Programowanie obiektowe - projekt

Symulacja rozwoju epidemii

Dokumentacja

Spis treści

1 Tematyka projektu **1**

2 Skład grupy projektowej **1**

3 Opis zadania symulacji w języku naturalnym **2**

4 Karty CRC poszczególnych klas **2**

5 Diagram przypadków użycia **8**

6 Diagram maszyny stanów **8**

7 Diagram klas **9**

8 Diagram obiektów **10**

9 Analiza czasownikowo-rzeczownikowa **11**

10 Diagramy sekwencji **12**

11 Uwagi **15**

1 Tematyka projektu

Program przedstawiający rozwój epidemii, pozwalający na zadanie jak największej liczby warunków początkowych.

2 Skład grupy projektowej

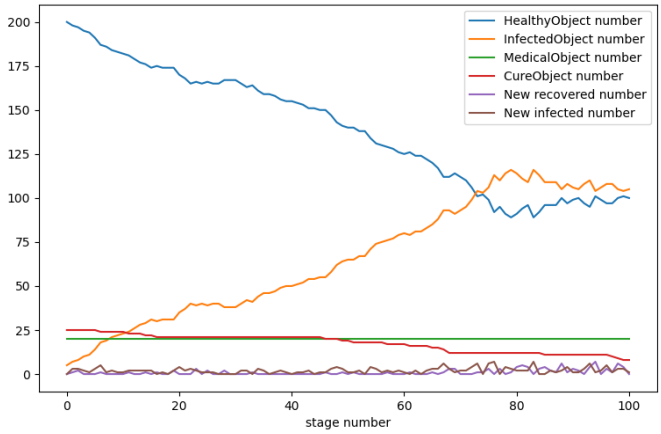
- Damian Gnieciak
- Miłosz Siemiński

Link do repozytorium: `https://github.com/Damian0401/ProjektPO`

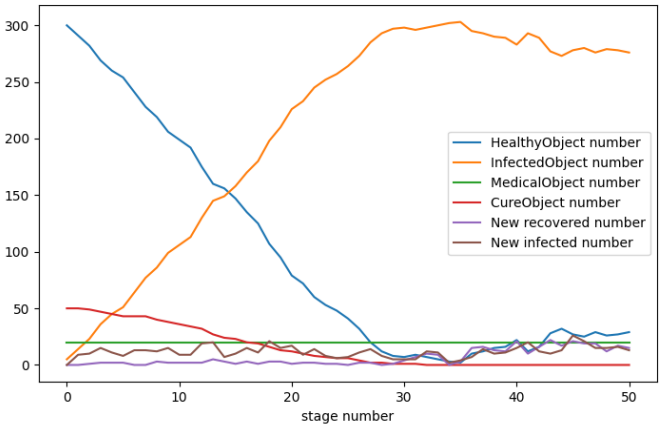
3 Opis zadania symulacji w języku naturalnym

Program przedstawia rozwój epidemii na dwuwymiarowej mapie, o dowolnej wielkości. Na początku symulacji na mapie zostają rozmieszczone w losowy sposób 4 rodzaje klas: zdrowi, zakażeni, lekarze oraz lekarstwa. Podczas każdej epoki symulacji zachodzą różne interakcje pomiędzy obiektami, np. lekarze leczą zakażonych oraz zostają poruszone w losowy sposób odpowiednie obiekty (obiekty klas dziedziczących po bazowej klasie lekarstwa nie mają możliwości poruszania się). Podczas każdej epoki zbierane są statystyki, które po pomyślnym zakończeniu całej symulacji zostają zapisane do pliku *.csv z nazwą odpowiadającą dacie oraz godzinie zakończenia symulacji.

Dane z przykładowych symulacji:



Rysunek 1: 100 epok, szansa za zakażenie: 10%, szansa na uleczenie lekarstwem: 50%



Rysunek 2: 50 epok, szansa za zakażenie: 50%, szansa na uleczenie lekarstwem: 50%

4 Karty CRC poszczególnych klas

Abstract	**ACureObject**	Superclasses: ASimulationObject Subclasses: Cure, DefectedCure
• get recovery chance		

Abstract	**AHealthyObject**	Superclasses: ASimulationObject, IMove Subclasses: HealthyHuman, ImmuneHealthyHuman
• check if object is infected		• AInfectedObject

Abstract	**AInfectedObject**	Superclasses: ASimulationObject, IMove Subclasses: InfectedHuman, MutatedInfectedHuman
• get infect chance • check if object is cured		• ACureObject

Abstract	**AMap**	Superclasses: JPanel Subclasses: Map
• carry out next stage • move objects • get stats		• ACureObject • AInfectedObject • AHealthyObject • AMedicalObject

