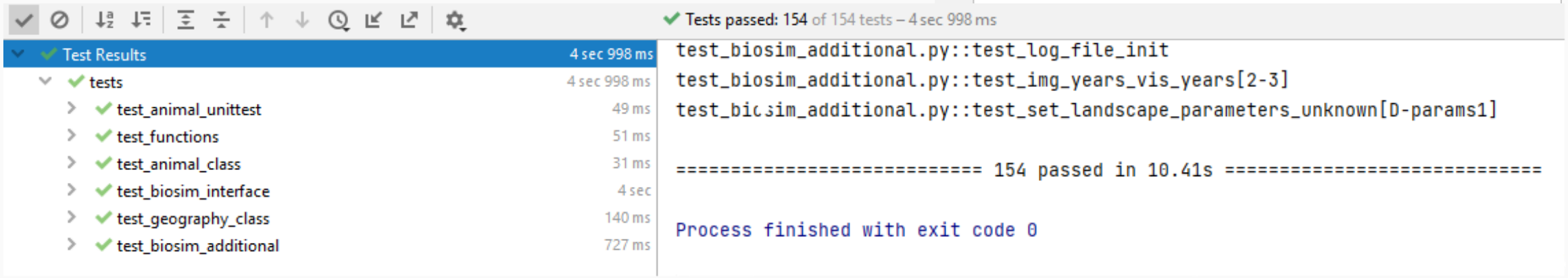


Is it trustworthy?

- How did we ensure quality?
  - Tests, pep8, flake8, tox, etc.
  - CI/CD pipline with GitLab for automatic tox test-run in docker environment

- Is code maintainable?
  - Classes with Sub-Classes (inheritance)
  - Parametrization and fixtures in test
  - Automatically generated code with Sphinx



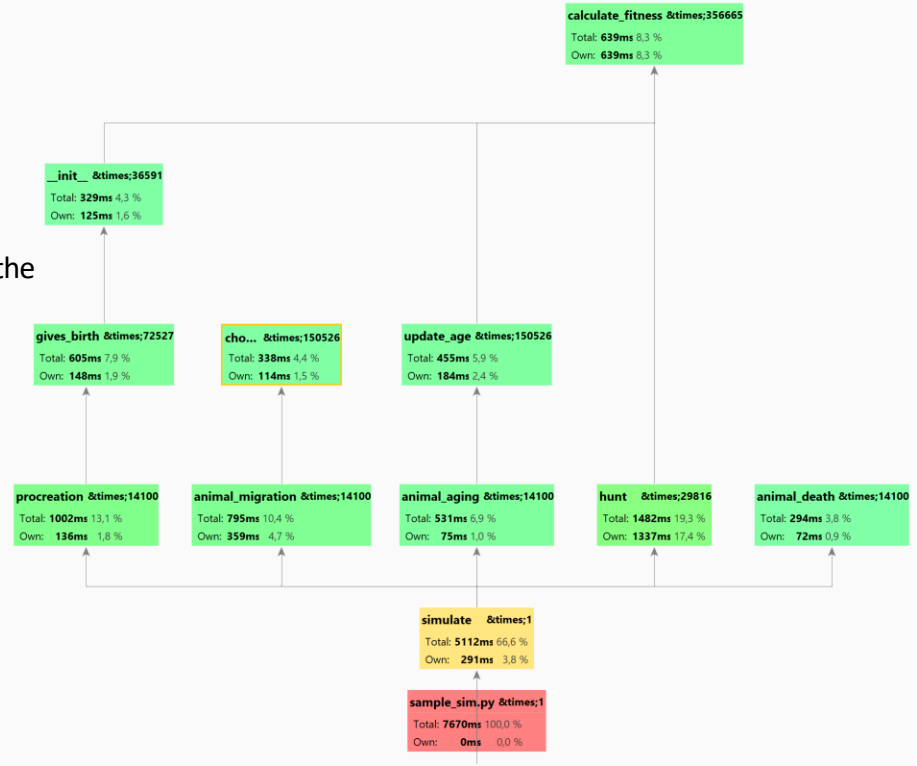


Productive code?

