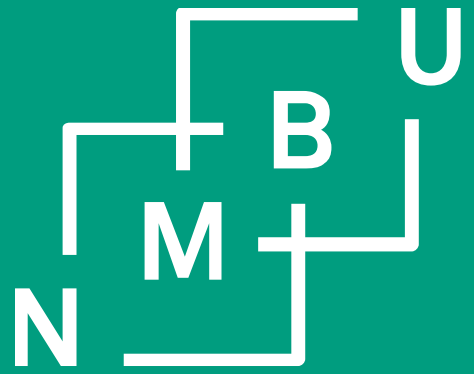
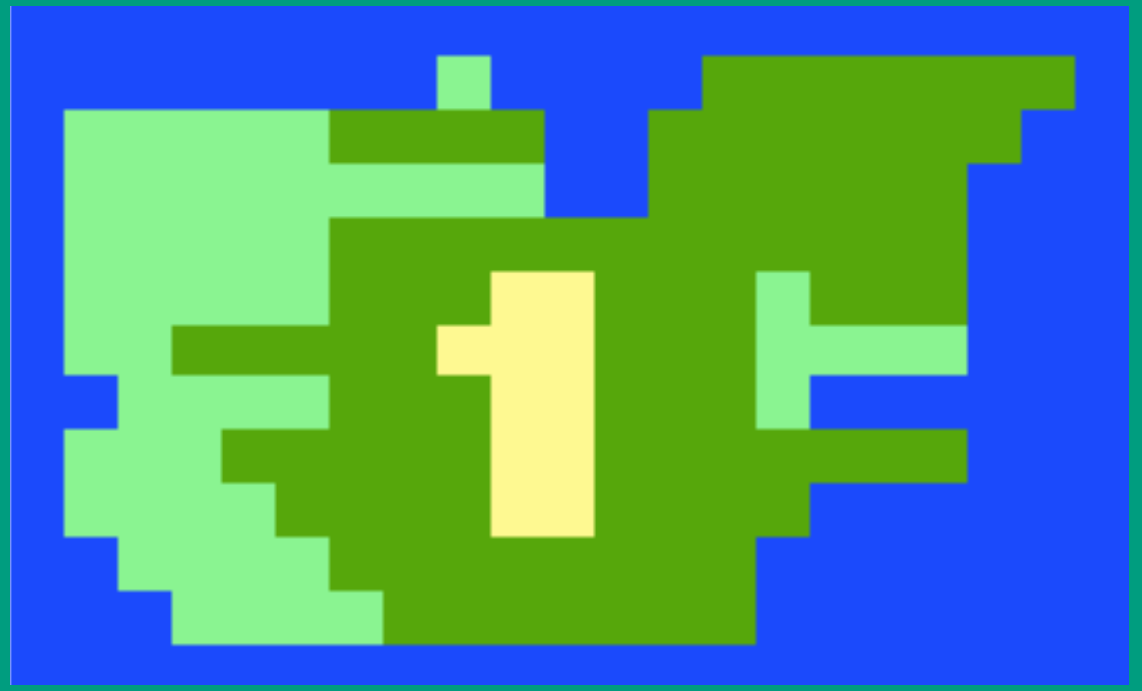