Abstract	**AMedicalObject**	Superclasses: ASimulationObject, IMove Subclasses: MedicalHuman, InexperiencedMedicalHuman
• check if cure is successful		• AInfectedObject

	Statistics	Superclasses: Subclasses:
• store statistics • get string with statistics		

Abstract	Superclasses: ACureObject, AHealthyObject, AInfectedObject, AMedicalObject **ASimulationObject** Subclasses:
• get x position • get y position • set x position • set y position • check if objects are equal • generate hash code	

	Superclasses: ACureObject **Cure** Subclasses:
• get recovery chance • get x position • get y position • set x position • set y position	

	Superclasses: ACureObject **DefectedCure** Subclasses:
• get decreased recovery chance • get x position • get y position • set x position • set y position	

	Superclasses: JFrame **Epidemic** Subclasses:
• start simulation • save stats • create map	• Statistics • Date • SimpleDateFormat • File • StackFile

HealthyHuman	Superclasses: AHealthyObject Subclasses:
• check if object is infected • get move range • get x position • get y position • set x position • set y position	• RandomGenerator • Math • AInfectedObject

ImmuneHealthyHuman	Superclasses: AHealthyObject Subclasses:
• check if object is infected (with decreased chance) • get move range • get x position • get y position • set x position • set y position	• RandomGenerator • Math

Interface IMove	Superclasses: Subclasses:
• get move range	

InexperiencedMedicalHuman	Superclasses: AMedicalObject Subclasses:
• get move range • check if cure is successful (with decreased chance) • get x position • get y position • set x position • set y position	• RandomGenerator • Math • AInfectedObject

InfectedHuman	Superclasses: AInfectedObject Subclasses:
• get infect chance • get move range • get x position • get y position • set x position • set y position • check if object is cured	• RandomGenerator • Math • ACureObject

Interface IObjectedFactory	Superclasses: Subclasses: ObjectFactory
• create infected object • create healthy object • create cure object • create medical object	• ACureObject • AHealthyObject • AInfectedObject • AMedicalObject

Map	Superclasses: AMap Subclasses:
• carry out next stage • move objects • get stats	• ACureObject • AHealthyObject • AInfectedObject • AMedicalObject

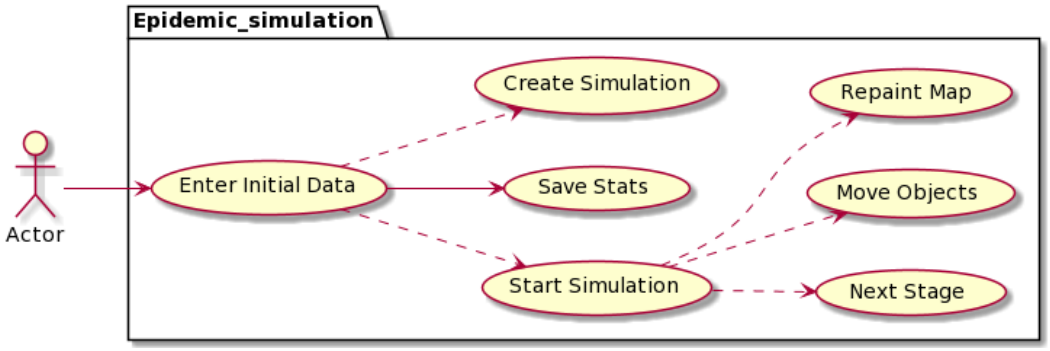
MedicalHuman	Superclasses: AMedicalHuman Subclasses:
• get move range • check if cure is successful • get x position • get y position • set x position • set y position	• Math • AInfectedObject

MutatedInfectedHuman	Superclasses: AInfectedObject Subclasses:
• get increased infect chance • get increased move range • get x position • get y position • set x position • set y position • check if object is cured	• RandomGenerator • Math • ACureObject