- Ease of use
  - Relatively simple user interface (list, dict,)
  - Only necessary to interact with the interface of the BioSim-class
  - Relatively comprehensive documentation
    - Examples and descriptions

- Performance
  - Results, simulations
  - Seems to be sufficiently fast
  - Room for improvements

Profiling – Call Graph



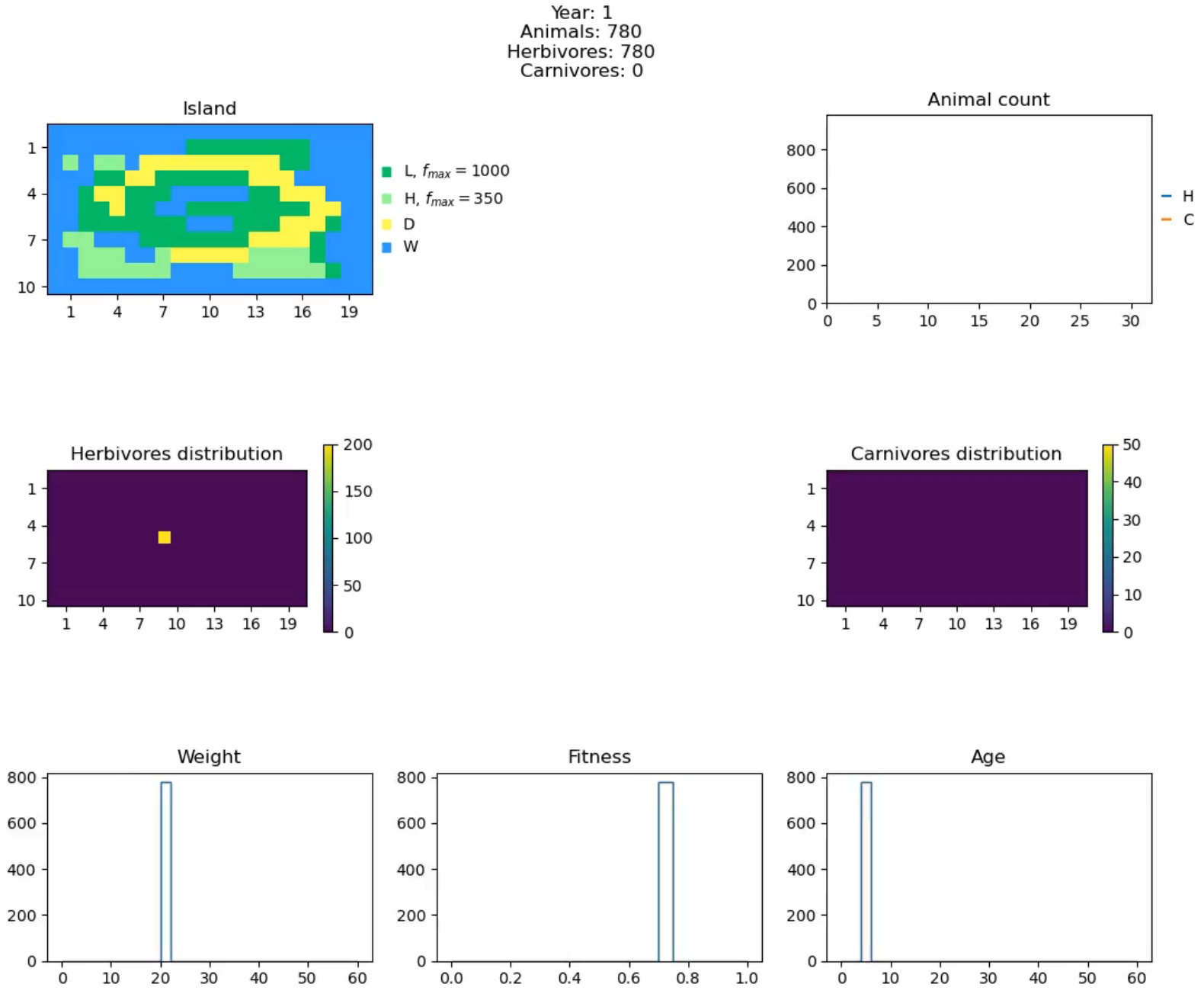
Sphinx – Documentation

Simulation results

Animation 2

- example_2.py

```
sim.set_animal_parameters('Herbivore', {'zeta': 3.2,
                                        'xi': 1.8})
sim.set_animal_parameters('Carnivore', {'a_half': 70,
                                        'phi_age': 0.5,
                                        'omega': 0.3,
                                        'F': 50,
                                        'DeltaPhiMax': 8.})
sim.set_landscape_parameters('L', {'f_max': 1000})
sim.set_landscape_parameters('H', {'f_max': 350})
```

