

InformatiCup - Team Stream

Christoph Schnell Jasper Lorenz Robin Schmöcker

Oktober 2019 - Januar 2020

Inhaltsverzeichnis

1	Einleitung	3
1.1	Vorwort	3
1.2	Schnellstart	3
2	Installation	4
2.1	Bauen der Jar Datei	4
2.2	Lokale Version mit GUI	4
2.3	Bereitstellung bei AWS	5
3	Benutzung	12
3.1	Benutzung über AWS	12
3.2	Benutzung der lokalen Version	12
3.2.1	Benutzung der graphischen Oberfläche	12
3.2.2	Benutzung ohne graphische Oberfläche	12
3.2.3	Beenden des Servers	12
3.2.4	Anzeige der Fehlermeldungen des Webservices	12
3.3	Benutzung des Testskripts	13
4	Lösungsidee	14
4.1	Entwicklung von Ideen	14
4.2	Verworfenen Ansätze	15
4.3	Umsetzung der Gesamtidee	15
4.4	Pathogen Klassifikationen	15
4.4.1	Aktives Pathogen	15
4.4.2	Starkes Pathogen	15
4.4.3	Langsames Pathogen	15
4.4.4	Schnelles Pathogen	16
4.4.5	Ignoriertes Pathogen	16
4.5	Bewertung	16
4.5.1	Runde beenden	16
4.5.2	Quarantäne	17
4.5.3	Medikament entwickeln	17
4.5.4	Impfung entwickeln	18

4.5.5	Medikament verteilen	18
4.5.6	Impfung verteilen	18
4.5.7	Zufallseignisse	18
4.5.8	Verbindungen sperren und Flughafen schließen	19
5	Software Architektur	20
5.1	Allgemein	20
5.2	UML Diagramm	20
5.3	Gamepaket	21
5.4	HTTP-Paket	21
5.5	Solverpaket	22
5.6	GUI-Paket	22
5.7	IO-Paket	23
6	Wartung und Weiterentwicklung	24
6.1	Die Bewertungsfunktion anpassen	24
6.2	Hinzufügen neuer Attributwerte	24
6.3	Hinzufügen neuer Konstanten	24
6.4	Hinzufügen neuer Events	24
6.5	Hinzufügen neuer Aktionen	25
7	Validierung	26
8	Reflexion	26
8.1	Software Architektur	26
8.2	Bibliotheks Nutzung	26
8.3	Programmierstil	26
8.4	Heuristik	27
8.5	Organisatorisch	27
9	FAQ	28
10	Anhang	30
10.1	Herausgefundene Daten über GI client	30
10.1.1	Pathogene	31
10.1.2	Städte	32
10.1.3	Seeds	36
10.1.4	Seeds mit Starteigenschaften	42
10.2	Wichtige Codeauszüge	42
	Glossar	45

1 Einleitung

1.1 Vorwort

Wenn eine Pandemie die Menschheit auszulöschen droht, dann braucht die Welt Mut, Stärke, Intelligenz, Weisheit und Geschick. Anders gesagt es braucht die Besten der Besten. Die F.L. Bauern unter den Softwareingenieuren. Die Turinge unter den Programmierern. Die Admiral Trips unter den Pathogenen. Eben die „Crème de la Crème“. Die Welt hat Glück, denn eine Gruppe junger Programmierer vereint genau diese Eigenschaften: Team Stream.

Ohne weitere Selbstbeweihräucherung nun zur Dokumentation, welche sich grob wie folgt gliedert. Zunächst wird erläutert, wie unsere Lösung zu installieren und im nachfolgenden Kapitel auch zu bedienen ist. Auf die grafische Oberfläche wird genauer eingegangen. Anschließend wird der theoretische Lösungsansatz dargelegt, der vielleicht erfrischenderweise mal kein neuronales Netz beinhaltet. Bevor die Umsetzung des Lösungsansatzes im Code im Detail beschrieben wird, wird die Software Architektur erläutert. Danach wird erklärt wie unser Programm weiterzuentwickeln ist und welche Schritte wir unternommen haben, um die Stärke unseres Programmes zu beweisen.

Zum Ende hin wird kurz reflektiert, was in dieser finalen Abgabe nicht optimal lief, bzw. was wir beim nächsten mal anders machen würden. Zum Schluss werden noch häufig gestellte Fragen zu unserem Projekt beantwortet und Daten dargelegt, die wir über die Spielversion des bereitgestellten Kommandozeilenwerkzeugs gefunden haben.

1.2 Schnellstart

Die schnellste und einfachste Weise den *GI Client* zu testen ist über den von uns bei AWS bereitgestellten Webservice. Um diesen nutzen zu können, muss der *GI Client* mit dem „-u“ Parameter und folglich der URL unseres Webservices aufgerufen werden.

Dies sieht z.B. unter Linux wie folgt aus.

```
\$ ./ic20_linux -u https://udi8pt9vo9.execute-api.us-east-1.amazonaws.com/default/
```

2 Installation

Dieses Kapitel erklärt die Installation unseres Webservices. Möchte man unseren Webservice ohne Installation nutzen, so können die Schritte unter 1.2 Schnellstart befolgt werden.

2.1 Bauen der Jar Datei

Sowohl für 2.2 Lokale Version mit GUI, als auch für 2.3 Bereitstellung bei AWS, muss zuerst die Jar Datei von unserem Projekt gebaut werden. Dazu muss die Oracle Java Version 8 und Maven installiert sein. Zum leichten Herunterladen von Git-Hub sollte auch noch Git installiert sein. Sind diese Grundlagen geschaffen, gilt es als erstes das Git-Repository zu klonen. Alternativ kann das Projekt auch als Zip-Datei von Git-Hub heruntergeladen und entpackt werden. Hier wird ein Beispiel für das Klonen mit HTTPS gezeigt.

```
\$ git clone https://github.com/jasZnerol/InformatiCup2020.git
```

Es sollte nun ein neuer Ordner namens „informatiCup2020“ vorhanden sein. In diesen muss man nun wechseln.

```
\$ cd informatiCup2020/
```

Zum Bauen muss folgender Maven-Aufruf aus dem Root-Verzeichnis des Projektes erfolgen.

```
\$ mvn package
```

Die Jar-Datei sollte nun in dem Unterordner „target“ liegen und „ic-webservice-20.jar“ heißen.

2.2 Lokale Version mit GUI

Um unser Programm lokal auszuführen, muss Oracle Java 8 installiert sein. Wurden die Schritte von 2.1 Bauen der Jar Datei befolgt so muss nur folgender Befehl zum Ausführen des Programms aufgerufen werden.

```
\$ java -jar target/ic-webservice-20.jar
```

War die Installation erfolgreich, so sollte sich an dieser Stelle die zur lokalen Version gehörende, GUI öffnen. Mehr Informationen finden sich unter 3.2 Benutzung der lokalen Version.

Da die Aufgabenstellung einen Befehl zum Bauen und Ausführen vorgibt, können die Befehle auch kombiniert werden.

