Projekt Programowanie Obiektowe

Kacper Jaskulski

Kwiecień-Czerwiec 2019

ETAP I

1 Skład grupy

Lider:Kacper Jaskulski

2 Język programowania

Implementacja projektu będzie wykonywana w Javie .

3 Opis projektu nr 1

Symulacja miasta od czasów starożytnych do nowocześności.Na mapie będą znajdować się 4 miasta.Każde z miast będzie miało statystyki: Ilość mieszkańców, poziom wojska, poziom nauki, poziom kultury, poziom gospodarki.Użytkownik przed rozpoczęciem symulacji wybiera w jaki sposób rozwija się każde z miast i jakie decyzje ma podejmować. Każda z decyzji będą wykonywane co turę(ok.10 lat).Sama zaś symulacja będzie trwać około 4 tysięcy lat. Decyzje będą dotyczyć poziomu rozwoju każdej ze statystyk danego miasta w określonej liczbie tur oraz tego czy ma ono zaatakować inne miasto by obniżyć jego statystyki lub je przejąć za pomocą swojego poziomu kultury.Każde z miast będzie musiało się rozwijać równomiernie pod względem każdej ze statystyk, gdyż niski poziom wojska może umożliwić innym miastom zniszczenie go, niski poziom gospodarki może prowadzić do głodu,niski poziom nauki do powstawania chorób, natomiast niski poziom kultury do przejęcia miasta przez wroga.W każdym z wieków(10 tur) miasto będzie produkowało odpowiednia ilosc punktow, zależną od przewagi rozwoju nad innymi miastami,ktore procentowo(stosunek podziału będzie wybrany przez użytkownika) zostaną podzielone między poszczególne atrybuty miasta).Pod koniec symulacji program wypisze do pliku podsumowanie rozwoju każdego z miast co wiek oraz w całym okresie symulacji.

4 Opis projektu nr 2

Symulacja transportu na morzu karaibskim.Trwa 30 lat.Każda z tur to miesiąć W symulacji biorą udział 3 frakcje: Hiszpanie, Anglicy oraz Piracii. Użytkownik ustanawia szlaki transportowe między miastami oraz ilosc towarów na statkach.Każdy z towarów ma określoną wartość i wagę.Ilośc i waga towarów wpływa na szybkość transportu surówców na statku.Może dochodzić do walk między statkami,które płyną w niedalekiej odległości od siebie. Kryterium walki statku określa się na podstawie tego, który z nich ma większe doświadczenie w przewozie towarów, walce oraz który z nich jest szybszy.Wygrany statek zatapia statek, który przegrał w bitwie co powoduje stworzenie przed daną frakcję statku słabszego na tej samej trasie, za który frakcja musi zapłacić.Każda ze frakcji ma taką samą ilośc statków,jednak to użytkownik decyduje o ich prędkośći ze względu na przewożone surowce oraz ich ilość.Symulacja na koniec porównuje ilość złota zarobionego przez każdą ze frakcji oraz ich roczne porównanie ich zarobków i eksportuje je do pliku tekstowego

ETAP II

5 Analiza czasownikowo-rzeczownikowa

Projekt symulacji transportu surowców przez statki pracujące dla danych nacji. Symulacja odbywa się na kwadratowej mapie wymiarze o *size* na której znajdują się miasta,których pozycja opisana jest przez parametry x oraz y. Statki poruszaja się między miastami po mapie traktując ją jako układ współrzędnych sprzedając zawartość ładownii. . W miastach dostają one nowe cele i uzupełniają ładownie.Statki mają statystyki: prędkość, atak,pojemność ładowni.