Norwegian University
of Life Sciences

Modelling the ecosystem of Rossumøya

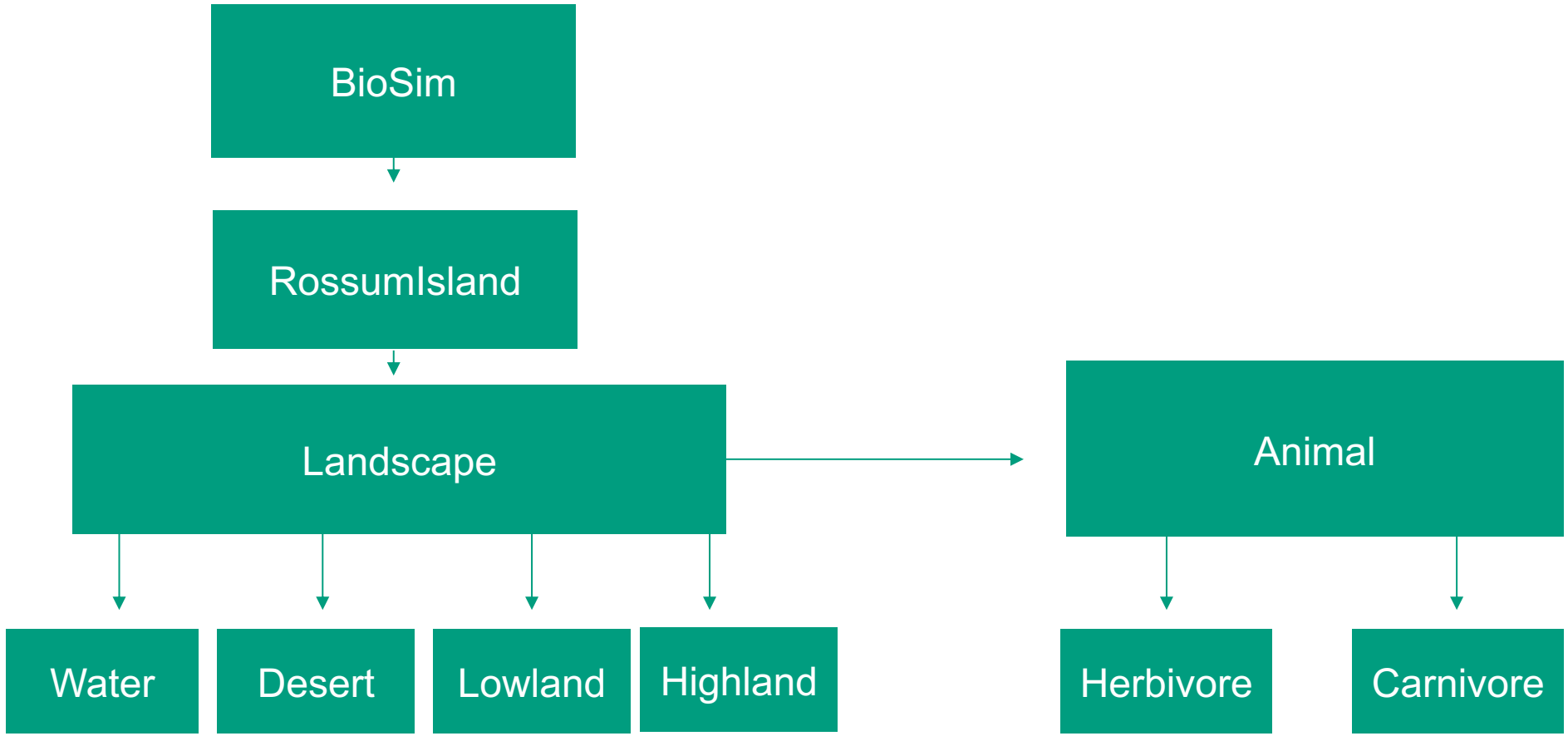
A stability study through population dynamics simulation

- Preparation and structure
- Reliability through testing
- Documentation with Sphinx
- Results

