

# Dokumentacja

Etalon Pierwotny Sigmy

Michał Musielak 281092

Mikołaj Bożejko 281163

<https://github.com/ComprexEu/EcosystemSimulation>

## Budowa i włączenie symulacji

- Plik jar należy stworzyć za pomocą komendy: **gradlew fatJar** w folderze z projektem, plik jar pojawi się w folderze EcosystemSimulation/build/libs
- Jara należy uruchomić w konsoli za pomocą komendy: **java -jar EcosystemSimulation-1.0.jar 50 50 5 1 0 0.2 0.2 0.2 0.2**, gdzie przykładowe liczby to argumenty które użytkownik może dostosować, kolejno:
  - wysokość mapy (musi być większa niż 3)
  - szerokość mapy (musi być większa niż 3 i jeżeli jest większa niż 70 mapa się nie wydrukuje)
  - liczba iteracji symulacji
  - opcja wyświetlenia symulacji w konsoli (jeśli zostanie podana "1" i szerokość mapy nie jest większa niż 70)
  - opcja zapisania danych do plików csv (jeśli zostanie podana "1")
  - szansa na spawn jelenia (w zakresie [0,1])
  - szansa na spawn owcy (w zakresie [0,1])
  - szansa na spawn rysia (w zakresie [0,1])
  - szansa na spawn wilka (w zakresie [0,1])

Nie ma ograniczeń co do maksymalnego rozmiaru mapy i liczby iteracji, ale nie zalecamy tworzenia map większych niż 300x300, bo już dla takich rozmiarów symulacja działa wolno.

**W wierszu polecen na windowsie 10 nie wyświetlają się poprawnie kolory! (sprawdzone dla wersji 10.0.19045.4529)**

**Temat:** Symulacja ekosystemu (w języku Java) - Mapa, będzie składała się z elementów klasy WaterTile oraz GrassTile (dziedziczących po Tile). Na polach GrassTile zostaną losowo wygenerowani agenci (gatunki różnych roślin, roślinożerców oraz drapieżników, które będą dziedziczyć kolejno po klasach: Plant, Herbivore i Predator). Rośliny na początku symulacji będą wyrosnięte i w momencie, gdy zostaną zjedzone przez roślinożercę, będą odrastać przez ustalony czas. Poszczególne gatunki roślin będą różniły się czasem rośnięcia oraz efektywnością zaspokajania głodu roślinożerców. Głównym celem zwierząt będzie przedłużenie gatunku. Aby zwierzęta rozmnożyły się, muszą zostać spełnione 2 warunki:

- muszą zaspokoić swoje potrzeby, którymi są spragnienie oraz głód
- musi się spotkać dwójka zwierząt różnych płci

Wszystkie zwierzęta będą zaspokajać spragnienie poprzez podejście do WaterTile na mapie. Na początku symulacji oraz po każdym rozmnożeniu potrzeby agentów resetują się.

Roślinożercy:

- będą zaspakajać głód poprzez jedzenie roślin
- gdy jakiś drapieżnik wejdzie w ich pole widzenia będą od niego uciekać

Drapieżnicy:

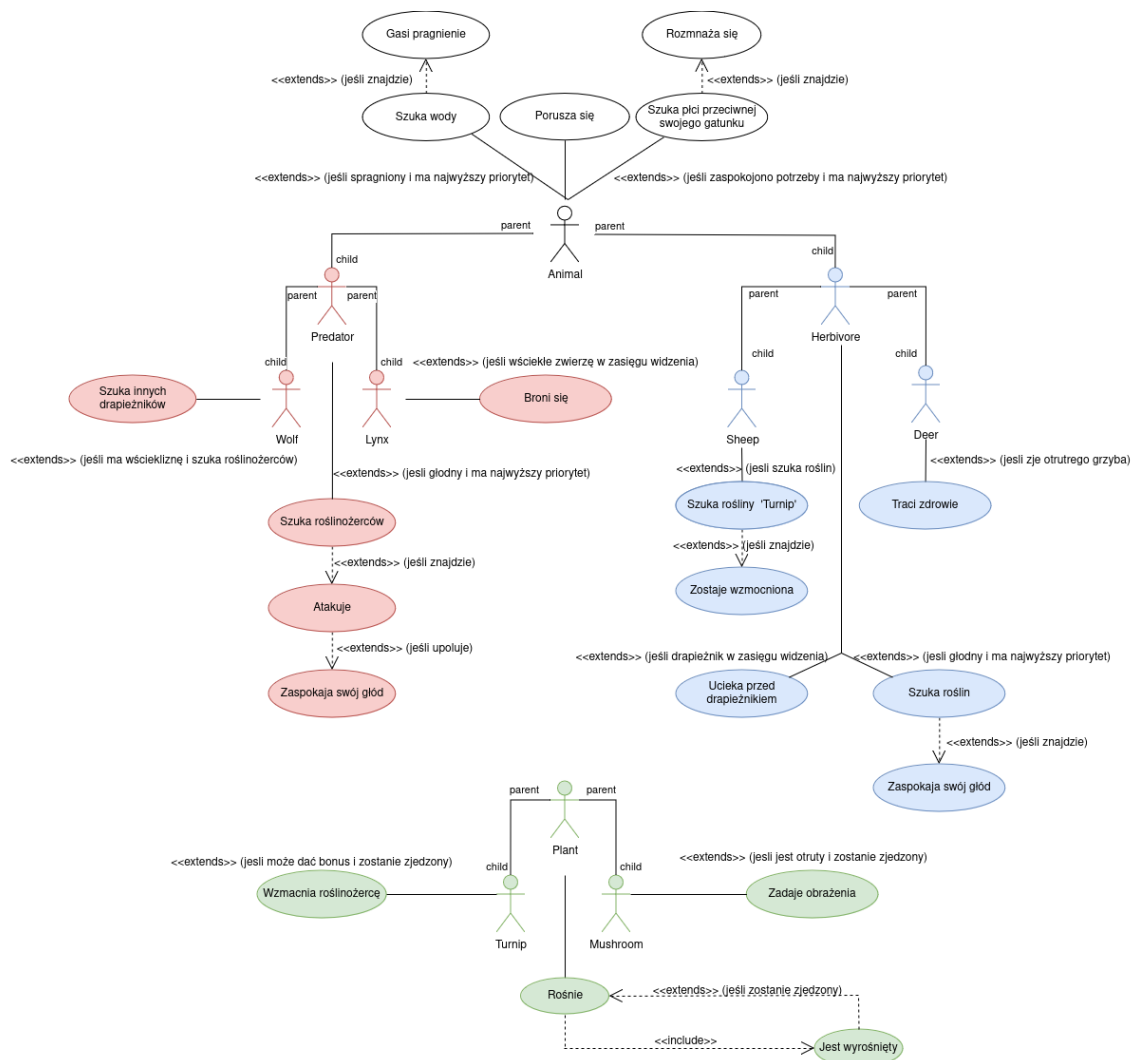
- ich pożywieniem są roślinożercy
- zwiększona prędkość poruszania się
- zwiększone pole widzenia