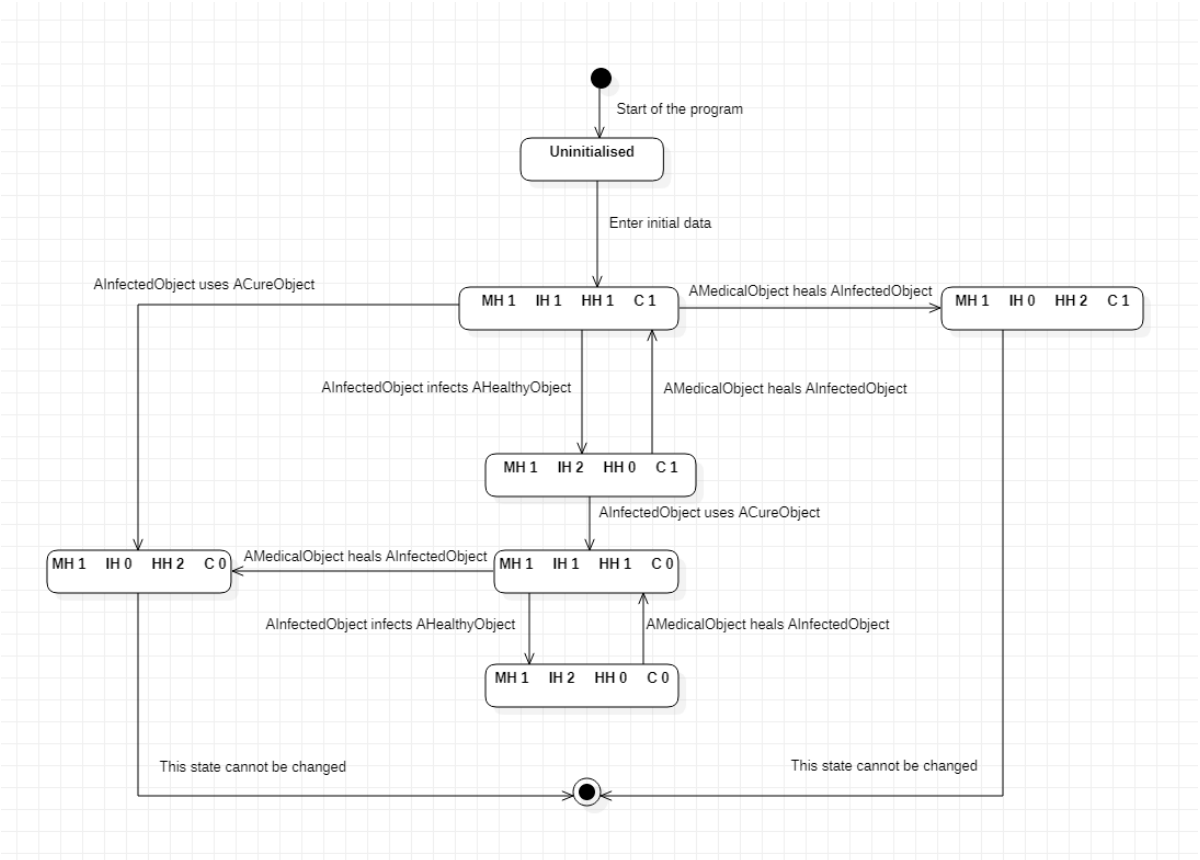
ObjectFactory	Superclasses: IObjectFactory Subclasses:
• create infected humans • create healthy humans • create cure objects • create medical humans	• ACureObject • AHealthyObject • AInfectedObject • AMedicalObject • RandomGenerator • MutatedInfectedHuman • MedicalHuman • InfectedHuman • InexperiencedMedicalHuman • ImmuneHealthyHuman • HealthyHuman • DefectedCure • Cure

RandomGenerator	Superclasses: Subclasses:
• get random position • get random chance • get random move	• Random