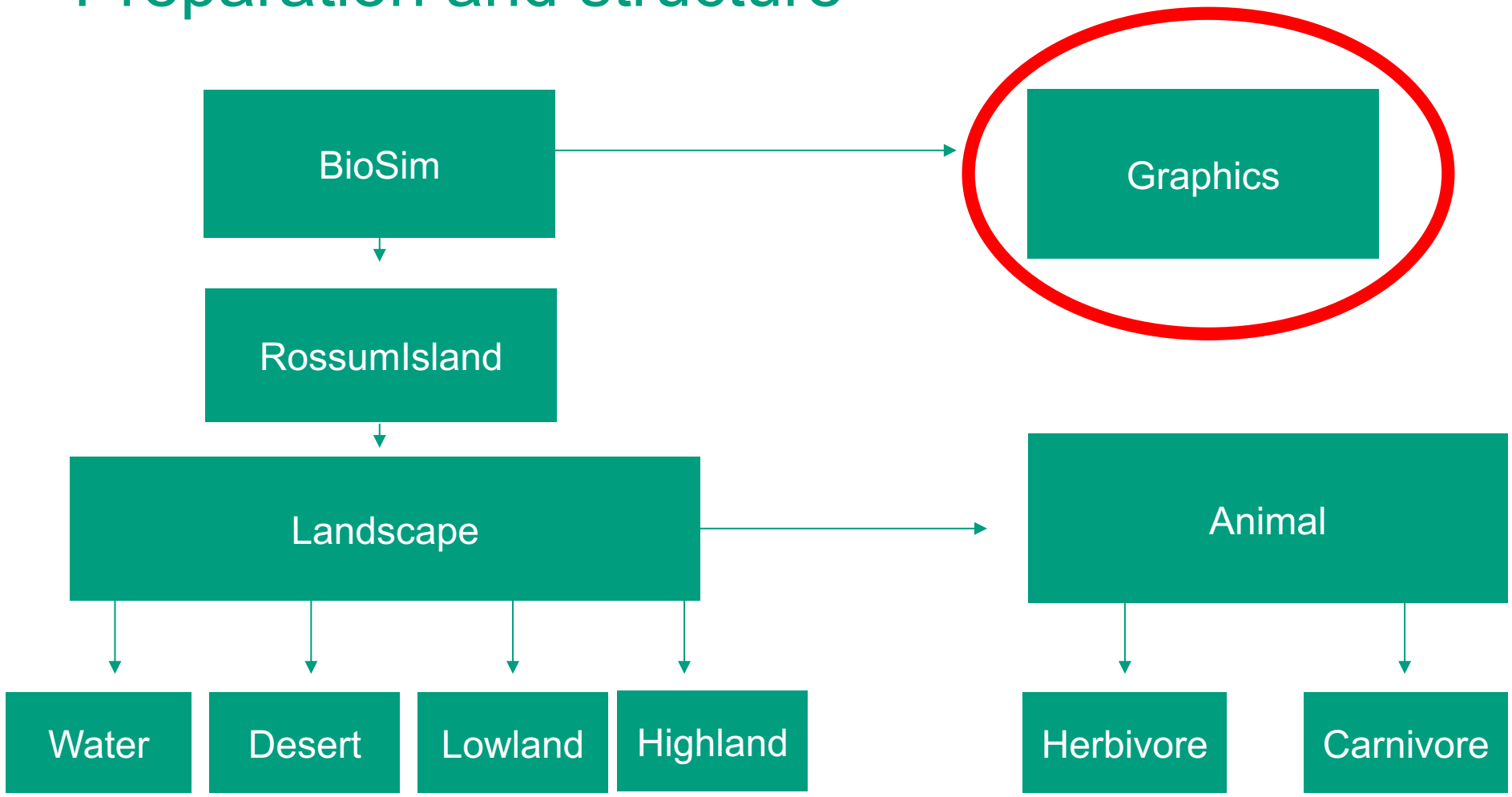


Sara Idris & Thorbjørn L Onsaker INF200 January 2021

Preparation and structure



Preparation and structure



Reliability through testing

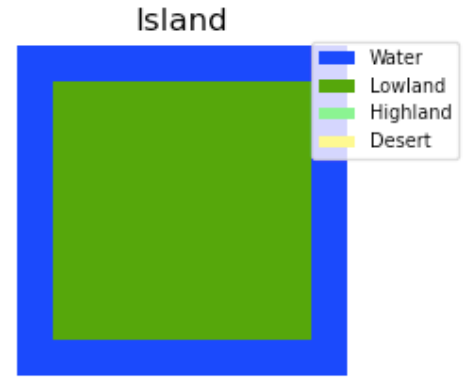
Testing with Pytest:

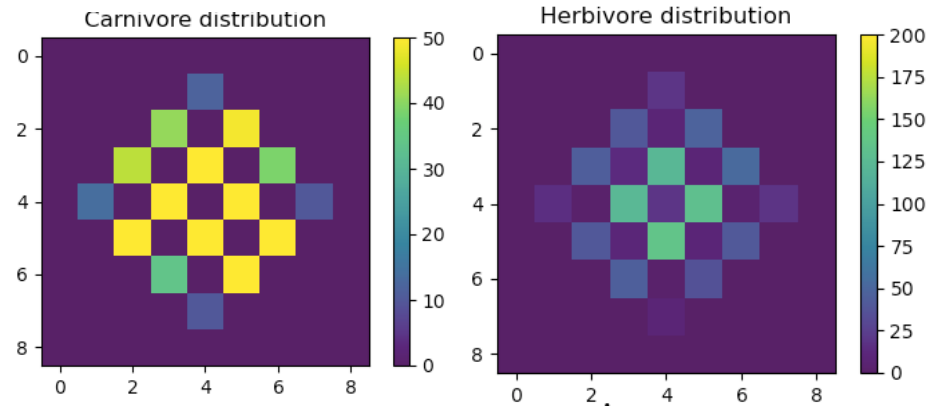
- Statistical
- Mocking
- Fixtures and parametrization

```python
def test_weight_normal_distributed():
    """
    Test that the weight given as default is normal distributed
    and passes the test with an alpha-value of 0.05.
    """

    alpha = 0.05
    herb_weights = [Herbivore().weight for _ in range(1000)]
    carn_weights = [Carnivore().weight for _ in range(1000)]
    result_herb = normaltest(herb_weights)
    result_carn = normaltest(carn_weights)
    assert alpha < result_herb[1]
    assert alpha < result_carn[1]
```

```python
def test_mate_method_and_offspring_weight(mocker):
    """Test that an offspring is of same class as parent and weight greater than 0."""

    mocker.patch('random.random', return_value=0)
    for _ in range(100):
        h = Herbivore(5, 50)
        c = Carnivore(5, 50)
        h_offspring = h.mate(100)
        c_offspring = c.mate(100)
        assert type(h_offspring) == Herbivore and h_offspring.weight > 0
        assert type(c_offspring) == Carnivore and c_offspring.weight > 0
```

Testing through visualization





Documentation with Sphinx

Math formulas for probability in the Python script:

```
if self.get_fitness() <= herb.get_fitness():
    break
elif (self.get_fitness() - herb.get_fitness()) < self.params['DeltaPhiMax']:
    p = ((self.get_fitness() - herb.get_fitness()) / self.params['DeltaPhiMax'])
else:
    p = 1
if random.random() < p:
    if wanted_food < herb.weight:
        self.weight += self.params['beta'] * wanted_food
        killed_herbs.append(herb)
```

More readable and clear through Sphinx

consumed_herbs(*herb_sorted*)

