NMBU

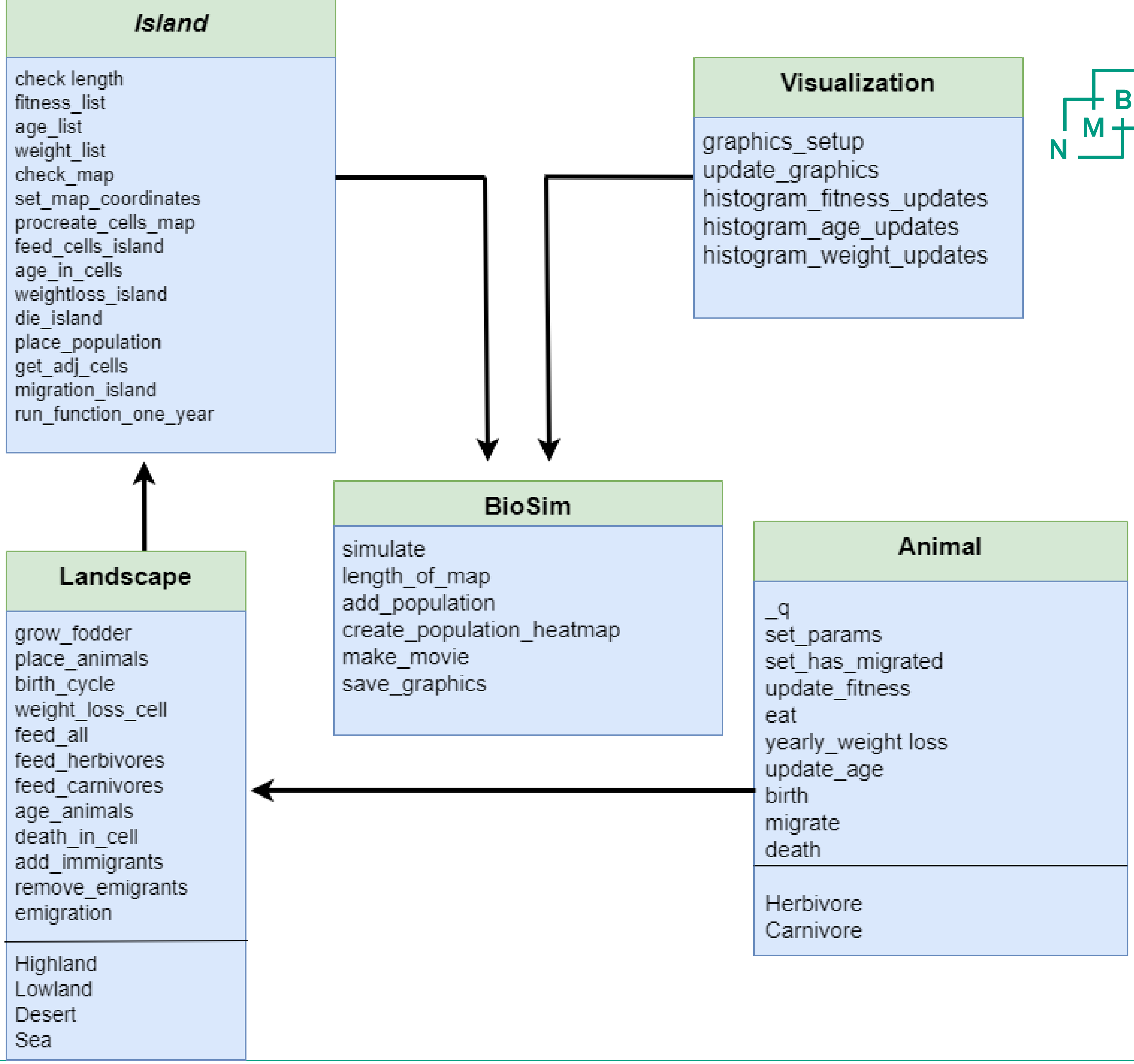
Norwegian University of Life Sciences

INF200 advanced programming

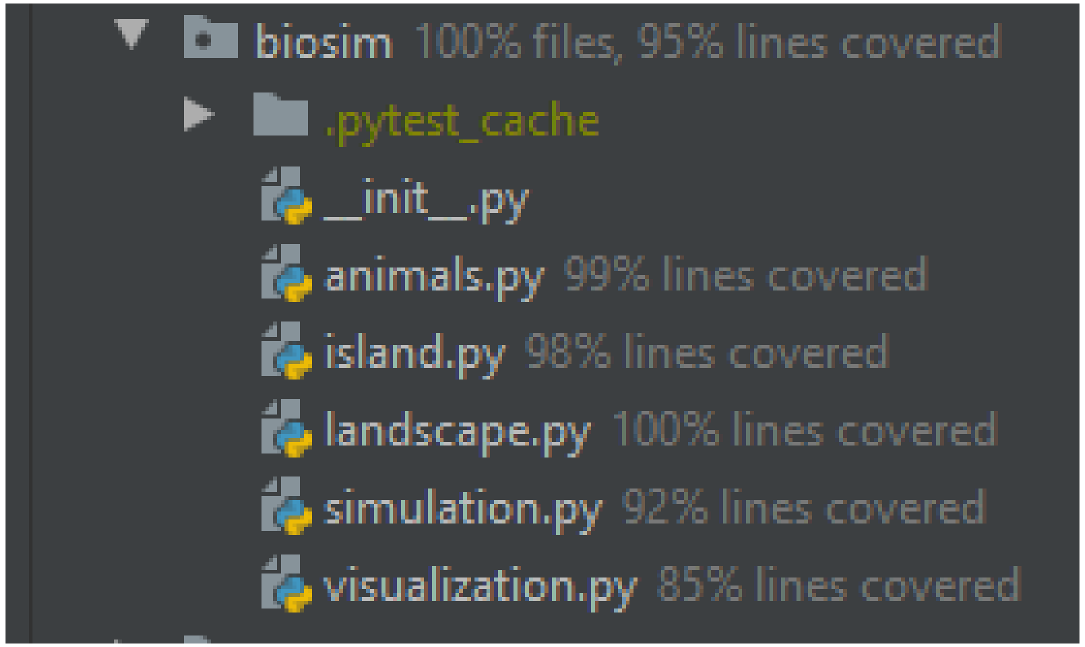
Haris Karovic & Isak Finnøy

23.06.2020

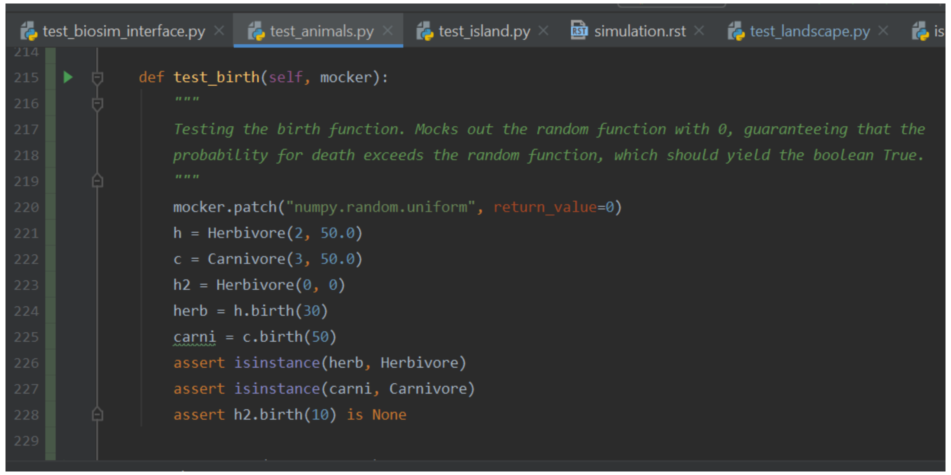
Class diagram that gives an overview of how our code was structured



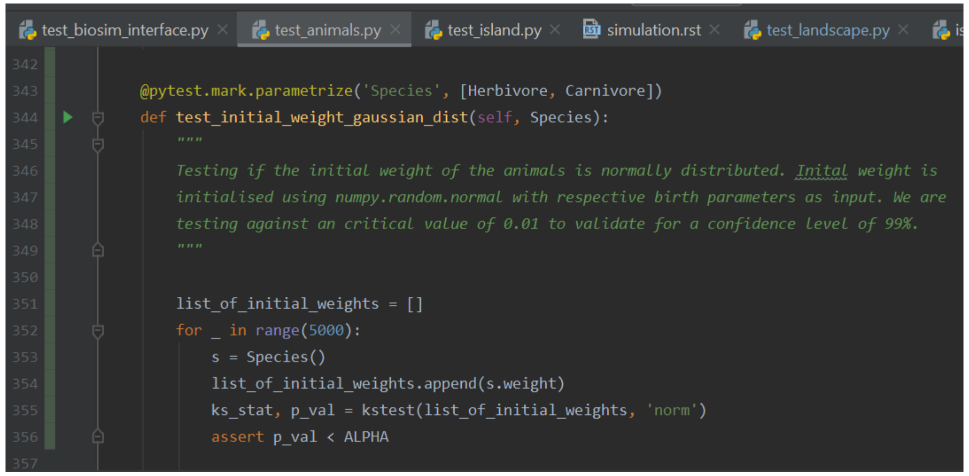
We managed to achieve a 95 % test coverage overall, using advanced techniques such as mocker, statistical tests and @pytest.mark.parametrize. We thus find our code thrustworthy.



```
biosim 100% files, 95% lines covered
    .pytest_cache
    __init__.py
    animals.py 99% lines covered
    island.py 98% lines covered
    landscape.py 100% lines covered
    simulation.py 92% lines covered
    visualization.py 85% lines covered
```