5 Diagram przypadków użycia

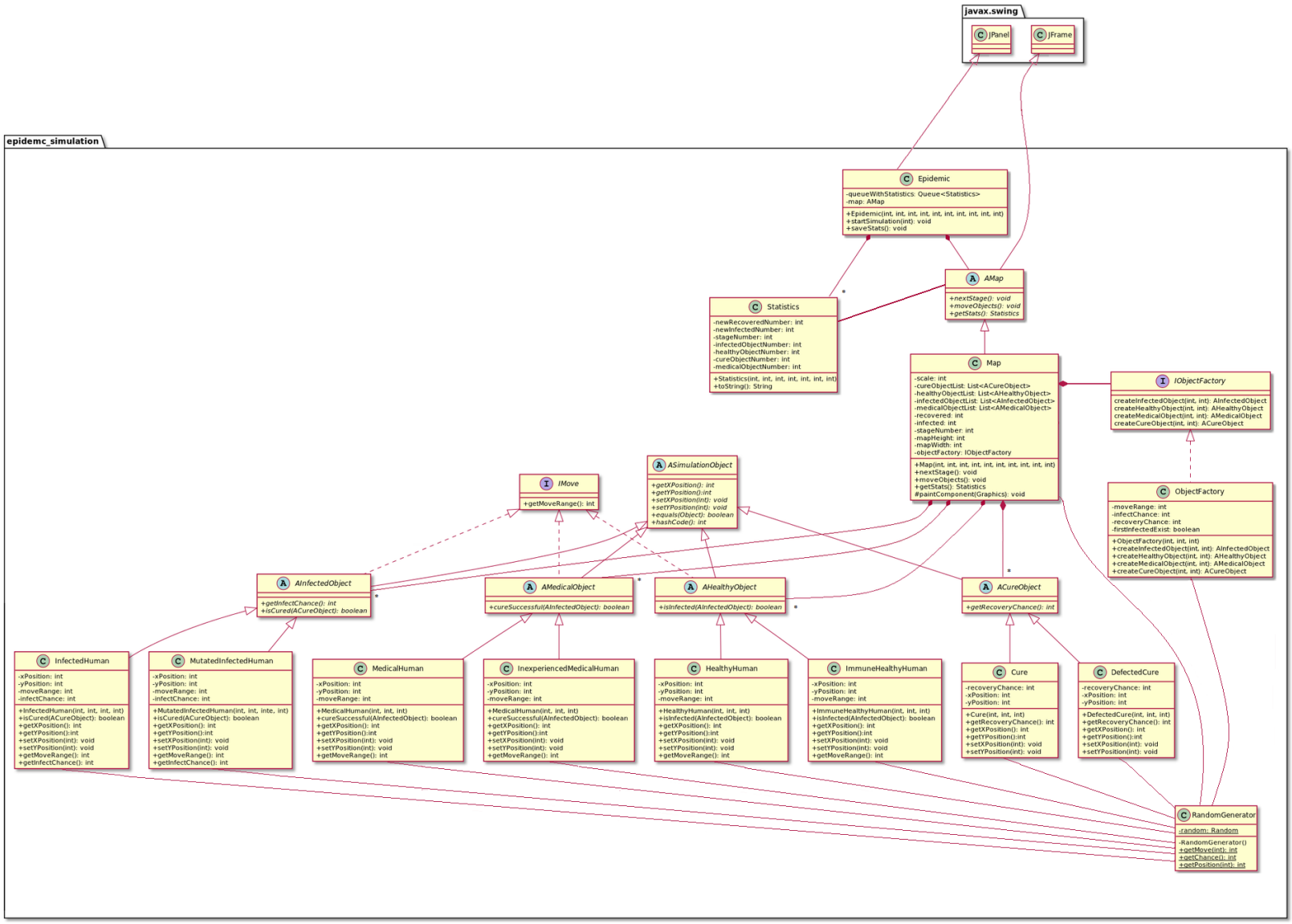


6 Diagram maszyny stanów



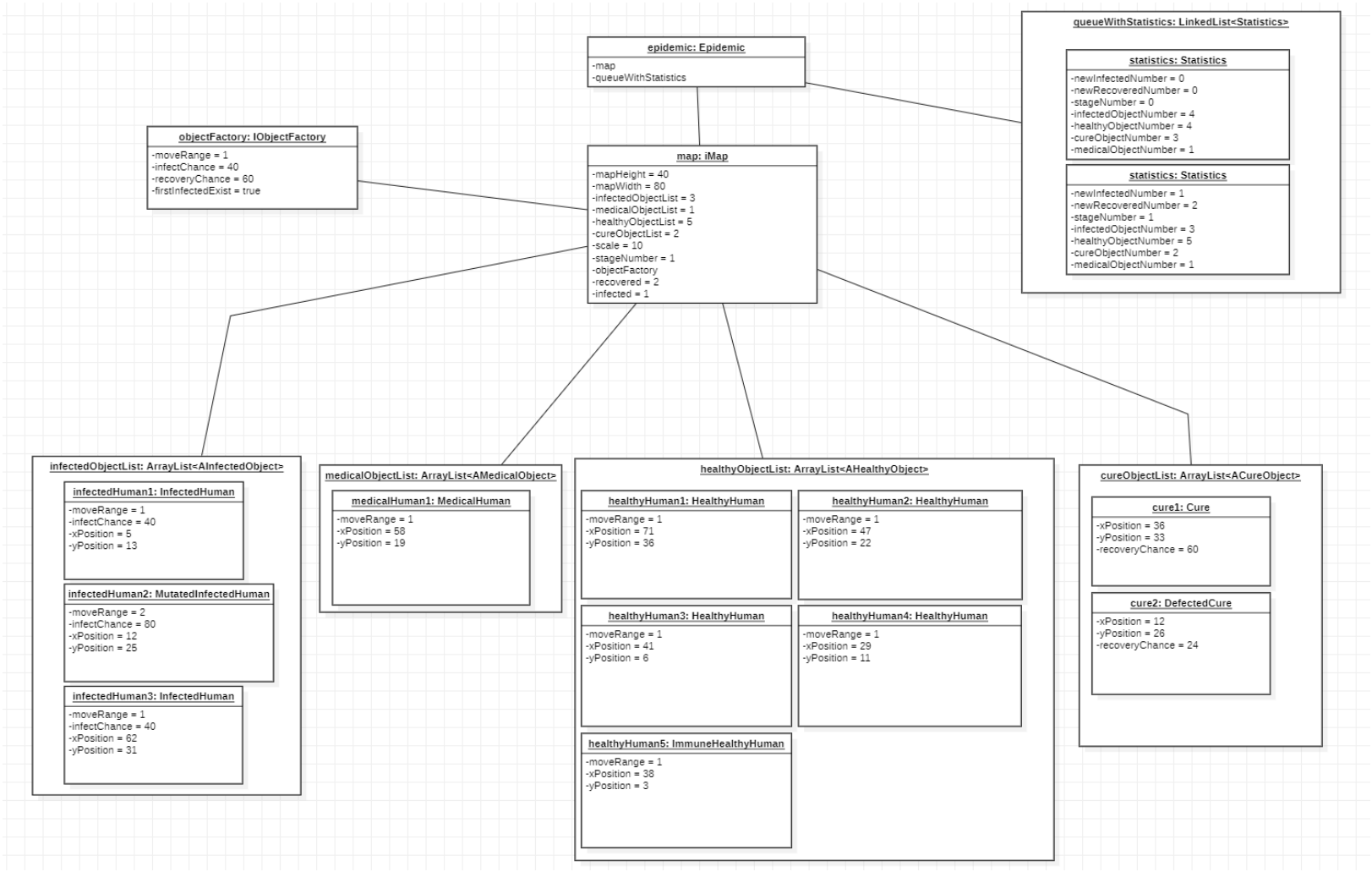
7 Diagram klas

Wersja w lepszej rozdzielczości:
https://github.com/Damian0401/ProjektPO/blob/master/Dokumentacja/DiagramyKlas/ClassDiagram.png



8 Diagram obiektów

Wersja w lepszej rozdzielczości:
https://github.com/Damian0401/ProjektPO/blob/master/Dokumentacja/DiagramyObiektow/ObjectDiagram.png



9 Analiza czasownikowo-rzeczownikowa

Potrzebny nam jest obiekt umożliwiający przeprowadzenie oraz zapisanie statystyk symulacji.