5.1 Zachowanie statków

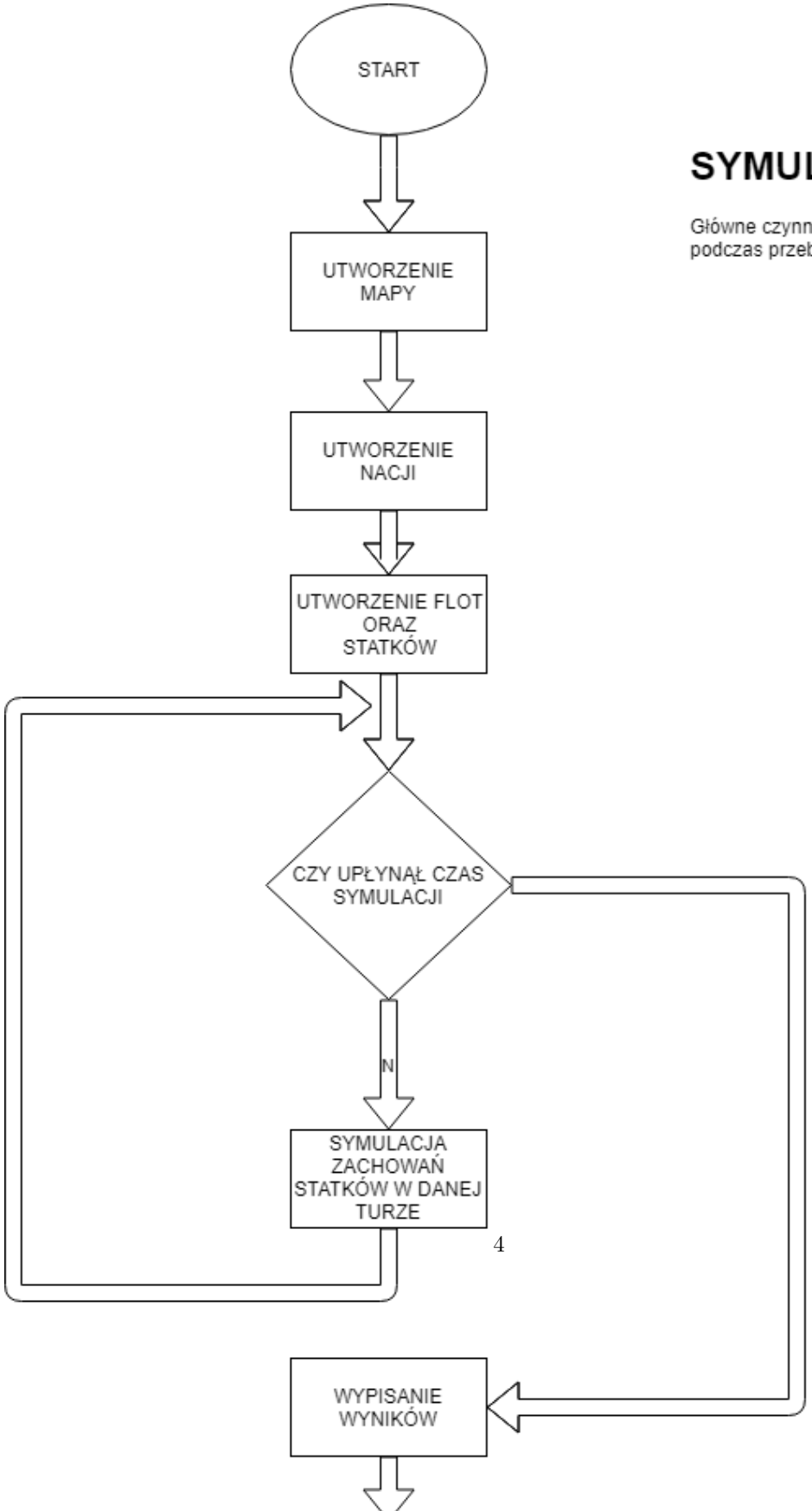
- Każdy ze statków płynie do swojego celu po najkrótszej prostej drodze.
- Jeśli statki znajdują się na tych samych współrzędnych odbywają walkę.Wygrywa statek z wyższą wartością ataku. Przegrany statek traci swój ładunek.
- Po dotarciu do celu stateksprzedaje zawartość swojej ładowni po cenie obowiązującej w mieście.
- W miescie statek uzupełnia ładownię oraz otrzymuje nowy cel.

