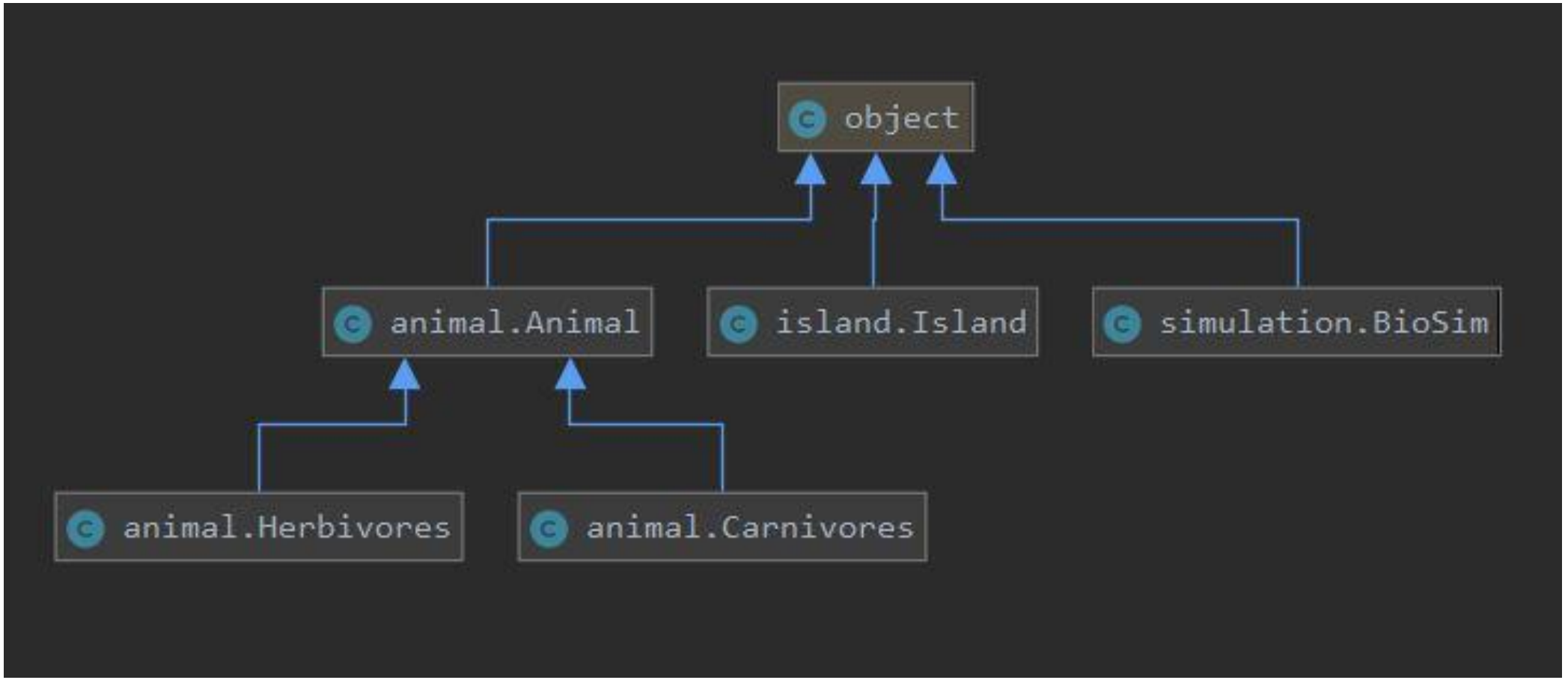
INF200 – Videregående programmering

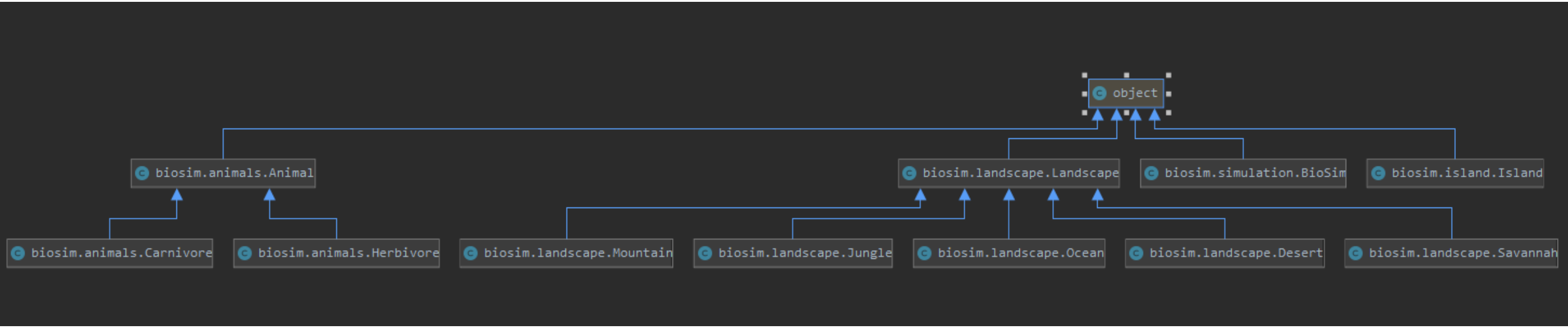
Markus O. Granheim & Rasmus Svebestad

28.jan.2020

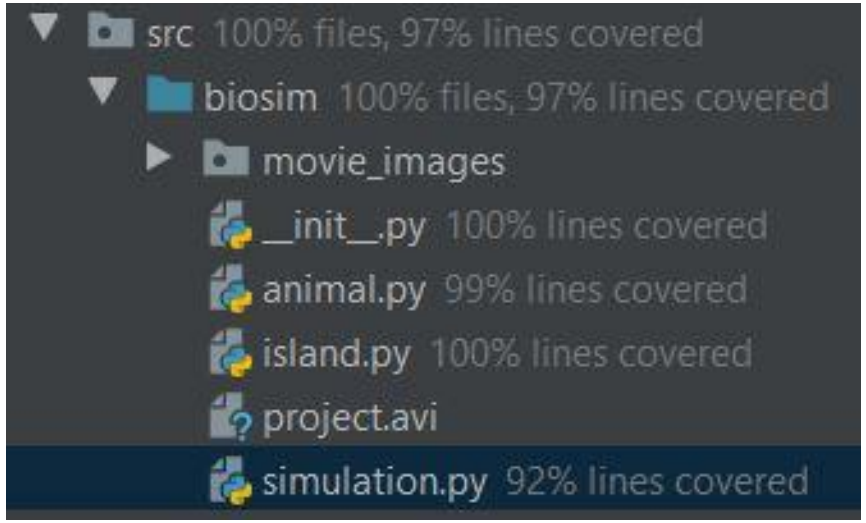
Klassediagramet vårt



Et eksempel på hvordan vi egentlig ønsket å organisere koden



Vi dekket 97% av koden med tester




```python
def make_movie(self):
    import glob

    img_array = []
    for filename in glob.glob(self._img_base + '*' + self._img_fmt):
        img = cv2.imread(filename)
        height, width, layers = img.shape
        size = (width, height)
        img_array.append(img)

    out = cv2.VideoWriter('project.avi', cv2.VideoWriter_fourcc(*'DIVX'), 6, size)

    for i in range(len(img_array)):
        out.write(img_array[i])
    out.release()
```

```python
def death(self, pos, animals):
    """
    Removes  from the animals list according to the formula for death
    :param pos: the position asked for
    :param animals: animal-dictionary
    """
    if pos in animals.keys():
        a = []
        for idx, animal in enumerate(animals[pos]):
            if animal['fitness'] == 0:
                a.append(idx)
            else:
                p = self.omega * (1 - animal['fitness'])
                if p >= random.random():
                    a.append(idx)
        for idx in sorted(a, reverse=True):
            del animals[pos][idx]
```

```python
def animal_distribution_for_plot(self):
    """Pandas DataFrame with animal count per species for each cell on island."""
    animal_list = np.zeros((self.island.rader, self.island.col, 2))
    for i in range(self.island.rader):
        for j in range(self.island.col):
            if (i, j) in self.island.herbs.keys():
                animal_list[i, j, 0] = len(self.island.herbs[(i, j)])
            if (i, j) in self.island.carns.keys():
                animal_list[i, j, 1] = len(self.island.carns[(i, j)])
    return animal_list
```

