# Projekt PO

# Etap V

# „Symulacja interakcji między zwierzętami”

Grupa 6:

Bartosz Hornicki

Tomasz Grochowski

# Spis narzędzi użytych przy tworzeniu projektu

* GitHub (<https://github.com/Horochov/pwrSymulatorZwierzat>)
* Algorytm OpenSimplexNoise (<https://gist.github.com/KdotJPG/b1270127455a94ac5d19>)
* Biblioteki języka Java
* Microsoft Word
* Narzędzie do tworzenia diagramów UML: (<https://online.visual-paradigm.com>)
* Dokumentacja Javy (<https://docs.oracle.com/en/java/>)
* Stack Overflow (<https://stackoverflow.com>)

# Wymagania funkcjonalne

1. Rozpoczęcie symulacji (użytkownik)
2. Ustawienie szybkości symulacji (użytkownik)
3. Przerwanie symulacji (użytkownik)
4. Poruszanie się zwierząt po siatce
5. Decyzja zwierząt o czynności

5.1. Poszukiwanie za pożywieniem

5.2. Atakowanie innego zwierzęcia

5.3. Ucieczka przed innym zwierzęciem

5.4. Swobodne poruszanie się

1. Dorastanie zwierząt
2. Zamiana zwierząt w padlinę po śmierci
3. Pola dostępne i zabronione dla zwierząt
4. Losowe zdarzenia

# Analiza czasownikowo-rzeczownikowa

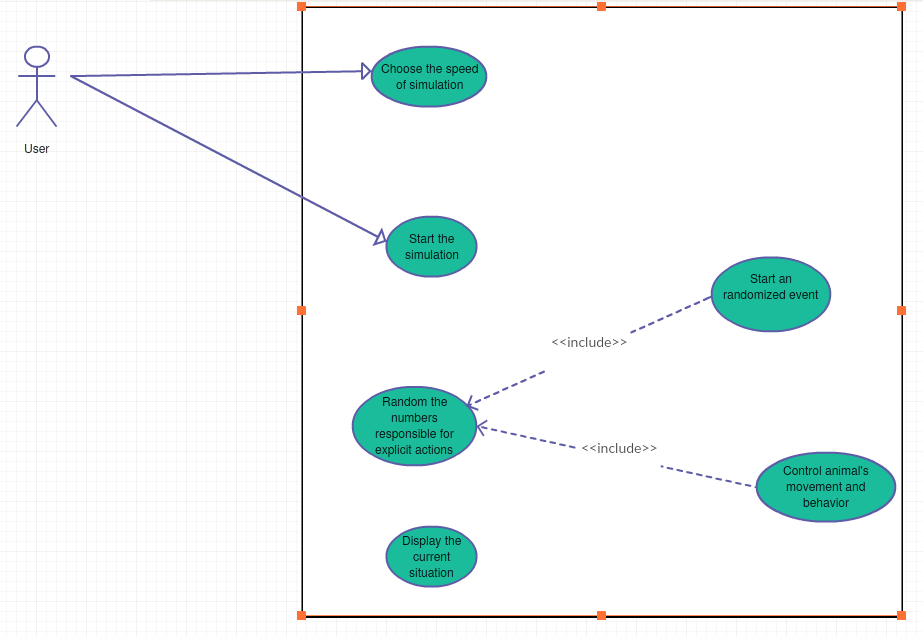
Celem projektu było stworzenie turowej symulacji zależności między roślinożercami (króliki) a drapieżnikami (wilki). Użytkownik może włączyć symulację, ustawić prędkość jej wykonywania i przerwać ją.

Zwierzęta poruszać się będą po terenie złożonym z kwadratowych pól. Zwierzę na podstawie swoich potrzeb zadecyduje jaką czynność wykonać - szukać za pożywieniem, atakować inne zwierzę bądź uciekać, swobodnie poruszać się lub odpoczywać. Każde zwierzę posiada wiek, który wpływa na jego statystyki. Zwierzę może umrzeć ze starości lub zostać zabite – zamienia się wtedy w padlinę, która jest pożywieniem.

Teren składa się z kwadratowych pól. Pola mogą być dostępne dla zwierząt, bądź nie, np. drzewo, kamień. Na polu może znajdować się surowiec - trawa, padlina lub woda.

