

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	Diagramy aktywności	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/ProjektP0>

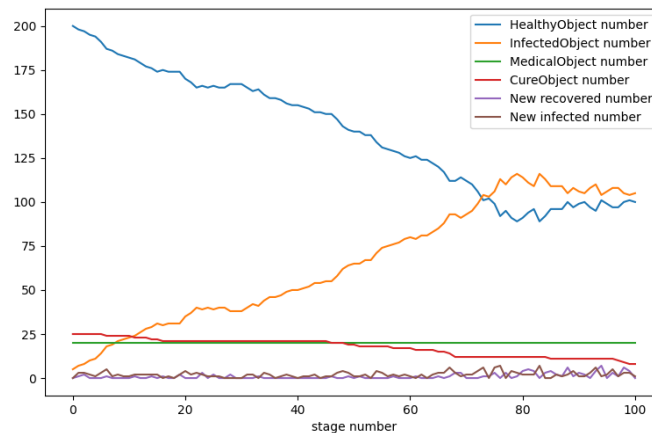
"The trouble with programmers is that you can never tell what a programmer is doing until it's too late."

-Seymour Cray

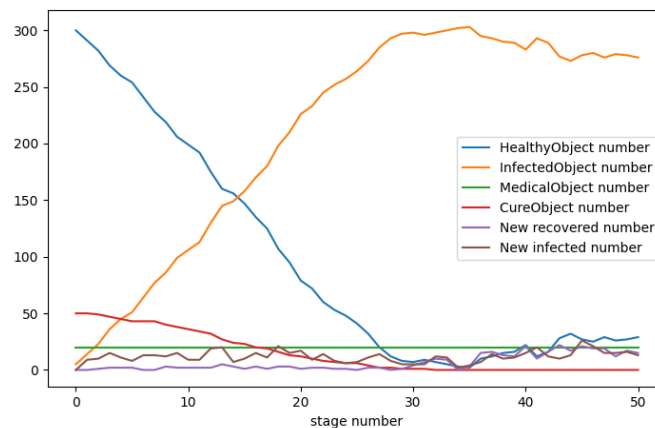
3 Opis zadania symulacji w języku naturalnym

Program przedstawia rozwój epidemii na dwuwymiarowej mapie, o dowolnej wielkości. Na początku symulacji na mapie zostają rozmieszczone w losowy sposób 4 rodzaje obiektów: 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
<ul style="list-style-type: none"> get recovery chance 		

Abstract	AHealthyObject	Superclasses: ASimulationObject, IMove Subclasses: HealthyHuman, ImmuneHealthyHuman
<ul style="list-style-type: none"> check if object is infected 	<ul style="list-style-type: none"> AInfectedObject 	

Abstract	AInfectedObject	Superclasses: ASimulationObject, IMove Subclasses: InfectedHuman, MutatedInfectedHuman
<ul style="list-style-type: none"> get infect chance check if object is cured 	<ul style="list-style-type: none"> ACureObject 	

Abstract	AMap	Superclasses: JPanel Subclasses: Map
<ul style="list-style-type: none"> carry out next stage move objects get stats 	<ul style="list-style-type: none"> ACureObject AInfectedObject AHealthyObject AMedicalObject 	

Abstract	AMedicalObject	Superclasses: ASimulationObject, IMove Subclasses: MedicalHuman, InexperiencedMedicalHuman
<ul style="list-style-type: none"> check if cure is successful 	<ul style="list-style-type: none"> AInfectedObject 	

	Statistics	Superclasses: Subclasses:
<ul style="list-style-type: none"> store statistics get string with statistics 		

Abstract <div> Superclasses: ACureObject, AHealthyObject, AInfectedObject, AMedicalObject ASimulationObject </div> <div>Subclasses:</div>	
<ul style="list-style-type: none"> • get x position • get y position • set x position • set y position • check if objects are equal • generate hash code 	

Cure <div> Superclasses: ACureObject </div> <div>Subclasses:</div>	
<ul style="list-style-type: none"> • get recovery chance • get x position • get y position • set x position • set y position 	

DefectedCure <div> Superclasses: ACureObject </div> <div>Subclasses:</div>	
<ul style="list-style-type: none"> • get decreased recovery chance • get x position • get y position • set x position • set y position 	

