

## **oop-project**

### **Propozycje projektu**

#### **Symulacja ekosystemu**

Symulacja środowiska naturalnego, składającego się ze zwierząt, roślin, etc.

Świat będzie dwuwymiarową siatką. Każda z komórek siatki będzie posiadała pewne właściwości a także będzie mogła zawierać jakiś organizm (roślinę lub zwierzę).

##### **Rośliny**

Rośliny będą organizmami bardziej pasywnymi, tzn. będą reagowały na zmiany środowiska w pewnym, tudzież ograniczonym, stopniu, np. jeżeli środowisko będzie miało odpowiednie nasłonecznienie i wilgotność, będą rosnąć; w przeciwnym wypadku, będą umierać. Będą miały również bardzo ograniczoną możliwość wpływania na środowisko czy inne organizmy.

Rośliny będą właściwością komórki. Komórka będzie miała poziom roślinności. Roślinożercy przebywający na tej komórce będą żywić się zmniejszając poziom roślinności w komórce.

##### **Zwierzęta**

Elementami wyróżniającymi organizmy zwierzęce będzie przede wszystkim możliwość zmiany swojej pozycji oraz większy zakres wpływu na swoje środowisko. Zwierzę będzie musiało w sposób aktywny zaspokajać swoje potrzeby fizjologiczne: głód, odpoczynek, etc. W sposób aktywny, tj. podejmować kroki do osiągnięcia celu, np. szukając pożywienia, unikać potencjalnych zagrożeń, etc.

Każdy gatunek będzie miał określone statystyki: wytrzymałość, siła i szybkość. Statystyki zwierząt tego samego gatunku mogą się delikatnie różnić. Jeśli wytrzymałość spadnie do 0 to osobnik umiera. Wytrzymałość można stracić podczas konfrontacji z innymi zwierzętami oraz można ją odzyskać odpoczywając i jedząc. Siła odpowiada za to jak dobrze zwierze radzi sobie w walce, a szybkość jak sprawnie przemieszcza się po świecie.

Będą różne typy zwierząt różniące się pod wieloma względami: roślinożercy i mięsożercy, samiec i samica, młode i dorosłe, jak dużo potomstwa będą produkować, jakich strategii będą używać aby przeżyć, etc.

Jeśli wytrzymałość spadnie do 0 to osobnik umiera. Wytrzymałość można stracić podczas konfrontacji z innymi zwierzętami oraz można ją odzyskać odpoczywając i jedząc. Siła odpowiada za to jak dobrze zwierze radzi sobie w walce, a szybkość jak sprawnie przemieszcza się po świecie.

**Gatunki** Organizmy zwierzęce będą dzielić się na **gatunki**. Rozmnażanie będzie się odbywać tylko w obrębie gatunku (brak crossbreedingu). Statystyki potomstwa będą generowane na podstawie statystyk rodziców.

Oprócz tego, gatunki mogą mieć różne wzorce zachowań. Przykładowo: roślinożerca po znalezieniu pożywienia wchodzi np. w stan JEDZ, który polega po prostu przemieszczeniu się w jego kierunku pożywienia i spożyciu go; mięsożerca natomiast po wypatrzeniu pożywienia (innego zwierzęcia) przejdzie w stan POLUJ, w którym zwierzę będzie starało się najpierw zabić swój cel. Po udanym polowaniu mięsożerca może zjeść swoją zdobycz, w wypadku porażki zwierzę poszuka innego celu. Z pozoru takie same stany, np. POLUJ mogą również różnić się pomiędzy gatunkami, np. gepard będzie polował samotnie, ale wilki będą polować w grupie, co będzie wymagało koordynacji i kooperacji pomiędzy przedstawicielami gatunku zarówno podczas polowania, jak i po (dzielenie się pożywieniem).

### **Skład grupy**

- Marcel Guzik (256317) (**lider**)
- Miłosz Skóra (252733)

# Symulacja ekosystemu

---

Symulacja **środowiska naturalnego**, składającego się ze **zwierząt, roślin**, etc.

**Świat** będzie dwuwymiarową siatką. Każda z komórek siatki będzie **posiadała pewne właściwości** a także będzie mogła **zawierać jakiś organizm** (roślinę lub zwierzę).

## Rośliny

**Rośliny** będą organizmami bardziej pasywnymi, tzn. **będą reagowały na zmiany środowiska** w pewnym, tudzież ograniczonym, stopniu, np. jeżeli **środowisko** będzie miało odpowiednie **nasłonecznienie i wilgotność**, będą **rosnąć**; w przeciwnym wypadku, będą **umierać**. Będą miały również bardzo **ograniczoną możliwość wpływania na środowisko czy inne organizmy**.

**Rośliny** będą właściwością **komórki**. **Komórka** będzie **miała poziom roślinności**. **Roślinożercy** przebywający na tej **komórce** będą **żywić** się **zmniejszając poziom roślinności w komórce**.

## Zwierzęta

Elementami wyróżniającymi **organizmy zwierzęce** będzie przede wszystkim **możliwość zmiany swojej pozycji** oraz większy zakres **wpływu na swoje środowisko**. **Zwierzę** będzie musiało w sposób aktywny **zaspokajać swoje potrzeby fizjologiczne: głód, odpoczynek**, etc. W sposób aktywny, tj. **podejmować kroki do osiągnięcia celu**, np. **szukając pożywienia, unikać potencjalnych zagrożeń**, etc.

Każdy **gatunek** będzie miał określone **statystyki: wytrzymałość, siła i szybkość**. **Statystyki zwierząt** tego samego **gatunku** mogą się delikatnie różnić. Jeśli **wytrzymałość** spadnie do 0 to osobnik **umiera**. **Wytrzymałość** można **stracić** podczas **konfrontacji** z innymi **zwierzętami** oraz można ją **odzyskać odpoczywając i jedząc**. **Siła** odpowiada za to jak dobrze **zwierze** radzi sobie w **walce**, a **szybkość** jak sprawnie **przemieszcza** się po **świecie**.

