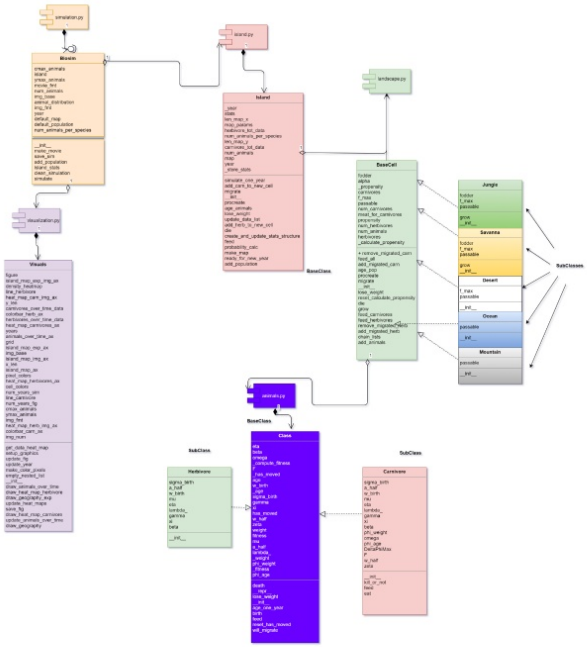
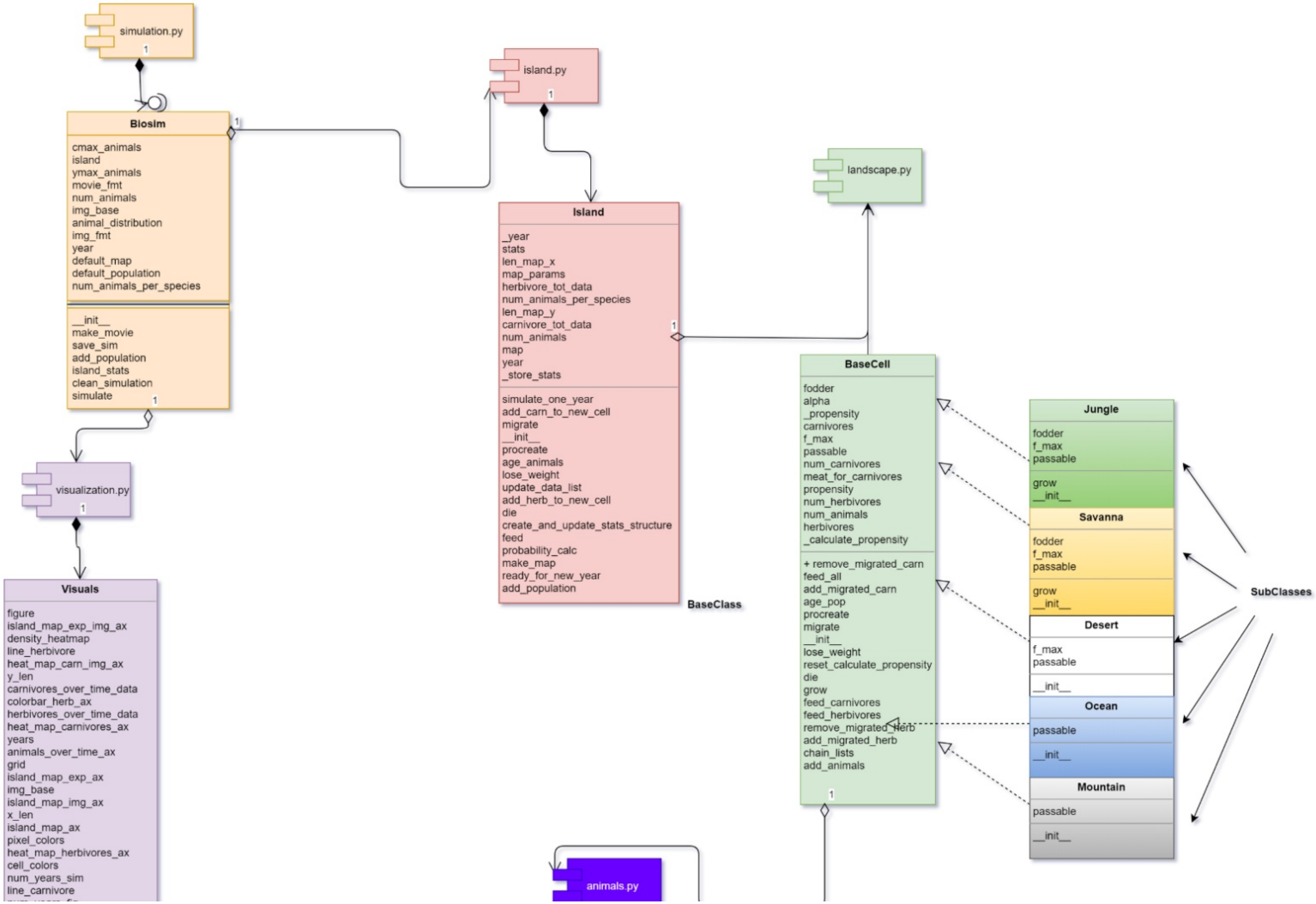
Population Dynamics Group 22