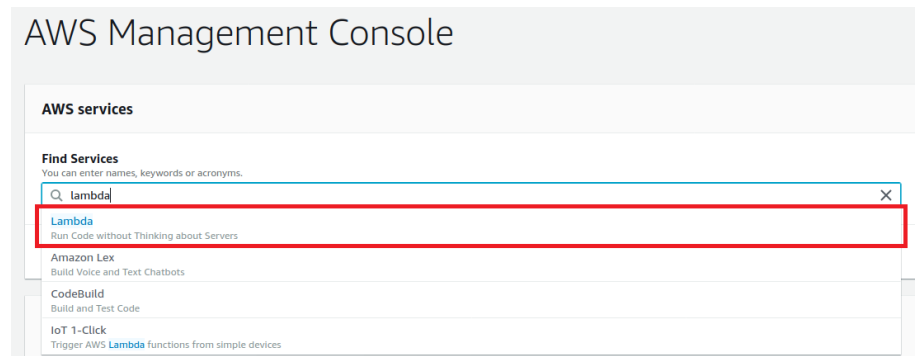
```
\$ mvn package && java -jar target/ic-webservice-20.jar
```

2.3 Bereitstellung bei AWS

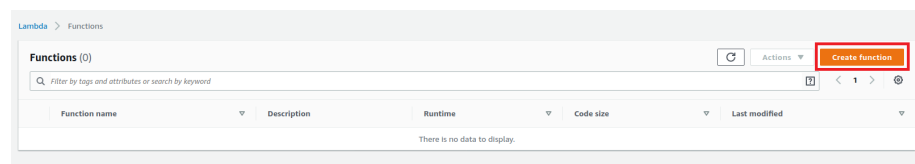
Wurden die Schritte von 2.1 Bauen der Jar Datei befolgt, gilt es die gebaute Jar bei AWS als Lambda-Funktion bereitzustellen. Dafür öffnet man die AWS-Konsole unter der folgenden URL:

`https://console.aws.amazon.com/console/home`

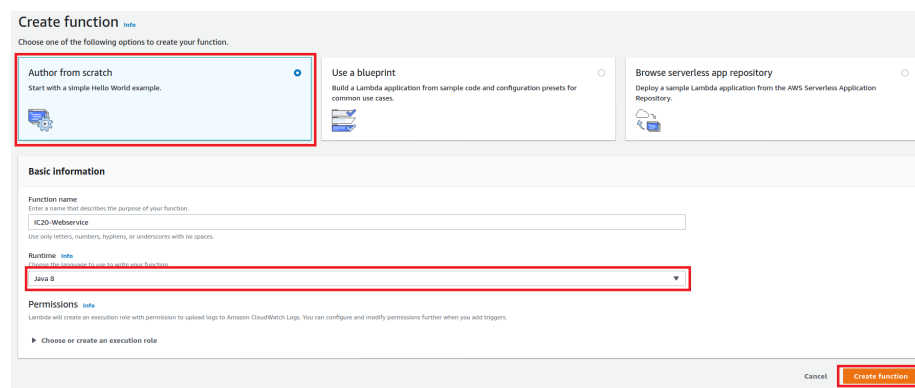
Als erstes sucht man in der Suchleiste nach „Lambda“.



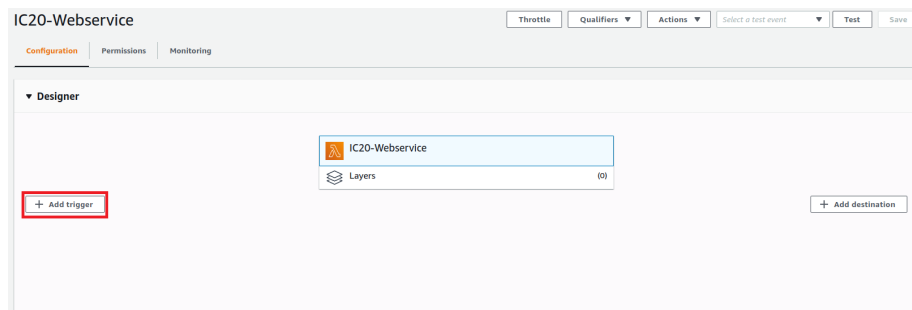
Dannach erstellt man eine neue Lambda-Funktion.



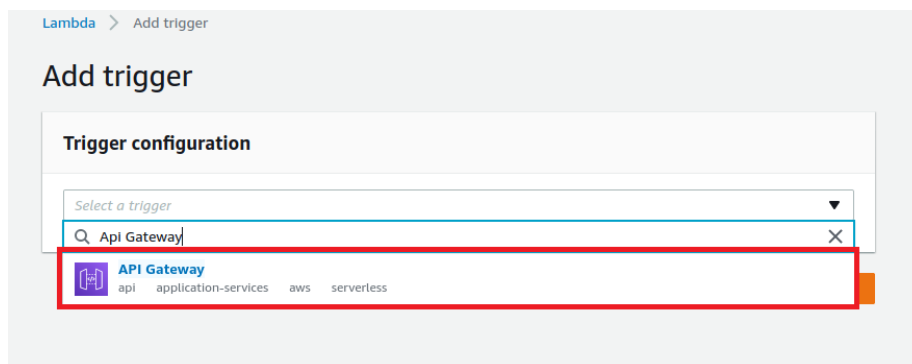
Nun gilt es die Optionen zum Erstellen festzulegen. Dabei wählen wir „Author from scratch“, einen beliebigen Namen und Java 8 als Laufzeit Umgebung.



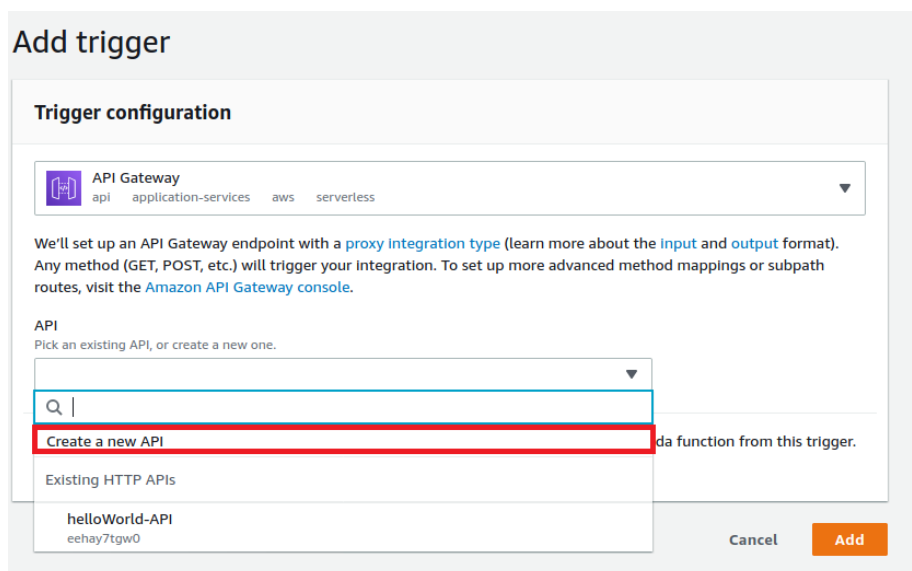
Jetzt muss ein Auslöser zur Funktion hinzugefügt werden.