Będą różne typy **zwierząt** różniące się pod wieloma względami: **roślinożercy** i **mięsożercy**, **samiec** i **samica**, **młode** i **dorośle**, jak dużo **potomstwa** będą **produkować**, jakich **strategii** będą **używać** aby **przeżyć**, etc.

## Gatunki

**Organizmy** zwierzęce będą **dzielić** się na **gatunki**. **Rozmnażanie** będzie się **odbywać** tylko w obrębie **gatunku** (brak crossbreedingu). **Statystyki potomstwa** będą **generowane** na podstawie **statystyk rodziców**.

Oprócz tego, **gatunki** mogą **mieć** różne **wzorce zachowań**. Przykładowo: **roślinożerca** po **znalezieniu pożywienia** wchodzi np. w **stan JEDZ**, który polega po prostu **przemieszczeniu** się w jego **kierunku pożywienia** i **spożyciu go**; **mięsożerca** natomiast po **wypatrzeniu pożywienia** (innego zwierzęcia) przejdzie w **stan POLUJ**, w którym **zwierzę** będzie **starało** się najpierw **zabić** swój **cel**. Po udanym **polowaniu mięsożerca** może **zjeść** swoją **zdobycz**, w wypadku **porażki zwierzę poszuka** innego **celu**. Z pozoru takie same **stany**, np. **POLUJ** mogą również **różnić** się pomiędzy **gatunkami**, np. **gepard** będzie **polował samotnie**, ale **wilki** będą **polować w grupie**, co będzie **wymagało koordynacji i kooperacji** pomiędzy **przedstawicielami gatunku** zarówno podczas **polowania**, jak i po (dzielenie się pożywieniem).

App	
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Akceptowanie wejścia użytkownika</li> <li>• Wykonywanie kroków symulacji</li> <li>• Zarządza obiektem klasy World</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> </ul>

World	
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Stan świata</li> <li>• Przechowywanie wszystkich obiektów klasy Animal i mapy</li> <li>• Wykonywanie kroku symulacji</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• Animal</li> <li>• Tile</li> </ul>

Abstract <div> <b>Tile</b>  Klasy potomne: ForestTile, DesertTile, WaterTile </div>	
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Przechowuje informacje na temat komórki</li> <li>• Bycie abstrakcją dla konkretnego typu komórki</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> </ul>

Abstract <div> <b>Animal</b>  Klasy potomne: Wold, Lion, Antelope, Hippo, Crocodile </div>	
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Symuluje jedno zwierze</li> <li>• Klasa rodzic wszystkich gatunków</li> <li>• Zawiera obiekt AiBehaviour do oddelegowywania zachowania zwierzęcia</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> </ul>

Wolf		Klasa rodzicielska: Animal
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Symuluje jednego osobnika z gatunku wilk</li> <li>• Porusza się po ForestTile</li> <li>• Poluje na roślinożerców</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> </ul>	

Lion		Klasa rodzicielska: Animal
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Symuluje jednego osobnika z gatunku lew</li> <li>• Porusza się po DesertTile</li> <li>• Poluje na roślinożerców</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> </ul>	

Antelope		Klasa rodzicielska: Animal
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Symuluje jednego osobnika z gatunku antylopa</li> <li>• Żywi się roślinnością</li> <li>• Porusza się po DesertTile, ForestTile</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> </ul>	

Hippo		Klasa rodzicielska: Animal
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Symuluje jednego osobnika z gatunku hipopotam</li> <li>• Porusza się po DesertTile</li> <li>• Żywi się roślinnością</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> </ul>	

<div> <div>Crocodile</div> <div>Klasa rodzicielska: Animal</div> </div>	
<div>Obowiązki</div> <ul style="list-style-type: none"> <li>• Symuluje jednego osobnika z gatunku krokodyl</li> <li>• Porusza się po WaterTile</li> <li>• Poluje na roślinożerców</li> </ul>	<div>Współpracownicy</div> <ul style="list-style-type: none"> <li>• World</li> </ul>

<div> <div>Abstract</div> <div>AiBehaviour</div> <div>Klasy potomne: AiBehaviourWolf, AiBehaviourLion, AiBehaviourCrocodile, AiBehaviourAntelope, AiBehaviourHippo</div> </div>	
<div>Obowiązki</div> <ul style="list-style-type: none"> <li>• Odpowiada za poruszanie się zwierzęcia</li> <li>• Odpowiada za zmianę stanów zwierzęcia</li> <li>• Zarządza zbiorem stanów do których zwierze może wejść</li> <li>• Uchwyt do AI zwierzęcia</li> </ul>	<div>Współpracownicy</div> <ul style="list-style-type: none"> <li>• IAiState</li> </ul>

<div> <div>AiBehaviourWolf</div> <div>Klasa rodzicielska: AiBehaviour</div> </div>	
<div>Obowiązki</div> <ul style="list-style-type: none"> <li>• Zachowanie AI wilka</li> <li>• Poluje poruszając się i aktywnie szukając zwierzyny</li> </ul>	<div>Współpracownicy</div> <ul style="list-style-type: none"> <li>• IAiState</li> </ul>

<div> <div>AiBehaviourLion</div> <div>Klasa rodzicielska: AiBehaviour</div> </div>	
<div>Obowiązki</div> <ul style="list-style-type: none"> <li>• Zachowanie AI lwa</li> <li>• Poluje poruszając się i aktywnie szukając zwierzyny</li> </ul>	<div>Współpracownicy</div> <ul style="list-style-type: none"> <li>• IAiState</li> </ul>

<div>Klasa rodzicielska: AiBehaviour</div> <div><b>AiBehaviourAntelope</b></div>	
<div><b>Obowiązki</b></div> <ul style="list-style-type: none"> <li>• Zachowanie AI antylopy</li> <li>• Jeżeli zaatakowana, ucieka</li> </ul>	<div><b>Współpracownicy</b></div> <ul style="list-style-type: none"> <li>• IAiState</li> </ul>