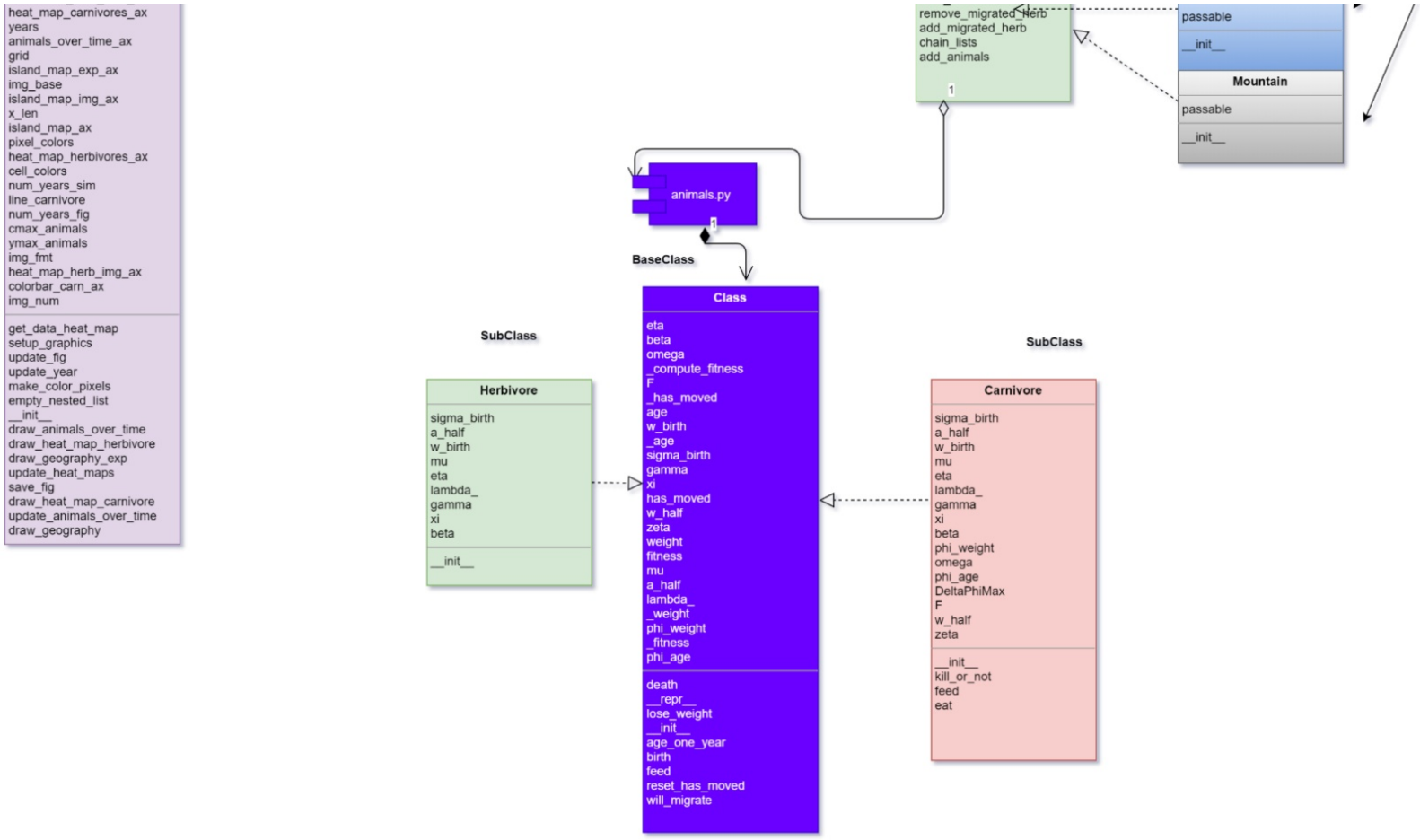
Jon-Mikkel Korsvik & Petter Hørtvedt



The project

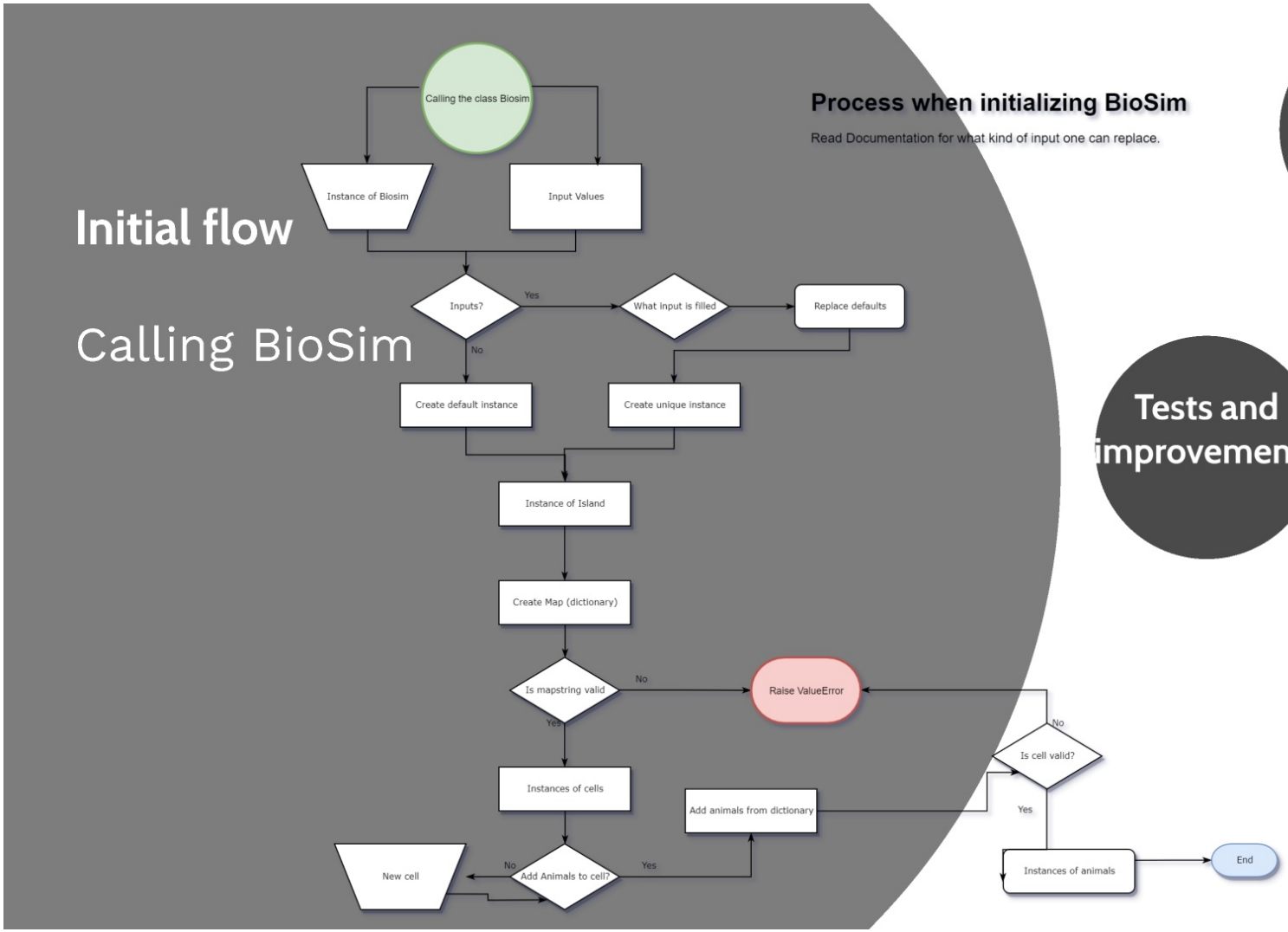

Ajit b Kumar


simulation.py
island.py
BioSim
cmax_animals
island
ymax_animals
movie_fmt
num_animals
img_base
animal_distribution
img_fmt
year
default_map
default_population
num_animals_per_species
__init__
make_movie
save_sim
add_population
island_stats
clean_simulation
simulate
landscape.py
Island
_year
stats
len_map_x
map_params
herbivore_tot_data
num_animals_per_species
len_map_y
carnivore_tot_data
num_animals
map
year
_store_stats
simulate_one_year
add_carn_to_new_cell
migrate
__init__
procreate
age_animals
lose_weight
update_data_list
add_herb_to_new_cell
die
create_and_update_stats_structure
feed
probability_calc
make_map
ready_for_new_year
add_population
BaseClass
BaseCell
fodder
alpha
_propensity
carnivores
f_max
passable
num_carnivores
meat_for_carnivores
propensity
num_herbivores
num_animals
herbivores
_calculate_propensity
+ remove_migrated_carn
feed_all
add_migrated_carn
age_pop
procreate
migrate
__init__
lose_weight
reset_calculate_propensity
die
grow
feed_carnivores
feed_herbivores
remove_migrated_herb
add_migrated_herb
chain_lists
add_animals
Jungle
fodder
f_max
passable
grow
__init__
Savanna
fodder
f_max
passable
grow
__init__
Desert
f_max
passable
__init__
Ocean
passable
__init__
Mountain
passable
__init__
SubClasses
visualization.py
Visuals
figure
island_map_exp_img_ax
density_heatmap
line_herbivore
heat_map_carn_img_ax
y_len
carnivores_over_time_data
colorbar_herb_ax
herbivores_over_time_data
heat_map_carnivores_ax
years
animals_over_time_ax
grid
island_map_exp_ax
img_base
island_map_img_ax
x_len
island_map_ax
pixel_colors
heat_map_herbivores_ax
cell_colors
num_years_sim
line_carnivore
animals.py


heat_map_carnivores_ax
years
animals_over_time_ax
grid
island_map_exp_ax
img_base
island_map_img_ax
x_len
island_map_ax
pixel_colors
heat_map_herbivores_ax
cell_colors
num_years_sim
line_carnivore
num_years_fig
cmax_animals
ymax_animals
img_fmt
heat_map_herb_img_ax
colorbar_carn_ax
img_num
get_data_heat_map
setup_graphics
update_fig
update_year
make_color_pixels
empty_nested_list
__init__
draw_animals_over_time
draw_heat_map_herbivore
draw_geography_exp
update_heat_maps
save_fig
draw_heat_map_carnivore
update_animals_over_time
draw_geography
remove_migrated_herb
add_migrated_herb
chain_lists
add_animals
1
passable
__init__
Mountain
passable
__init__
animals.py
1
BaseClass
SubClass
Herbivore
sigma_birth
a_half
w_birth
mu
eta
lambda_
gamma
xi
beta
__init__
Class
eta
beta
omega
_compute_fitness
F
_has_moved
age
w_birth
_age
sigma_birth
gamma
xi
has_moved
w_half
zeta
weight
fitness
mu
a_half
lambda_
_weight
phi_weight
_fitness
phi_age
death
__repr__
lose_weight
__init__
age_one_year
birth
feed
reset_has_moved
will_migrate
SubClass
Carnivore
sigma_birth
a_half
w_birth
mu
eta
lambda_
gamma
xi
beta
phi_weight
omega
phi_age
DeltaPhiMax
F
w_half
zeta
__init__
kill_or_not
feed
eat