Symulacja będzie się kończyła po upływie ustalonego czasu lub gdy wszystkie zwierzęta wymrą.

Celem symulacji będzie zbadanie wpływu różnych parametrów (na przykład: szybkość, pole widzenia, liczność gatunku przy inicjalizacji) na zmianę populacji zwierząt i pokazanie wyników na wykresach.

Symulacja zostanie zwizualizowana za pomocą konsoli.

Poniżej przedstawiono odpowiednio diagram przypadków użycia oraz diagramy obiektów. Dla prostszego zrozumienia przypadków, diagramy obiektów zostały zrobione dla limitowanej liczby agentów oraz bez uwzględnienia niektórych (losowych) zachowań zwierząt.



Rysunek 1: Diagram przypadków użycia

Decyzje podejmowane przez zwierzęta będą miały następującą hierarchię:  
Drapieżnicy:

- szukanie płci przeciwnej jeśli osiągnie wymaganą wartość saturacji i spragnienia
- szukanie roślinożerców jeśli wartość saturacji jest mniejsza lub równa od wartości spragnienia oraz nie szuka płci przeciwnej
- szukanie wody jeśli wartość spragnienia jest mniejsza od wartości saturacji oraz nie szuka płci przeciwnej

Roślinożercy:

- uciekanie od drapieżnika jeśli drapieżnik w polu widzenia
- szukanie płci przeciwnej jeśli osiągnie wymaganą wartość saturacji i spragnienia i w polu widzenia nie ma drapieżnika
- szukanie roślin jeśli wartość saturacji jest mniejsza lub równa od wartości spragnienia oraz nie szuka płci przeciwnej i w polu widzenia nie ma drapieżnika
- szukanie wody jeśli wartość spragnienia jest mniejsza od wartości saturacji oraz nie szuka płci przeciwnej i w polu widzenia nie ma drapieżnika

Zwierzęta będą zbliżać się/oddalać do/od znalezionej wcześniej celu. Jeśli nie znajdą celu, będą poruszać się w sposób losowy. Po przemieszczeniu się, zmieniają się statystyki zwierząt i roślin. Co każdą iterację, wartości saturacji i spragnienia zwierząt zmniejszają się o określoną wartość. Zmiany statystyk (zmiana saturacji i spragnienia, a także atakowanie, jedzenie itp.) dzieją się w tej samej iteracji co ruch, ale za ruch dodatkowo poza zmianą pozycji uznajemy rozmnażanie się.

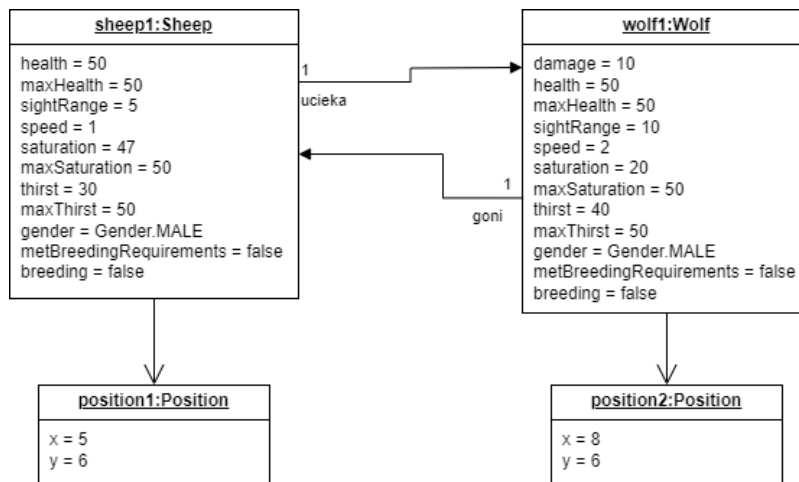
Zwierzęta mają możliwość zregenerowania swojego poziomu zdrowia. Wszystkie zwierzęta regenerują 5 punktów zdrowia, gdy osiągną warunki rozmnożenia się, ponadto drapieżnicy regenerują połowę swojego maksymalnego zdrowia po upolowaniu innego zwierzęcia.