<div>Klasa rodzicielska: AiBehaviour</div> <div><b>AiBehaviourHippo</b></div>	
<div><b>Obowiązki</b></div> <ul style="list-style-type: none"> <li>• Zachowanie AI hipopotama</li> <li>• Jeżeli jest atakowany, walczy z drapieżnikami</li> </ul>	<div><b>Współpracownicy</b></div> <ul style="list-style-type: none"> <li>• IAiState</li> </ul>

<div>Klasa rodzicielska: AiBehaviour</div> <div><b>AiBehaviourCrocodile</b></div>	
<div><b>Obowiązki</b></div> <ul style="list-style-type: none"> <li>• Zachowanie AI krokodyla</li> <li>• Poluje chowając się w wodzie</li> </ul>	<div><b>Współpracownicy</b></div> <ul style="list-style-type: none"> <li>• IAiState</li> </ul>

<div>Interface</div> <div><b>IAiState</b></div>	
<div><b>Obowiązki</b></div> <ul style="list-style-type: none"> <li>• Interfejs stanu zachowania zwierzęcia</li> <li>• Klasy implementujące ten interfejs zawierają czynność którą wykonuje dane zwierze</li> <li>• Klasy implementujące ten interfejs zawierają wszelkie referencje potrzebne zwierzęciu do wykonania czynności</li> </ul>	<div><b>Współpracownicy</b></div> <ul style="list-style-type: none"> <li>• World</li> <li>• IAiBehaviour</li> </ul>



Abstract	
<b>AiStateSleep</b>	
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Stan snu zwierzęcia</li> <li>• Regeneruje energię zwierzęcia</li> <li>• Zmniejsza obszar widzenia zwierzęcia</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> <li>• IAiBehaviour</li> </ul>

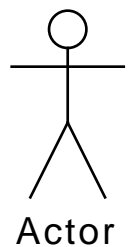
Abstract	
<b>AiStateEatPlant</b>	
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Stan w którym zwierze pożywia się roślinami</li> <li>• Zmniejsza roślinność w komórce w której jest zwierze</li> <li>• Zwiększa najedzenie</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> <li>• IAiBehaviour</li> </ul>

Abstract	
<b>AiStateLookForFood</b>	
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Stan w którym zwierze poszukuje roślinności do zjedzenia</li> <li>• Po znalezieniu pożywienia przechodzi w stan AiStateEatPlant</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> <li>• IAiBehaviour</li> </ul>

Abstract	
<b>AiStateHunt</b>	
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Stan w którym zwierze poluje</li> <li>• Odpowiada za walkę drapieżnika ze zwierzyną</li> <li>• Po upolowaniu zwierzyny przechodzi w stan AiStateEatCorpse</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> <li>• IAiBehaviour</li> </ul>

Abstract	
<b>AiStateEatCorpse</b>	
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Stan w którym zwierze pożywia się upolowaną zdobyczą</li> <li>• Zmniejsza ilość mięsa w komórce</li> <li>• Zwiększa najedzenie</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> <li>• IAiBehaviour</li> </ul>

Abstract	
<b>AiStateCopulate</b>	
<b>Obowiązki</b> <ul style="list-style-type: none"> <li>• Stan w którym para zwierząt się rozmnaża</li> <li>• Odpowiada za znalezienie partnera</li> <li>• Odpowiada za zbliżenie się do partnera</li> <li>• Po ukończeniu kopulacji pojawia się nowy osobnik danego gatunku</li> </ul>	<b>Współpracownicy</b> <ul style="list-style-type: none"> <li>• World</li> <li>• IAiBehaviour</li> </ul>



System

Rozpocznij symulacje

sterowanie przebiegiem  
czasu

Poruszaj kamerę po  
wiecie

Zakocz symulacje

Zapisz symulacje

Wczytaj symulacje

Zapauzuj symulacje

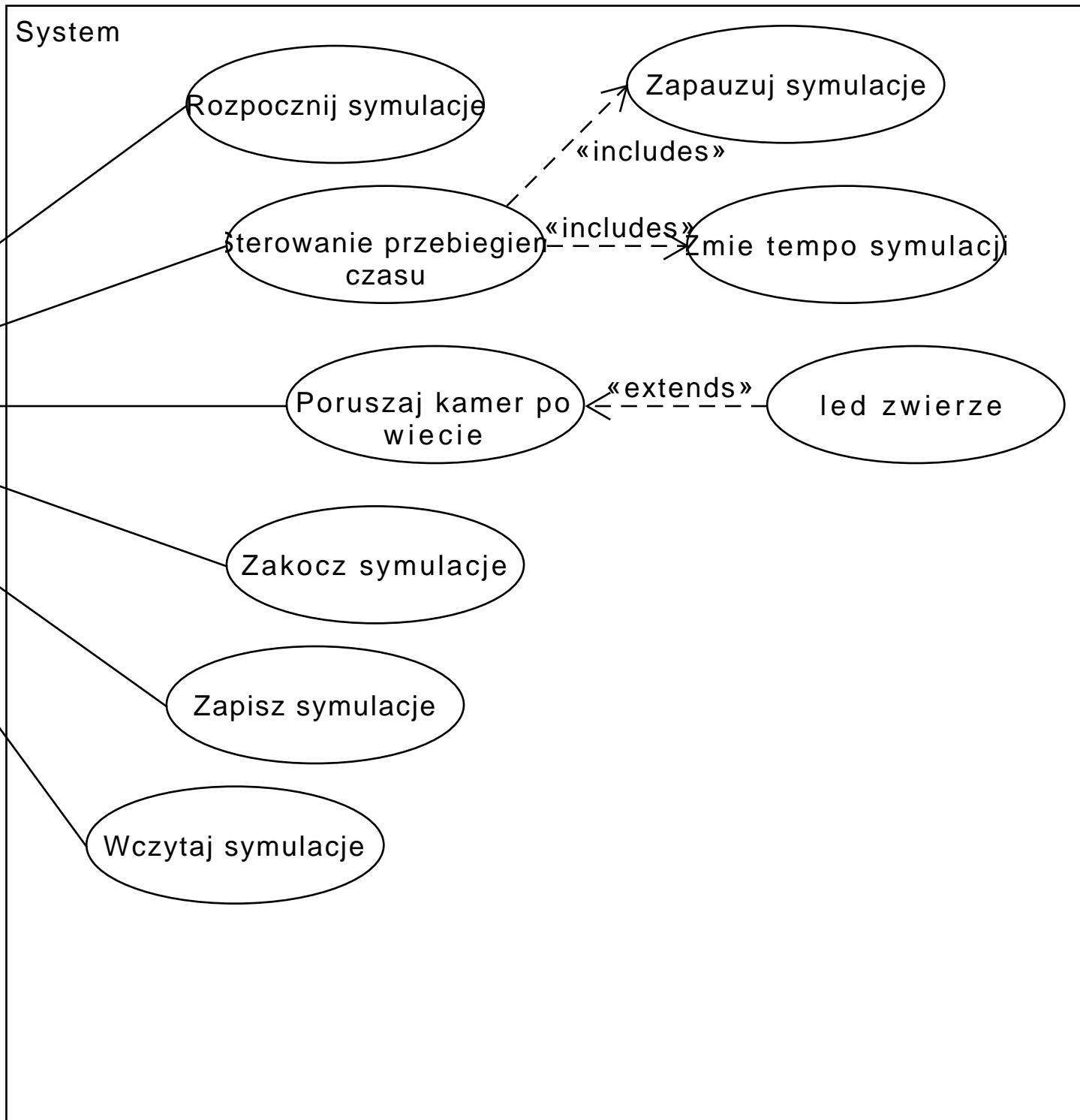
«includes»