Epidemic <div> Superclasses: JFrame </div> <div>Subclasses:</div>	
<ul style="list-style-type: none"> • start simulation • save stats • create map 	<ul style="list-style-type: none"> • Statistics • Date • SimpleDateFormat • File • StackFile

<div> <div>HealthyHuman</div> <div> Superclasses: AHealthyObject Subclasses: </div> </div>	
<ul style="list-style-type: none"> • check if object is infected • get move range • get x position • get y position • set x position • set y position 	<ul style="list-style-type: none"> • RandomGenerator • Math • AInfectedObject

<div> <div>ImmuneHealthyHuman</div> <div> Superclasses: AHealthyObject Subclasses: </div> </div>	
<ul style="list-style-type: none"> • check if object is infected (with decreased chance) • get move range • get x position • get y position • set x position • set y position 	<ul style="list-style-type: none"> • RandomGenerator • Math

<div> <div>Interface</div> <div>IMove</div> <div> Superclasses: Subclasses: </div> </div>	
<ul style="list-style-type: none"> • get move range 	

<div> <div>InexperiencedMedicalHuman</div> <div> Superclasses: AMedicalObject Subclasses: </div> </div>	
<ul style="list-style-type: none"> • get move range • check if cure is successful (with decreased chance) • get x position • get y position • set x position • set y position 	<ul style="list-style-type: none"> • RandomGenerator • Math • AInfectedObject

<div> <div>InfectedHuman</div> <div> Superclasses: AInfectedObject Subclasses: </div> </div>	
<ul style="list-style-type: none"> • get infect chance • get move range • get x position • get y position • set x position • set y position • check if object is cured 	<ul style="list-style-type: none"> • RandomGenerator • Math • ACureObject

<div> <div>Interface IObjectedFactory</div> <div> Superclasses: Subclasses: ObjectFactory </div> </div>	
<ul style="list-style-type: none"> • create infected object • create healthy object • create cure object • create medical object 	<ul style="list-style-type: none"> • ACureObject • AHealthyObject • AInfectedObject • AMedicalObject

<div> <div>Map</div> <div> Superclasses: AMap Subclasses: </div> </div>	
<ul style="list-style-type: none"> • carry out next stage • move objects • get stats 	<ul style="list-style-type: none"> • ACureObject • AHealthyObject • AInfectedObject • AMedicalObject

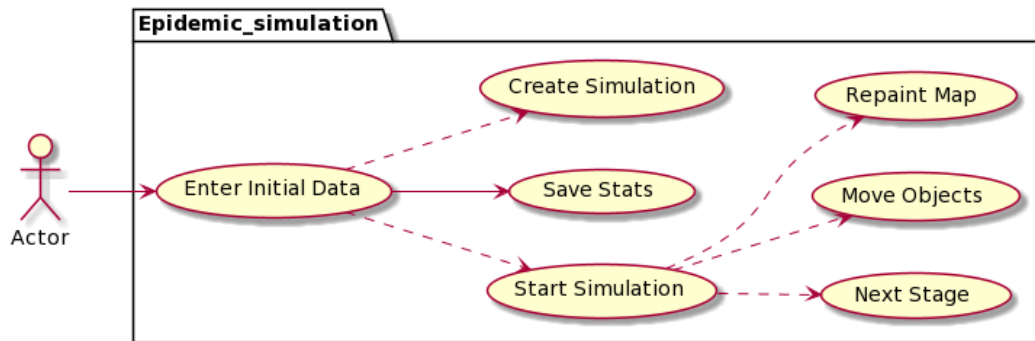
<div> <div>MedicalHuman</div> <div> Superclasses: AMedicalHuman Subclasses: </div> </div>	
<ul style="list-style-type: none"> • get move range • check if cure is successful • get x position • get y position • set x position • set y position 	<ul style="list-style-type: none"> • Math • AInfectedObject

<div> <div> MutatedInfectedHuman </div> <div> Superclasses: AInfectedObject Subclasses: </div> </div>	
<ul style="list-style-type: none"> • 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 	<ul style="list-style-type: none"> • RandomGenerator • Math • ACureObject