5.2 Parametry symulacji

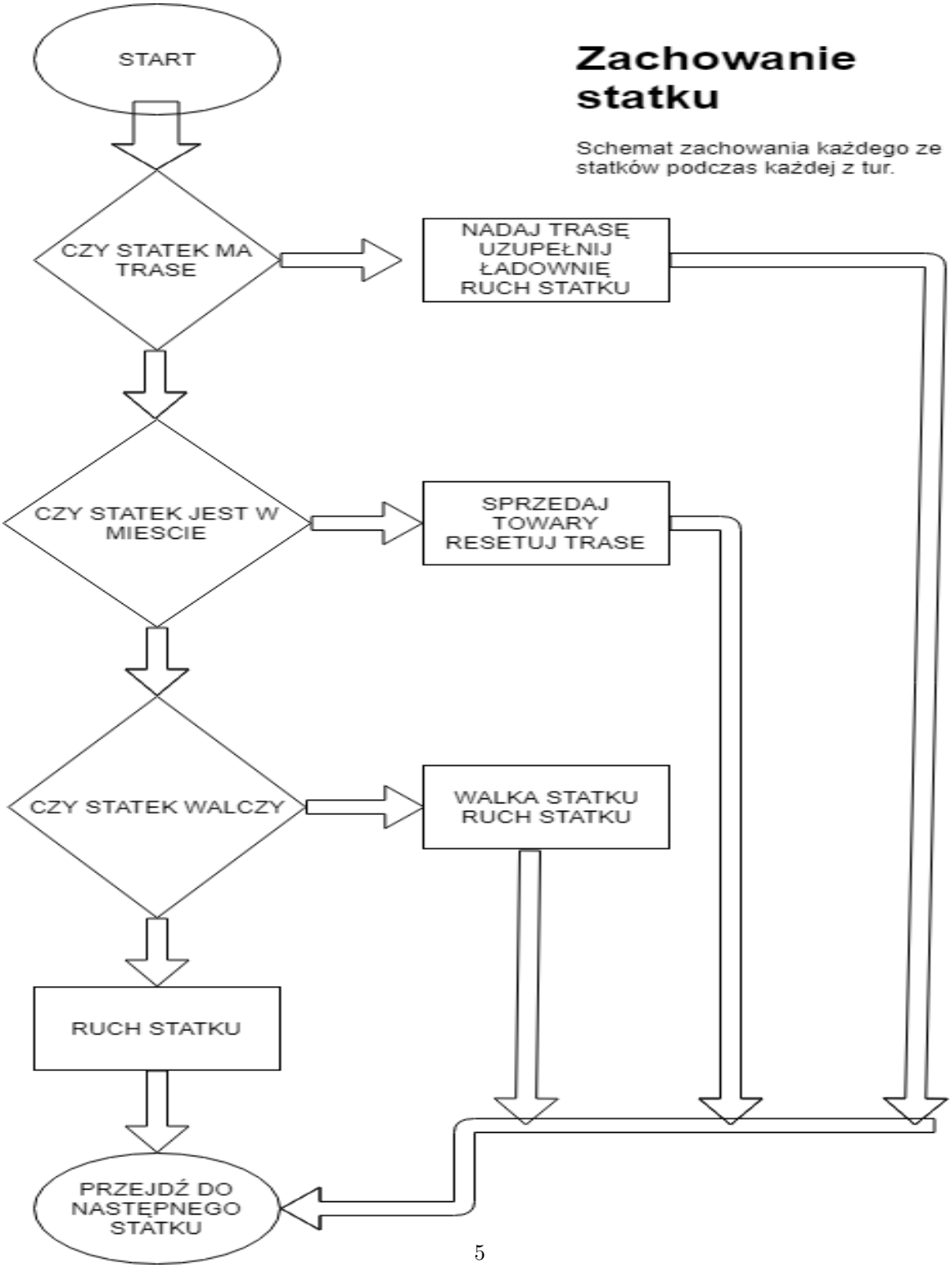
- Wielkość mapy
- Ilość miast
- Statystyki miast
- Ilość nacji
- Ilość statków
- Statystyki statków
- Czas symulacji(liczba "tickow")

