

Symulacja Battle Royal

Autorzy:

- **Bartosz Futoma**
- **Mateusz Piętko**

1. Opis projektu

Oprogramowanie będzie symulowało Battle Royal, czyli walkę pomiędzy różnymi bohaterami. System będzie napisany w języku Java z wykorzystaniem narzędzia Maven. W naszym programie mapa będzie mogła mieć dowolny rozmiar zdefiniowany przez użytkownika. Liczba wygenerowanych bohaterów, którzy będą brać udział w walce również będzie dostosowywalna. Każdy bohater będzie miał modyfikowalne w trakcie trwania symulacji atrybuty opisujące ilość życia oraz wartość siły ataku. Atrybuty te będzie można modyfikować poprzez zadawanie obrażeń lub zbieranie z mapy specjalnych obiektów, które będą dodawały życie lub siłę ataku. Każdy obiekt będzie również opisywany przez swoją nazwę oraz szybkość poruszania się.

2. Analiza czasownikowo-rzeczownikowa

Symulacja będzie wywoływała odpowiednie metody dla wszystkich obiektów, tak aby obiekty te mogły reagować na otoczenie, poruszać się oraz żeby ich pozycja na mapie mogła być aktualizowana.

Zlokalizowane rzeczowniki: symulacja, obiekt, bohater, mapa, licznik statystyk

Zlokalizowane czasowniki: rysuje (stan symulacji oraz mape), porusza się (bohater), atakuje (bohater), rozwija się (bohater), umiera (bohater), znika (obiekt), pojawia się (obiekt), losuje (mape), zapisuje (stan symulacji), aktualizuje (statystyki symulacji), aktualizuje (stan mapy)

Połączenia rzeczowników z czasownikami:

Symulacja: rysuje (stan symulacji oraz mape), losuje (mape)

Obiekt: znika (obiekt), pojawia się (obiekt)

Bohater: porusza się (bohater), atakuje (bohater), rozwija się (bohater), umiera (bohater)

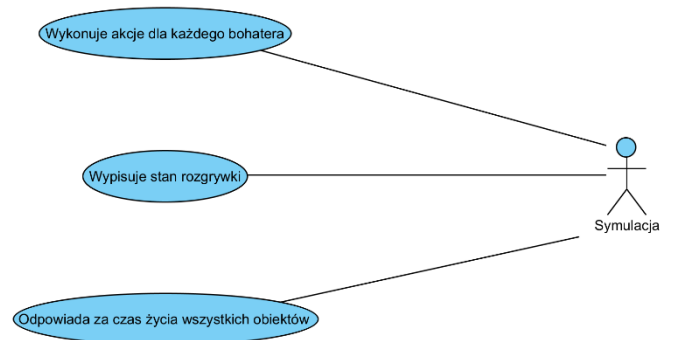
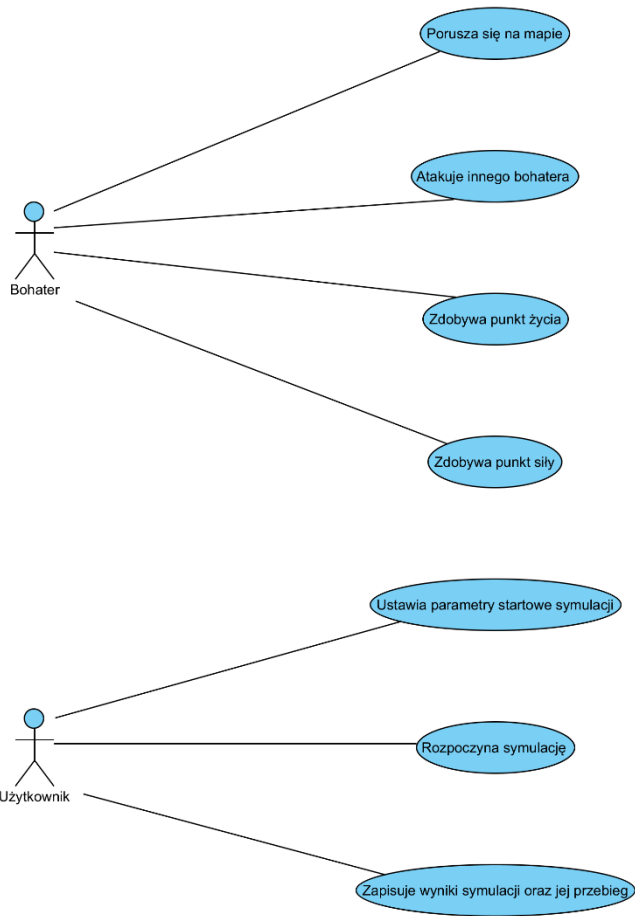
Mapa: aktualizuje (stan mapy)