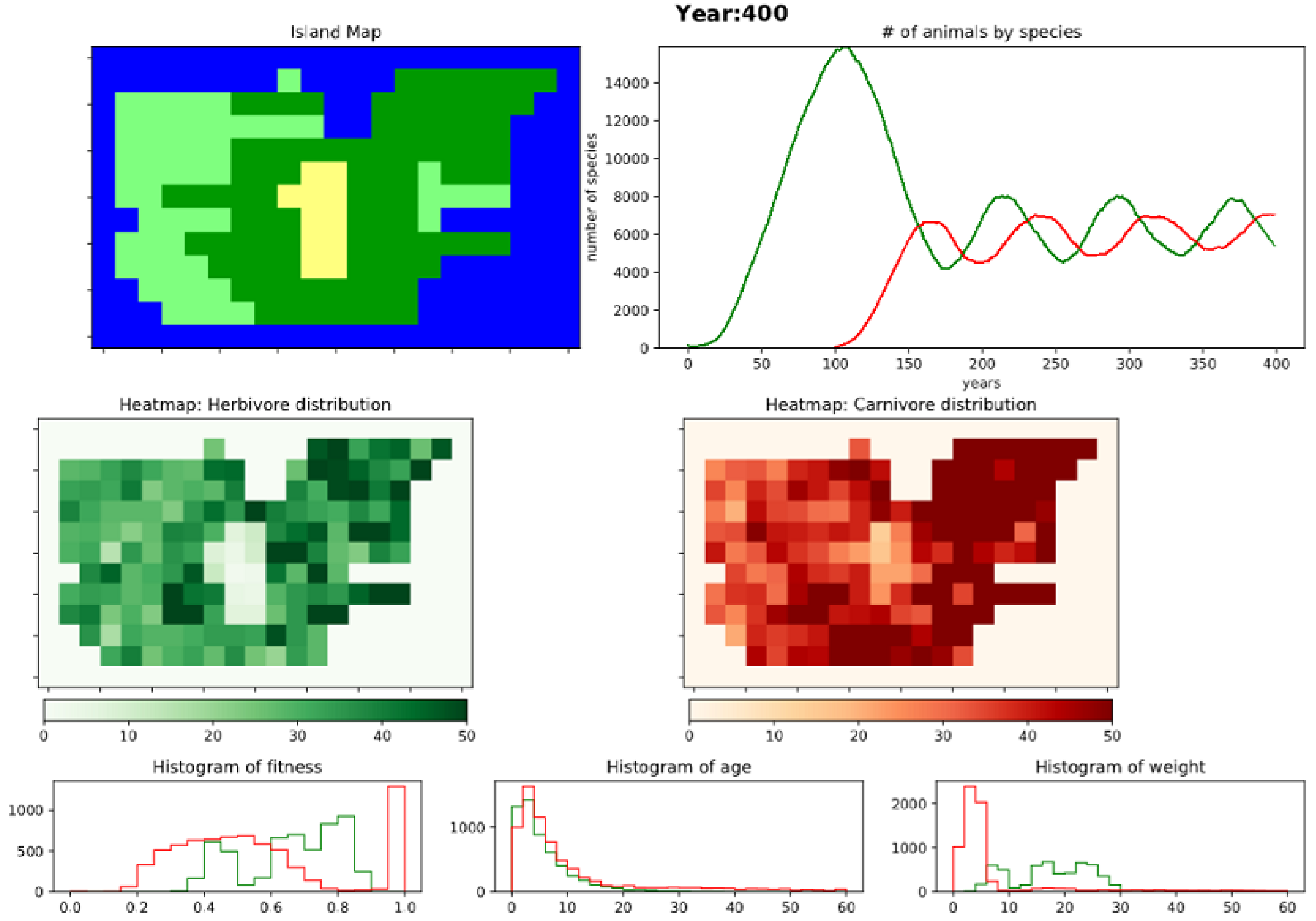


```python
    def test_birth(self, mocker):
        """
        Testing the birth function. Mocks out the random function with 0, guaranteeing that the
        probability for death exceeds the random function, which should yield the boolean True.
        """
        mocker.patch("numpy.random.uniform", return_value=0)
        h = Herbivore(2, 50.0)
        c = Carnivore(3, 50.0)
        h2 = Herbivore(0, 0)
        herb = h.birth(30)
        carni = c.birth(50)
        assert isinstance(herb, Herbivore)
        assert isinstance(carni, Carnivore)
        assert h2.birth(10) is None
```



```python
    @pytest.mark.parametrize('Species', [Herbivore, Carnivore])
    def test_initial_weight_gaussian_dist(self, Species):
        """
        Testing if the initial weight of the animals is normally distributed. Inital weight is
        initialised using numpy.random.normal with respective birth parameters as input. We are
        testing against an critical value of 0.01 to validate for a confidence level of 99%.
        """

        list_of_initial_weights = []
        for _ in range(5000):
            s = Species()
            list_of_initial_weights.append(s.weight)
            ks_stat, p_val = kstest(list_of_initial_weights, 'norm')
            assert p_val < ALPHA
```