6 Diagramy

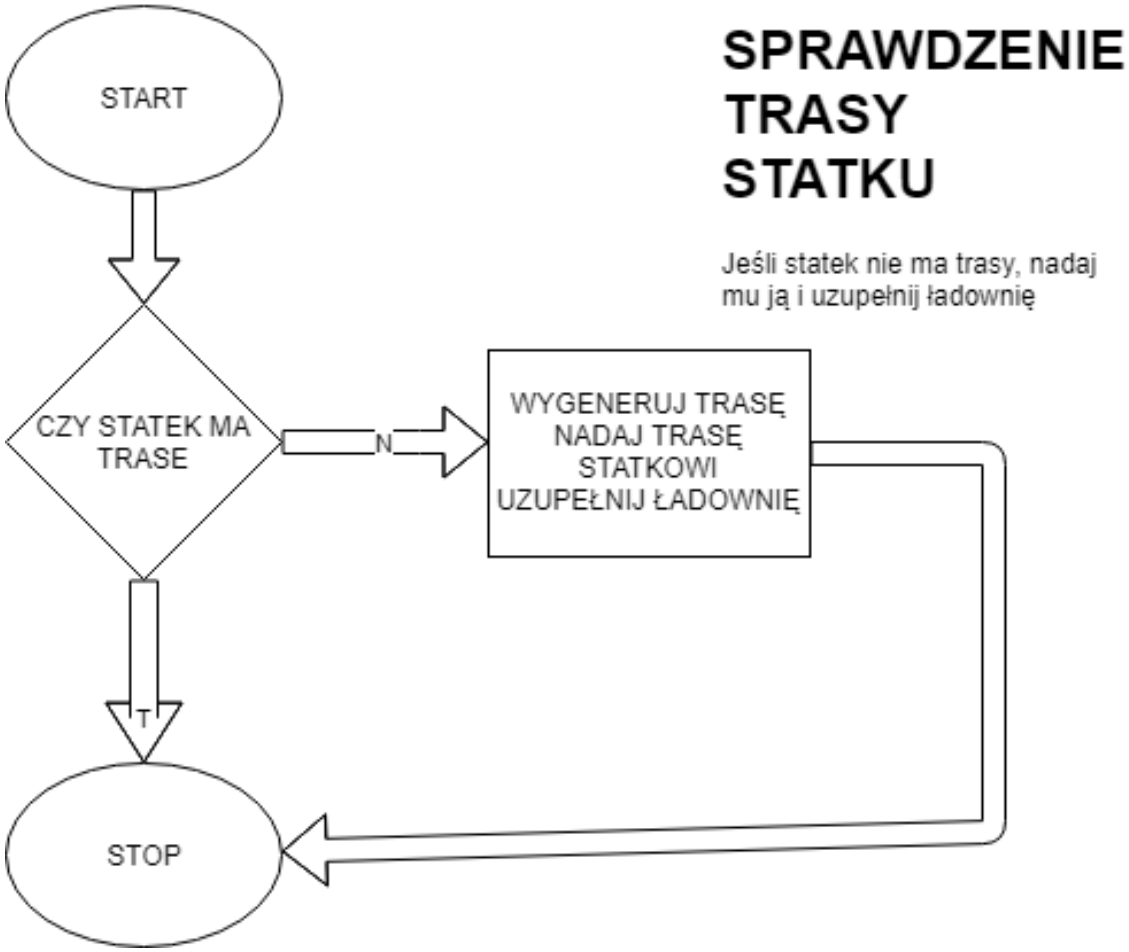
6.1 Diagram przebiegu symulacji