Wir wollen eine neu erstellte API als Auslöser wählen. Darum wählen wir als Auslöser das „API Gateway“.



Danach direkt „Create new API“.



Die zu erstellende API soll eine Rest-API sein. Als Sicherheitseinstellung wählen wir „öffentlich“.

Trigger configuration

API Gateway
api application-services aws serverless

We'll set up an API Gateway endpoint with a [proxy integration type](#) (learn more about the [input](#) and [output](#) format). Any method (GET, POST, etc.) will trigger your integration. To set up more advanced method mappings or subpath routes, visit the [Amazon API Gateway console](#).

API

Pick an existing API, or create a new one.

Create a new API

Choose a template

☐ HTTP API
Beta
Ideal for building REST APIs that proxy to Lambda functions.

☒ REST API
Ideal for building REST APIs where you want full control over the request and response with API management capabilities.

What are we creating in this template?
This template has the all inclusive set of features needed to build, manage, and publish REST APIs. Required to use IAM authentication or API keys.
To learn more about various features supported go to [API Gateway Documentation](#).

Security

Configure the security mechanism for your API endpoint. [Learn more](#).

Open

Warning: Your API endpoint will be publicly available and can be invoked by all users.

► Additional settings

Lambda will add the necessary permissions for Amazon API Gateway to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Cancel Add

Damit die Lambda-Funktion noch unseren Code ausführt, muss jetzt die gebaute Jar noch hochgeladen werden. Dafür wählt man „Upload“. Nachdem die gebaute Jar aus dem target Ordner hochgeladen wurde, muss noch ein AWS-Handler festgelegt werden. Dieser ist in unserem HTTP-Package zu finden. Der Handler wird auf „app.http.AWShandler::handleRequest“ gesetzt.

Function code [info](#)

Code entry type

Upload a zip or jar file

Function package

☒ Upload ☐ _c_web-service-20.jar (386.2 kB)
For files larger than 10 MB, consider uploading using Amazon S3.

Runtime

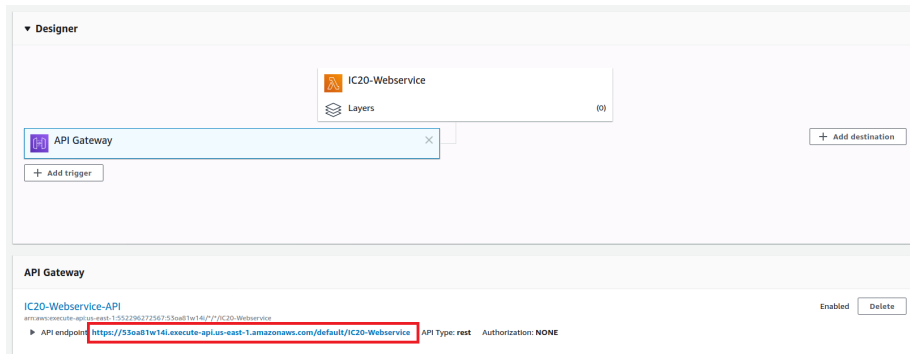
Java 8

Handler [info](#)

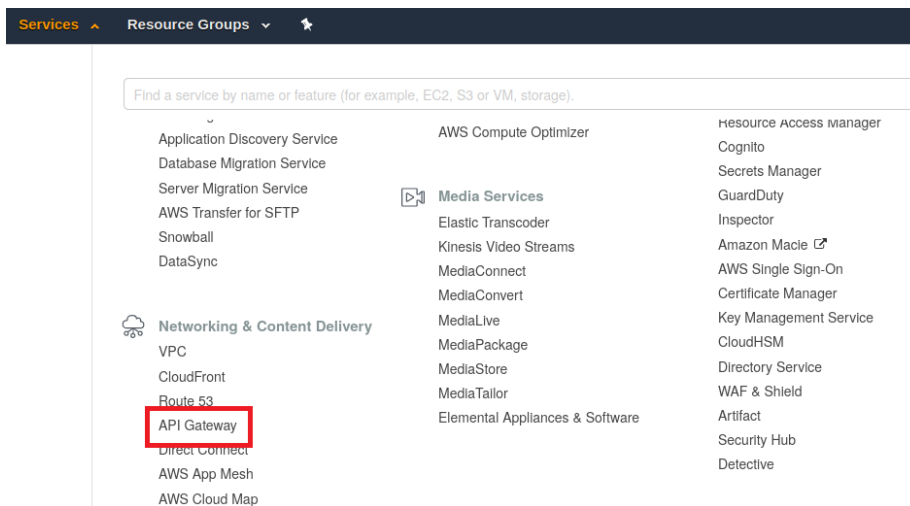
app.http.AWShandler::handleRequest

Die Einstellung muss man nun noch speichern und dann ist das Bereitstellen auf

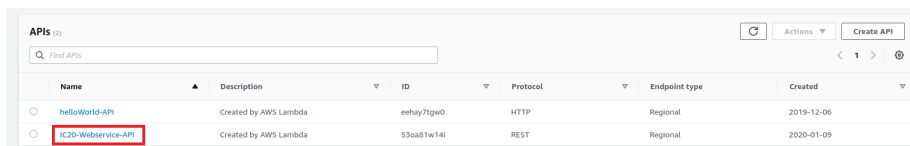
AWS fast abgeschlossen. Die URL des neuen Webservice findet sich unter der Rest-Api als „endpoint“.



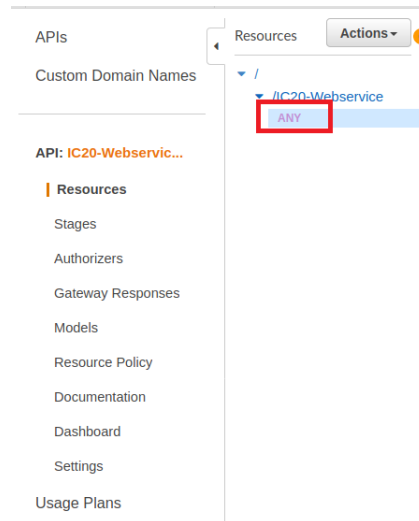
Um die Ausführung zu gewährleisten, muss eine weitere Einstellung der API geändert werden. Dazu suchen wir zuerst die API in der AWS-Konsole.