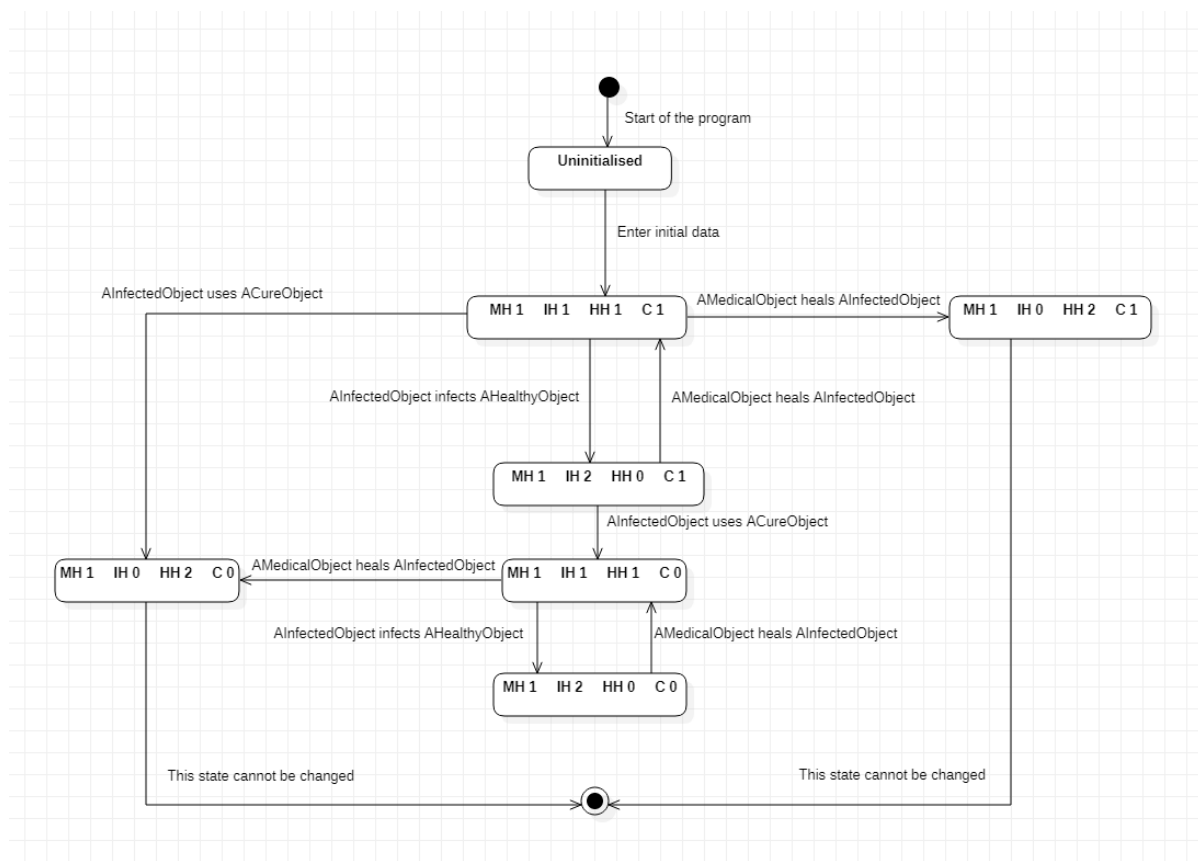
<div> <div> ObjectFactory </div> <div> Superclasses: IObjectFactory Subclasses: </div> </div>	
<ul style="list-style-type: none"> • create infected humans • create healthy humans • create cure objects • create medical humans 	<ul style="list-style-type: none"> • ACureObject • AHealthyObject • AInfectedObject • AMedicalObject • RandomGenerator • MutatedInfectedHuman • MedicalHuman • InfectedHuman • InexperiencedMedicalHuman • ImmuneHealthyHuman • HealthyHuman • DefectedCure • Cure

<div> <div> RandomGenerator </div> <div> Superclasses: Subclasses: </div> </div>	
<ul style="list-style-type: none"> • get random position • get random chance • get random move 	<ul style="list-style-type: none"> • Random