Initial flow

Calling BioSim



Flow charts

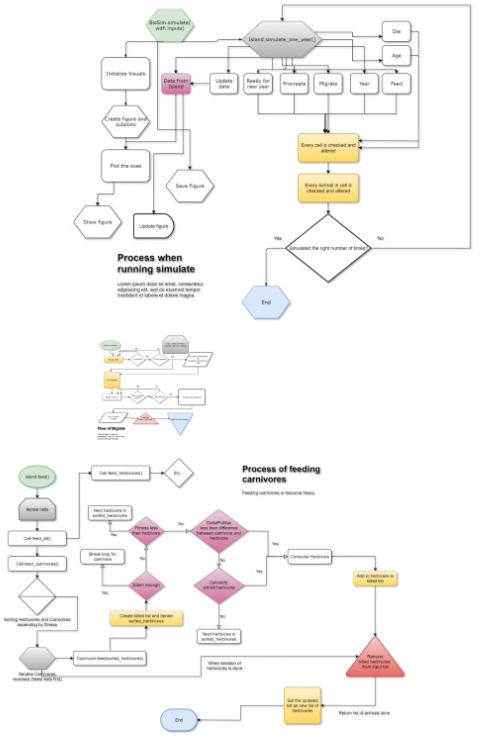
Visuals

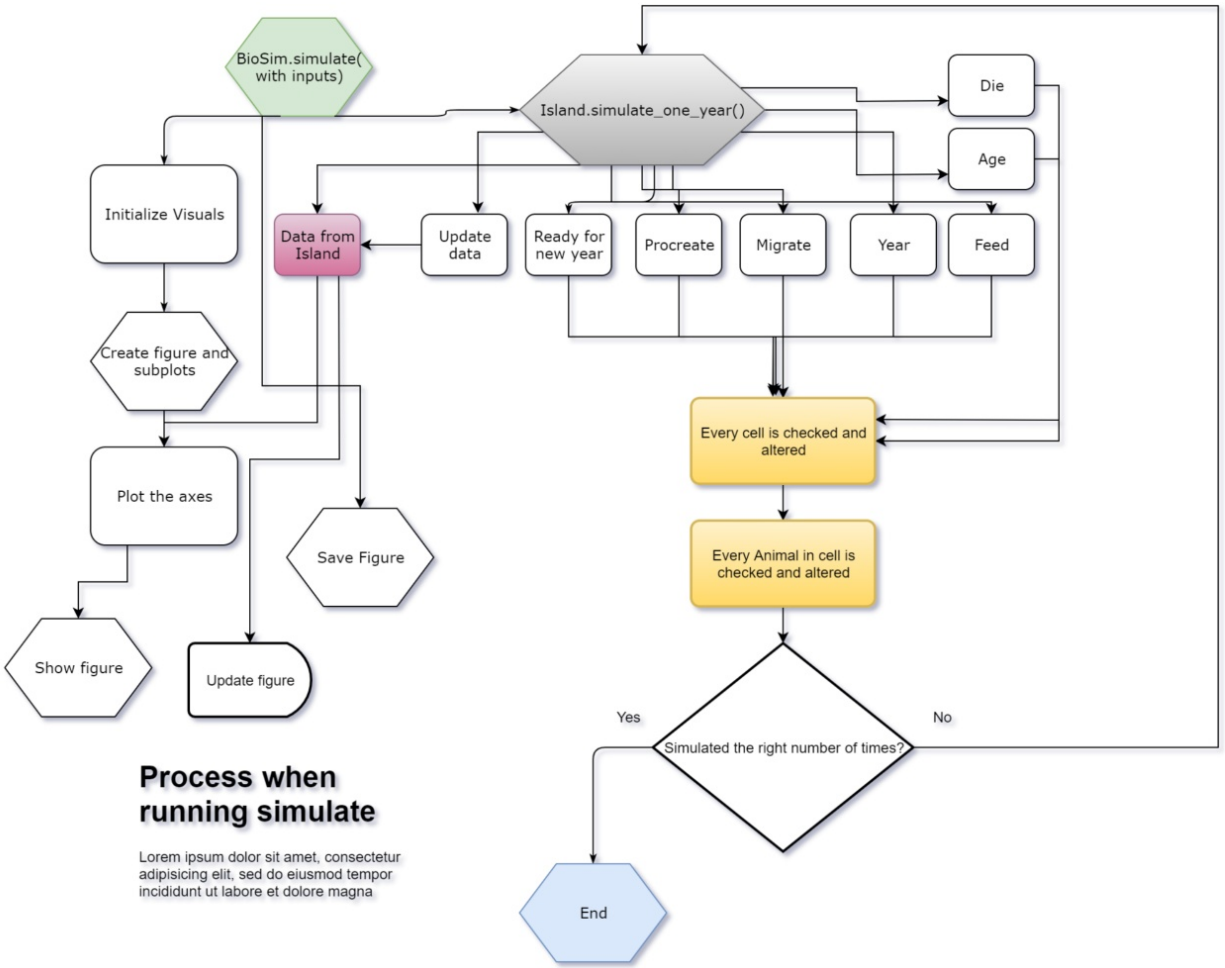
Tests and improvements

Extra Features

Flow of some methods

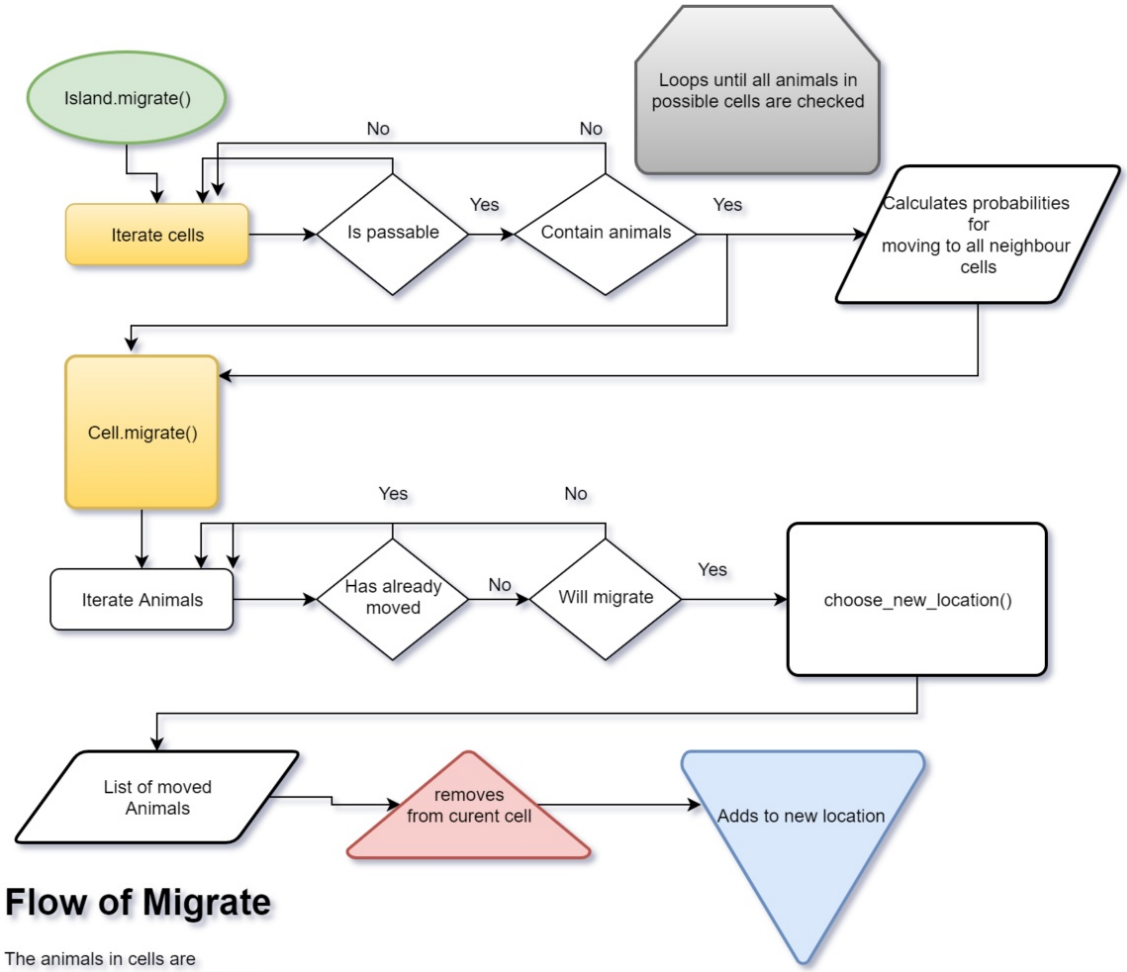
Primary and difficult methods





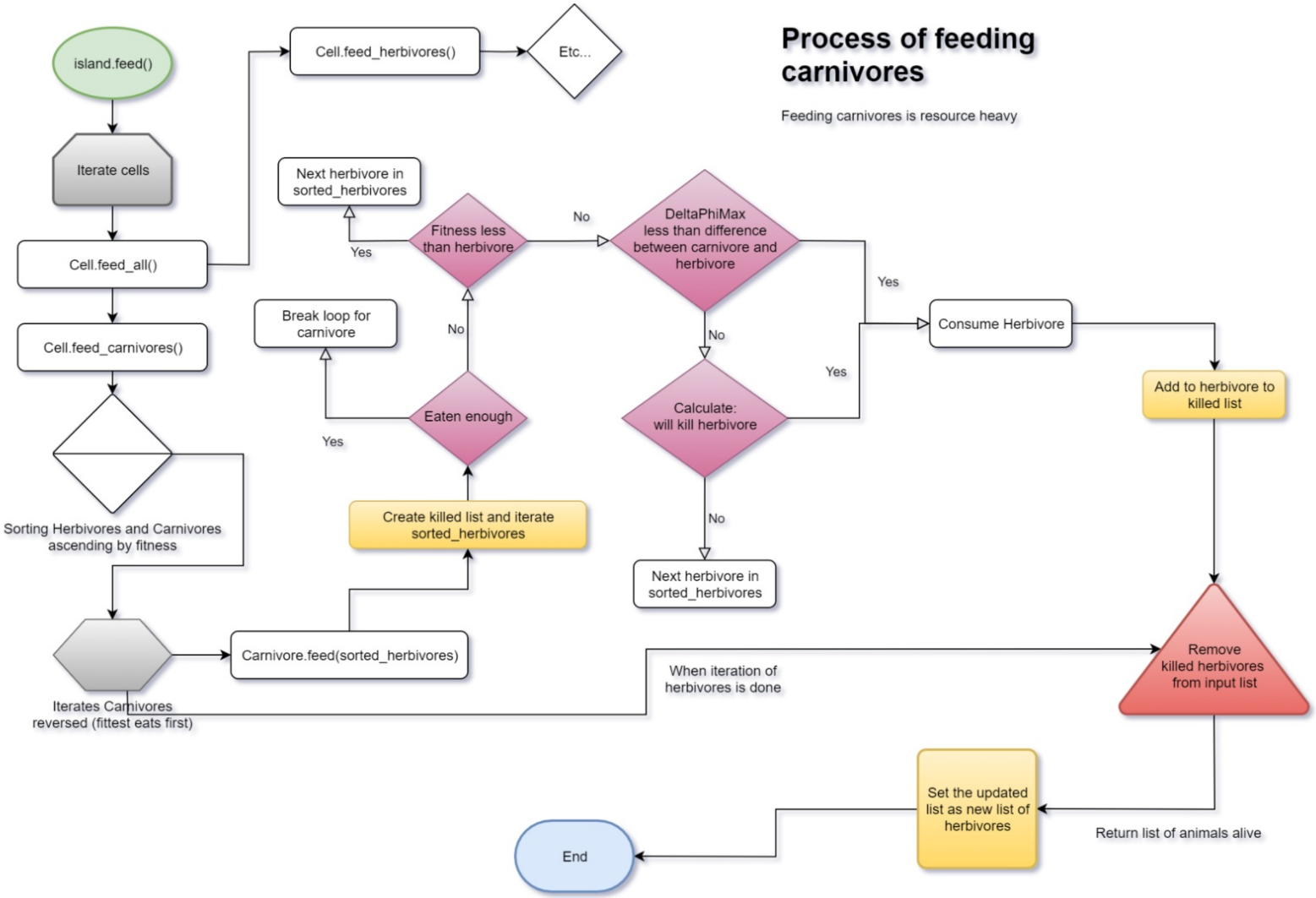
Process when running simulate

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna



Flow of Migrate

The animals in cells are separeted in two list, and so are the list of moved animals


Process of feeding carnivores
Feeding carnivores is resource heavy
island.feed()
Iterate cells
Cell.feed_all()
Cell.feed_herbivores()
Etc...
Cell.feed_carnivores()
Sorting Herbivores and Carnivores ascending by fitness
Iterates Carnivores reversed (fittest eats first)
Carnivore.feed(sorted_herbivores)
Next herbivore in sorted_herbivores
Fitness less than herbivore
Yes
No
Break loop for carnivore
Eaten enough
Yes
No
Create killed list and iterate sorted_herbivores
DeltaPhiMax less than difference between carnivore and herbivore
No
Calculate: will kill herbivore
Yes
No
Next herbivore in sorted_herbivores
Yes
Consume Herbivore
Add to herbivore to killed list
When iteration of herbivores is done
Remove killed herbivores from input list
Return list of animals alive
Set the updated list as new list of herbivores
End

Test Coverage

Coverage report: 92%

Module ↓	statements	missing	excluded	coverage
src\biosim__init__.py	2	0	0	100%
src\biosim\animals.py	278	20	0	93%
src\biosim\island.py	163	12	0	93%
src\biosim\landscape.py	216	6	0	97%
src\biosim\simulation.py	134	33	0	75%
src\biosim\visualization.py	161	7	0	96%
Total	**954**	**78**	**0**	**92%**