Jeśli rośliny zostaną zjedzone przez roślinożercę - jego saturacja zostanie zwiększona o określoną wartość, a roślina zacznie spowrotem wyrastać przez określoną ilość iteracji.

Dodatkowo, poszczególne zwierzęta i rośliny będą miały swoje charakterystyczne cechy:

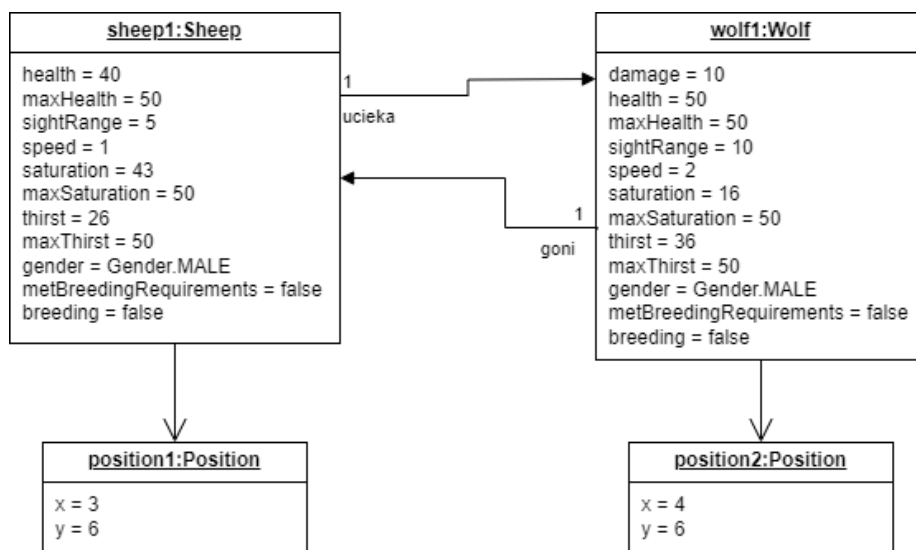
- Mushroom - ma szansę być otruty i zabrać punkty życia
- Turnip - ma szansę dać różny bonus w zależności od tego przez kogo zostanie zjedzony, w naszym przypadku będzie to tylko Sheep, Deer nie dostaje bonusu
- Sheep - może jeść tylko Turnipa, a jedząc go może dostać pewien wylosowany bonus
- Deer - je mushroomy oraz turnipy, może zostać otruty, ale nie jest aplikowany na niego bonus z turnipa
- Wolf - atakuje roślinożerców, ale może dostać wścieklizny i przez to atakować wszystkich
- Lynx - atakuje roślinożerców, jeśli dookoła znajdzie się wściekły Wolf, zada mu obrażenia

Poniżej przedstawiono niektóre z wyżej opisanych przypadków na diagramach obiektów. Dla ułatwienia, zakładamy, że pomiędzy obiektami jest trawa, więc mogą się swobodnie poruszać.



Rysunek 2: Diagram obiektów - ucieczka przed drapieżnikiem

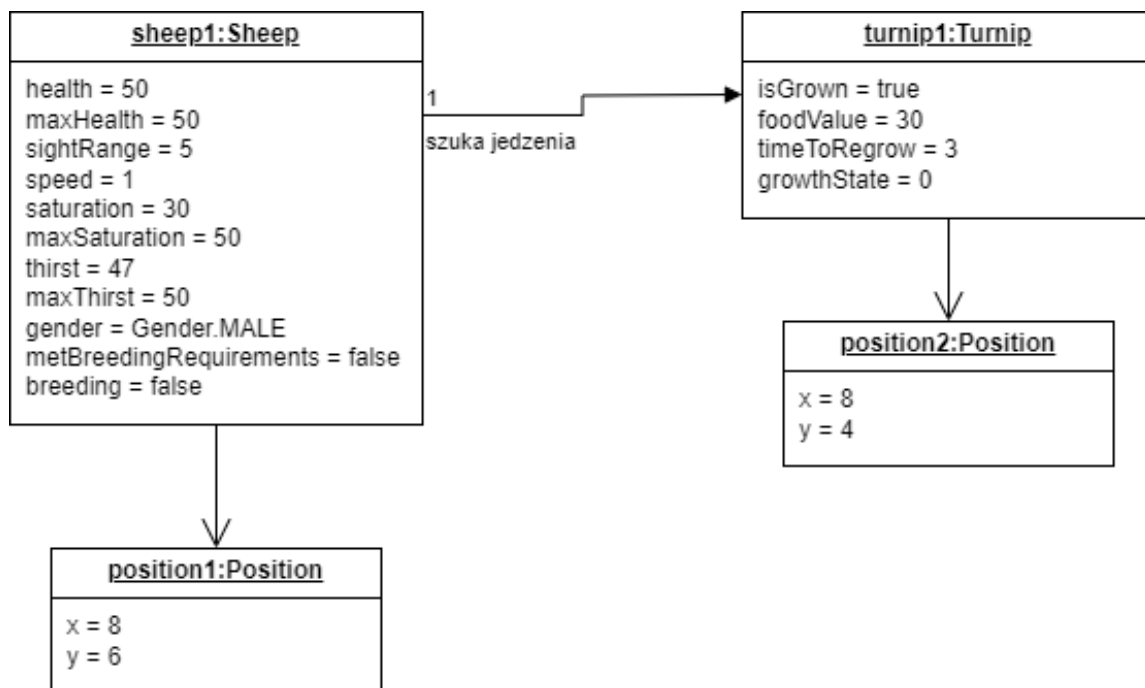
Według hierarchii, dla roślinożercy został spełniony warunek ucieczki, za to dla drapieżnika, warunek szukania (gonienia) roślinożercy. Poniżej przedstawiono sytuację po 2 iteracjach symulacji.