5 Diagram przypadków użycia



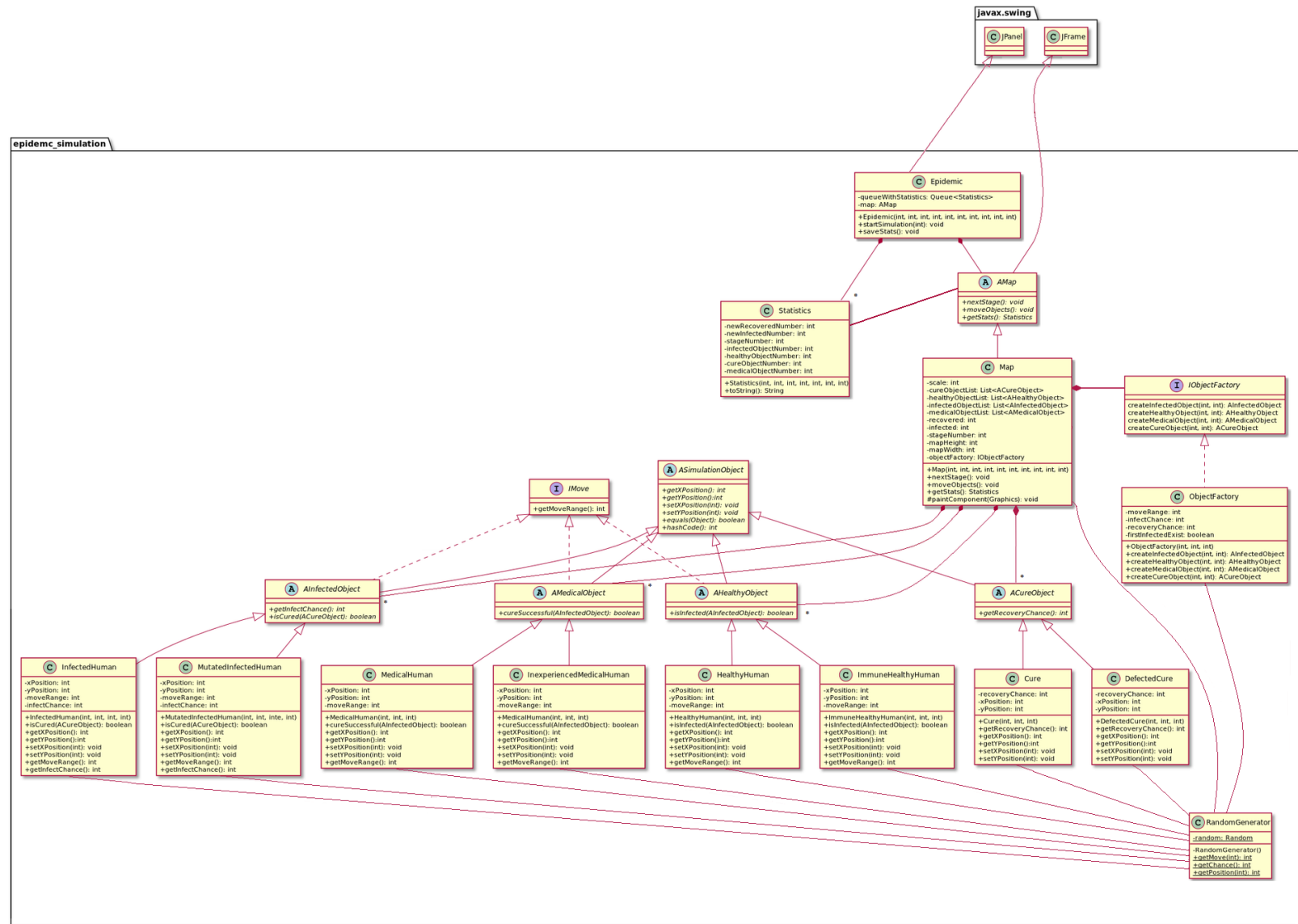
6 Diagram maszyny stanów



7 Diagram klas

Wersja w lepszej rozdzielczości:

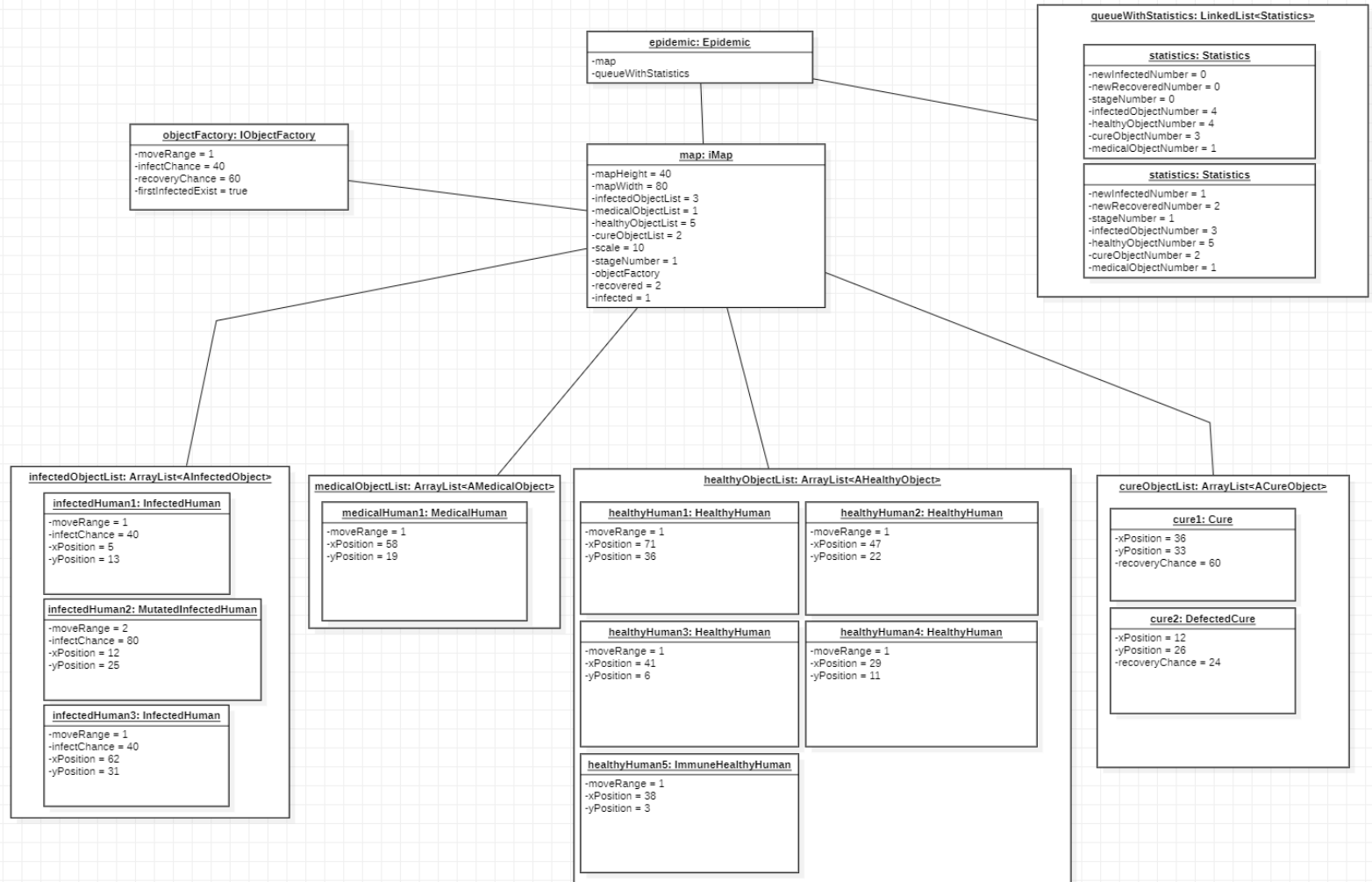
<https://github.com/Damian0401/ProjektPO/blob/master/Dokumentacja/DiagramyKlas/ClassDiagram.png>



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**.