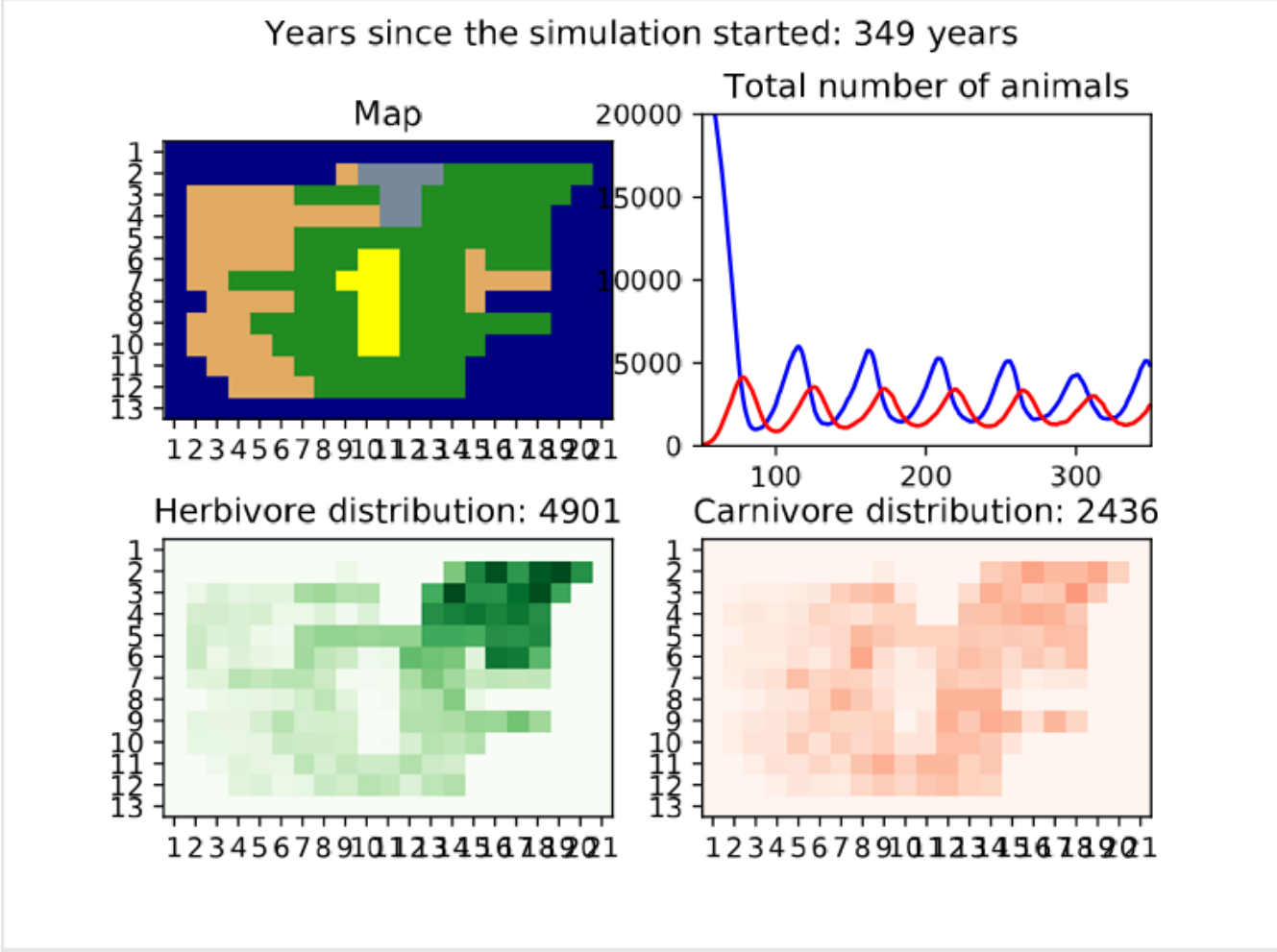
Vi klarte ikke å implentere mocker i testene våre dette måtte vi kompensere for ved andre hjelpemidler

```python
def test_carnivore_leaves_food(self):
    """Testing that the carnivores stops eating when it has received f amount of food"""
    animal_list = [{'loc': (1, 1), 'pop': [{'species': 'Carnivore', 'age': 5, 'weight': 53.3},
                                           {'species': 'Carnivore', 'age': 5, 'weight': 53.3},
                                           {'species': 'Herbivore', 'age': 500, 'weight': 13.3},
                                           {'species': 'Herbivore', 'age': 500, 'weight': 13.3},
                                           {'species': 'Herbivore', 'age': 500, 'weight': 13.3},
                                           {'species': 'Herbivore', 'age': 500, 'weight': 13.3}]}]
    a = Island('OOO\nOJO\nOOO')
    b = Carnivores()
    c = Herbivores()
    a.add_animals(animal_list)
    b.set_new_params({'beta': 1, 'F': 5, 'DeltaPhiMax': 0.0001})
    c.calculate_fitness((1, 1), a.herbs)
    b.calculate_fitness((1, 1), a.carns)
    b.carnivores_eat((1, 1), a, a.carns)
    assert len(a.herbs[(1, 1)]) == 2
    assert a.carns[(1, 1)][0]['weight'] == 58.3
    assert a.carns[(1, 1)][1]['weight'] == 58.3
```

```python
def test_breeding_carnivores(self):
    """Tests if carnivores can give birth,
     and that the children get added to the same tile as their parents"""
    added_list = [{'loc': (3, 3), 'pop': [{'species': 'Carnivore', 'age': 20, 'weight': 17.3},
                                          {'species': 'Carnivore', 'age': 30, 'weight': 19.3},
                                          {'species': 'Carnivore', 'age': 10, 'weight': 107.3},
                                          {'species': 'Carnivore', 'age': 1, 'weight': 20.3}]}]
    a = self.jungle_island_animals()[0]
    b = Carnivores()
    a.set_food((3, 3))
    a.add_animals(added_list)
    len_list1 = len(a.carns[(3, 3)])
    b.calculate_fitness((3, 3), a.carns)
    for _ in range(10):
        b.breeding((3, 3), a, a.carns)
    len_list2 = len(a.carns[(3, 3)])
    assert len_list2 > len_list1
```