Rysunek 3: Diagram obiektów - ucieczka przed drapieżnikiem - drapieżnik atakuje

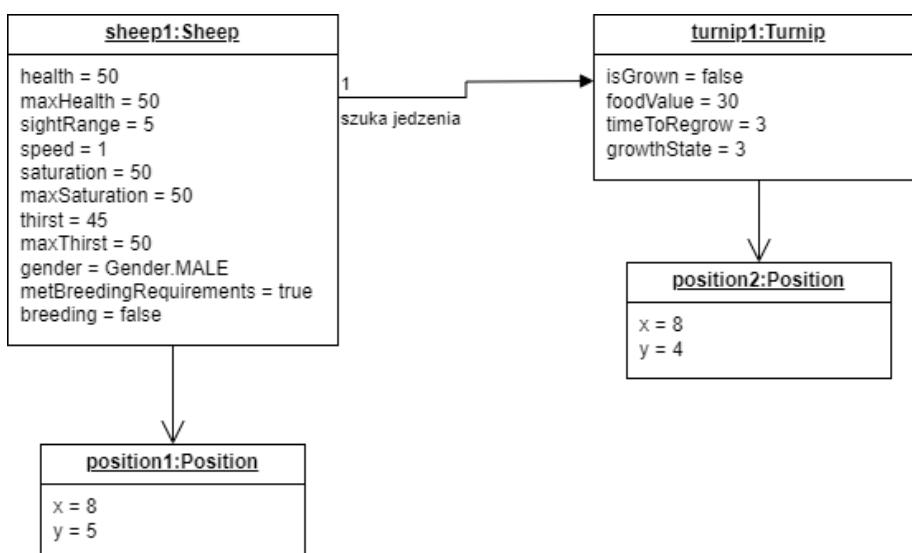
Można zauważyć, że wartość zmiennej health zmniejszyła się o 10. Jest to spowodowane tym, że jeśli drapieżnik znajduje się obok roślinożercy, którego atakuje, zadaje mu obrażenia ( $health -= damage$ ). Drapieżnik dalej goni roślinożercę, lecz ponieważ jest on szybszy, roślinożercy nie uda się uciec.

Następnym przypadkiem jest szukanie rośliny przez roślinożercę, diagram znajduje się na kolejnej stronie.



Rysunek 4: Diagram obiektów - szukanie rośliny

Wartość saturacji jest mniejsza niż wartość spragnienia. Jedzenie jest w polu widzenia, więc roślinożerca zbliża się do niego. Poniżej przedstawiona została sytuacja po jednej iteracji, a co za tym stoi, przy zjedzeniu Turnipa przez owcę (saturacja zwiększyła się o 30, ale jest to więcej niż maxSaturation, więc zwiększyła się do wartości maxSaturation). Wartość zmiennej growthState dla Turnipa zwiększyła się do 3. W każdej następnej iteracji będzie się zmniejszała co 1, aż osiągnie 0, wtedy roślina będzie wyrosnięta.

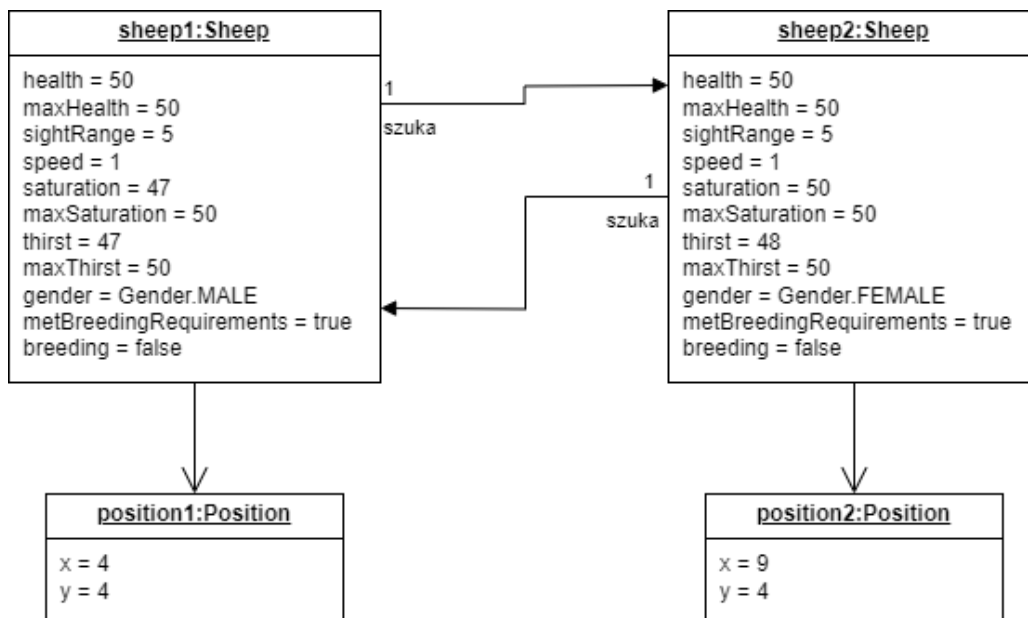


Rysunek 5: Diagram obiektów - szukanie rośliny - roślina zjedzona

Przypadek szukania wody jest analogiczny. Pierwotny pomysł zrobienia terenu jako dwuwymiarowej tablicy obiektów GrassTile i WaterTile dziedziczących po Tile