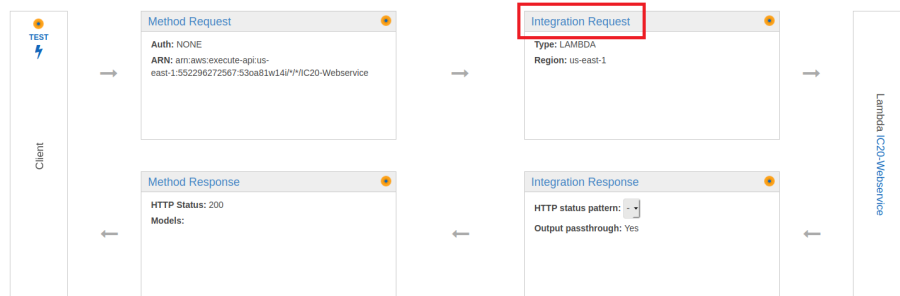
Wir wählen unsere gerade erstellte API.



Danach muss „Any“ ausgewählt werden.



Anschließend die Integrationsanforderung.



Nun muss die Checkbox Lambda-Proxy **deaktiviert** werden.

Provide information about the target backend that this method will call and whether the incoming request data should be modified.

Integration type ☒ Lambda Function ⓘ
☐ HTTP ⓘ
☐ Mock ⓘ
☐ AWS Service ⓘ
☐ VPC Link ⓘ

Use Lambda Proxy integration ☒ ⓘ

Lambda Region us-east-1 ⓘ

Lambda Function IC20-Webservice ⓘ

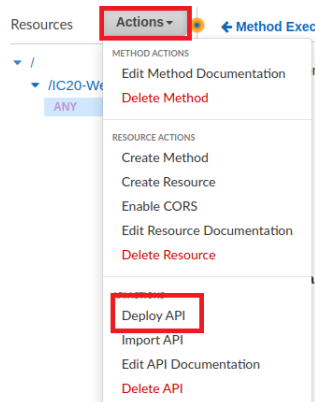
Execution role ⓘ

Invoke with caller credentials ☐ ⓘ

Credentials cache Do not add caller credentials to cache key ⓘ

Use Default Timeout ☒ ⓘ

Um die Änderungen Wirksam zu machen, muss die API neu bereitgestellt werden. Dazu wählt man erst Aktionen aus und in dem Drop-Down-Menu „bereitstellen“.



Als nächstes wählt man die Stufe „default“ und stellt sie bereit.

Deploy API

Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

Deployment stage

default

Deployment description

Cancel
Deploy

Nachdem die Einstellung gespeichert ist, kann die API genutzt werden.

default Stage Editor

Delete Stage
Configure Tags

Invoke URL: <https://530a81w14i.execute-api.us-east-1.amazonaws.com/default>

Settings
Logs/Tracing
Stage Variables
SDK Generation
Export
Deployment History
Documentation History
Canary

Cache Settings

Enable API cache

Default Method Throttling

Choose the default throttling level for the methods in this stage. Each method in this stage will respect these rate and burst settings. Your current account level throttling rate is 10000 requests per second with a burst of 5000 requests. [Read more about API Gateway throttling](#)

Enable throttling

Rate 10000 requests per second

Burst 5000 requests

Web Application Firewall (WAF) [Learn more.](#)

Select the Web ACL to be applied to this stage.

Web ACL
Create Web ACL

Client Certificate

Select the client certificate that API Gateway will use to call your integration endpoints in this stage.

Certificate None

Save Changes

Will man nun den *GI Client* zum neuen Webservice aufrufen, geht das (z.B. unter Linux) wie folgt.

```
\$ ./ic20_linux -u <API endpoint>
```

3 Benutzung

3.1 Benutzung über AWS

Die Benutzung des bei AWS deployten Webservices entspricht der Beschreibung unter 1.2 Schnellstart. Sollte man das Programm selber deployt haben, muss lediglich die URL des eigenen Webservices verwendet werden.

3.2 Benutzung der lokalen Version

Nachdem die Installationsanleitung unter 2.1 Bauen der Jar Datei befolgt wurde, lässt sich unser Programm mit folgendem Befehl lokal ausführen.

```
\$ java -jar target/ic_webservice-20.jar
```

3.2.1 Benutzung der graphischen Oberfläche

Man kann nun ein Spiel schrittweise automatisch, sowie selber manuell spielen. Ein blauer Ausgabertext gibt dabei Informationen über den aktuellen Zustand an. Startet man an dieser Stelle den *GI Client*, so sollte der Text ausgegeben werden, dass ein Spiel stattgefunden hat. Nun kann man durch den „Auto turn“ Knopf einen Zug von der Heuristik ausführen lassen. Optional lässt sich im Textfeld „amount“ eine gewisse Anzahl von automatischen Zügen angeben. Die Tooltips geben hierbei weitere Informationen darüber an, was welche Knöpfe machen und welche Informationen zum Ausführen der Aktion gebraucht werden.

3.2.2 Benutzung ohne graphische Oberfläche

Bekommt der Server mehr als eine Anfrage gleichzeitig oder wird das GUI über das rote X geschlossen, so schließt sich das GUI und der Server wechselt in einen Modus, in dem alle Anfragen mit unserer Heuristik beantwortet werden. Alle weiteren einkommenden Spiele, sowie das Spiel, welches derzeit in der GUI war, wird mit unserer Heuristik gespielt.

3.2.3 Beenden des Servers

Im GUI lässt sich der Server über den „Quit“ Knopf beenden. Wurde der Server im Hintergrund gestartet, und das GUI ist nicht mehr offen, so kann der Server mit einer einfachen Anfrage an localhost:50123 beendet werden, solange diese nicht vom Typ Post ist. Dazu kann beispielsweise im Browser localhost:50123 in das Adressfeld eingegeben werden.

3.2.4 Anzeige der Fehlermeldungen des Webservices

Da Fehlermeldungen über „stderr“ ausgegeben werden, muss der Webservice über eine Kommandozeile/Bash gestartet werden damit diese angezeigt werden. Startet man ihn über einen Doppelklick auf die Jar Datei so kann man sie nicht sehen.

3.3 Benutzung des Testskripts

Das Testskript, welches als `Test.py` im Wurzelverzeichnis unseres Projekts liegt, bietet verschiedene Arten die lokale als auch die von uns deployte Version zu testen. Testen heißt in dem Fall, dass man beliebig viele Spiele, auch parallel, spielt und eine Ausgabe über die aktuelle Winrate erhält. Um es zu benutzen, muss Python Version 3 installiert sein. Es funktioniert sowohl auf Windows, als auch auf Linux. Vor der Nutzung muss der *GI Client* in den gleichen Ordner des Test Skripts gelegt werden.

Um sich alle Möglichkeiten, um das Skript auszuführen, von dem Skript selber ausgeben zu lassen, führt man das Skript mit der Flag „-h“ oder „-help“ aus.

```
\$ python3 Test.py -h
```

Führt man das Skript ohne zusätzliche Flags aus, werden die ersten 100 Seeds, in 4 Threads, über den lokalen Webservice auf localhost:50123 gespielt. Dem entsprechend muss dieser gestartet sein. Wie man diesen startet, findet sich unter 2.2 Lokale Version mit GUI.