Bilde av en 350 år lang simulering

- Aksene burde vært penere
- Det siste året blir ikke vist pga. en feil
- Kunne indikert tydeligere hvilken graf som følger hvilken type dyr, dette kunne vi gjort med legend



Oversikt over tidsbruken til simuleringen

Name	Call Count	Time (ms)	Own Time (ms)
<built-in method builtins.input>	1	29829 29,9 %	29829 29,9 %
calculate_fitness	573300	20345 20,4 %	14640 14,7 %
carnivores_eat	81900	8815 8,9 %	7154 7,2 %
migration_calculations	300	10127 10,2 %	7123 7,2 %
fetch_naturetype	4842433	3858 3,9 %	3858 3,9 %
<built-in method math.exp>	15448974	3845 3,9 %	3845 3,9 %
breeding	163800	7949 8,0 %	3451 3,5 %
migration_calculations	300	6353 6,4 %	3405 3,4 %
<method 'update' of 'dict' objects>	7060068	2314 2,3 %	2314 2,3 %
add_animals	58058	3730 3,7 %	2192 2,2 %
death	163800	2503 2,5 %	2026 2,0 %
<method 'normal' of 'mtrand.RandomState' objects	435350	1752 1,8 %	1752 1,8 %
<method 'random' of '_random.Random' objects>	11483890	1712 1,7 %	1712 1,7 %
<built-in method builtins.sorted>	625829	2192 2,2 %	1640 1,6 %
tot_weight_herbivores	524108	1604 1,6 %	1532 1,5 %
loss_of_weight	163800	1540 1,5 %	1502 1,5 %

Sammendrag

- Vi har en kode som visualiserer en simulering
- Vi mener koden er til å stole på, på grunn av den gode dekningen fra testene
- Koden går tregt og kunne vært mer oversiktelig