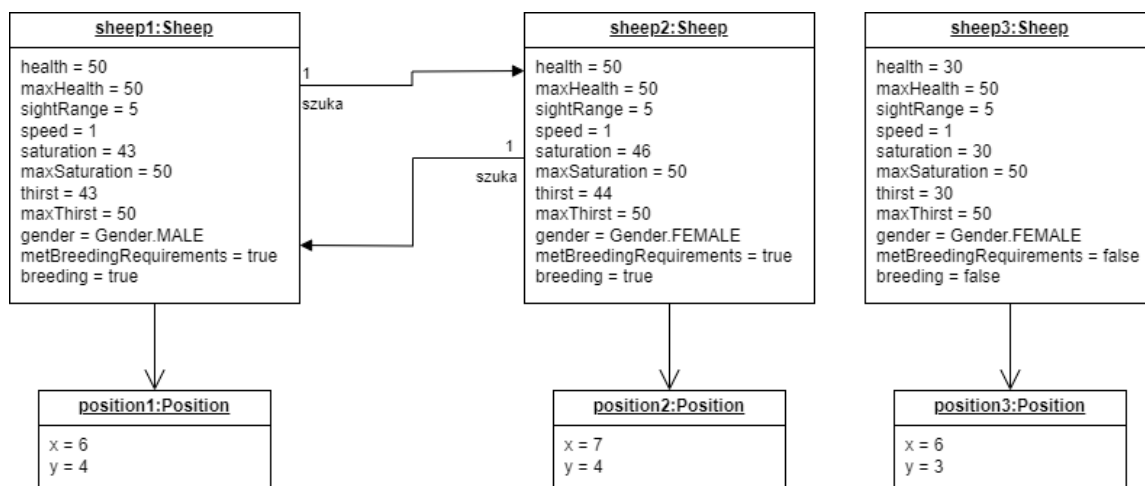
zmienił się. Zostanie ona zaimplementowana jako tablica enumów (enum Terrain z polami WATER oraz GRASS). Z tego powodu, przedstawienie tego przypadku na diagramie obiektów nie jest możliwe.

Kolejnym przypadkiem jest szukanie płci przeciwnej w celu rozmnożenia się:



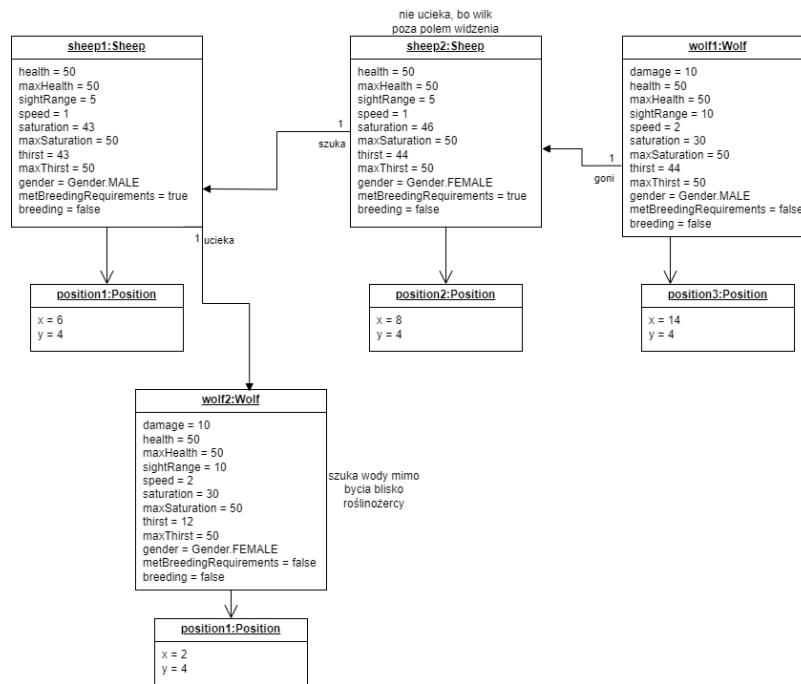
Rysunek 6: Diagram obiektów - szukanie gatunku płci przeciwnej

Zostały spełnione warunki do rozmnożenia się. W tym przypadku owce zbliżają się do siebie i jak do siebie dojdą (po dwóch iteracjach), obok owcy płci żeńskiej pojawi się nowa owca (po kolejnej iteracji).