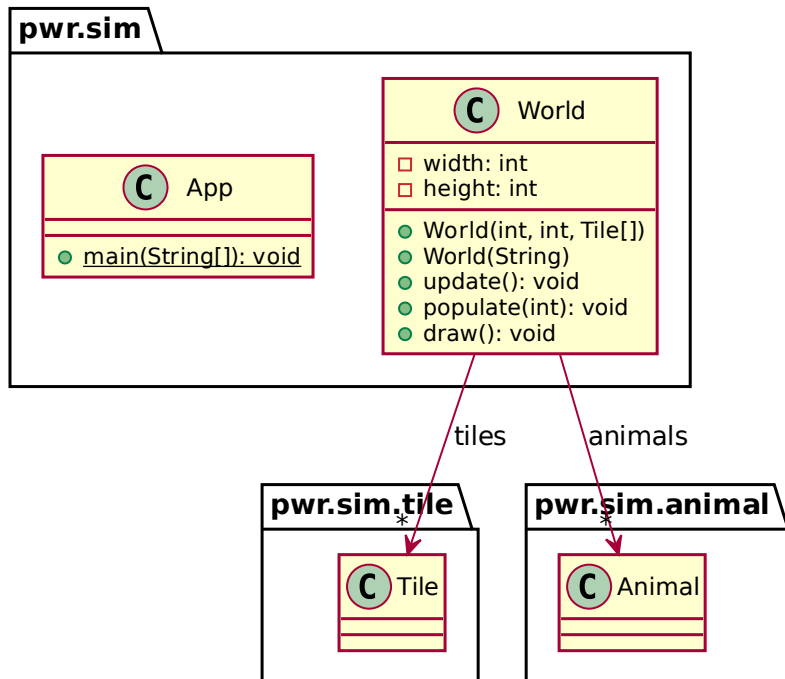
«includes»

Zmień tempo symulacji

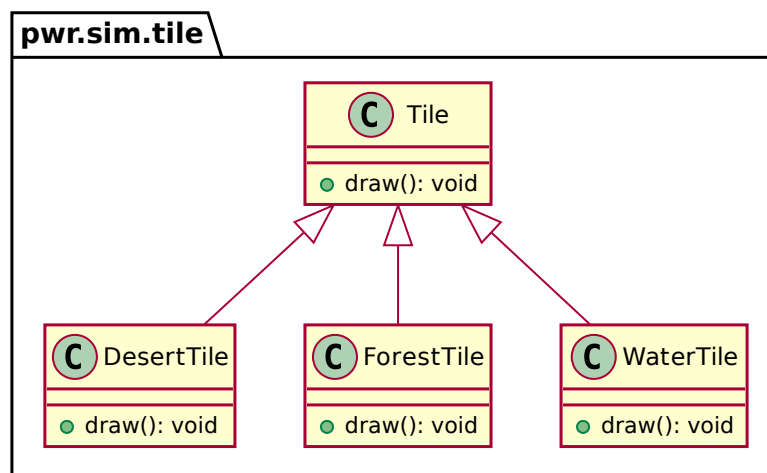
«extends»