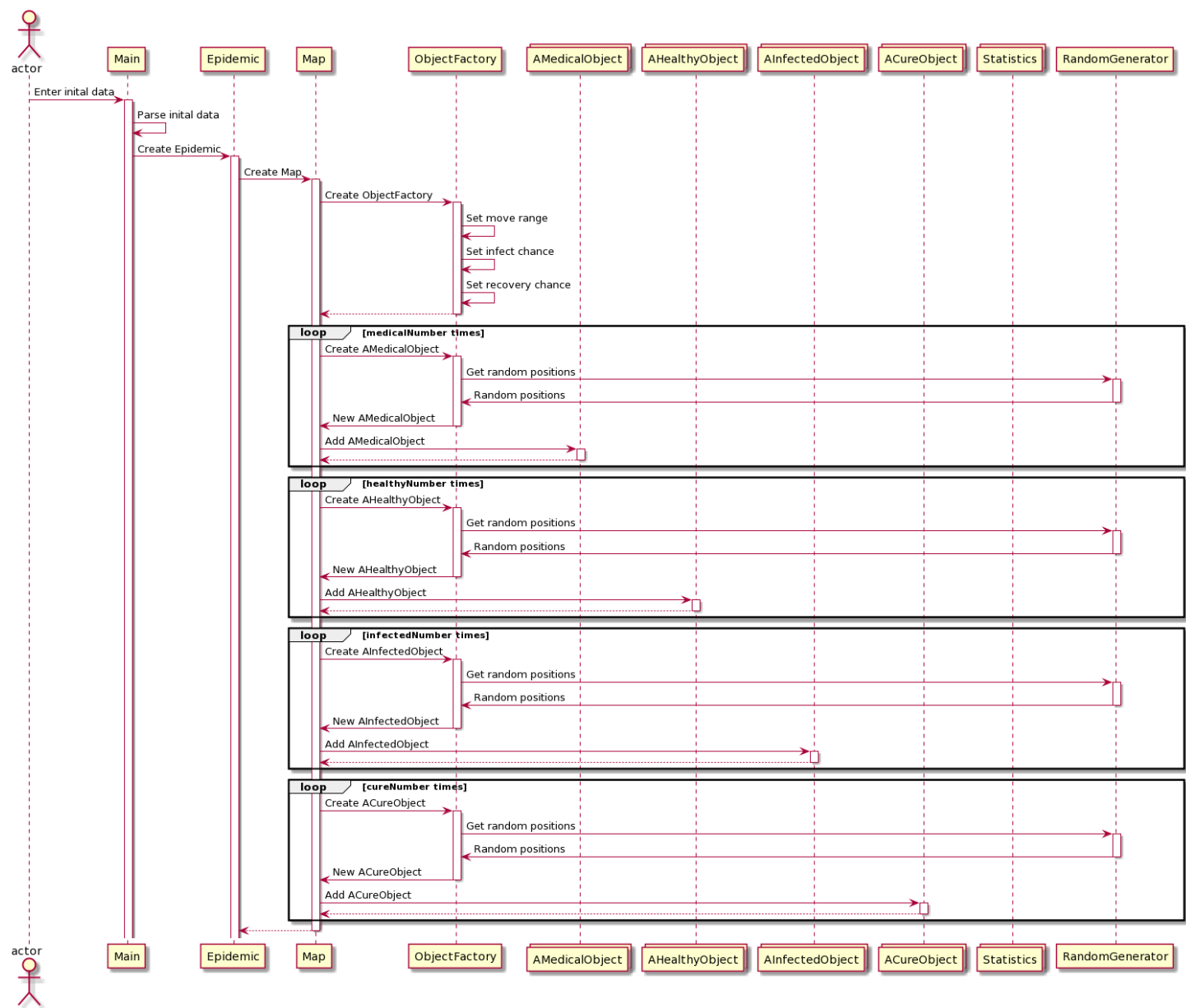
Ponadto potrzebujemy **obiektu** odpowiedzialnego za **leczenie zakażonych** oraz **obiektu** odpowiedzialnego za **przechowywanie statystyk** poszczególnych epoki.

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/ProjektP0/blob/master/Dokumentacja/DiagramySekwancji/SequencyDiagram.png>