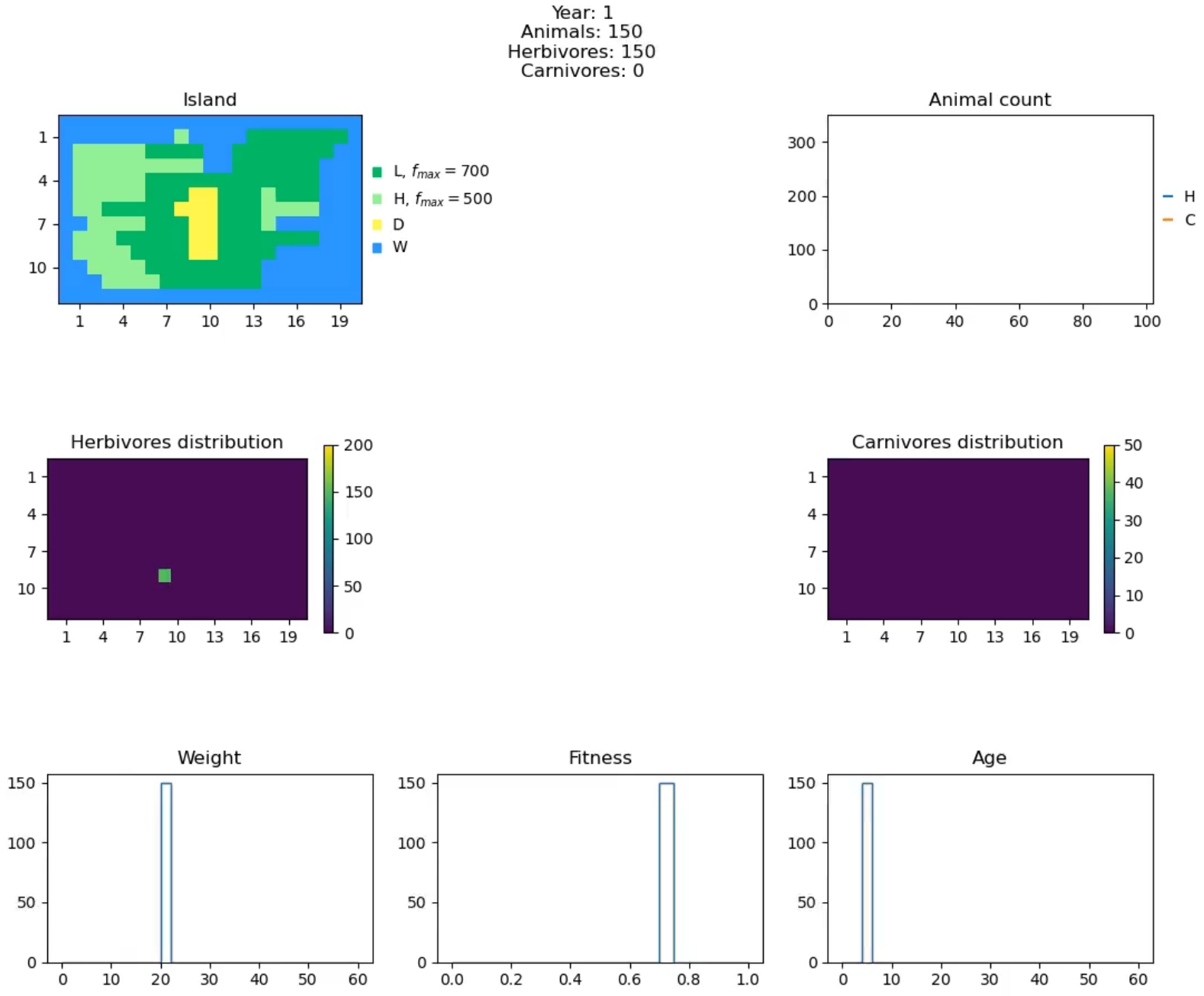


Simulation results

Animation 3

- check_sim.py

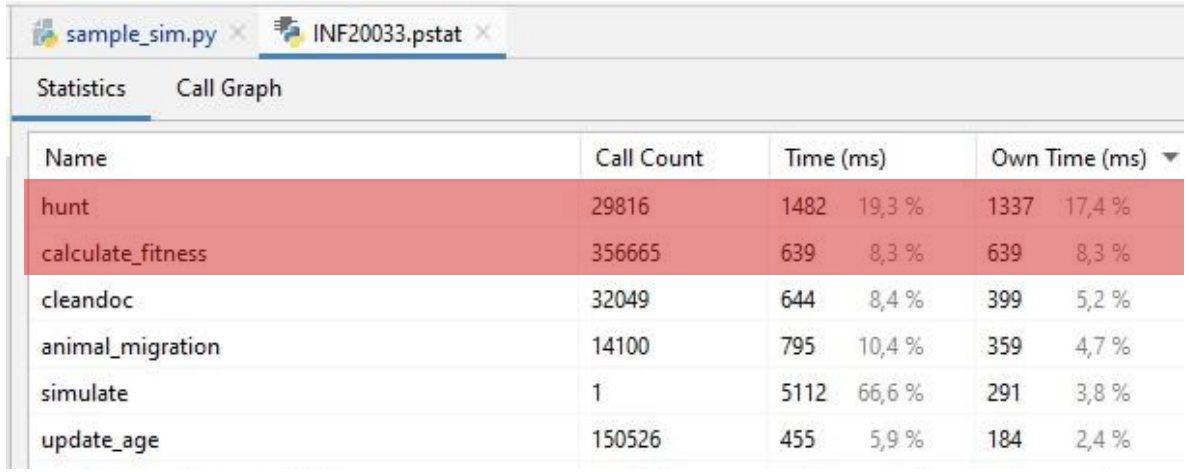
```
sim.set_animal_parameters('Herbivore', {'zeta': 3.2,
                                        'xi': 1.8})
sim.set_animal_parameters('Carnivore', {'a_half': 70,
                                        'phi_age': 0.5,
                                        'omega': 0.3,
                                        'F': 65,
                                        'DeltaPhiMax': 9.})
sim.set_landscape_parameters('L', {'f_max': 700})
```



Further Development

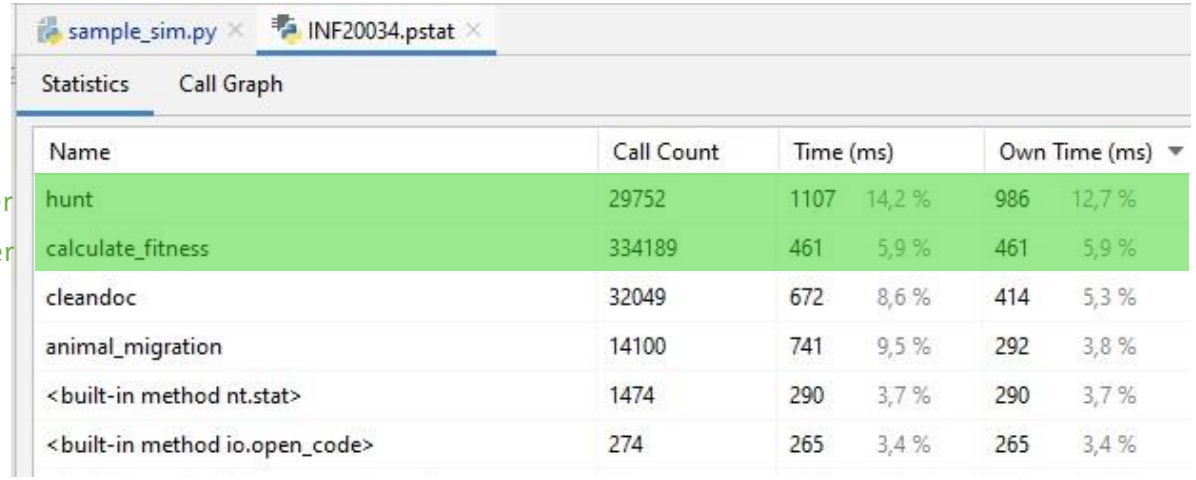
- Error for invalid animal and geography parameters
- Ensure high quality tests
- Separate Island class (?)
- Profiling and optimization
  - List comprehensions
  - Break out of loops when criteria is met
  - Limit function-calls
  - "Lazy computations"
  - Start with
    - hunt
    - calculate_fitness
    - animal_migration
  - Restructuring?

sample_sim.py (graphics disabled) - before optimization



Name	Call Count	Time (ms)	Own Time (ms)
hunt	29816	1482 19,3 %	1337 17,4 %
calculate_fitness	356665	639 8,3 %	639 8,3 %
cleandoc	32049	644 8,4 %	399 5,2 %
animal_migration	14100	795 10,4 %	359 4,7 %
simulate	1	5112 66,6 %	291 3,8 %
update_age	150526	455 5,9 %	184 2,4 %

sample_sim.py (graphics disabled) - after optimization



	Name	Call Count	Time (ms)	Own Time (ms)
~ 27% faster	hunt	29752	1107 14,2 %	986 12,7 %
~ 28% faster	calculate_fitness	334189	461 5,9 %	461 5,9 %
	cleandoc	32049	672 8,6 %	414 5,3 %
	animal_migration	14100	741 9,5 %	292 3,8 %
	<built-in method nt.stat>	1474	290 3,7 %	290 3,7 %
	<built-in method io.open_code>	274	265 3,4 %	265 3,4 %