Rysunek 7: Diagram obiektów - szukanie gatunku płci przeciwnej - nowo urodzona owca

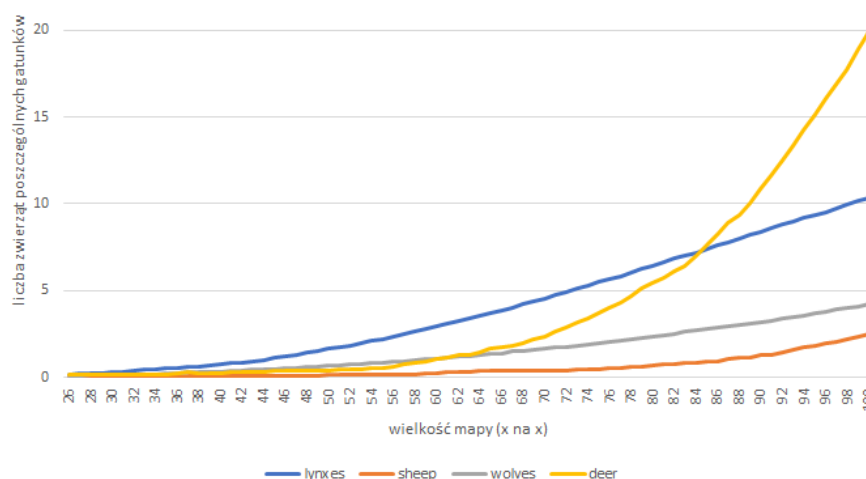
Na kolejnej stronie dodatkowo przedstawiono bardziej złożony przypadek:



Rysunek 8: Diagram obiektów - złożony przypadek

Poniżej przedstawiono badania poszczególnych parametrów na wpływ symulacji.

Na początku przedstawiono wpływ wielkości mapy na liczbę zwierząt (rozważamy mapy kwadratowe). Każda symulacja trwała 80 iteracji. Dla wszystkich symulacji prawdopodobieństwo pojawienia się każdego z gatunków przy inicjalizacji było jednakowe i wynosiło 10%. Dla każdej wielkości wykonano 20 pomiarów i wyniki znajdujące się na poniższym wykresie są ich średnią.



Rysunek 9: Wykres zależności ilości zwierząt od wielkości mapy

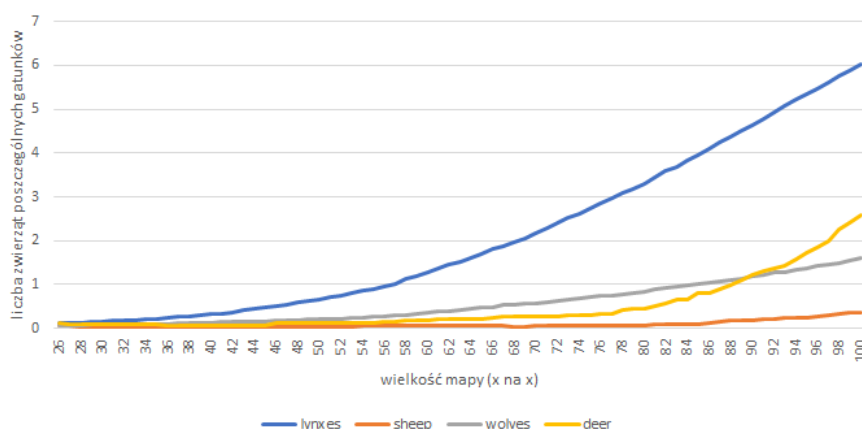
Wyniki dla małych wielkości mapy zostały pominięte ze względu na fakt, że w prawie każdym przypadku kończą się one zerową liczbą zwierząt każdego gatunku.



Dzieje się tak ponieważ drapieżnicy szybko mogą upolować roślinożerców, a następnie umrzeć z głodu.

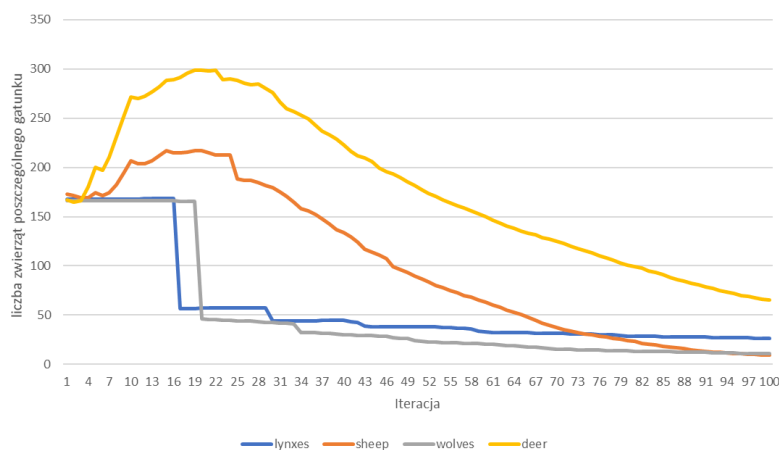
Zwierzęta oraz rośliny mają większość parametrów ściśle określonych. Jednym z nich jest chociażby pole widzenia. Przez to, że zwierzęta po inicjalizacji symulacji znajdują się w 4 różnych ćwiartkach, niektórzy drapieżnicy mogą nigdy nie zobaczyć roślinożercy i stosunkowo szybko umrzeć z głodu. Nasuwa się jeszcze wniosek, że dla środkowych wartości wielkości mapy, rysie są najmocniejsze.