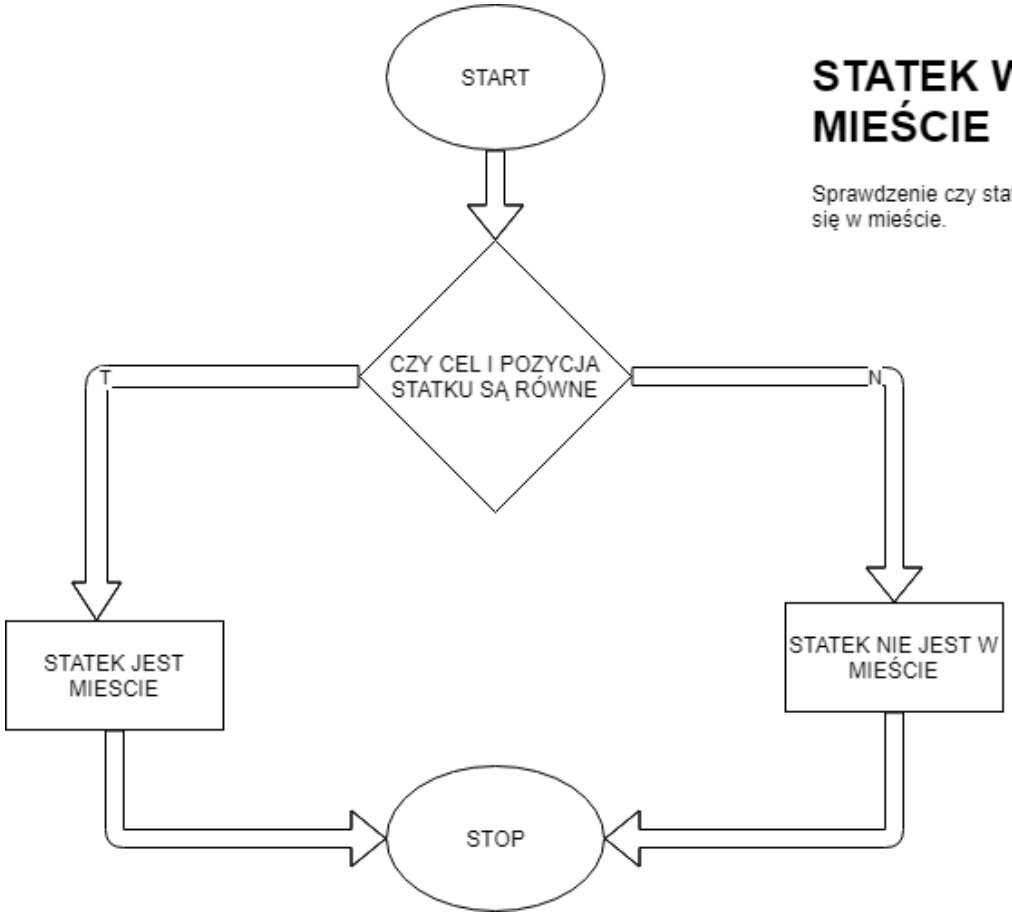
6.2 Diagram zachowań statków w każdej z tur