Licznik statystyk: zapisuje (stan symulacji), aktualizuje (statystyki symulacji)

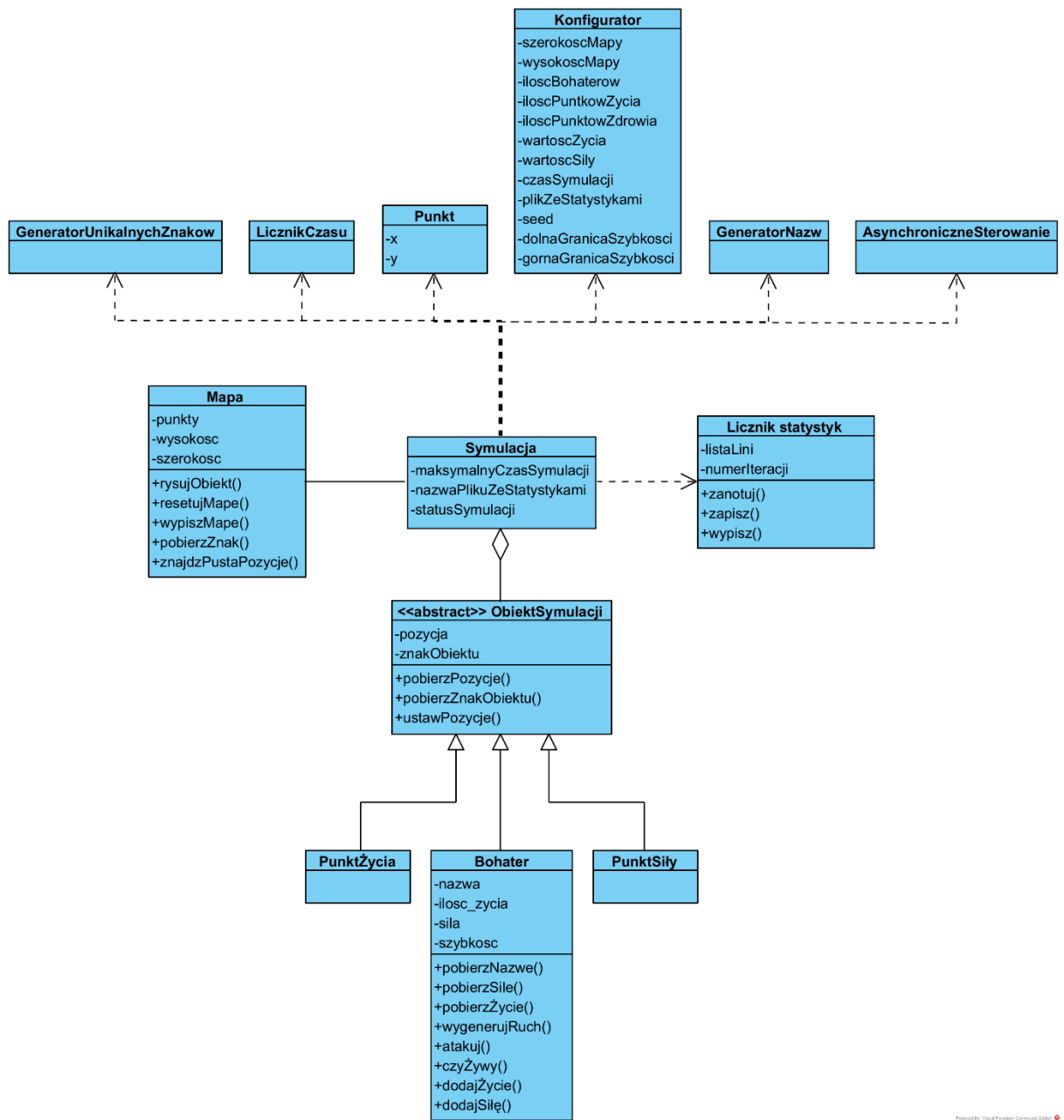
3. Karty CRC dla klas

Symulacja	Obiekt, Bohater Mapa, Licznik Statystyk
Rysuje stan map	
Losuje mape oraz obiekty	
Odpowiada za liczenie statystyk	
Odpowiada za ruchy bohaterów	
Odpowiada za generowanie obiektów	
Obiekt	Punkt życia, Punkt siły, Bohater
Aktualizuje statystyki bohatera	
Bohater	Mapa, Obiekt
Aktualizuje swoją pozycję na mapie	
Atakuje innego bohatera	
Zbiera obiekty	
Porusza się	
Mapa	Obiekt, Bohater
Przechowuje informacje na temat obiektów i bohaterów	
Rysuje mape	
Licznik statysyk	Symulacja
Ustawia początkowe wartości dla statystyk	
Aktualizuje statystyki w kolejnych iteracjach	
Zapisuje statystyki do pliku w wybranym formacie	

