

Programowanie Obiektowe - Projekt, Etap 3

Mikołaj Chmielecki, Jakub Mroziński

27 kwietnia 2020

1 Analiza czasownikowo - rzeczownikowa

Symulacja łąki

Język: Java

Symulacja polega na pokazaniu łąki (o ustawionym przez użytkownika rozmiarze) po której poruszają się zwierzęta. Będą nimi: krowy, owce, wilki, koty i myszy. Na początku działania symulacji zostaną one rozmieszczone na losowych polach w losowej liczebności (jednak maksymalna i minimalna początkowa liczba osobników każdego gatunku będzie ustalona przed rozpoczęciem symulacji). Każdy gatunek zwierząt będzie się poruszał po łące z charakterystyczną prędkością i z czasem będzie się starzeć i umierać ze starości. Aby zapobiec wymarciu gatunku przy spotkaniu dwóch zwierząt przeciwnej płci powstanie trzecie o zerowym wieku i posiadające jedną z dwóch płci (50% prawdopodobieństwa).

Zwierzęta różnych gatunków przy spotkaniu na jednym polu będą mogły wchodzić ze sobą w interakcje. Koty po spotkaniu z myszami zjadają je. Wilki, w celu zyskania pożywienia, mogą zaatakować wszystkie zwierzęta, ale prawdopodobieństwo przeprowadzenia skutecznego ataku nie jest stuprocentowe we wszystkich przypadkach: z myszą – 100%, z kotem – 80%, z owcą – 60%, z krową – 40%. Jeżeli atak skończy się niepowodzeniem to zwierzęta rozchodzą się osłabione. Poza tymi przypadkami na jednym polu nie może przebywać więcej niż jedno zwierzę.

Na wolnych polach podczas działania symulacji będzie się pojawiało losowo rozmieszczane pożywienie potrzebne zwierzętom do przetrwania. Będzie to trawa (dla krów i owiec) oraz ser (dla myszy). Gdy jakieś zwierzę napotka na pożywienie którym nie może się pożywić to zostaje ono zniszczone. Przy krawędzi łąki będzie usytuowany wodopój (jeden lub więcej – ustala użytkownik przed rozpoczęciem symulacji), z którego będą mogły w każdej chwili korzystać aby zaspokoić pragnienie.

Symulacja zakończy się gdy wszystkie zwierzęta zginą lub gdy któryś z gatunków osiągnie ustaloną przed rozpoczęciem symulacji liczebność. Po zakończeniu zostaną pokazane statystyki symulacji dla każdego gatunku: maksymalna liczba zwierząt, całkowita liczba zwierząt, liczba zabitych oraz czas trwania symulacji. Statystyki te zostaną również zapisane do pliku, którego nazwa zostanie podana na początku.

2 Karty CRC klas

Classname: Animal	
Superclass: none	
Subclass(es): Cat, Cow, Mouse, Sheep, Wolf	
Responsibilities: Contains common parameters and operations for all animals	Collaboration: Simulation

Classname: Cat	
Superclass: Animal	
Subclass(es): none	
Responsibilities: Contains cat specific operations	Collaboration: none

Classname: Cow	
Superclass: Animal	
Subclass(es): none	
Responsibilities: Contains cow specific operations	Collaboration: none

Classname: Sheep	
Superclass: Animal	
Subclass(es): none	
Responsibilities: Contains sheep specific operations	Collaboration: none

Classname: Mouse	
Superclass: Animal	
Subclass(es): none	
Responsibilities: Contains mouse specific operations	Collaboration: none

Classname: Wolf	
Superclass: Animal	
Subclass(es): none	
Responsibilities: Contains wolf specific operations	Collaboration: none

Classname: Meadow	
Superclass: none	
Subclass(es): none	
Responsibilities: Stores meadow state information	Collaboration: Field, Simulation

Classname: Field	
Superclass: none	
Subclass(es): none	
Responsibilities: Stores information about the content of a given field in a meadow	Collaboration: Feed, Waterhole, Meadow

Classname: Waterhole	
Superclass: none	
Subclass(es): none	
Responsibilities: Stores information about waterholes	Collaboration: Field