Inicjalizacja symulacji oraz utworzenie obiektów

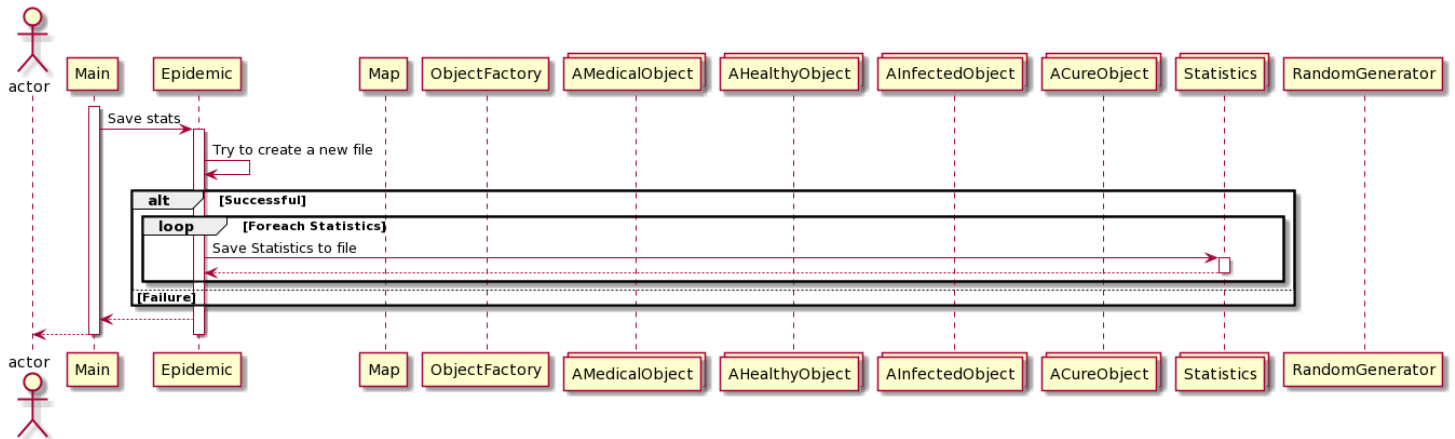


```

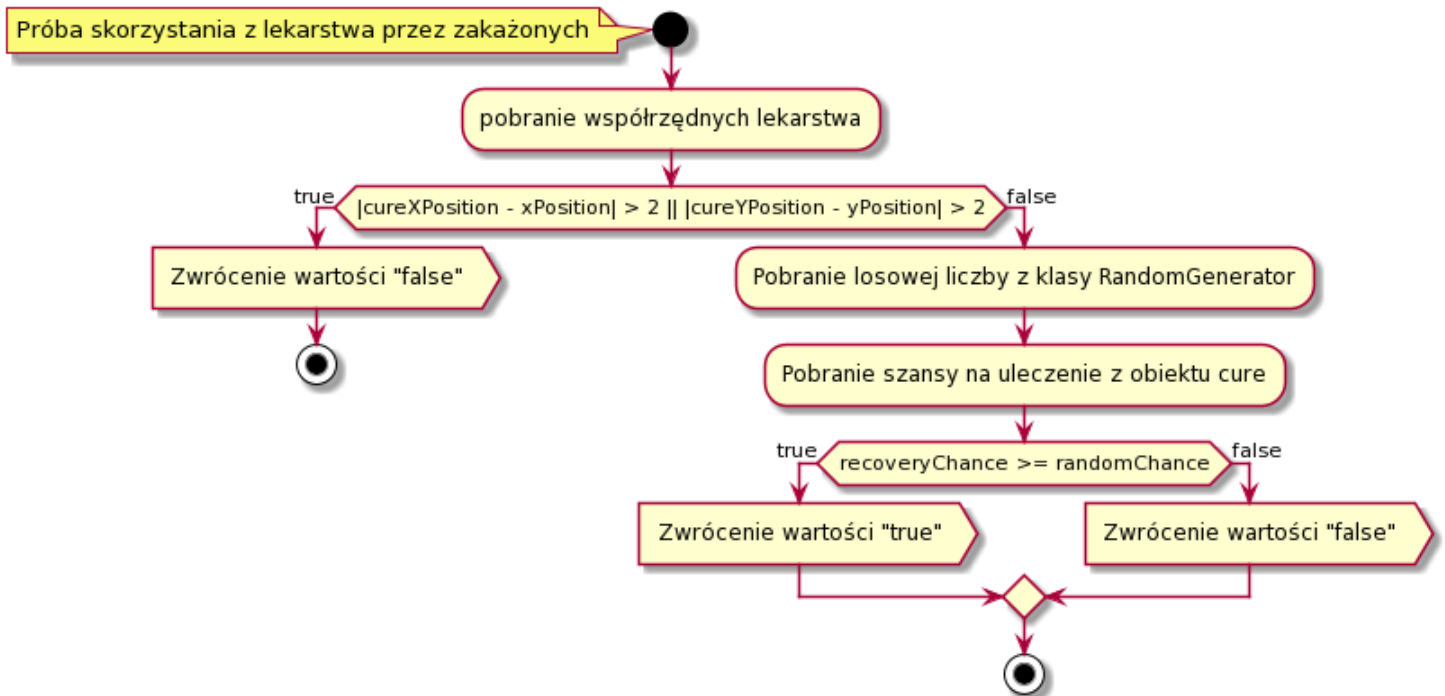
sequenceDiagram
    actor actor
    participant Main
    participant Epidemic
    participant Map
    participant ObjectFactory
    participant AMedicalObject
    participant AHealthyObject
    participant AInfectedObject
    participant ACureObject
    participant Statistics
    participant RandomGenerator

    actor->>Main: Start simulation
    activate Main
    Main->>Epidemic: Next stage
    deactivate Main
    activate Epidemic
    Epidemic->>Map: [stageNumber times]
    activate Map
    Map->>ObjectFactory: loop [Foreach AMedicalObject]
    activate ObjectFactory
    ObjectFactory->>AMedicalObject: loop [Foreach AInfectedObject]
    activate AMedicalObject
    AMedicalObject->>AInfectedObject: Try to cure an AInfectedObject
    activate AInfectedObject
    alt [AMedicalObject is MedicalHuman]
        AInfectedObject->>AInfectedObject: Check if AInfectedObject is close enough
        AInfectedObject-->>AMedicalObject: Result
    else [AMedicalObject is InexperiencedMedicalHuman]
        AInfectedObject->>AInfectedObject: Check if AInfectedObject is close enough
        AInfectedObject-->>AMedicalObject: Result
        AInfectedObject->>RandomGenerator: Get random chance
        RandomGenerator-->>AInfectedObject: Chance
    end
    AInfectedObject-->>AMedicalObject: Result
    alt [Result = "true"]
        AMedicalObject->>ObjectFactory: Get positions
        ObjectFactory-->>AMedicalObject: Positions
        AMedicalObject->>ObjectFactory: Create AHealthyObject
        ObjectFactory-->>AMedicalObject: New AHealthyObject
        AMedicalObject->>ObjectFactory: Add AHealthyObject
        ObjectFactory-->>AMedicalObject: 
        AMedicalObject->>AInfectedObject: Remove AInfectedObject
        AInfectedObject-->>AMedicalObject: 