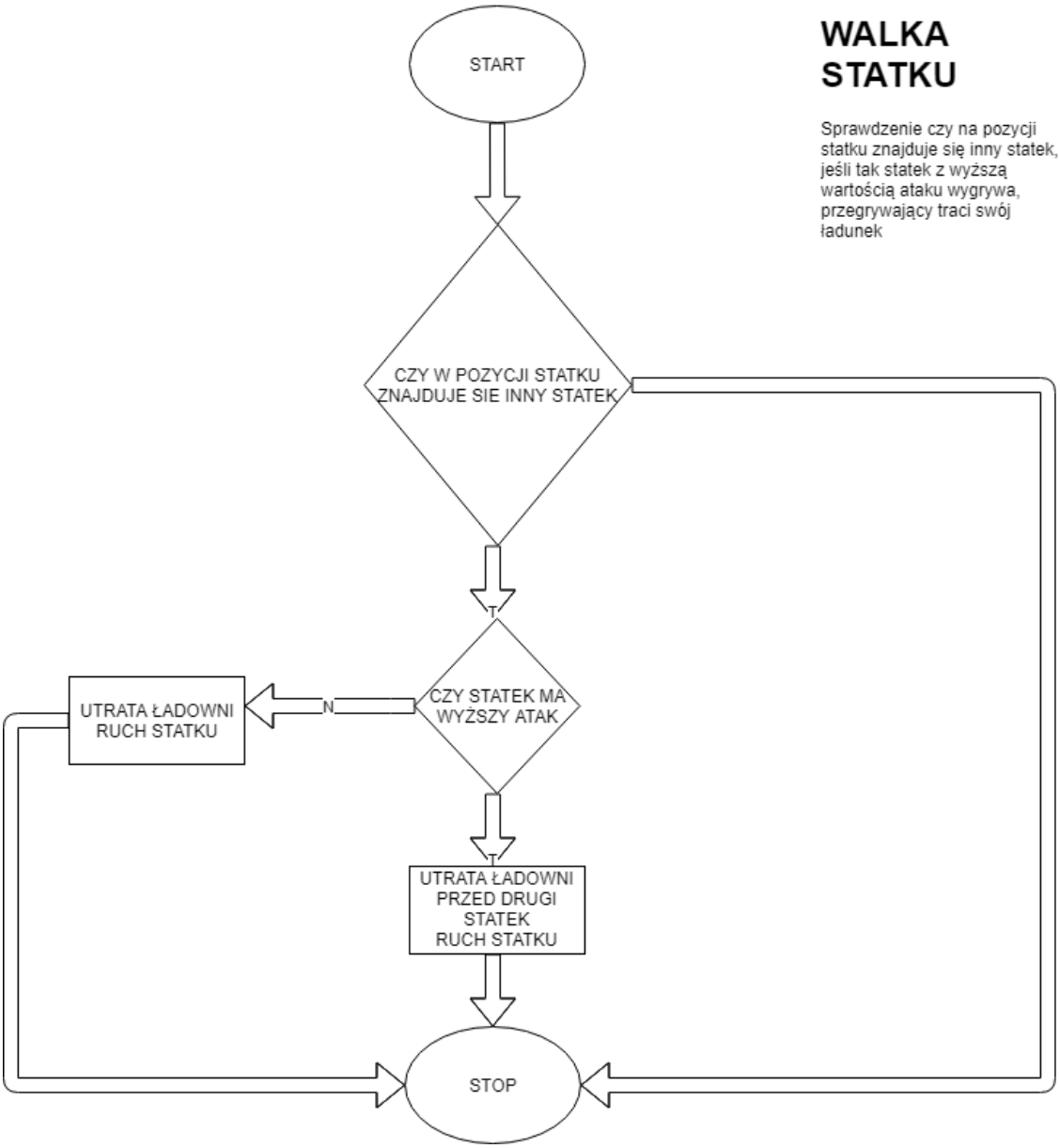
6.3 Diagram sprawdzenia czy statek ma trase



6.4 Diagram sprawdzenia czy statek jest w miescie



6.5 Diagram sprawdzenia czy statek walczy



7 Karty CRC

Classname:Map	
Superclass:*none* Subclass(es):*none*	
Responsibilities:	Collaboration:
Zawiera listę miast i nacji Udostępnia pozycję miast Udostępnia nacje	Nation City

Classname:Nation	
Superclass:*none* Subclass(es):*none*	
Responsibilities:	Collaboration:
Zawiera flotę Tworzy flotę Zawiera skarbiec Udostępnia flotę	Fleet

Classname:Fleet	
Superclass:*none* Subclass(es):*none*	
Responsibilities:	Collaboration:
Zawiera flotę statków Udostępnia statki	Ship