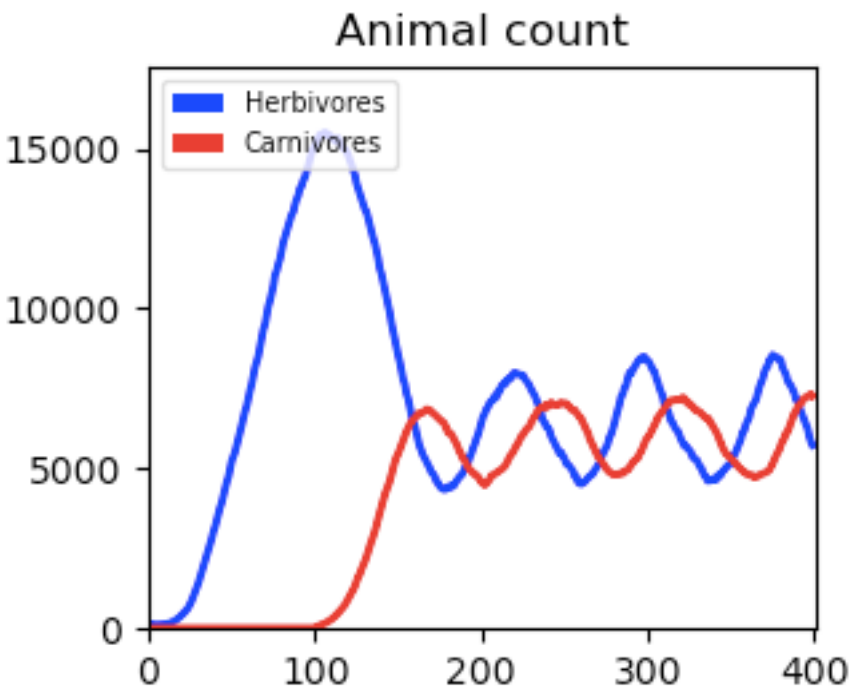
Decides whether a carnivore kills and eat a herbivore.

Carnivores will kill a herbivore with probability

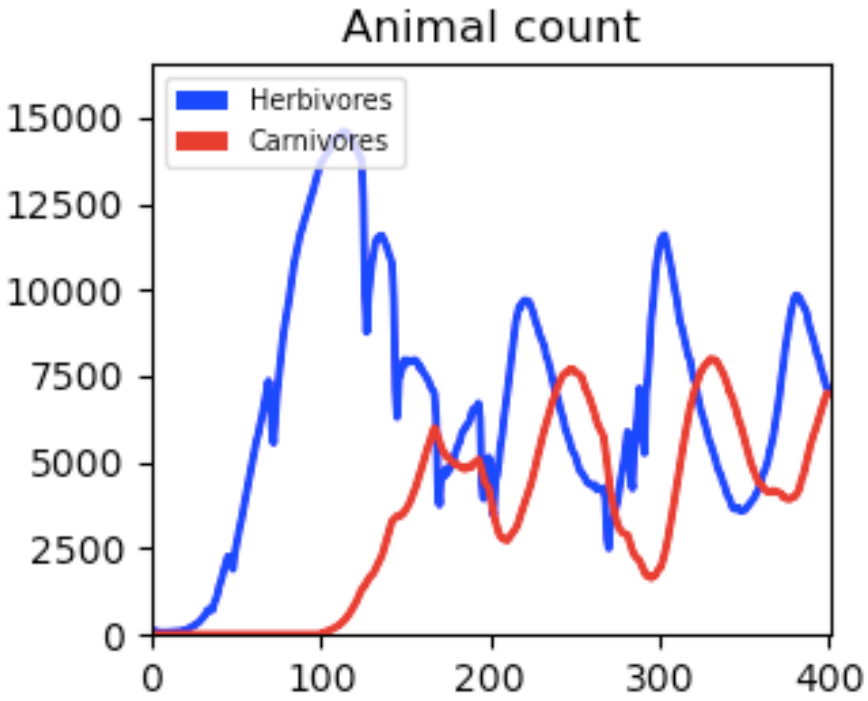
$$p = \begin{cases} 0 & if \Phi_{carn} \leq \Phi_{herb} \\ \frac{\Phi_{carn} - \Phi_{herb}}{\Delta\Phi_{max}} & if 0 < \Phi_{carn} - \Phi_{herb} < \Delta\Phi_{max} \\ 1 & otherwise. \end{cases}$$

Results

Result of simulation



With our additional feature:
Pythonvirus disease (Pyvid)



Sara Idris	said@nmbu.no	NMBU
Thorbjørn L Onsaker	thon@nmbu.no	NMBU