    else [Result = "false"]
    end
    AMedicalObject-->>ObjectFactory: 
    deactivate AMedicalObject
    ObjectFactory-->>Map: 
    deactivate ObjectFactory
    Map->>AInfectedObject: loop [Foreach AInfectedObject]
    activate AInfectedObject
    AInfectedObject->>AInfectedObject: Check if AInfectedObject is cured
    AInfectedObject->>ACureObject: Check if ACureObject is close enough
    ACureObject-->>AInfectedObject: Result
    alt [Result = "true"]
        AInfectedObject->>RandomGenerator: Get random chance
        RandomGenerator-->>AInfectedObject: Random chance
        AInfectedObject->>RandomGenerator: Get recovery chance
        RandomGenerator-->>AInfectedObject: Recovery chance
        AInfectedObject->>AInfectedObject: Compare chances
        AInfectedObject-->>AInfectedObject: 
    else [Result = "false"]
    end
    AInfectedObject-->>AInfectedObject: Result
    alt [Result = "true"]
        AInfectedObject->>ObjectFactory: Get positions
        ObjectFactory-->>AInfectedObject: Positions
        AInfectedObject->>ObjectFactory: Create AHealthyObject
        ObjectFactory-->>AInfectedObject: New AHealthyObject
        AInfectedObject->>ObjectFactory: Add AHealthyObject
        ObjectFactory-->>AInfectedObject: 
        AInfectedObject->>AInfectedObject: Remove AInfectedObject
        AInfectedObject-->>AInfectedObject: 
        AInfectedObject->>ACureObject: Remove ACureObject
        ACureObject-->>AInfectedObject: 
    else [Result = "false"]
    end
    AInfectedObject-->>AInfectedObject: [Result = "false"]
    deactivate AInfectedObject
    AInfectedObject-->>Map: 
    deactivate AInfectedObject
    Map-->>Epidemic: 
    deactivate Map
    Epidemic-->>actor: 
    deactivate Epidemic
  
```


Zapisanie statystyk symulacji do pliku



11 Diagramy aktywności



Próba leczenia zainfekowanego przez InexperiencedMedicalHuman