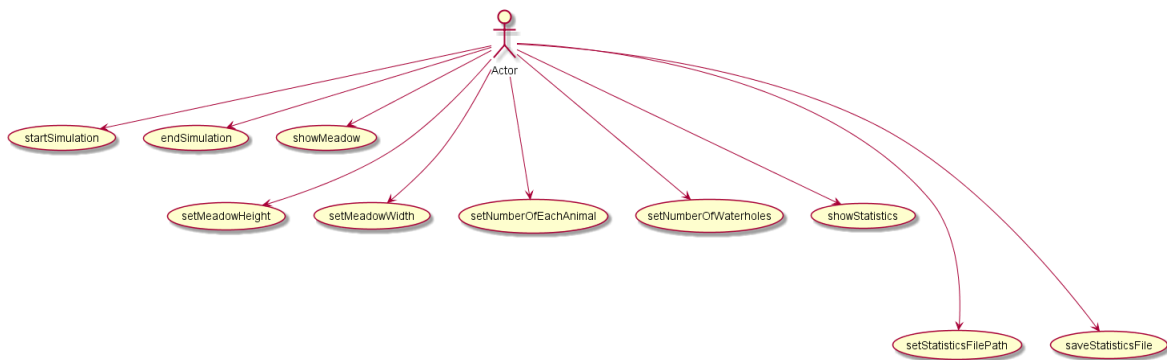
Classname: Feed	
Superclass: none	
Subclass(es): none	
Responsibilities: Stores information about food on meadow	Collaboration: Field

Classname: Parameters	
Superclass: none	
Subclass(es): none	
Responsibilities: It communicates with the user, sets and stores the initial parameters	Collaboration: Control

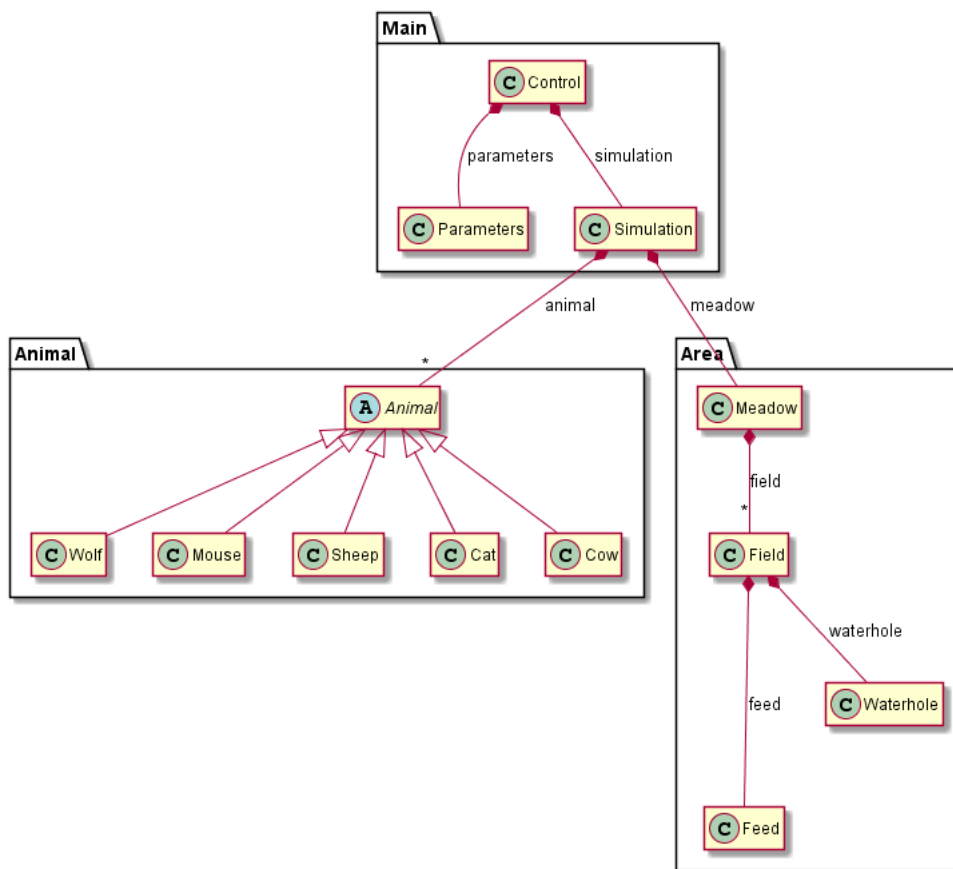
Classname: Simulation	
Superclass: none	
Subclass(es): none	
Responsibilities: Controls animals and meadow	Collaboration: Meadow, Animal, Control

Classname: Control	
Superclass: none	
Subclass(es): none	
Responsibilities: Is starting and ending the simulation. It also has a main loop that iterates through the simulation	Collaboration: Simulation, Parameters

3 Diagram przypadków użycia



4 Diagram obiektów



5 Diagramy klas