Will man den deployten Webservice testen, fügt man die Flag „-aws“ hinzu.

```
\$ python3 Test.py -aws
```

Sollte man die Anzahl der Threads, also die Anzahl der gleichzeitig laufenden Spiele ändern wollen, fügt man die Flag `--threads %AnzahlDerThreads` hinzu.

```
\$ python3 Test.py --threads 10
```

Wenn man Spielseeds aus einem bestimmten Bereich (z.B. von 1 bis 1000) spielen will, so fügt man die Flag `--range %Start %Ende` hinzu.

```
\$ python3 Test.py --range 1 1000
```

Zusätzlich ist es möglich Seeds in einer Datei anzugeben. Diese Datei muss im gleichen Ordner wie `Test.py` liegen und `seeds.txt` heißen. Die Seeds müssen zeilenweise in der Datei liegen. Durch das Hinzufügen der flag „-file“ werden die Seeds aus der Datei genutzt. Gibt man zu dieser Flag noch eine „range“ an, so wird diese ignoriert.

```
\$ python3 Test.py -file
```

Da unsere Heuristik für Aktionen mit gleichen Wert nicht deterministisch ist und der *GI Client* einen gewissen Zufallsfaktor beinhaltet, lässt sich die Flag „-consistency“ hinzufügen, welche fortlaufend die gleichen Seeds spielt, damit sich diese vergleichen lassen. Will man z.B. die Einheitlichkeit der Winrate zwischen den Seeds 60 und 70 testen, sieht der Aufruf folgendermaßen aus.

```
\$ python3 Test.py -consistency --range 60 70
```

Die Flags lassen sich beliebig kombinieren. So kann man z.B. den deployten Webservice auf folgende Weise effizient auf eine konsistente Winrate für die ersten 1000 Seeds testen.

```
\$ python3 Test.py -aws -consistency --threads 8 --range 1 1000
```

4 Lösungsidee

4.1 Entwicklung von Ideen

Um ein Gefühl für das Spiel zu bekommen, haben wir die GUI entwickelt. Diese ermöglichte es uns Spiele selber zu spielen und das Verhalten der Pathogene genauer kennenzulernen.

Außerdem haben wir ein Skript geschrieben, welches uns alle Informationen über sämtliche Städte und alle gefundenen Pathogene in den ersten 436 Seeds liefert. Zeitgleich haben wir dadurch eine Sammlung der Seeds erstellt, welche es uns ermöglichte fast beliebige Startzustände zum selbstständigen Testen unserer Heuristik zu wählen. Die Liste mit den Seeds findet sich unter 10 Anhang

Um mit diesem Grundgerüst an Informationen und Möglichkeiten effiziente Wege zum Bekämpfen der Pathogene zu finden, begann die Entwicklung der Heuristik.

Die Lösungsidee der finalen Lösung ist eine lineare Auswertung aller möglichen Aktionen. Dies findet nach jeder ausgeführten Aktion erneut statt.

Die Heuristik basiert dabei auf einzelnen Ideen, die inkrementell hinzugefügt wurden. Ob eine Verbesserung vorliegt, wurde ständig mit unserem Winrate-Testskript überprüft. Man könnte also von „Winrate driven Development“ sprechen. Das soll heißen, dass zunächst die einfachsten Ideen implementiert wurden, wie z.B. 4.5.2 Quarantäne zum Schutz der restlichen Weltbevölkerung. Häufig entstanden Ideen aus Problemsituation von einzelnen Seeds. Z.B. Seed 4 und 4.5.2 Quarantäne zum Schutz der Bevölkerung der Stadt unter Quarantäne.

Funktionierte nun eine Idee für diesen einen Seed, so konnte man sie für mehrere Spiele mit dem erstellten Testskript testen. Eine nützliche Idee sollte minimale Verbesserungen an der Winrate hervorheben. War dieser abschließende Test erfolgreich, wurde der Prozess mit der nächsten Idee fortgesetzt.

Dabei ist der größte Aufwand das Entwickeln der Ideen und das Finden von typischen Problemsituationen. Eine Vorgehensweise, welche das Finden von Problemsituationen ermöglichte, war das Testen von mehreren Seeds und das anschließende manuelle Spielen von verlorenen Seeds. Konnte man den Seed auf einfache Weise manuell gewinnen galt es zu vergleichen, was die Heuristik anders tat als man selber und diese Idee anschließend umzusetzen.

Dabei stellte sich schnell ein Problem heraus. Durch das Wählen anderer Aktionen wird der folgende Spielablauf massiv verändert, da die Aktionen den internen Zufallsgenerator des *GI Clients* beeinflussen.

Insgesamt ist es uns so gelungen eine effektive Heuristik zu entwickeln und ein umfassendes Verständnis für den internen Spielablauf des *GI Client* zu entwickeln. Wobei es dennoch einige ungeklärte Fragen gibt, z.B. was die „globalen Events“, wie z.B. „largeScalePanic“ konkret bewirken.

4.2 Verworfenе Ansätze

Zunächst erschien die Aufgabe wie ein größeres Rucksackproblem. Jede Aktion kostet eine gewisse Anzahl an Punkten, bringt aber auch einen bestimmten Wert mit sich. Es gilt also die Menge an Aktionen mit maximalem Wert zu finden. Dies stellte sich aber schnell als problematisch heraus. Eine Lösung für das Rucksackproblem ist eine Menge von Aktionen. Problematisch ist es nun eine solche Menge aufgrund möglicher Abhängigkeiten der Aktionen untereinander zu bewerten. Ein Beispiel ist eine Menge, die nur die Aktionen „Medikament entwickeln“ und „Impfstoff entwickeln“ enthält. Hier ist der Wert der Menge nicht die Summe der beiden Aktionen, da es weniger wertvoll ist ein Medikament zu entwickeln, wenn bereits ein Impfstoff entwickelt wird, oder vice versa.

4.3 Umsetzung der Gesamtidee

Um eine möglichst gute Aktion zu ermitteln, wird jeder möglichen Aktion eine Wertung zugewiesen. Um die beste Aktion zu ermitteln, werden alle Aktionen verglichen und die Aktion mit der größten Wertung wird ausgewählt. Hierbei befinden sich die Wertungen nicht in einer gleichmäßig verteilten Größenordnung. Weiterhin werden viele Aktionen ausgeschlossen indem sie null Punkte bekommen.

4.4 Pathogen Klassifikationen

Um Pathogene allgemein beurteilen zu können, werden die Pathogene anhand ihrer Eigenschaften klassifiziert und in Gruppen zusammengefasst.

4.4.1 Aktives Pathogen

Global gesehen ist ein aktives Pathogen ein Pathogen, welches noch mindestens eine Stadt infiziert hat. Lokal gesehen ist ein aktives Pathogen ein Pathogen, welches die betrachtete Stadt infiziert hat.

4.4.2 Starkes Pathogen

Ein starkes Pathogen tötet innerhalb so weniger Runden die Hälfte der Weltbevölkerung, dass das Entwickeln von Medikamenten oder Impfstoffen zu lange dauert, um eine Niederlage zu verhindern. Um als starkes Pathogen klassifiziert zu werden, muss das Produkt der numerischen Repräsentation der Werte für „Infectivity“, „Lethality“, 6 - „Duration“ und „Mobility“ größer oder gleich 220 sein.