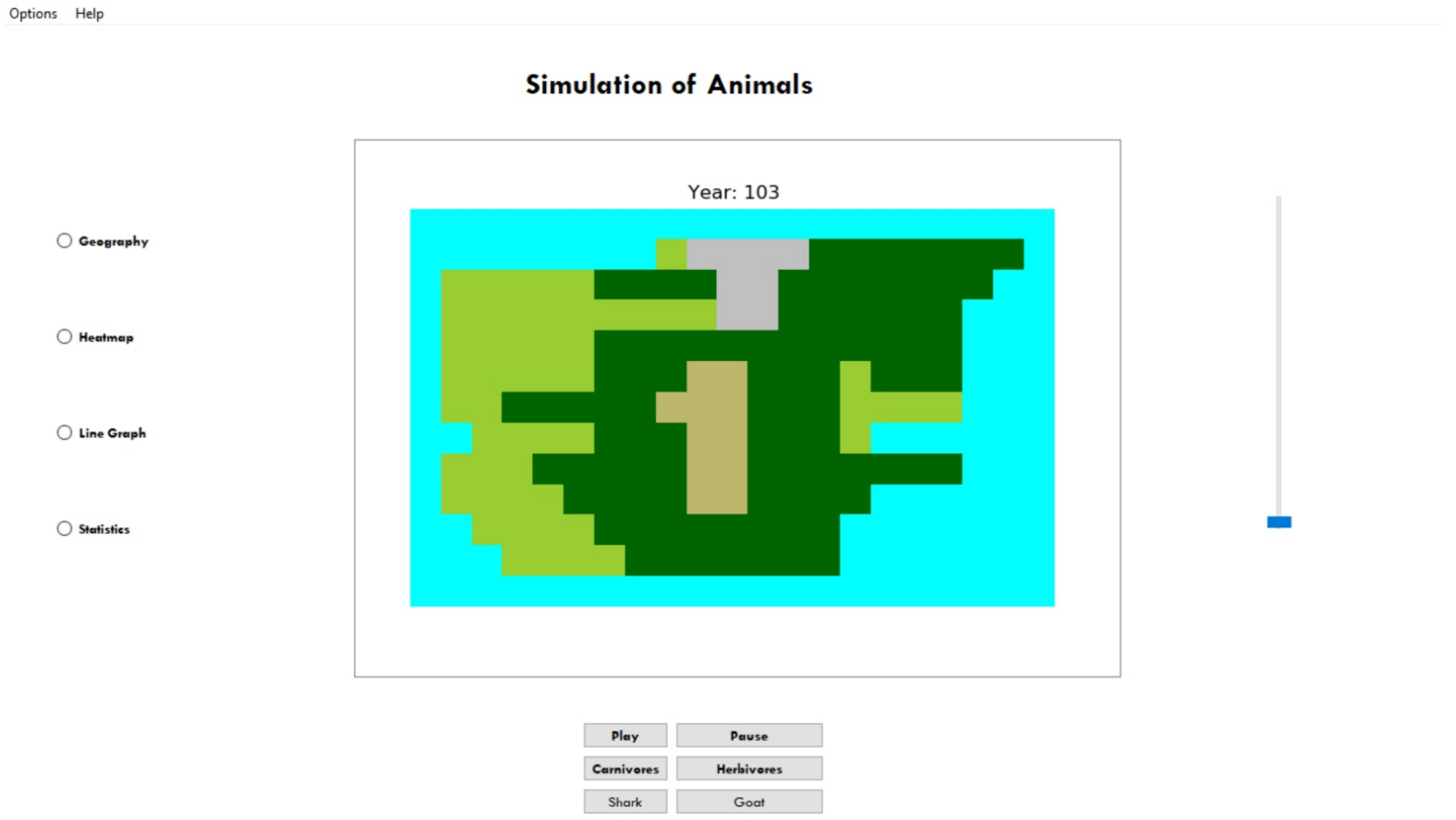
coverage.py v4.5.3, created at 2020-01-22 12:12

Improvements

1: Add new animals
2: Create animal dictionary
3: Create GUI
4: Optimize feeding
5: Continue optimizing migration
6: JIT - compilate more functions/methods
7: Make instance of Visuals in BioSIm
8: Set x-axis of line graph specifically
9: More elif and else staments (optimizing)
10: Use seaborn for pretty plots

Improvements

1: Add new animals
2: Create animal dictionary
3: Create GUI
4: Optimize feeding
5: Continue optimizing migration
6: JIT - compilate more functions/methods
7: Make instance of Visuals in BioSIm
8: Set x-axis of line graph specifically
9: More elif and else staments (optimizing)
10: Use seaborn for pretty plots


Options
Help
Simulation of Animals
Year: 103
Geography
Heatmap
Line Graph
Statistics
Play
Pause
Carnivores
Herbivores
Shark
Goat