Idź za zwierzętem

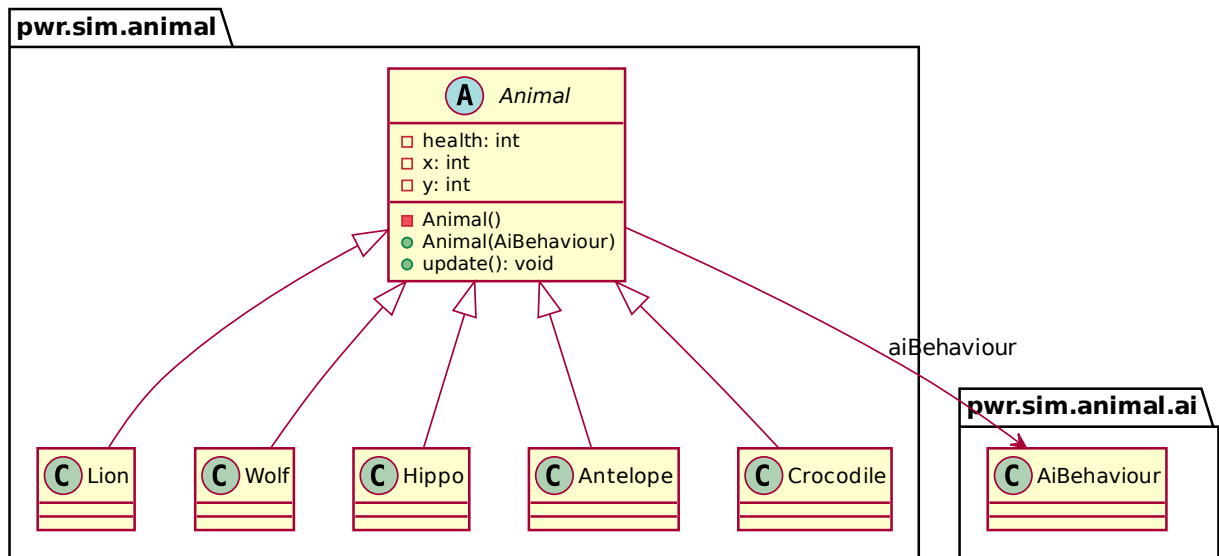




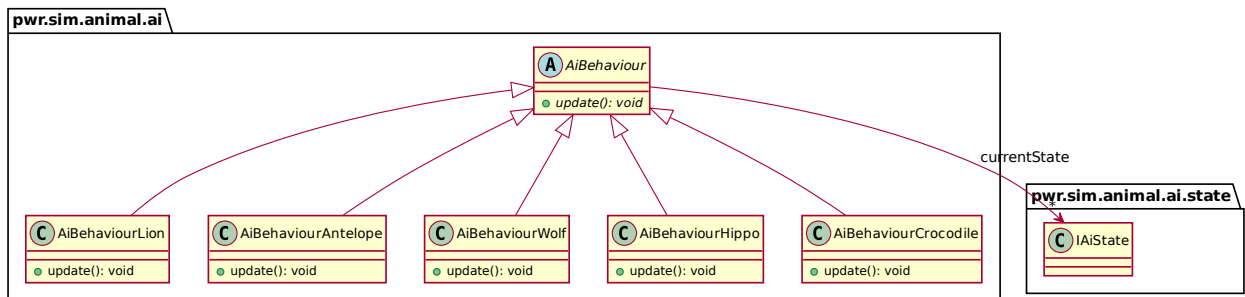
UMLDoclet 2.0.9, PlantUML 1.2020.08



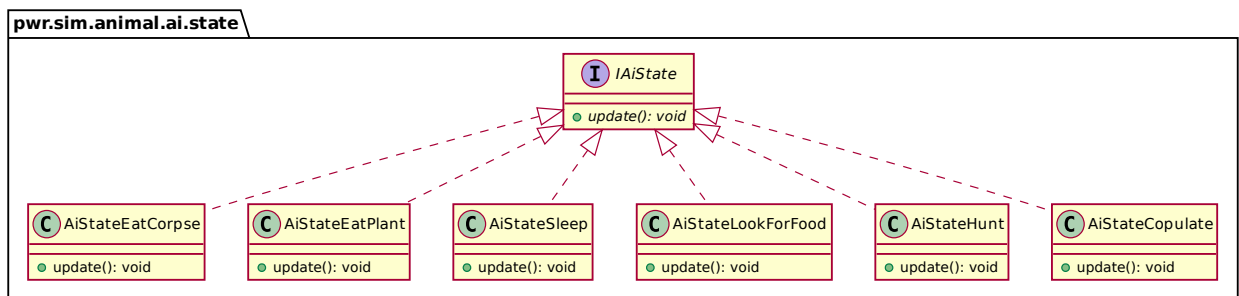
UMLDoclet 2.0.9, PlantUML 1.2020.08



UMLDoclet 2.0.9, PlantUML 1.2020.08

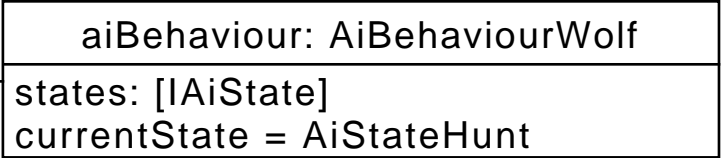
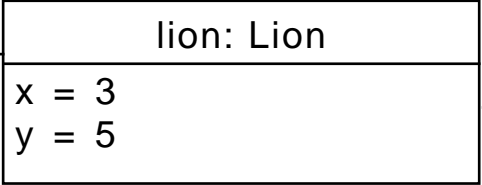
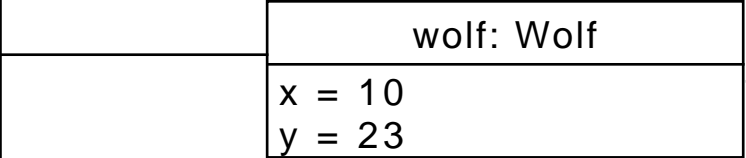
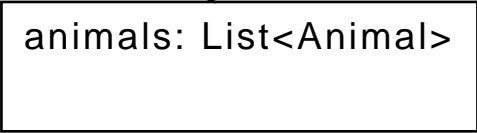
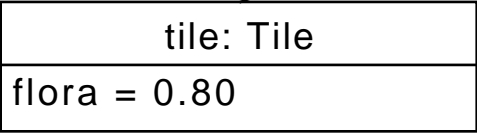
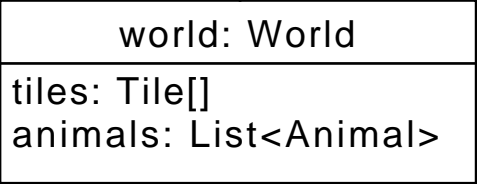
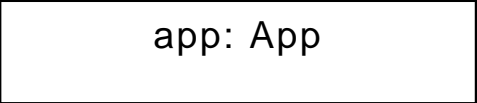


UMLDoclet 2.0.9, PlantUML 1.2020.08



UMLDoclet 2.0.9, PlantUML 1.2020.08

Diagramy zostały wygenerowane przez narzędzie UMLDoclet. Niestety nie jest ono w stanie umieścić wszystkich paczek i klas na jednym diagramie.