Die Implementierung zum erkennen von starken Pathogenen findet sich in der Methode „doQuarantine“ unter 10.2 Wichtige Codeauszüge.

4.4.3 Langsames Pathogen

Ein langsames Pathogen infiziert langsam Städte und ihre Bevölkerung. Um als langsames Pathogen klassifiziert zu werden, muss das Produkt der numerischen Repräsentation der Werte für „Infectivity“ und „Mobility“ kleiner oder gleich 6 sein.

Die Implementierung zum Erkennen von langsamen Pathogenen befindet sich in der Methode „doDevVaccine“ unter 10.2 Wichtige Codeauszüge.

4.4.4 Schnelles Pathogen

Ein schnelles Pathogen infiziert schnell viele Städte und ihre Bevölkerung. Um als schnelles Pathogen klassifiziert zu werden, muss das Produkt der numerischen Repräsentation der Werte für „Infectivity“ und „Mobility“ größer 6 sein.

Die Implementierung zum Erkennen von schnellen Pathogenen findet sich in der Methode „doDevMedication“ unter 10.2 Wichtige Codeauszüge.

4.4.5 Ignoriertes Pathogen

Pathogene werden ignoriert, wenn sich vermuten lässt, dass diese am Abklingen sind und keine Gefahr mehr darstellen oder sie kein *aktives Pathogen* mehr sind. Da das Programm zustandslos sein soll, muss geraten werden, ob ein Pathogen am Abklingen ist. Die wichtigste Bedingung dafür ist, dass es ein gewisses Mindestalter besitzt. Dieses Mindestalter beträgt bei uns zehn Runden. Als zweiten Aspekt gibt es zwei kleinere Bedingung, wovon mindestens eine erfüllt sein muss, damit ein Pathogen als am Abklingen spezifiziert wird. Entweder hat das Pathogen im Durchschnitt weniger als 10% der Bevölkerung aller infizierten Städte infiziert oder maximal 5 Städte insgesamt infiziert. Sollte man jedoch genug Punkte haben und ein Pathogen hat weniger als fünf Städte infiziert, wird dieses doch nicht ignoriert. Dies dient einem schnellen Sieg, da häufig am Ende einer Partie ein Pathogen noch in wenigen Städten für mehrere Runden verweilt. Das sorgt für einen langsameren Sieg, aber kann auch dafür sorgen, dass neue Pathogene ausbrechen.

Die Implementierung zum Erkennen von aktiven Pathogenen findet sich in der Methode „ignorePathogenThisRound“ unter 10.2 Wichtige Codeauszüge.

4.5 Bewertung

Abhängig von der Art der Aktion, fließen unterschiedliche Faktoren in die Bewertung der Aktion mit ein. Soll eine Aktion nicht ausgeführt werden, so wird ihr die Wertung 0 zugewiesen. Im Folgenden eine Erläuterung, wie jede Aktion bewertet wird.

4.5.1 Runde beenden

Runden zu beenden ist eine wichtige Aktion, da man in gewissen Situationen lieber Punkte sparen will. Ihr wird stets ein Score von 1 zugewiesen. Das „Warten“ durch die Aktion entsteht durch die Methode „ignorePathogenThisRound“. So wird „Runde beenden“ ausgeführt, wenn nicht genug Punkte für die Zufallsereignisse gesammelt haben, aber alle Pathogene als *ignoriertes Pathogen* bewertet wurden. Außerdem werden einige Aktionen nicht ausgeführt, wenn eine gewisse Punktzahl unterschritten wird, da immer die Gefahr eines Ausbruchs eines starken Pathogens besteht. Für dessen fortführbare Quarantäne werden mindestens 40 Punkte gebraucht.

4.5.2 Quarantäne

Die Quarantäne ist das wichtigste Hilfsmittel gegen ein *starkes Pathogen*. Ein *starkes Pathogen* tötet innerhalb so weniger Runden die Hälfte der Weltbevölkerung, dass das Entwickeln von Medikamenten oder Impfstoffen zu lange dauert, um eine Niederlage zu verhindern. Selbst geschlossene Verbindungen oder Flughäfen verhindern nicht die Infektion aller Städte in nur einer Runde. Beispiele für solche Viren wären „Admiral Trips“ oder „N5-10“. Bei der Bewertung einer Quarantäne Aktion wird zwischen drei Zielen der Quarantäne unterschieden.

1. **Quarantäne zum Schutz der restlichen Weltbevölkerung:** Befindet sich ein *starkes Pathogen* in genau einer Stadt, so muss diese Stadt unter Quarantäne gesetzt werden. Dies verhindert die Ausbreitung des Pathogens. Solange das Pathogen in der Stadt aktiv ist und noch nicht alle Infizierten getötet hat oder diese geheilt sind, bleibt die Stadt unter Quarantäne. Unter der speziellen Bedingung, dass alle Städte bereits mit einem Pathogen infiziert sind, muss die Stadt jedoch nicht unter Quarantäne gesetzt werden. Da in jeder Stadt nur ein Pathogen gleichzeitig sein kann, ist die Ausbreitung des Pathogens nicht möglich.
Die Implementierung dieser Evaluation findet sich unter 10.2 Wichtige Codeauszüge.
2. **Quarantäne zum Schutz der Bevölkerung der Stadt unter Quarantäne:** Hat sich ein *starkes Pathogen*, trotz des ersten Quarantäneziels, in mehrere Städte ausgebreitet (praktisch passiert das nur, wenn ein *starkes Pathogen* in zwei Städten gleichzeitig zu Beginn des Spiels aktiv ist), so ist die einzige, sinnvolle Aktion die größte, nicht infizierte Stadt zu beschützen. Durch eine Quarantäne der größten Stadt werden alle Einwohner der Stadt geschützt bis keine Stadt mehr infiziert ist.
Die Implementierung dieser Evaluation findet sich unter 10.2 Wichtige Codeauszüge.
3. **Quarantäne eines neu auftretenden Pathogens:** Sind alle Pathogene am Abklingen und ein neues Pathogen taucht auf, wird die Stadt mit dem Pathogen unter Quarantäne gesetzt. Dies verhindert die Ausbreitung neuer Pathogene in der Endphase.
Die Implementierung dieser Evaluation findet sich unter 10.2 Wichtige Codeauszüge.

4.5.3 Medikament entwickeln

Medikamente sind unter anderem die wichtigsten Hilfsmittel, um die Pandemie zu besiegen. Jedoch hat Quarantäne stets Vorrang gegenüber der Entwicklung von Medikamenten. Somit wird am Anfang zunächst geprüft, ob es noch laufende Quarantäneevents gibt. Ist dies der Fall, so werden keine Medikamente entwickelt. Dies dient einerseits dazu, dass keine Punkte ausgegeben werden, welche für die Quarantäne benötigt werden. Gleichzeitig sorgt es dafür, dass keine Medikamente für ein

Pathogen unter Quarantäne entwickelt werden. Ist dies jedoch nicht der Fall, sind nun die Eigenschaften des betrachteten Pathogens entschieden. Erfüllt dieses die Eigenschaften eines schnellen Pathogens, so werden Medikamente entwickelt. Ansonsten gilt es zu betrachten, ob bereits ein gewisser Anteil der gesamten Bevölkerung mit einem langsamen Pathogen infiziert ist, was das Entwickeln von Medikamenten selbst für *langsames Pathogen* wieder sinnvoll macht.