Year:400
Island Map
# of animals by species
number of species
years
Heatmap: Herbivore distribution
Heatmap: Carnivore distribution
Histogram of fitness
Histogram of age
Histogram of weight

Plots and visualization of the fauna of Rossumøya

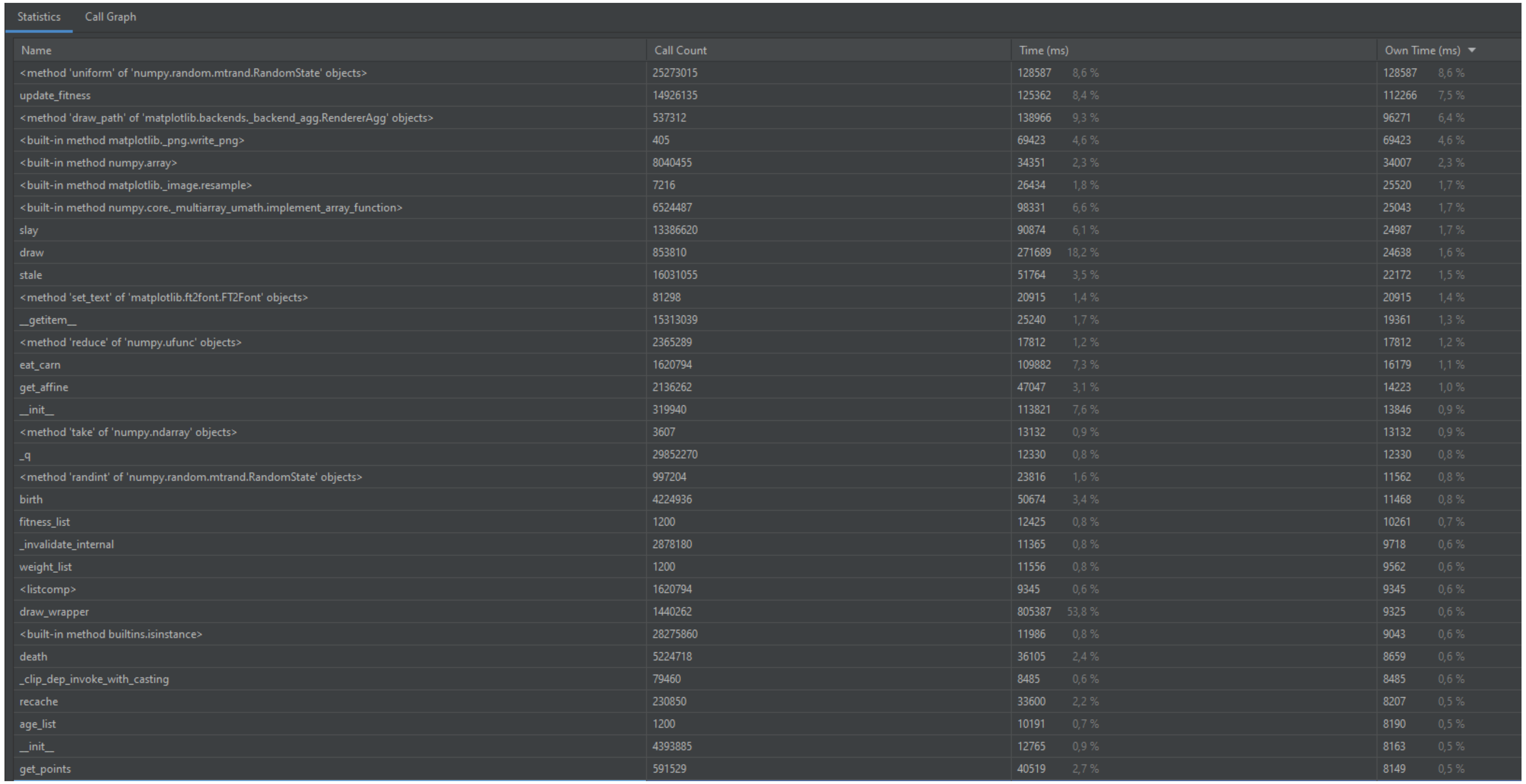
- As shown in the previous slide, the visuals are quite consistent with the expected results.
- We think the plots gives a good overview of the fauna of Rossumøya.
- We should have used legend to indicate which species is what color.
- We could have used a larger color bar to enhance