Ponadto potrzebujemy obiektu zarządzającego obiektami symulacji. Chcemy aby pozwalał na poruszanie obiektami, przeprowadzenie kolejnej epoki oraz pobranie statystyk symulacji.

Będzie też nam potrzebny obiekt umożliwiający tworzenie obiektów symulacji.

Potrzebujemy obiektu, który może zostać zakażony oraz obiektu który może zostać uleczony.

Potrzebujemy obiektu przechowującego szanse na uleczenie oraz umożliwiającego wyłuskanie szansy na zakażenie.

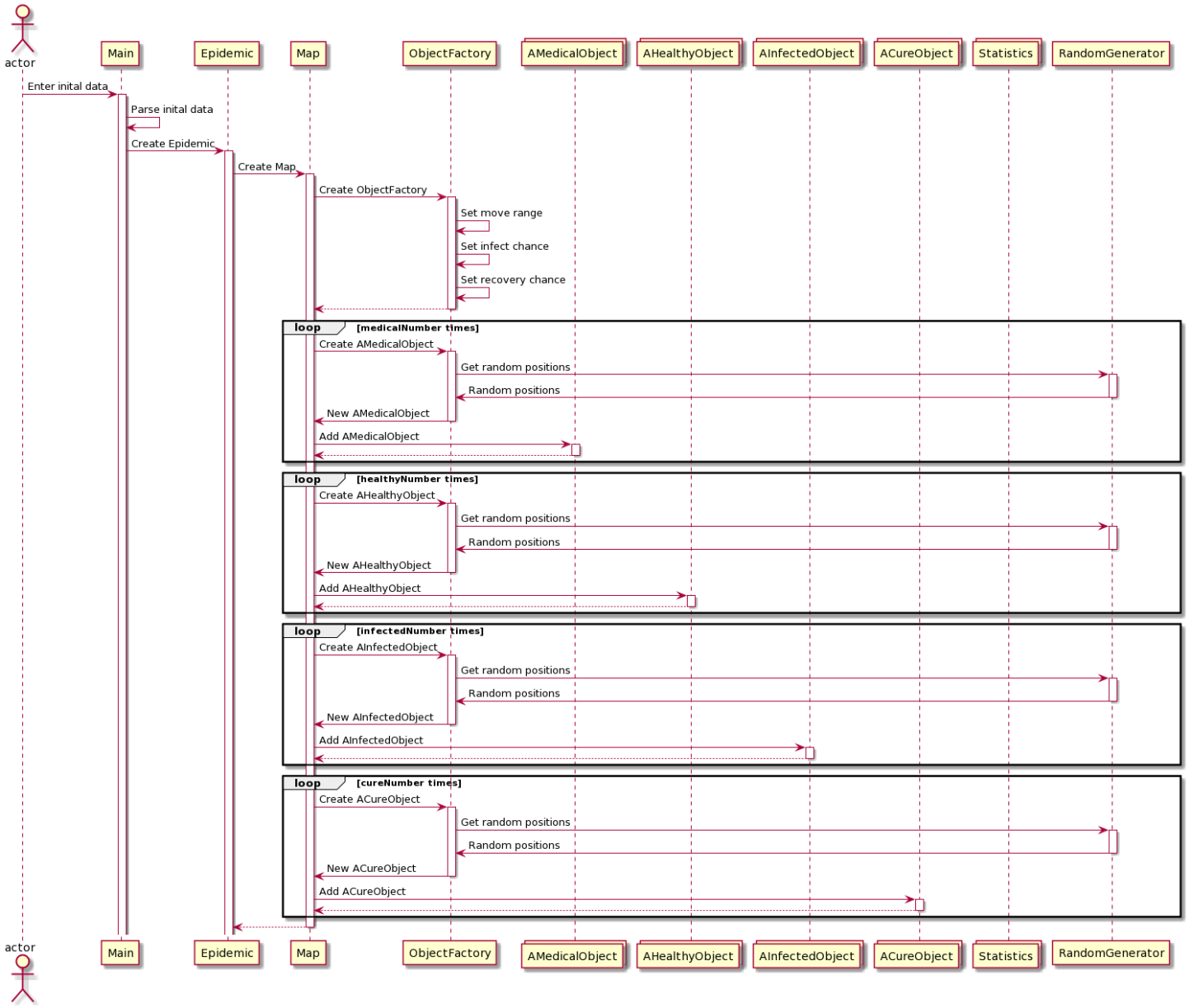
Ponadto potrzebujemy obiektu odpowiedzialnego za leczenie zakażonych oraz obiektu odpowiedzialnego za przechowywanie statystyk poszczególnych epoko.