## Etap 4 - logika symulacji

### Diagramy sekwencji

- StateCopulate

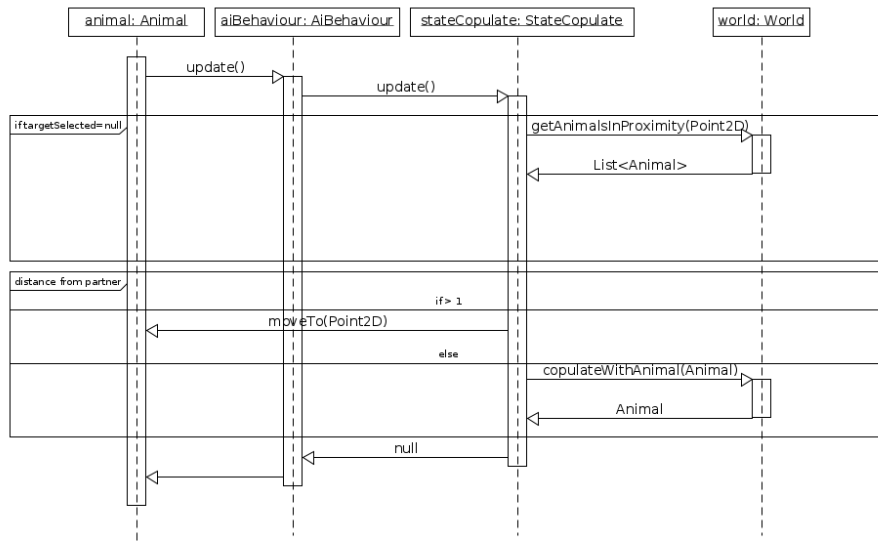


Figure 1: copulate

- StateHunt
- StateLookForFood

Jak można zauważyć stany `Copulate` i `Hunt` mają bardzo podobną strukturę - obiekt stanu wysyła do `World` zapytanie o listę innych zwierząt w pobliżu, którą filtrują wg. swoich predykatów. Tak podobną funkcjonalność można by wydzielić do oddzielnej klasy.

Zmiana aktualnego stanu bez wypychania nowego stanu na stos będzie realizowana przez zwracanie nowego stanu przez `IAiState::update()`.