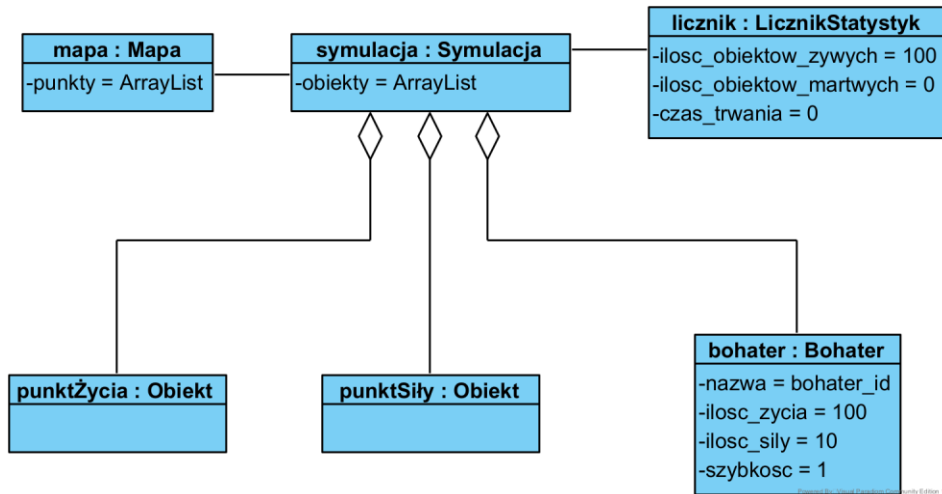
4. Diagram use-case



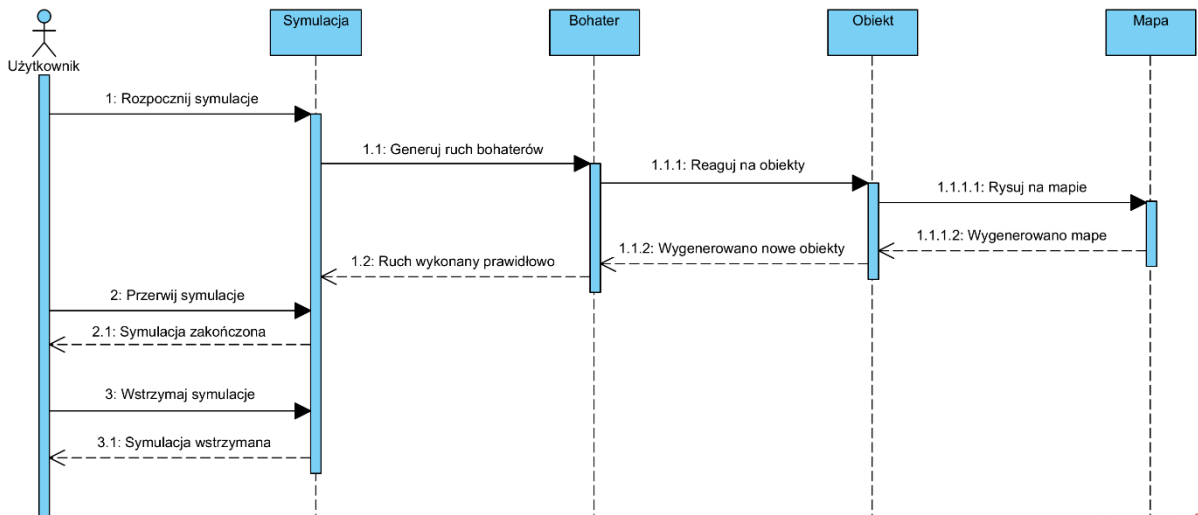
5. Diagram klas



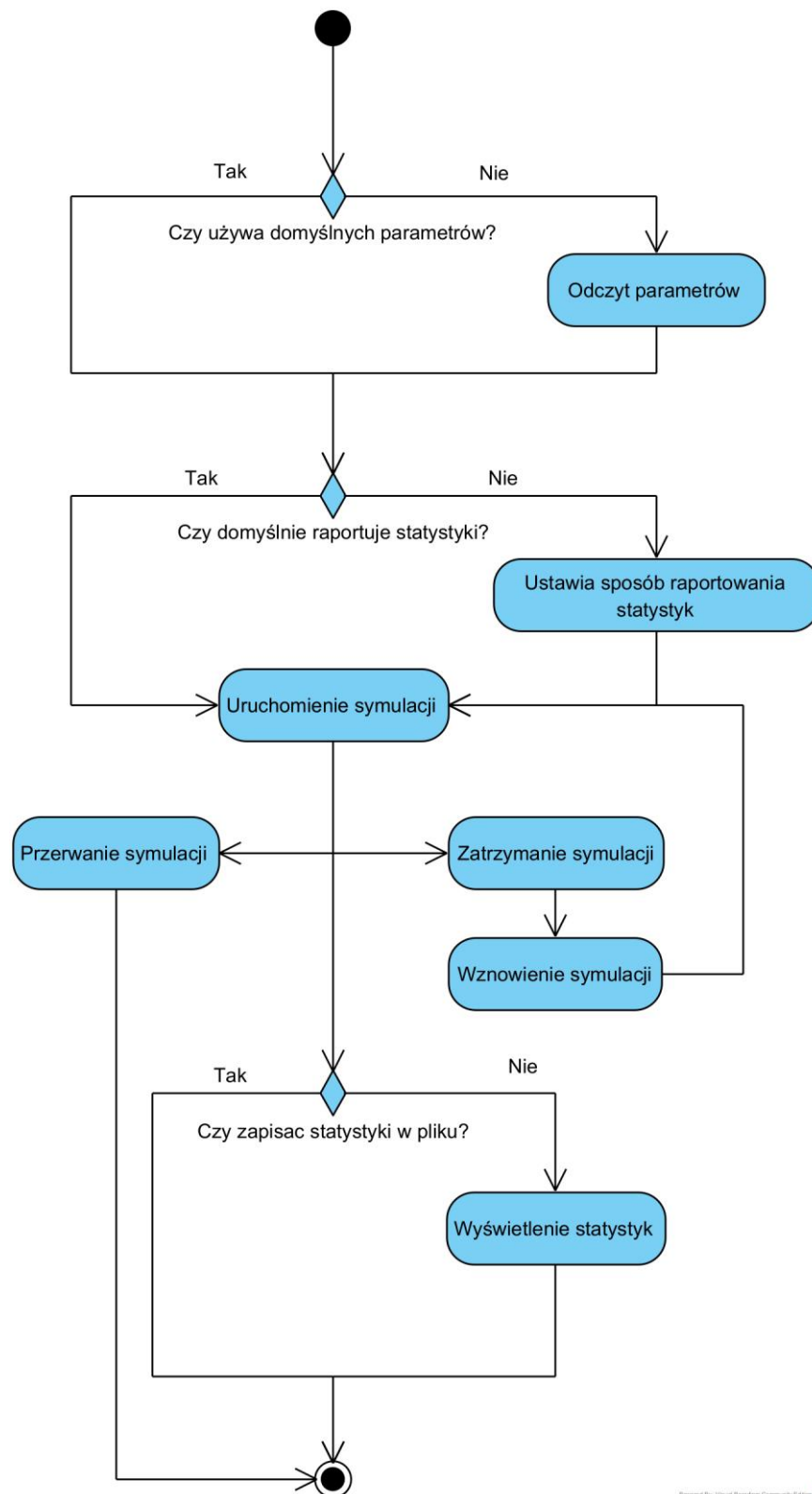
6. Diagram obiektów



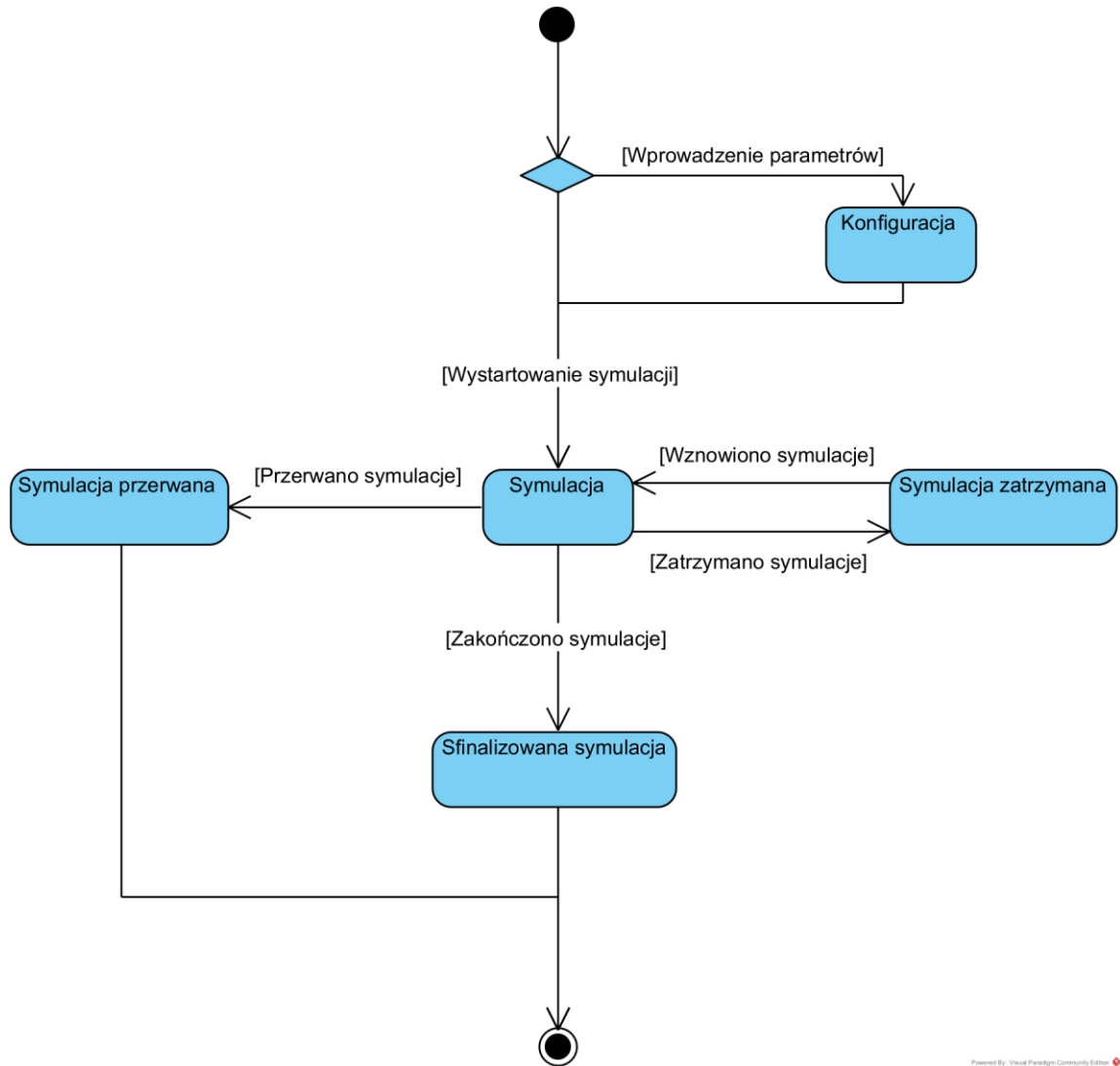
7. Diagram sekwencji



8. Diagram aktywności



9. Diagram stanów



10. Testowanie programu

Program należy uruchomić z konsoli za pomocą komendy `java -jar` podając ścieżkę do pliku „Main-1.0-jar-with-dependencies.jar”. Program uruchamia się w stanie `paused`, aby uruchomić symulację należy postępować zgodnie z tym co mówi menu programu. Przy uruchomieniu programu można dostarczyć opcjonalne przełączniki tj.:

- „-mw [int]” (map width) oznacza szerokość mapy
- „-mh [int]” (map height) oznacza wysokość mapy
- „-ip [int]” (init players) oznacza początkową ilość bohaterów
- „-ih [int]” (init health points) oznacza początkową ilość punktów życia
- „-is [int]” (init strength points) oznacza początkową ilość punktów siły
- „-lb [int]” (lower bound) oznacza dolną granicę dla szybkości bohatera
- „-up [int]” (upper bound) oznacza górną granicę dla szybkości bohatera
- „-sd [int]” (seed) oznacza wartość seed dla generatora liczb losowych
- „-h [int]” (init health value) początkowa ilość życia każdego bohatera
- „-s [int]” (init strength value) początkowa ilość siły każdego bohatera
- „-t [int]” (time) czas trwania symulacji
- „-sf [string]” (stats file) nazwa pliku do którego zostaną zapisane statystyki.

Przykład wywołania:

```
Java -jar Main-1.0-jar-with-dependencies.jar -mw 40 -mh 30 -ip 45 -ih 10 -is 10 -up 2 -h 100 -s 10 -t 360 -sf stats.txt
```

W trakcie trwania symulacji można wybierać opcje z menu, które należy zatwierdzić enterem! Może się również zdarzyć, że po zakończeniu symulacji trzeba będzie nacisnąć enter, aby program ostatecznie się zakończył (jest umieszczony stosowny komunikat).

11. Javadoc

Javadoc jest dostępny w folderze Javadoc