Epidemic	Przeprowadzenie symulacji
Epidemic	Zapisanie statystyk
AMap	Poruszanie obiektami
AMap	Przeprowadzenie epoki symulacji
AMap	Pobranie statystyk
IObjectFactory	Tworzenie obiektów
AHealthyObject	Zostanie zakażonym
AInfectedObject	Zostanie uleczonym
ACureObject	Przechowuje szanse na zakażenie
ACureObject	Wyłuskanie szansy na zakażenie
AMedicalObject	Leczenie zakażonych
Statistics	Przechowywanie statystyk

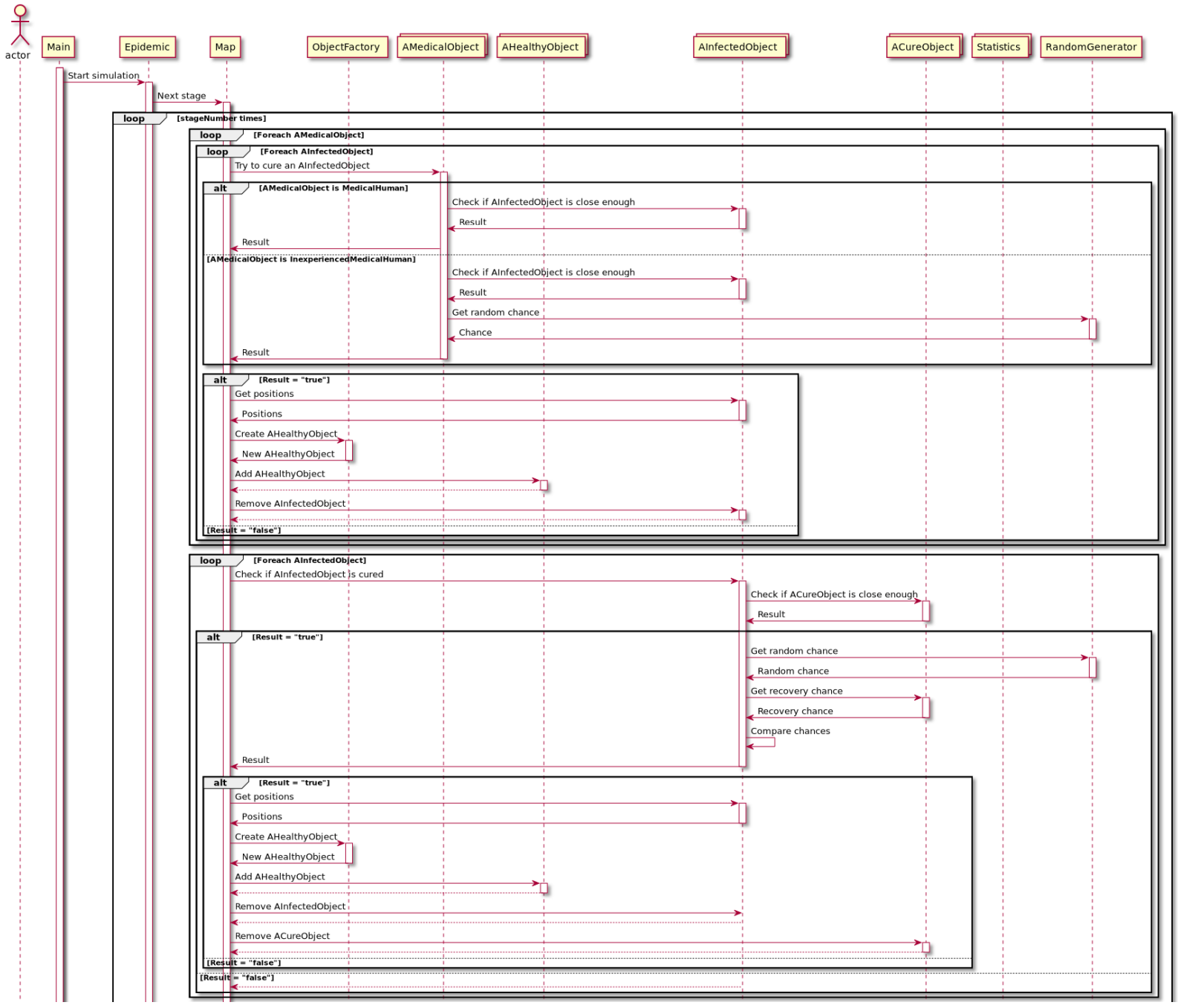
10 Diagramy sekwencji

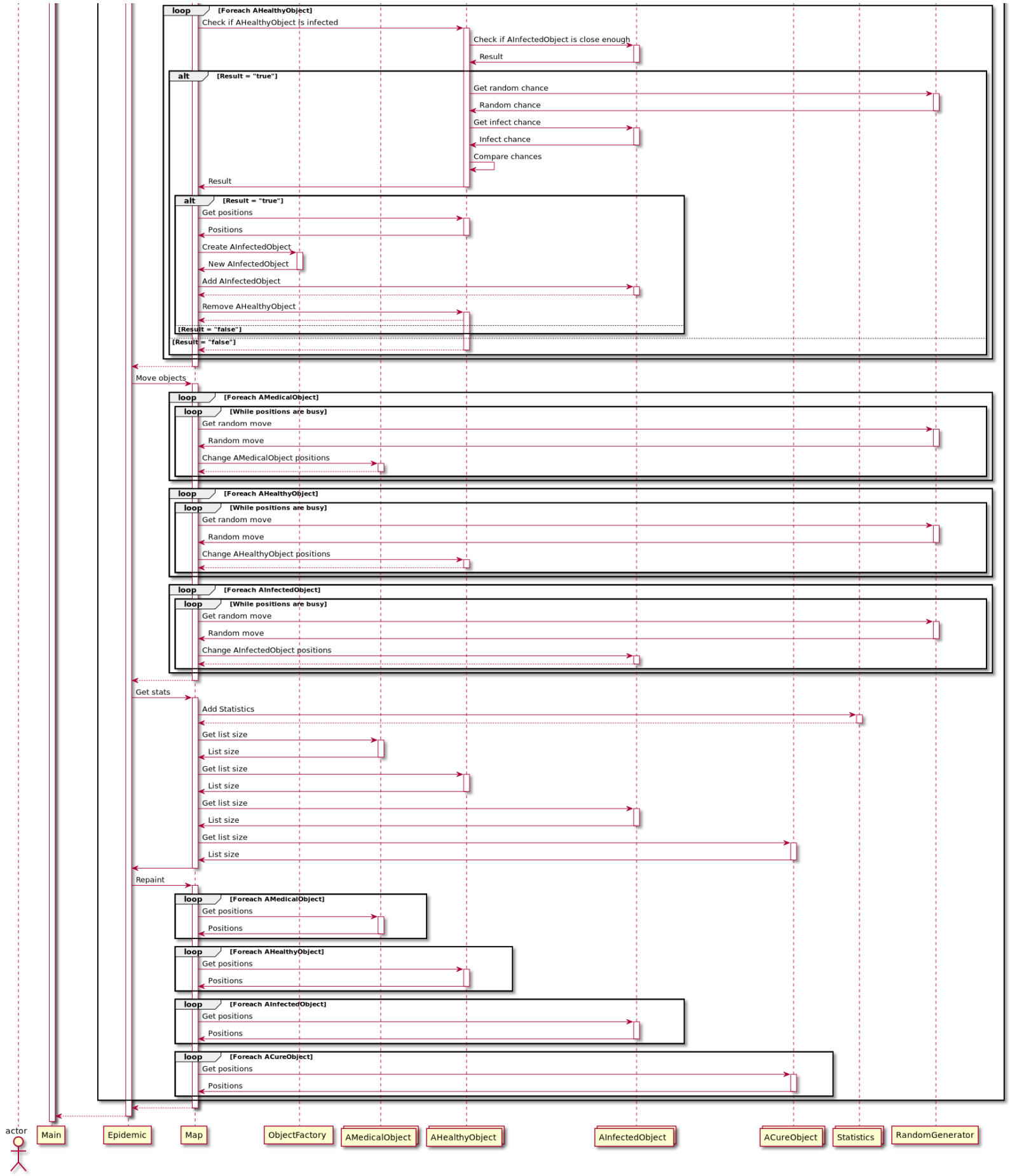
Wersja w lepszej rozdzielczości:
`https://github.com/Damian0401/ProjektPO/blob/master/Dokumentacja/DiagramySekwancji/SequencyDiagram.png`