### Diagramy aktywności

- Hunt
- Copulate

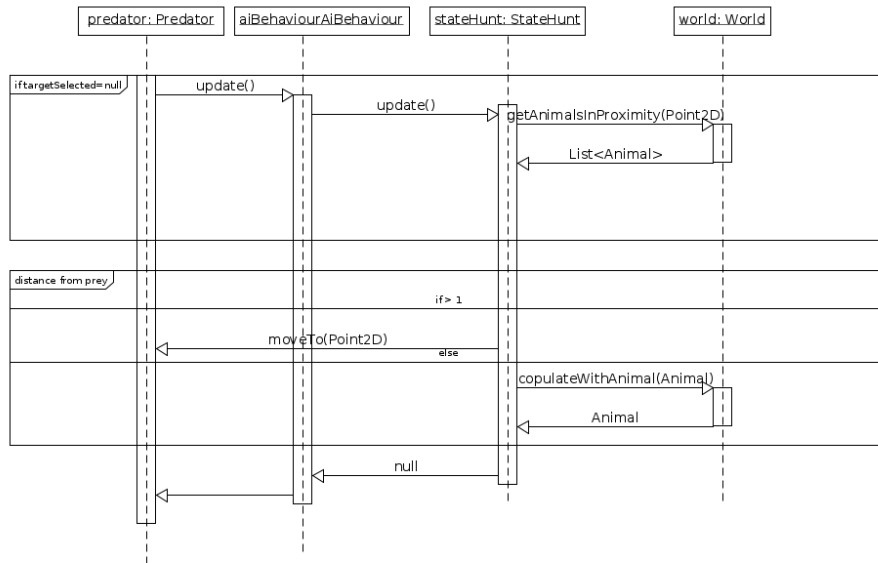


Figure 2: hunt

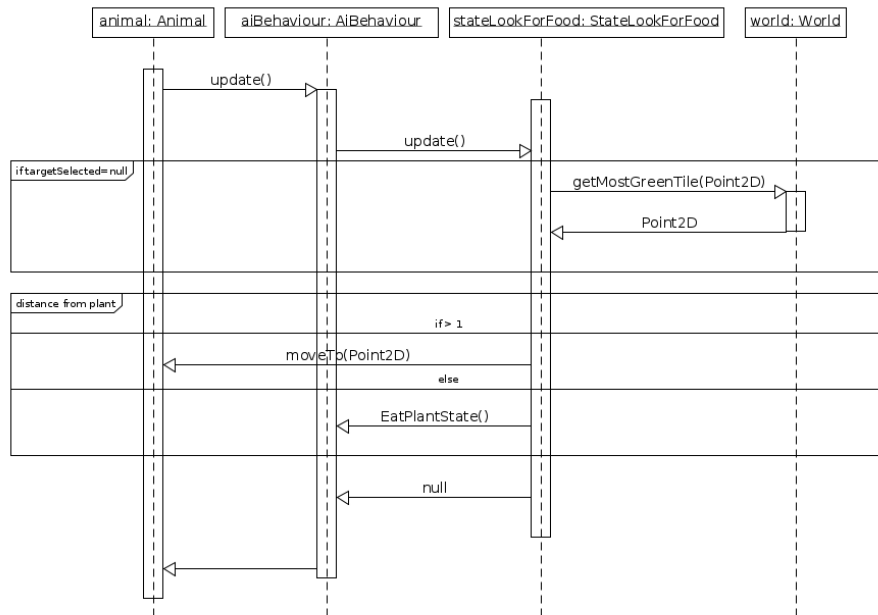


Figure 3: lookforfood



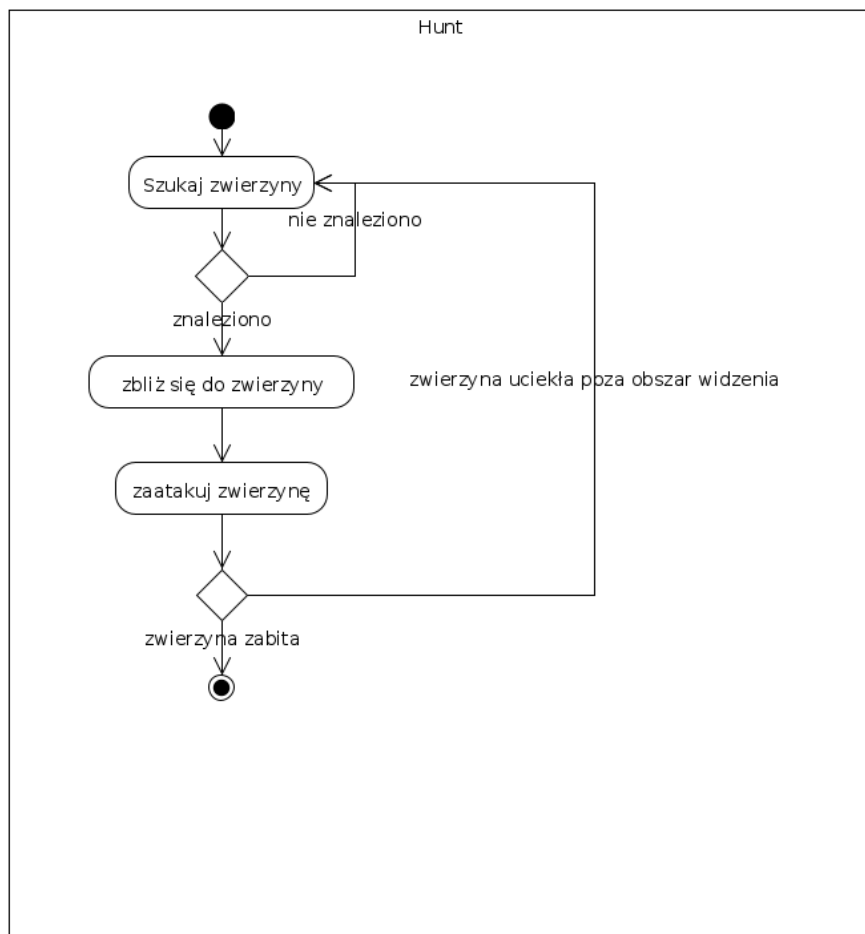


Figure 4: hunt

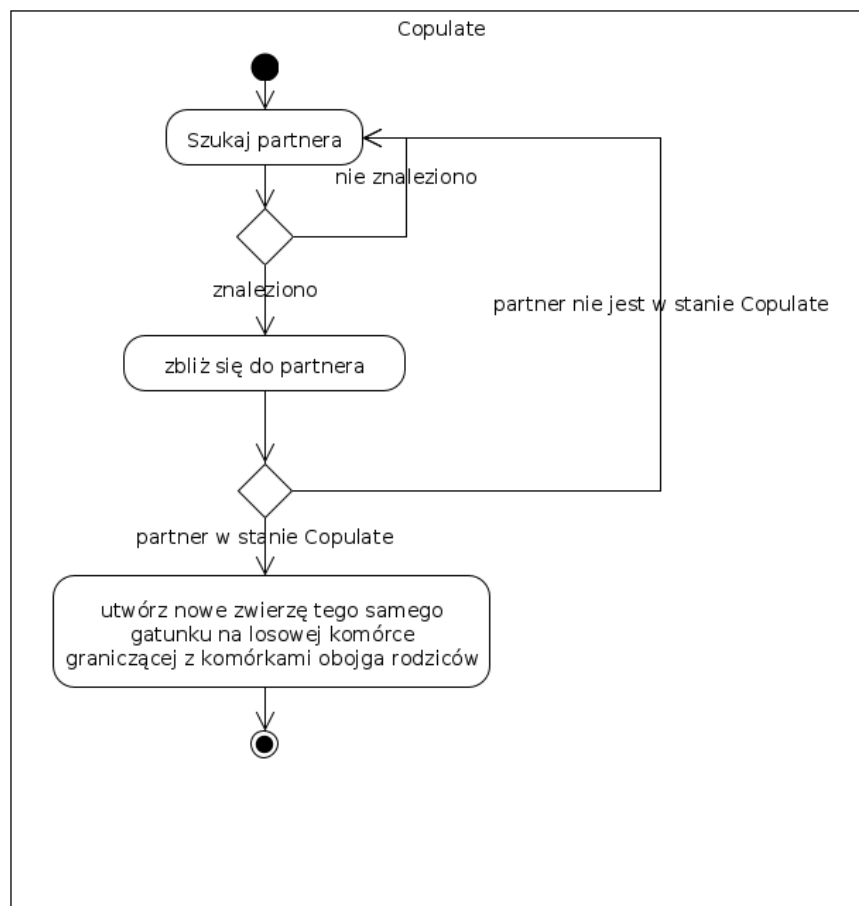


Figure 5: copulate

## Diagramy maszyny stanów

Zachowanie zwierząt będzie używało podejścia automatu ze stosem. Przykład: Podczas normalnego funkcjonowania (zwykła maszyna stanów) pewne zwierzęta będą celem polowań innych zwierząt. W takim wypadku, gdy zwierzę zauważy lub usłyszy drapieżnika znajdującego się w odległości mniejszej niż pewna określona odległość graniczna, zwierzę powinno wejść w stan ucieczki, a po jego zakończeniu (oddaleniu się od drapieżnika na odległość większą niż graniczna) powinno wrócić do stanu poprzedniego. Zostanie do tego wykorzystany stos; wejście w stan ucieczki będzie wepchnięciem go na stos, a wyjście z niego będzie wypchnięciem go ze stosu.