W symulacji mogą wystąpić losowe zdarzenia mające wpływ na teren, grupę zwierząt lub pojedynczego osobnika.

# Diagram UML przypadków użycia



# Karty CRC

|  |  |
| --- | --- |
| Class name:Main | |
| Superclasses: | |
| Subclasses: | |
| Responsibilities:   * Contains main function * Manages user input * Keeps game running | Collaborators:   * Game |

|  |  |
| --- | --- |
| Class name: Game | |
| Superclasses: | |
| Subclasses: | |
| Responsibilities:   * Creates terrain * Manages animal placement | Collaborators:   * Animal * Tile * Textures * Position * OpenSimplexNoise |

|  |  |
| --- | --- |
| Class name: IAnimal interface | |
| Superclasses: | |
| Subclasses: Animal | |
| Responsibilities:   * Declares common animal functions | Collaborators: |

|  |  |
| --- | --- |
| Class name: AnimalStats | |
| Superclasses: | |
| Subclasses: | |
| Responsibilities:   * Stores information about animal statistics. | Collaborators: |

|  |  |
| --- | --- |
| Class name: Animal abstract | |
| Superclasses: IAnimal | |
| Subclasses: Wolf, Hare | |
| Responsibilities:   * constructs "animal" objects * generally exhibits animals capabilities | Collaborators:   * AnimalStats |

|  |  |
| --- | --- |
| Class name: Hare | |
| Superclasses: Animal | |
| Subclasses: | |
| Responsibilities:   * defines animal's statistics * defines animal's reactions and behaviour * defines animal's needs | Collaborators: |

|  |  |
| --- | --- |
| Class name: Wolf | |
| Superclasses: Animal | |
| Subclasses: | |
| Responsibilities:   * defines animal's statistics * defines animal's reactions and behaviour * defines animal's needs | Collaborators: |

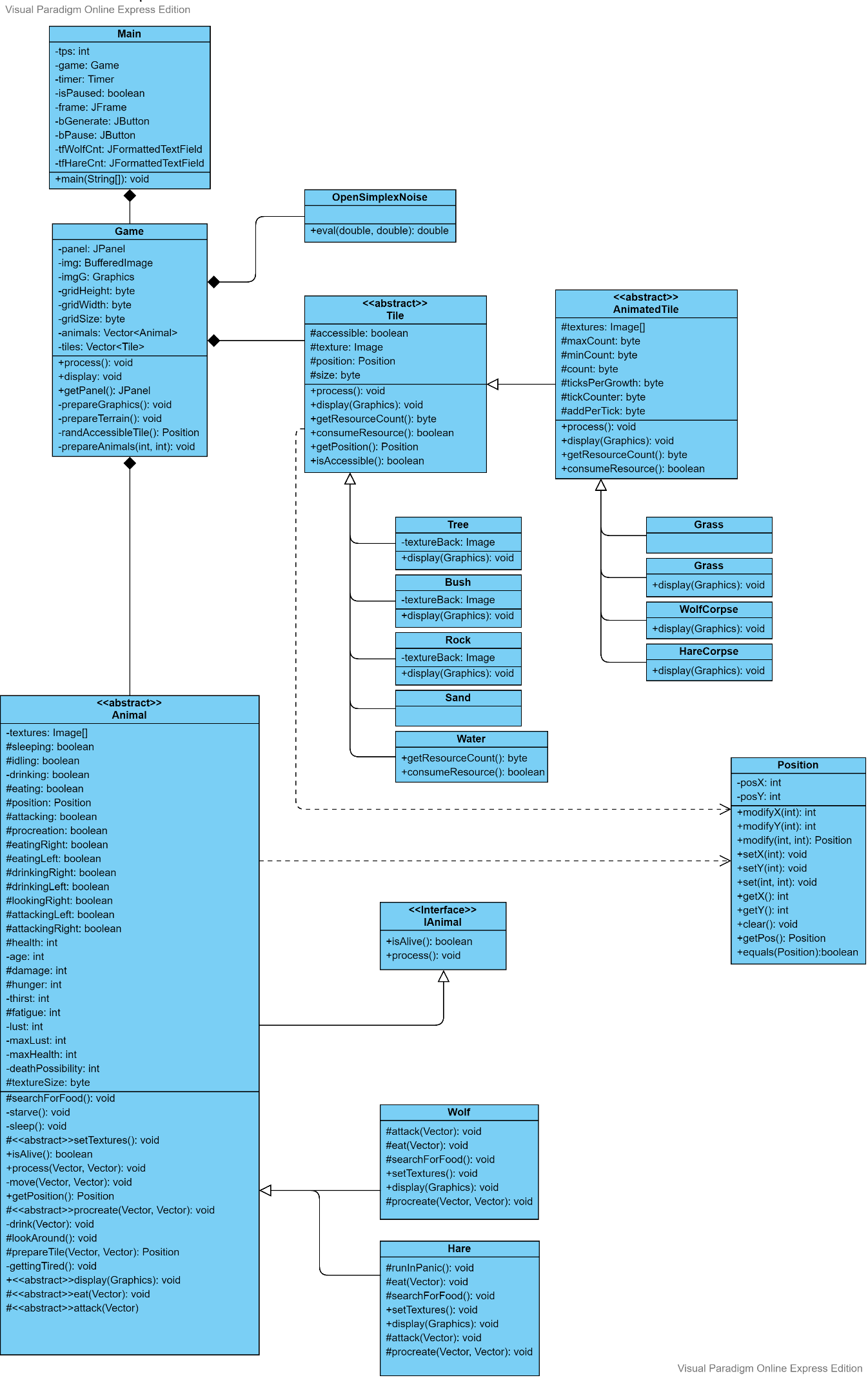
|  |  |
| --- | --- |
| Class name: Tile | |
| Superclasses: | |
| Subclasses: | |
| Responsibilities:   * displays the type of ground within * shows the resources specific tile contains * allows animals to move around | Collaborators:   * Position * Textures * ResourceType |