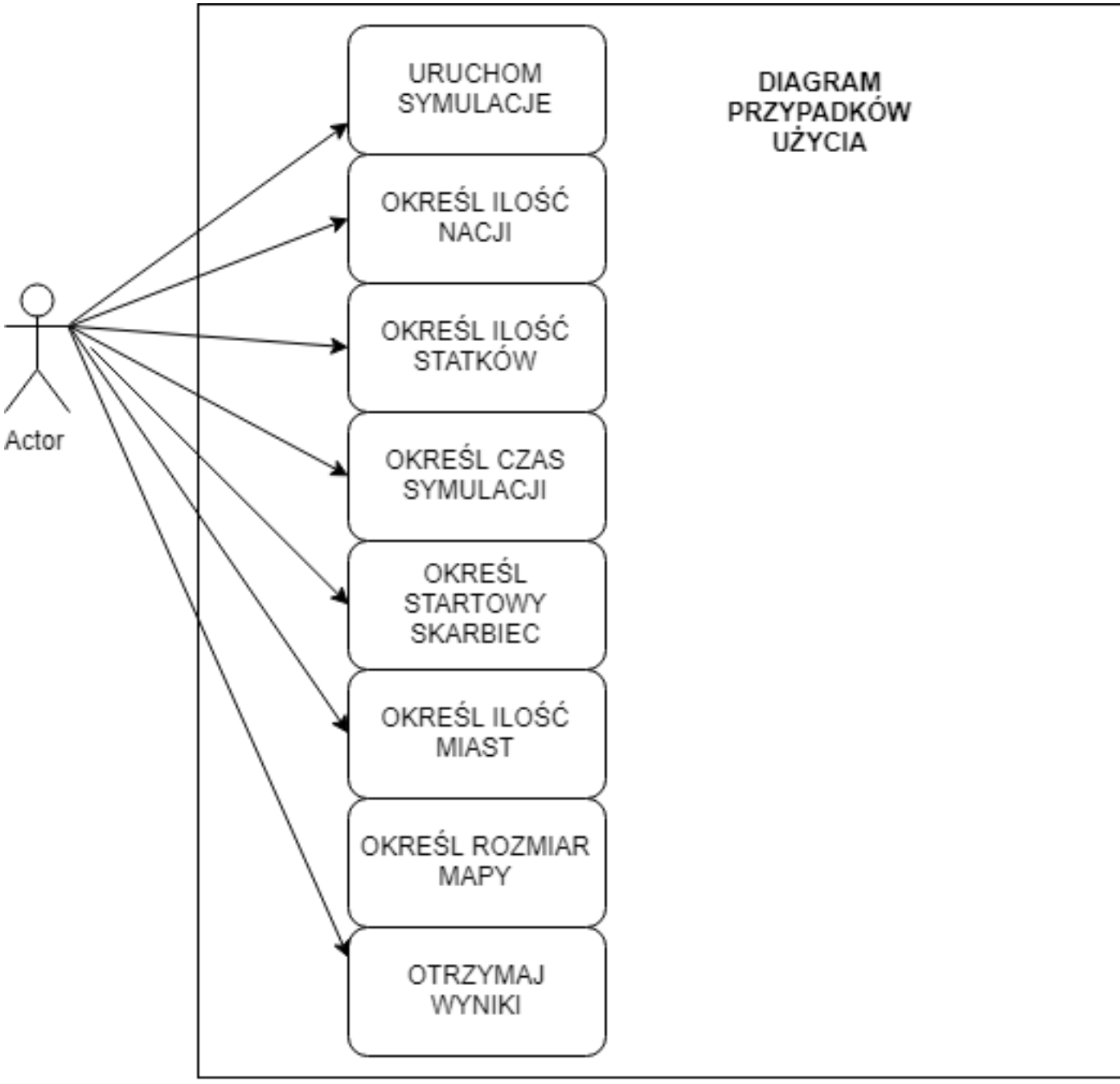
Classname:Ship	
Superclass:*none* Subclass(es):*none*	
Responsibilities:	Collaboration:
Zawiera informację o statkach Dodaje i usuwa trasę statków Udostępnia informację o statkach Zawiera metody operujące na statkach Sprawdza czy statek ma trasę Sprawdza czy statek jest w miescie Sprawdza czy statek walczy Przesuwa statki Zmienia zawartość ekwipunku i trasę statków	 Mapa Nation Ship

Classname:Simulation	
Superclass:*none* Subclass(es):*none*	
Responsibilities:	Collaboration:
Tworzy obiekty na podstawie losowych statystyk Zarządza symulacją	Map Nation City Ship Randomize