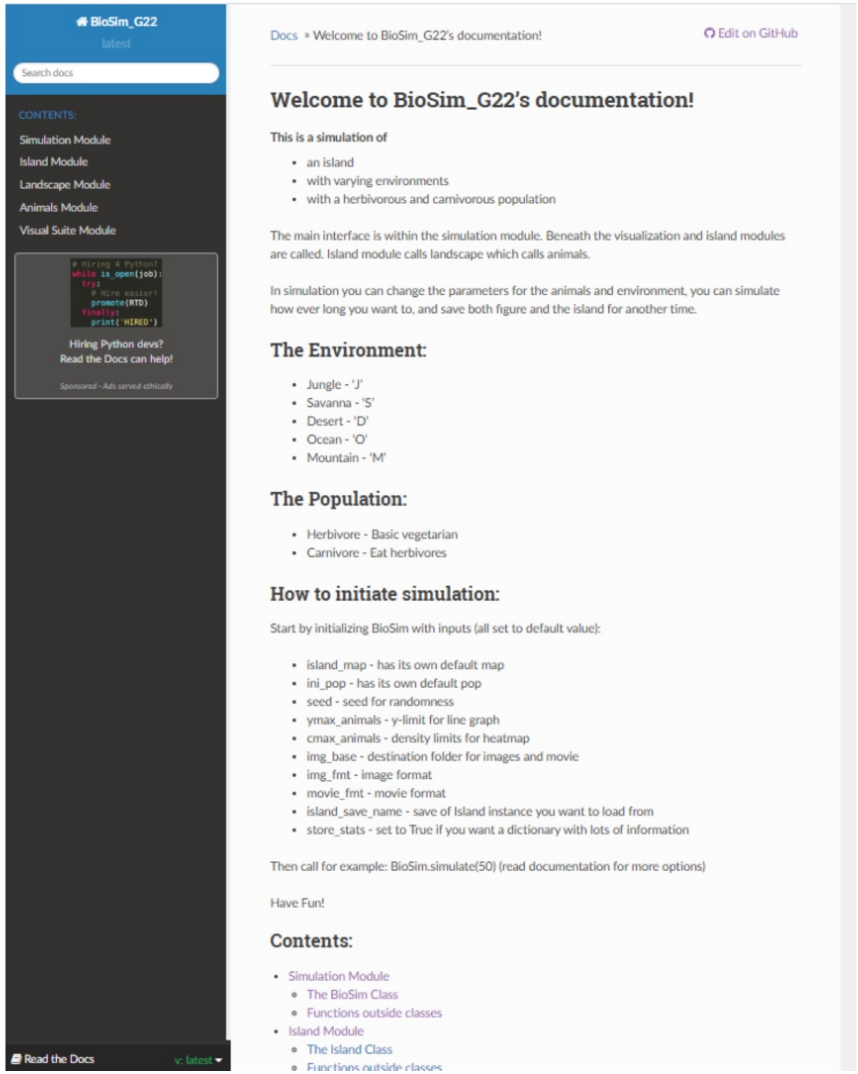
Documentation

README.md

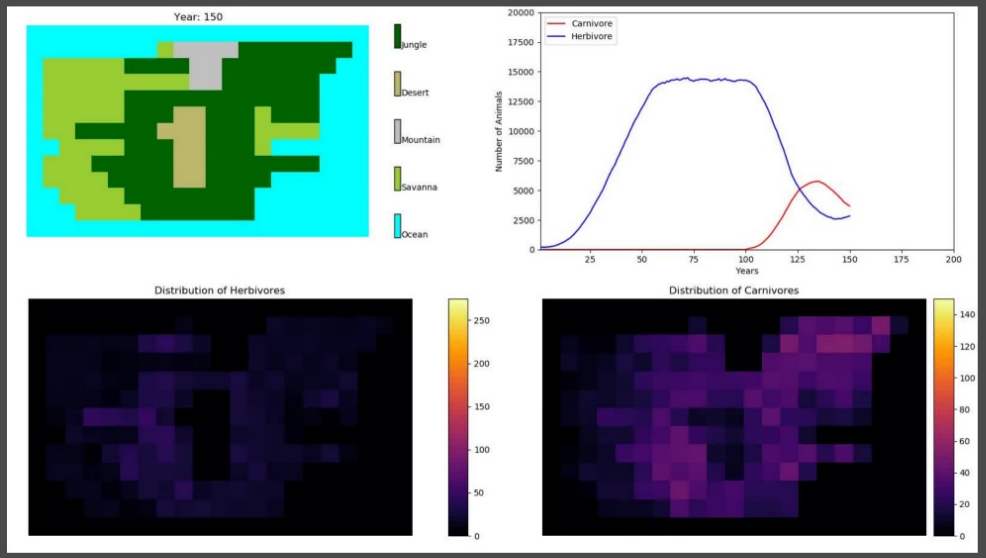
BioSim_G22_jkorsvik_petterho

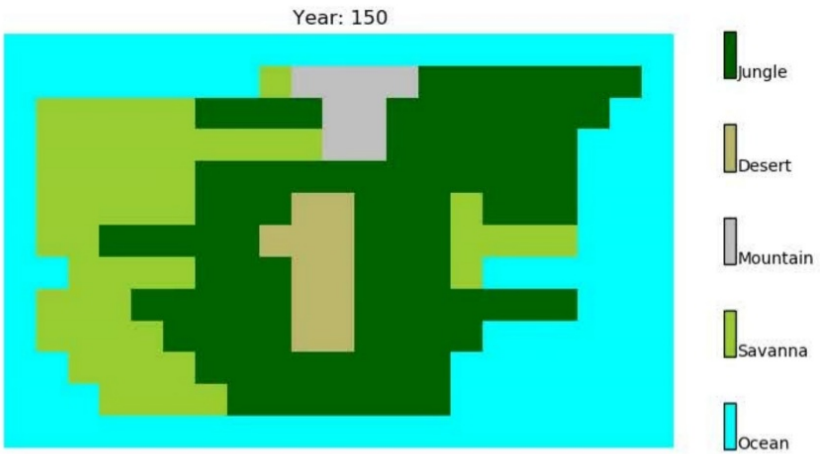
Documentation link

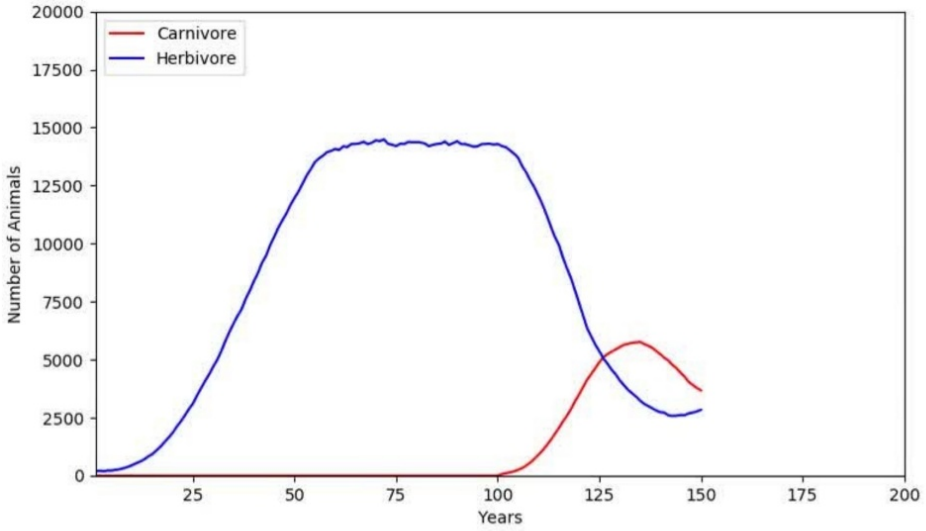
- Latest version of Documentation


BioSim_G22
latest
Search docs
CONTENTS:
Simulation Module
Island Module
Landscape Module
Animals Module
Visual Suite Module
Hiring Python devs?
Read the Docs can help!
Sponsored - Ads served ethically
Read the Docs
v: latest
Docs » Welcome to BioSim_G22's documentation!
Edit on GitHub
Welcome to BioSim_G22's documentation!
This is a simulation of
an island
with varying environments
with a herbivorous and carnivorous population
The main interface is within the simulation module. Beneath the visualization and island modules are called. Island module calls landscape which calls animals.
In simulation you can change the parameters for the animals and environment, you can simulate how ever long you want to, and save both figure and the island for another time.
The Environment:
Jungle - 'J'
Savanna - 'S'
Desert - 'D'
Ocean - 'O'
Mountain - 'M'
The Population:
Herbivore - Basic vegetarian
Carnivore - Eat herbivores
How to initiate simulation:
Start by initializing BioSim with inputs (all set to default value):
island_map - has its own default map
ini_pop - has its own default pop
seed - seed for randomness
ymax_animals - y-limit for line graph
cmax_animals - density limits for heatmap
img_base - destination folder for images and movie
img_fmt - image format
movie_fmt - movie format
island_save_name - save of Island instance you want to load from
store_stats - set to True if you want a dictionary with lots of information
Then call for example: BioSim.simulate(50) (read documentation for more options)
Have Fun!
Contents:
Simulation Module
The BioSim Class
Functions outside classes
Island Module
The Island Class

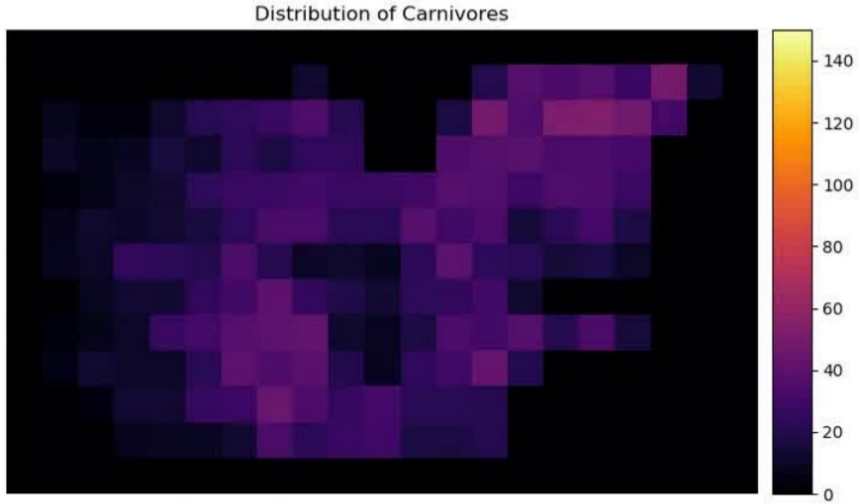
The Visuals


Year: 150
Jungle
Desert
Mountain
Savanna
Ocean
Carnivore
Herbivore
Number of Animals
Years
Distribution of Herbivores
Distribution of Carnivores


Year: 150
Jungle
Desert
Mountain
Savanna
Ocean


Carnivore
Herbivore
Number of Animals
Years
0
2500
5000
7500
10000
12500
15000
17500
20000
25
50
75
100
125
150
175
200


Distribution of Herbivores
0
50
100
150
200
250


Distribution of Carnivores
0
20
40
60
80
100
120
140

Extra Features

01 Just-in-time fitness calculation

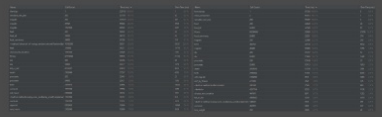
02 Saving instance of island and data

03 Storing data of dead, born and alive animals

```
def create_and_update_static_structure(self):
    """Creates data structure for stats"""
    all_herbs = self.num_animals_per_species['Herbivore']
    all_carns = self.num_animals_per_species['Carnivore']

    for pos in self.map.keys():
        self.stats[self.year] = {'Herbivore': {'Death': {pos: 0},
                                               'Birth': {pos: 0},
                                               'Alive': all_herbs},
                                 'Carnivore': {'Death': {pos: 0},
                                               'Birth': {pos: 0},
                                               'Alive': all_carns}
                                 }
```