Rozważmy dla porównania przypadek ze zwiększonym polem widzenia dla zwierząt:



Rysunek 10: Wykres zależności ilości zwierząt od wielkości mapy - pole widzenia zwiększone o 5

Powyższy wykres udowadnia więc fakt, że pole widzenia negatywnie wpływa na drapieżników. Na normalnych danych z symulacji, większość z nich po prostu umiera z głodu. Poniżej przedstawiono przykład takiej symulacji dla rozmiaru mapy 100x100 oraz dla 100 iteracji:

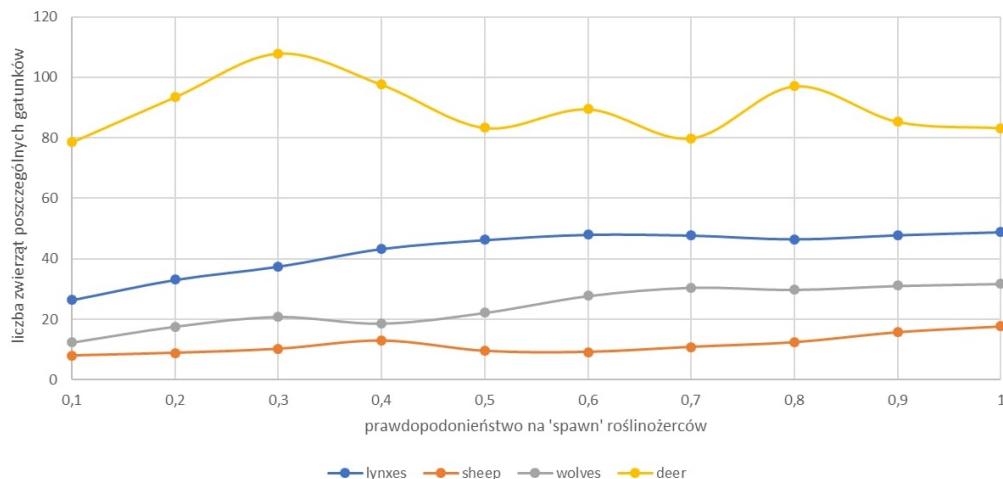


Rysunek 11: Wykres zależności liczby zwierząt od iteracji - rozmiar mapy 100x100

Widać, że w pewnym momencie gwałtownie spada liczba rysi i wilków. Są to zwierzęta, które pojawiły się za daleko potencjalnego pożywienia lub było go niewystarczająco.

Następne badania miały na celu sprawdzenie jaki wpływ na populacje zwierząt ma początkowa liczba zwierząt danego typu.

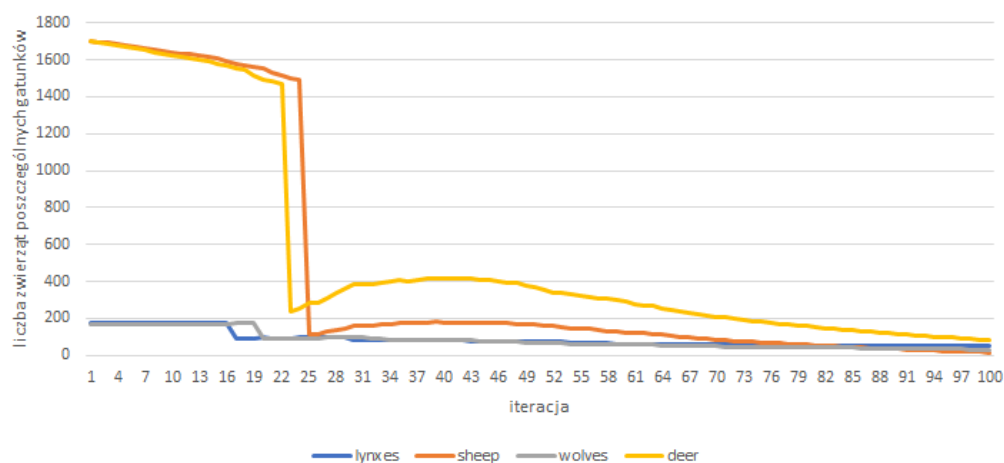
Ustawienia symulacji: rozmiar: 100x100, liczba iteracji: 100, szansa na 'spawn' drapieżników: 0.1, liczba powtórzeń każdej symulacji: 20



Rysunek 12: Wykres zależności końcowej populacji zwierząt od parametrów odpowiedzialnych za 'spawn' roślinożerców