Inicjalizacja symulacji oraz utworzenie obiektów

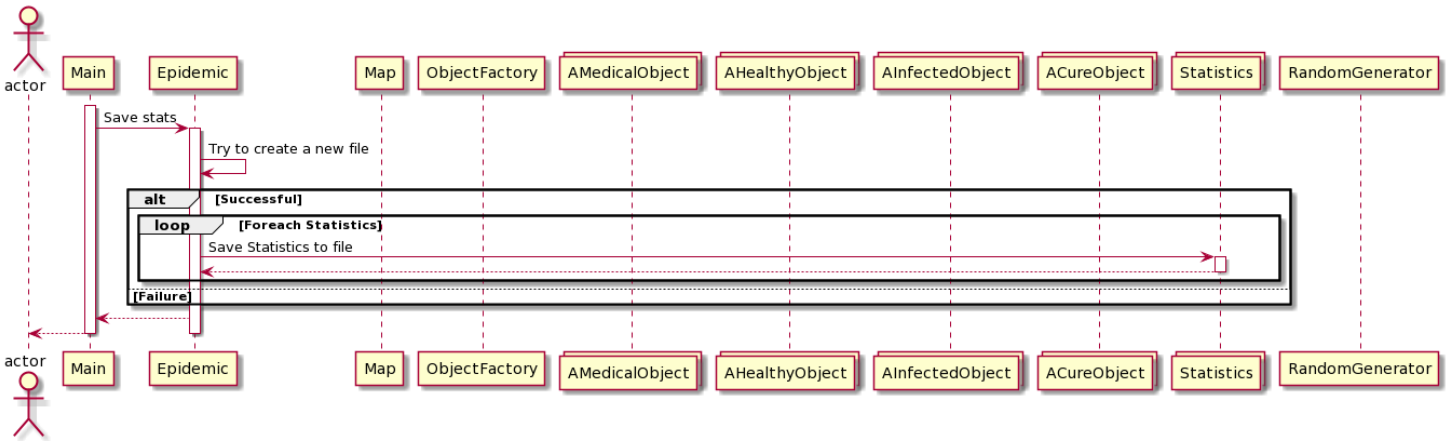


Przebieg symulacji




loop [Foreach AHealthyObject]
Check if AHealthyObject is infected
Check if AInfectedObject is close enough
Result
alt [Result = "true"]
Get random chance
Random chance
Get infect chance
Infect chance
Compare chances
Result
alt [Result = "true"]
Get positions
Positions
Create AInfectedObject
New AInfectedObject
Add AInfectedObject
Remove AHealthyObject
[Result = "false"]
[Result = "false"]
Move objects
loop [Foreach AMedicalObject]
loop [While positions are busy]
Get random move
Random move
Change AMedicalObject positions
loop [Foreach AHealthyObject]
loop [While positions are busy]
Get random move
Random move
Change AHealthyObject positions
loop [Foreach AInfectedObject]
loop [While positions are busy]
Get random move
Random move
Change AInfectedObject positions
Get stats
Add Statistics
Get list size
List size
Get list size
List size
Get list size
List size
Get list size
List size
Repaint
loop [Foreach AMedicalObject]
Get positions
Positions
loop [Foreach AHealthyObject]
Get positions
Positions
loop [Foreach AInfectedObject]
Get positions
Positions
loop [Foreach ACureObject]
Get positions
Positions
actor
Main
Epidemic
Map
ObjectFactory
AMedicalObject
AHealthyObject
AInfectedObject
ACureObject
Statistics
RandomGenerator

Zapisanie statystyk symulacji do pliku



11 Uwagi

Wszystkie diagramy oraz wykresy wykorzystane w dokumentacji znajdują się pod adresem:
`https://github.com/Damian0401/ProjektPO/tree/master/Dokumentacja`

Programy/technologie wykorzystane podczas realizacji projektu:

- Gradle
- Git/GitHub
- PlantUML
- StarUML

"The trouble with programmers is that you can never tell what a programmer is doing until it's too late."

-Seymour Cray