- Antelope

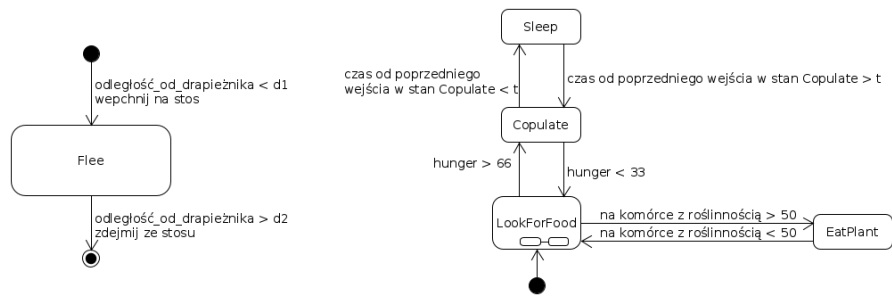


Figure 6: antelope

- Hippo

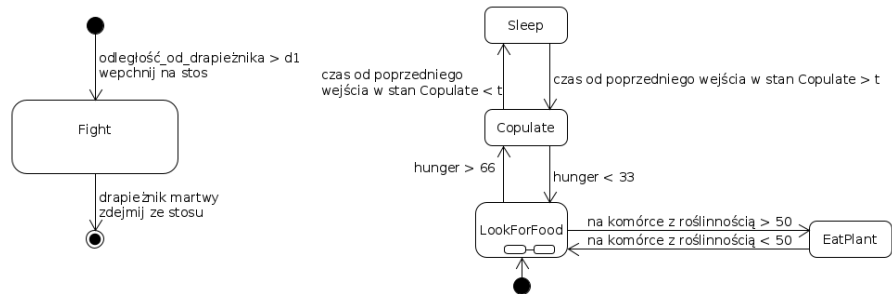


Figure 7: hippo

- Wolf
- Lion
- Crocodile

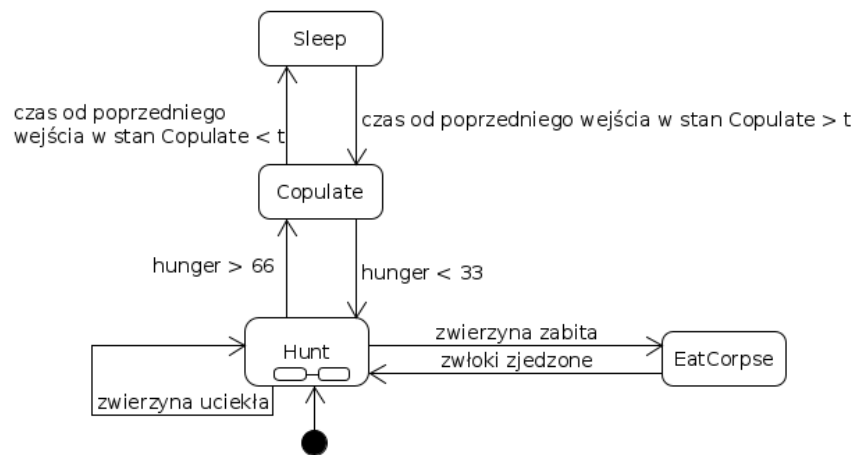


Figure 8: wolf

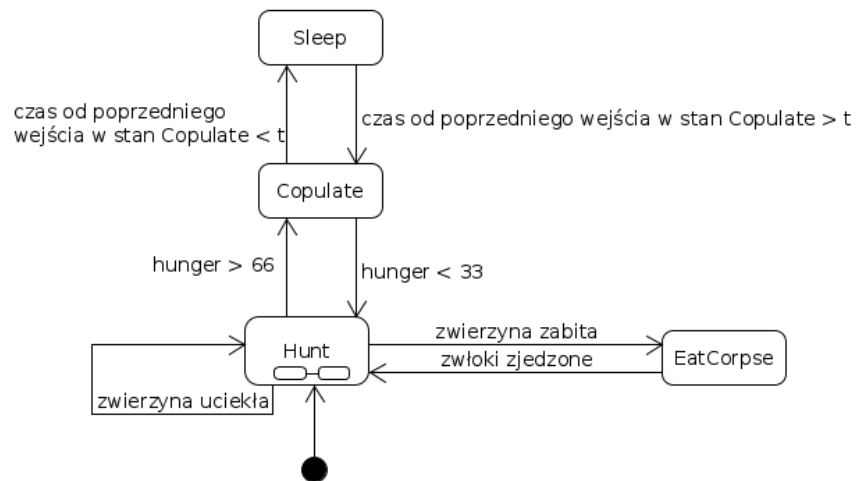


Figure 9: lion

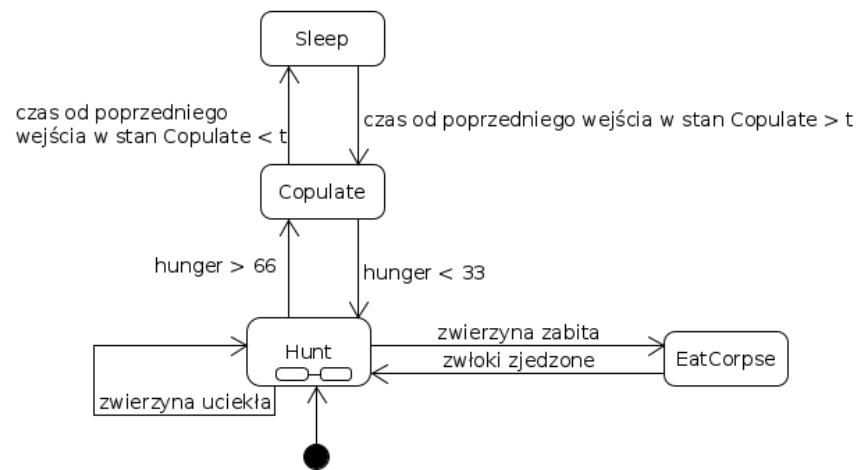


Figure 10: crocodile