|  |  |
| --- | --- |
| Class name: OpenSimplexNoise | |
| Superclasses: | |
| Subclasses: | |
| Responsibilities:   * Generates random noise map | Collaborators: |

|  |  |
| --- | --- |
| Class name: Position | |
| Superclasses: | |
| Subclasses: | |
| Responsibilities:   * Encapsulates coordinates | Collaborators: |

|  |  |
| --- | --- |
| Class name: Textures | |
| Superclasses: | |
| Subclasses: | |
| Responsibilities:   * Stores terrain tile textures. | Collaborators: |

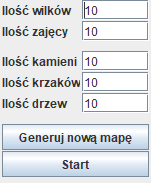
|  |  |
| --- | --- |
| Class name: ResourceType enumeration | |
| Superclasses: | |
| Subclasses: | |
| Responsibilities:   * Informs about resource type | Collaborators: |

Diagram UML klas

# Opis interfejsu użytkownika



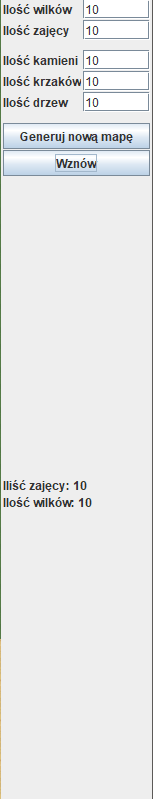
Wygląd okna. Większą część zajmuje plansza składająca się z terenów (trawy, piasku, wody), przeszkód (kamienie, krzaki, drzewa) i zwierząt (wilki i zające). Plansza generuje się z domyślnymi wartościami – po 10 przeszkód i zwierząt każdego rodzaju.



Liczbę zwierząt i przeszkód można zmienić poprzez wpisanie wartości w odpowiednie pole. W takim wypadku należy także kliknąć „Generuj nową mapę”.

Przycisk „Start” włącza symulację, zmieniając wyświetlany przez siebie tekst na „Pauza”. Po ponownym kliknięciu przycisku symulacja zostanie wstrzymana, a tekst przycisku zmieni się na „Wznów”.

Symulację można przerwać w dowolnym momencie także poprzez kliknięcie przycisku „Generuj nową mapę”. Jest to równoznaczne z przygotowaniem kolejnej symulacji do uruchomienia.



Wraz z uruchomieniem symulacji pojawią się dwa pola z liczbą żyjących zwierząt danego gatunku.

Symulacja zakończy się, gdy pozostanie tylko jedno zwierzę.



W momencie zakończenia symulacji liczba zwierząt zamienia się w powyższą wiadomość – w sytuacji gdy wygrał wilk wiadomość zostaje adekwatnie zmieniona.