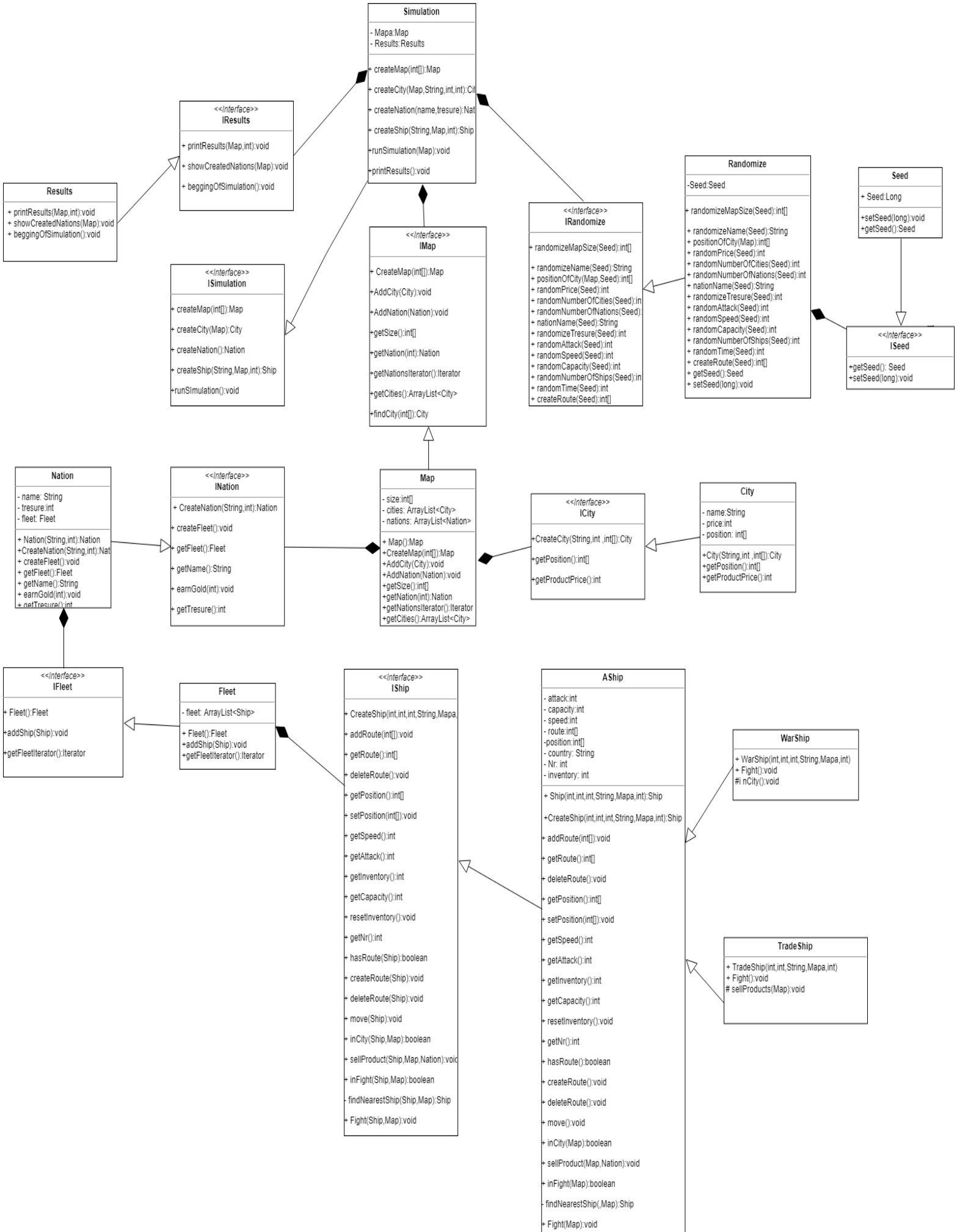
Classname:Randomize	
Superclass:*none* Subclass(es):*none*	
Responsibilities:	Collaboration:
Tworzy losowe statystyki obiektów	Map Simulation

Classname:Results	
Superclass:*none* Subclass(es):*none*	
Responsibilities:	Collaboration:
Produkuje wyniki i zapisuje do pliku	Map Nation City Ship

8 Diagram przypadków użycia



9 Diagramy klas



ETAP III

Implementacja w kodzie
https://github.com/anachim5/43ttjhj-

ETAP IV

Implementacja w kodzie, javadocs, finalny wygląd
https://github.com/anachim5/43ttjhj-