4.5.4 Impfung entwickeln

Impfungen sind mit Medikamenten die wichtigsten Hilfsmittel, um die Pandemie zu besiegen. Jedoch hat Quarantäne stets Vorrang gegenüber der Entwicklung von Impfungen. Somit wird am Anfang zunächst geprüft, ob es noch laufende Quarantäneereignisse gibt. Ist dies der Fall, so werden keine Impfungen entwickelt. Dadurch werden keine Punkte ausgegeben, welche für die Quarantäne benötigt werden. Gleichzeitig sorgt es dafür, dass keine Impfungen für ein Pathogen unter Quarantäne entwickelt werden. Ist dies jedoch nicht der Fall, sind nun die Eigenschaften des betrachteten Pathogens entschieden. Erfüllt dieses die Eigenschaften eines langsamen Pathogens, so werden Impfungen entwickelt. Ansonsten gilt es zu betrachten, ob weniger als 10% der Gesamtbevölkerung in uninfizierten Städten leben. In diesem Fall würde man Impfstoff für *schnelles Pathogen* entwickeln, da sich dieses nicht schnell ausbreiten kann.

4.5.5 Medikament verteilen

Die Wertung zur Verteilung eines Impfstoffs in einer Stadt wird proportional zur Einwohnerzahl, welche mit dem Pathogen infiziert sind, skaliert.

Dabei wird darauf geachtet, dass stets genug Punkte übrig bleiben, um eine Stadt für zwei Runden unter Quarantäne zu setzen, falls ein *starkes Pathogen* ausbrechen sollte. Dementsprechend werden nur Medikamente verteilt, wenn mindestens 30 Punkte zur Verfügung stehen.

4.5.6 Impfung verteilen

Die Wertung zur Verteilung eines Impfstoffs in einer Stadt wird proportional zur Einwohnerzahl, welche nicht mit dem Pathogen infiziert sind, skaliert. Gegen tödlichere Pathogene wird bevorzugt geimpft, da von ihnen eine größere Gefahr ausgeht. Wurde in Städten bereits Impfstoff verteilt, so wird nicht erneut Impfstoff verteilt, da bereits alle Einwohner tot oder immun gegen das Pathogen sind.

Dabei wird darauf geachtet, dass stets genug Punkte übrig bleiben, um eine Stadt für zwei Runden unter Quarantäne zu setzen, falls ein *starkes Pathogen* ausbrechen sollte. Dementsprechend werden nur Medikamente verteilt, wenn mindestens 25 Punkte zur Verfügung stehen.

4.5.7 Zufallseignisse

Die vier Zufallseignisse in einer Stadt (Einfluss geltend machen, Neuwahlen, Hygienemaßnahmen, Informationskampagne) werden proportional zur Einwohnerzahl

und zum Verbesserungspotential bewertet. Das Verbesserungspotential ist die Anzahl an Stufen, die das jeweilige Attribut noch steigern kann. Wäre die Stärke der Wirtschaft „o“ dann wäre das Verbesserungspotential zwei.

Weiterhin werden diese Aktionen nur dann gewählt, wenn genügend Punkte vorhanden sind, damit in folgenden Runden dadurch keine wichtigere Aktion blockiert wird. Dazu muss die „doRerolls“ Bedingung erfüllt sein. Diese ist erfüllt, wenn gerade jedes Pathogen als *ignoriertes Pathogen* bewertet wurde. Da wir aber Änderungen der Heuristik stets mit dem Testen der Winrate validiert haben und Erfahrungswerte durch die Zufallsereignisse keine Verbesserung ergeben haben, sind die Konstanten hierfür auf 0 gesetzt, d.h. diese Ereignisse werden mit 0 Punkten bewertet.

4.5.8 Verbindungen sperren und Flughafen schließen

Das Testen dieser Aktionen hat ergeben, dass die Effekte der Aktionen, unabhängig von der aktuellen Spielsituation, nicht mit anderen Maßnahmen, wie der Quarantäne, konkurrieren können. Selbst bei geschlossenem Flughafen oder geschlossenen Verbindungen breiten sich Pathogene kaum beeinträchtigt aus. Bei Quarantäne tritt dieses Problem nicht auf und es befindet sich in einer gleichen Größenordnung der Punkte. Deshalb werden diese beiden Aktionen mit jeweils 0 Punkten bewertet.

5.3 Gamepaket

Wie bereits erwähnt, ist das Gamepaket zunächst für das Parsen und Speichern, der vom *GI Client* bereitgestellten Informationen, zuständig. Die Klassen *City* und *Pathogen* speichern jeweils die Information über eine Stadt bzw. ein Pathogen. Dabei werden keine zusätzlichen Informationen erzeugt, die nicht von dem *GI Client* bereitgestellt werden. Die Werte, wie „Infectivity“ von Pathogenen oder „Hygiene“ von Städten werden als Enumerate repräsentiert, welche eine numerische Repräsentation haben. Dessen Implementierung findet sich in dem Enumerate „Scale“ unter 10.2 Wichtige Codeauszüge.

Zudem werden Aktionen und Events als eigene Klassen dargestellt. Aktionen und Events enthalten alle Informationen, die eine Aktion bzw. ein Event haben kann, also als ein „generisches Event“ bzw. als „generische Aktion“. Sollte nun z.B. das Event „Flughafen schließen“ kein Pathogen beinhalten, so ist das Pathogen zu diesem Event als „null“ gespeichert. Um nun eindeutig den Typ des Events oder der Aktion zu bestimmen, gibt es jeweils ein zugehöriges Enumerate. Einmal das „ActionType“ und das „EventType“ Enumerate, welches jedem Event beim Erstellen zugewiesen wird.

Städte und Pathogene werden als Map mit ihrem Namen als Schlüssel gespeichert, während Events als Map mit ihrem Typ als Schlüssel gespeichert werden.

Zusätzliche Informationen, welche nicht vom *GI Client* direkt erzeugt werden aber trotzdem im Gamepaket gespeichert werden, sind die Map von *ignoriertes Pathogen* und die Gesamtanzahl der Bevölkerung. Die Map der *ignoriertes Pathogen* wird jede Runde neu erzeugt, da sich jede Runde ändern kann, welches Pathogen ignoriert werden soll. Die Gesamtanzahl der Bevölkerung wird an mehreren Stellen in der Heuristik benötigt, weshalb es effizienter ist diese einmal zu speichern, anstatt jedes Mal die Summe der Bevölkerung jeder Stadt zu bilden.