04 Optimized simulation



01 Just-in-time fitness calculation

```python
@jit
def fitness_calculation(
        phi_age, age, a_half,
        phi_weight, weight, w_half
                        ):
    r"""
    Calculates fitness by sigmoid multiplication

    .. math::
        \Phi =\left\{\begin{matrix}0
         &  , w\leq 0 & \\
         q^-_{weight}*q^+_{age}&, else  &
        \end{matrix}\right.

    .. math::
        q^\pm(x, x_{\frac{1}{2}},\phi)=\frac{1}{1+e^{\pm\phi(x-x_\frac{1}{2

    Parameters
    ----------
    phi_age : float
    age : int
    a_half : float
    phi_weight : float
    weight : float
    w_half  : float

    Returns
    -------
    float
        Value between 0 and 1 representing fitness

    """
    pos_q_age = phi_age * (age - a_half)
    neg_q_weight = - (phi_weight * (weight - w_half))

    return 1/(1 + math.exp(pos_q_age)) * 1/(1 + math.exp(neg_q_weight))

```

02 Saving instance of island and data

```python
def save_sim(simulation, name):
    """
    Save a state of Simulation

    Parameters
    ----------
    simulation : object
        Instance of Simulation
    name : str
        Save name

    Returns
    -------

    """
    with open(name + '.pkl', 'wb') as f:
        pickle.dump(simulation, f, pickle.HIGHEST_PROTOCOL)


def load_sim(name):
    """
    Loads state of Simulation

    Parameters
    ----------
    name : file
        Name of file

    Returns
    -------
    Loaded file of Simulation

    """
    with open(name + '.pkl', 'rb') as f:
        return pickle.load(f)
```

03 Storing data of dead, born and alive animals

```
127     def create_and_update_stats_structure(self):
128         """Creates data structure for stats"""
129         all_herbs = self.num_animals_per_species['Herbivore']
130         all_carns = self.num_animals_per_species['Carnivore']
131 
132         for pos in self.map.keys():
133             self.stats[self.year] = {'Herbivore': {'death': {pos: []},
134                                                    'birth': {pos: []},
135                                                    'alive': all_herbs},
136                                      'Carnivore': {'death': {pos: []},
137                                                    'birth': {pos: []},
138                                                    'alive': all_carns}
139                                      }
140 
```

04 Optimized simulation

Name	Call Count	Time (ms)	Own Time (ms)
island.py	1	228102 100,0 %	1 0,0 %
simulate_one_year	200	227387 99,7 %	42 0,0 %
migrate	200	101819 44,6 %	567 0,2 %
migrate	28742	96648 42,4 %	3938 1,7 %
migrate	1791558	69594 30,5 %	2307 1,0 %
feed	200	64603 28,3 %	24 0,0 %
feed_all	54600	64579 28,3 %	43 0,0 %
feed_carnivores	54600	61388 26,9 %	310 0,1 %
<method 'binomial' of 'numpy.random.mtrand.RandomStat	16185593	58351 25,6 %	58351 25,6 %
feed	319486	54231 23,8 %	14770 6,5 %
choose_new_location	1791558	52806 23,2 %	7587 3,3 %
fitness	62714345	39878 17,5 %	21134 9,3 %
die	200	32366 14,2 %	49 0,0 %
die	54600	32317 14,2 %	1479 0,6 %
kill_or_not	6023247	30819 13,5 %	8033 3,5 %
death	2183048	27267 12,0 %	4393 1,9 %
procreate	200	22445 9,8 %	23 0,0 %
procreate	54600	22421 9,8 %	1385 0,6 %
birth	2181673	20894 9,2 %	10454 4,6 %
cumsum	1791558	19994 8,8 %	1490 0,7 %
will_move	2183048	18722 8,2 %	1951 0,9 %
<built-in method numpy.core._multiarray_umath.implemen	1791560	18167 8,0 %	914 0,4 %
cumsum	1791558	17252 7,6 %	1474 0,6 %
sigmoid	8508430	16966 7,4 %	16966 7,4 %
rand_move	2183048	15896 7,0 %	3056 1,3 %

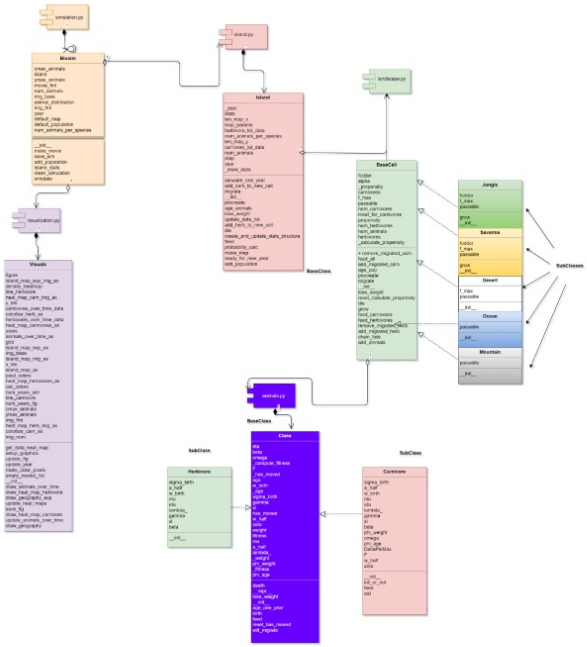
Name	Call Count	Time (ms)	Own Time (ms)
bioscript.py	1	97326 100,0 %	0 0,0 %
clean_simulation	2	95515 98,1 %	1 0,0 %
simulate_one_year	200	95397 98,0 %	3 0,0 %
feed	200	32628 33,5 %	38 0,0 %
feed_all	92400	32590 33,5 %	65 0,1 %
fitness	60290666	29680 30,5 %	25786 26,5 %
feed_carnivores	92400	28251 29,0 %	374 0,4 %
migrate	200	22496 23,1 %	345 0,4 %
feed	466702	22130 22,7 %	9634 9,9 %
migrate	40443	18003 18,5 %	1785 1,8 %
die	200	15737 16,2 %	53 0,1 %
die	92400	15683 16,1 %	1496 1,5 %
procreate	200	15149 15,6 %	35 0,0 %
procreate	92400	15113 15,5 %	1619 1,7 %
death	2933916	13758 14,1 %	3081 3,2 %
birth	2930604	13331 13,7 %	8285 8,5 %
will_migrate	3158289	9131 9,4 %	3063 3,1 %
sort_by_fitness	277200	7621 7,8 %	105 0,1 %
<built-in method builtins.sorted>	280134	7536 7,7 %	984 1,0 %
<lambda>	4357114	6533 6,7 %	1196 1,2 %
choose_new_location	464191	6511 6,7 %	1352 1,4 %
kill_or_not	4060682	5250 5,4 %	3470 3,6 %
<built-in method numpy.core._multiarray_umath.implemen	548022	5161 5,3 %	291 0,3 %
cumsum	464191	4641 4,8 %	337 0,3 %
lose_weight	200	4606 4,7 %	29 0,0 %

Name	Call Count	Time (ms) ▼	Own Time (ms)
island.py	1	228102 100,0 %	1 0,0 %
simulate_one_year	200	227387 99,7 %	42 0,0 %
migrate	200	101819 44,6 %	567 0,2 %
migrate	28742	96648 42,4 %	3938 1,7 %
migrate	1791558	69594 30,5 %	2307 1,0 %
feed	200	64603 28,3 %	24 0,0 %
feed_all	54600	64579 28,3 %	43 0,0 %
feed_carnivores	54600	61388 26,9 %	310 0,1 %
<method 'binomial' of 'numpy.random.mtrand.RandomStat	16185593	58351 25,6 %	58351 25,6 %
feed	319486	54231 23,8 %	14770 6,5 %
choose_new_location	1791558	52806 23,2 %	7587 3,3 %
fitness	62714345	39878 17,5 %	21134 9,3 %
die	200	32366 14,2 %	49 0,0 %
die	54600	32317 14,2 %	1479 0,6 %
kill_or_not	6023247	30819 13,5 %	8033 3,5 %
death	2183048	27267 12,0 %	4393 1,9 %
procreate	200	22445 9,8 %	23 0,0 %
procreate	54600	22421 9,8 %	1385 0,6 %
birth	2181673	20894 9,2 %	10454 4,6 %
cumsum	1791558	19994 8,8 %	1490 0,7 %
will_move	2183048	18722 8,2 %	1951 0,9 %
<built-in method numpy.core._multiarray_umath.implemen	1791560	18167 8,0 %	914 0,4 %