### Wnioski:

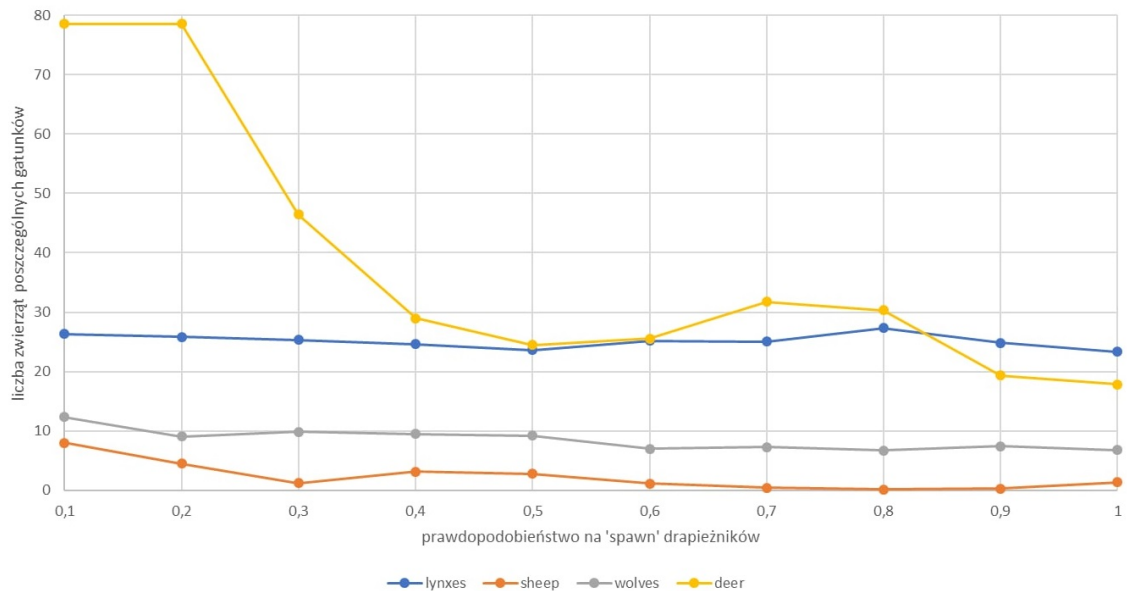
- Zwiększenie początkowej populacji roślinożerców ma pozytywny wpływ na końcową liczbę drapieżników. Wynika to z faktu, że przy większej ilości roślinożerców, drapieżnicy mają więcej jedzenia.
- Końcowa populacja roślinożerców minimalnie się zwiększa wraz ze wzrostem ich liczby na początku symulacji. Mimo, że jest ich zdecydowanie więcej przy prawdopodobieństwie 1, tylko dla niektórych z nich starczy jedzenia (prawdopodobieństwo na 'spawn' roślin jest stałe). Bardzo dobrze pokazuje to poniższy wykres.



Rysunek 13: Wykres zależności liczby zwierząt od iteracji - prawdopodobieństwo na 'spawn' roślinożerców wynosi 1

Widać, że w pewnym momencie gwałtownie spada liczba owiec i jeleni. Są to zwierzęta, które pojawiły się za daleko potencjalnego pożywienia lub nie miały do niego dostępu, bo było go niewystarczająco.

Ustawienia symulacji: rozmiar: 100x100, liczba iteracji: 100, szansa na 'spawn' roślinożerców: 0.1, liczba powtórzeń każdej symulacji: 20

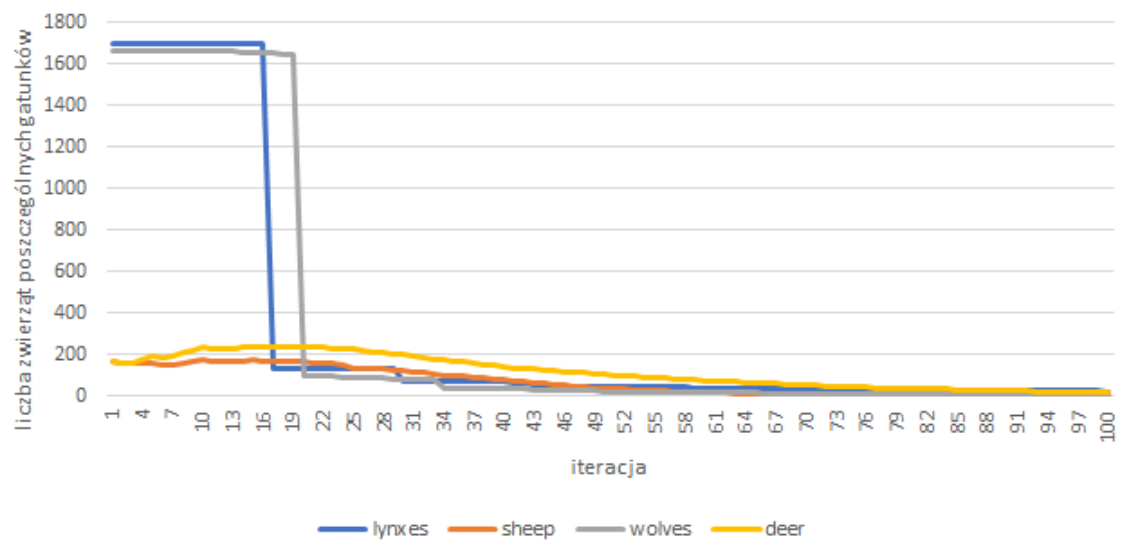


Rysunek 14: Wykres zależności końcowej populacji zwierząt od parametrów odpowiedzialnych za 'spawn' drapieżników

### Wnioski:

- Zwiększenie początkowej populacji drapieżników ma znaczący wpływ na populację roślinożerców. Ich ilość maleje ponad czterokrotnie.
- Końcowa populacja rysi jest stała wraz ze wzrostem prawdopodobieństwa. Dzieje się tak dlatego, że tylko dla niektórych z nich starcza jedzenia, a wielu z nich umiera już na początku poprzez brak roślinożerców w polu widzenia.
- Końcowa populacja wilków ma charakter malejący. Ponieważ wilki mogą dostać wścieklizny - im ich więcej tym więcej wilków z wścieklizną atakujących siebie nawzajem.

Na poniższym wykresie można zauważyć gwałtowny spadek liczby drapieżników omówiony wyżej. Później zaś, ich ilość zmniejsza się stopniowo, nie dla każdego starcza jedzenia.



Rysunek 15: Wykres zależności liczby zwierząt od iteracji - prawdopodobieństwo na 'spawn' drapieżników wynosi 1