visibility of herbivore density in different cells

- We did however have a bug in the simulation:

if vis_years is larger than img_years, it will update the plots with an interval twice the size of what vis_years was defined as.

The performance of our code



Name	Call Count	Time (ms)		Own Time (ms) ▼	
<method 'uniform' of 'numpy.random.mtrand.RandomState' objects>	25273015	128587	8,6 %	128587	8,6 %
update_fitness	14926135	125362	8,4 %	112266	7,5 %
<method 'draw_path' of 'matplotlib.backends._backend_agg.RendererAgg' objects>	537312	138966	9,3 %	96271	6,4 %
<built-in method matplotlib._png.write_png>	405	69423	4,6 %	69423	4,6 %
<built-in method numpy.array>	8040455	34351	2,3 %	34007	2,3 %
<built-in method matplotlib._image.resample>	7216	26434	1,8 %	25520	1,7 %
<built-in method numpy.core._multiarray_umath.implement_array_function>	6524487	98331	6,6 %	25043	1,7 %
slay	13386620	90874	6,1 %	24987	1,7 %
draw	853810	271689	18,2 %	24638	1,6 %
stale	16031055	51764	3,5 %	22172	1,5 %
<method 'set_text' of 'matplotlib.ft2font.FT2Font' objects>	81298	20915	1,4 %	20915	1,4 %
__getitem__	15313039	25240	1,7 %	19361	1,3 %
<method 'reduce' of 'numpy.ufunc' objects>	2365289	17812	1,2 %	17812	1,2 %
eat_carn	1620794	109882	7,3 %	16179	1,1 %
get_affine	2136262	47047	3,1 %	14223	1,0 %
__init__	319940	113821	7,6 %	13846	0,9 %
<method 'take' of 'numpy.ndarray' objects>	3607	13132	0,9 %	13132	0,9 %
_q	29852270	12330	0,8 %	12330	0,8 %
<method 'randint' of 'numpy.random.mtrand.RandomState' objects>	997204	23816	1,6 %	11562	0,8 %
birth	4224936	50674	3,4 %	11468	0,8 %
fitness_list	1200	12425	0,8 %	10261	0,7 %
_invalidate_internal	2878180	11365	0,8 %	9718	0,6 %
weight_list	1200	11556	0,8 %	9562	0,6 %
<listcomp>	1620794	9345	0,6 %	9345	0,6 %
draw_wrapper	1440262	805387	53,8 %	9325	0,6 %
<built-in method builtins.isinstance>	28275860	11986	0,8 %	9043	0,6 %
death	5224718	36105	2,4 %	8659	0,6 %
_clip_dep_invoke_with_casting	79460	8485	0,6 %	8485	0,6 %
recache	230850	33600	2,2 %	8207	0,5 %
age_list	1200	10191	0,7 %	8190	0,5 %
__init__	4393885	12765	0,9 %	8163	0,5 %
get_points	591529	40519	2,7 %	8149	0,5 %

Playing video