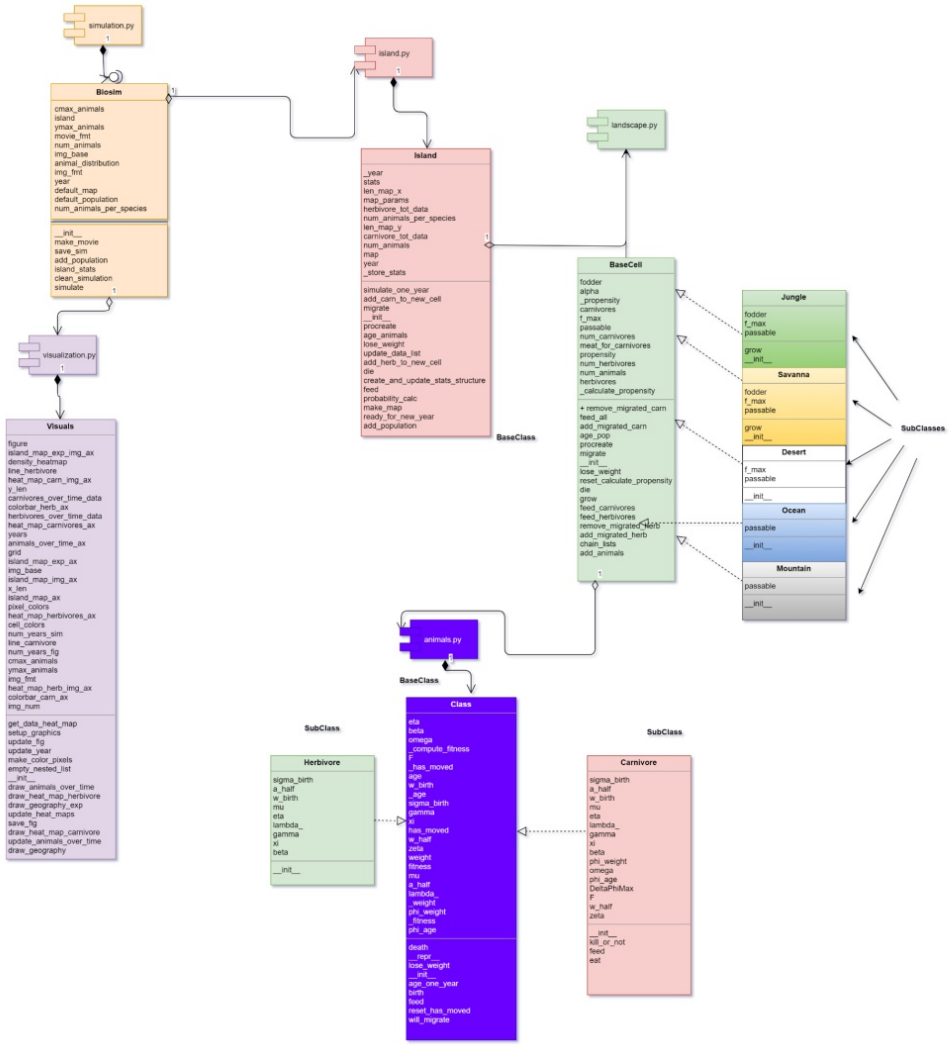
Name	Call Count	Time (ms) ▼	Own Time (ms)
bioscript.py	1	97326 100,0 %	0 0,0 %
clean_simulation	2	95515 98,1 %	1 0,0 %
simulate_one_year	200	95397 98,0 %	3 0,0 %
feed	200	32628 33,5 %	38 0,0 %
feed_all	92400	32590 33,5 %	65 0,1 %
fitness	60290666	29680 30,5 %	25786 26,5 %
feed_carnivores	92400	28251 29,0 %	374 0,4 %
migrate	200	22496 23,1 %	345 0,4 %
feed	466702	22130 22,7 %	9634 9,9 %
migrate	40443	18003 18,5 %	1785 1,8 %
die	200	15737 16,2 %	53 0,1 %
die	92400	15683 16,1 %	1496 1,5 %
procreate	200	15149 15,6 %	35 0,0 %
procreate	92400	15113 15,5 %	1619 1,7 %
death	2933916	13758 14,1 %	3081 3,2 %
birth	2930604	13331 13,7 %	8285 8,5 %
will_migrate	3158289	9131 9,4 %	3063 3,1 %
sort_by_fitness	277200	7621 7,8 %	105 0,1 %
<built-in method builtins.sorted>	280134	7536 7,7 %	984 1,0 %
<lambda>	4357114	6533 6,7 %	1196 1,2 %
choose_new_location	464191	6511 6,7 %	1352 1,4 %
kill_or_not	4060682	5250 5,4 %	3470 3,6 %

Population Dynamics Group 22

Jon-Mikkel Korsvik & Petter Hørtvedt



The project


simulation.py
island.py
landscape.py
Biosim
cmax_animals
island
ymax_animals
movie_fmt
num_animals
img_base
animal_distribution
img_fmt
year
default_map
default_population
num_animals_per_species
__init__
make_movie
save_sim
add_population
island_stats
clean_simulation
simulate
Island
_year
stats
len_map_x
map_params
herbivore_tot_data
num_animals_per_species
len_map_y
carnivore_tot_data
num_animals
map
year
_store_stats
simulate_one_year
add_carn_to_new_cell
migrate
__init__
procreate
age_animals
lose_weight
update_data_list
add_herb_to_new_cell
die
create_and_update_stats_structure
feed
probability_calc
make_map
ready_for_new_year
add_population
BaseCell
fodder
alpha
_propensity
carnivores
f_max
passable
num_carnivores
meat_for_carnivores
propensity
num_herbivores
num_animals
herbivores
_calculate_propensity
+ remove_migrated_carn
feed_all
add_migrated_carn
age_pop
procreate
migrate
__init__
lose_weight
reset_calculate_propensity
die
grow
feed_carnivores
feed_herbivores
remove_migrated_herb
add_migrated_herb
chain_lists
add_animals
Jungle
fodder
f_max
passable
grow
__init__
Savanna
fodder
f_max
passable
grow
__init__
Desert
f_max
passable
__init__
Ocean
passable
__init__
Mountain
passable
__init__
SubClasses
BaseClass
visualization.py
Visuals
figure
island_map_exp_img_ax
density_heatmap
line_herbivore
heat_map_carn_img_ax
y_len
carnivores_over_time_data
colorbar_herb_ax
herbivores_over_time_data
heat_map_carnivores_ax
years
animals_over_time_ax
grid
island_map_exp_ax
img_base
island_map_img_ax
x_len
island_map_ax
pixel_colors
heat_map_herbivores_ax
cell_colors
num_years_sim
line_carnivore
num_years_fig
cmax_animals
ymax_animals
img_fmt
heat_map_herb_img_ax
colorbar_carn_ax
img_num
get_data_heat_map
setup_graphics
update_fig
update_year
make_color_pixels
empty_nested_list
__init__
draw_animals_over_time
draw_heat_map_herbivore
draw_geography_exp
update_heat_maps
save_fig
draw_heat_map_carnivore
update_animals_over_time
draw_geography
animals.py
BaseClass
Class
eta
beta
omega
_compute_fitness
F
_has_moved
age
w_birth
_age
sigma_birth
gamma
xi
has_moved
w_half
zeta
weight
fitness
mu
a_half
lambda_
_weight
phi_weight
_fitness
phi_age
death
__repr__
lose_weight
__init__
age_one_year
birth
feed
reset_has_moved
will_migrate
SubClass
Herbivore
sigma_birth
a_half
w_birth
mu
eta
lambda_
gamma
xi
beta
__init__
SubClass
Carnivore
sigma_birth
a_half
w_birth
mu
eta
lambda_
gamma
xi
beta
phi_weight
omega
phi_age
DeltaPhiMax
F
w_half
zeta
__init__
kill_or_not
feed
eat