5.4 HTTP-Paket

Das HTTP-Paket beinhaltet den Server für die Kommunikation mit dem *GI Client*. Es besteht aus drei Klassen. Dem *AWSHandler*, dem *GameServer* und dem *GameExchange*.

Der *AWSHandler* ist für die Kommunikation mit dem *GI Client* zuständig, wenn die Jar Datei als Lambda Funktion bei AWS bereitgestellt wurde. Das Spiel wird eingelesen, evaluiert und die Aktion, welche von unserer Heuristik bestimmt wurde, wird an den *GI Client* zurückgesendet.

Der *GameServer* und das *GameExchange* sind für die Kommunikation mit dem *GI Client* zuständig, wenn die Jar Datei lokal ausgeführt wird.

Der *GameServer* ist ein HTTP Server, welcher auf localhost:50123 angesprochen werden kann. Bekommt er eine POST Anfrage, so wird diese in einem neuen Thread behandelt. Dies ermöglicht das parallele Spielen von Spielen. Zuerst wird bei der Behandlung der Anfrage das Spiel eingelesen und in unsere Objektstruktur umgewandelt. Hierbei wird, neben dem Gameobjekt, auch ein neues *GameExchange* Objekt erstellt. Das *GameExchange* Objekt bündelt die Schnittstelle zum Antworten an den *GI Client* und das gepackte Game Objekt. Letzendlich wird die Art der Behandlung

der Anfrage ausgewertet. Hierbei gibt es drei mögliche Arten eine Antwort für den *GI Client* zu generieren.

1. Das GUI ist geschlossen oder es werden mehrere Spiele gleichzeitig gespielt. In beiden Fällen wird automatisch eine Antwort mit der Heuristik bestimmt.
2. Im GUI wurde gesagt, dass eine Aktion mehrmals ausgeführt werden soll. Beispielsweise sollen die größten 10 Städte geimpft werden. Gilt (1.) nicht so werden diese Aktionen abgearbeitet.
3. Das GUI wird benutzt, um eine Aktion zu finden.

5.5 Solverpaket

Das Solverpaket beinhaltet drei Komponenten. Die ActionHeuristic- und die ConstantSetup-Klasse, sowie das GameEvaluator-Interface. Die ActionHeuristic oder auch Heuristik ist das „Herzstück“ des Programms. Ihre Aufgabe ist es für das aktuelle Spiel, aus allen möglichen Aktionen, die Beste zu finden und dann auszuführen.

Wie diese Auswertung im Detail passiert, findet sich unter 4.5 Bewertung.

Die ConstantSetup-Klasse lädt die Konstanten der Heuristik, wie z.B. ein Threshold, der überprüft, ob ein Pathogen unter Quarantäne gesetzt werden muss.

Hier findet sich gleichzeitig die wesentliche Nutzung der FileHandler-Klasse, da die Konstanten in der „constants.txt“ Datei im Resources-Ordner liegen.

Das GameEvaluator-Interface dient dazu, um dynamisch Antworten auf eine bestimmte Spielsituation zu finden. Hierzu muss die „solve“ Methode implementiert werden. Die „solve“ Methode bekommt ein Game Objekt übergeben und gibt ein Action Objekt als Antwort auf diese Spielsituation zurück. Eine Beispielimplementierung für das GameEvaluator Interface wäre eine Methode, welche immer als Aktion Medikamente in der Stadt mit den meisten Infizierten Einwohner verteilt. Dadurch kann eine dynamische Aktion direkt mehrfach an den GameServer weitergegeben werden. Diese wird dann bei der Anfrage des *GI Clients* ausgewertet.

5.6 GUI-Paket

Das GUI-Paket enthält als einzige Klasse den GUI Kontroller. Der GUI Kontroller enthält überwiegend „On-Action-Methoden“ für die einzelnen GUI Elemente („Buttons“, „Choiceboxes“, etc.).

Die GUI wird nicht im Code selbst erzeugt, sondern aus einer FXML Datei geladen, welche die unveränderlichen Informationen der GUI enthält, wie z.B. Struktur, Positionen der Elemente, Texte, Tooltips, etc. Die GUI basiert dabei auf dem JavaFX Paket. Dieses ist in der Standardbibliothek der genutzten Javaversion enthalten.

Der Zweck der GUI ist das manuelle Spielen von einzelnen Spielen. Sie schließt sich, sobald das Programm mehr als zwei, zeitgleich laufende Spiele erkennt.

Die GUI bietet die Visualisierung des einen aktuellen Spielzustandes, wobei sich einzelne Pathogene, Städte und weitere Informationen gefiltert anzeigen lassen. Mehr dazu findet sich unter 3.2 Benutzung der lokalen Version

5.7 IO-Paket

Das IO-Paket enthält nur die Klasse `FileHandler`. Über den `FileHandler` wird die Interaktion (Lesen, Schreiben) mit Dateien ermöglicht. Die dafür zuständigen Methoden sind `readFile` oder `writeFile`. Es wird beispielsweise zum Lesen der Konstanten oder dem Schreiben des exportierten Bilds benutzt.

6 Wartung und Weiterentwicklung

6.1 Die Bewertungsfunktion anpassen

Die Bewertungsfunktion ist zwar das Herzstück unserer theoretischen Überlegungen, wurde aber bis zum Ende kontinuierlich verbessert, was eine zukünftige Verbesserung nicht ausschließt.

Um die Heuristik anzupassen oder zu erweitern, muss die Bewertung einzelner Aktionen in der „getScore“ Methode angepasst oder hinzugefügt werden. In der Methode sollte man nun nach dem Typ der Aktion in der switch-Anweisung suchen und die Modifikation der Variable „score“ anpassen.

6.2 Hinzufügen neuer Attributwerte

Um neue Attributwerte zusätzlich zu den aktuellen, wie bspw. „-“ oder „o“, müssen folgende Schritte durchgeführt werden.

1. Im Enum „Scale“ das neue Zeichen eintragen. Siehe 10.2 Wichtige Codeauszüge
2. In der Methode „parse“ von Scale die String Repräsentation festlegen.
3. In der Methode „getValue“ von Scale den numerischen Wert festlegen.
4. In der Methode „toString“ von Scale den von der Methode „parse“ geparsen String zum neuen Wert angeben.

6.3 Hinzufügen neuer Konstanten

Um eine neue Konstante in der Heuristik verwenden zu können, muss diese in die Resourcendateien „src/main/resources/constants.txt“ im folgenden Format pro Zeile eingetragen werden:

Name Wert

Name kann eine beliebig lange Zeichenkette sein und **Wert** kann beliebige Double Werte annehmen. Leerzeilen und Zeilen die mit „/“ beginnen, werden beim Einlesen der Konstanten ignoriert. Um auf die Konstante zugreifen zu können, muss die Methode „constants.get(**Name**)“ benutzt werden.

6.4 Hinzufügen neuer Events

Um ein neues Event hinzuzufügen, müssen folgende Schritte durchgeführt werden:

1. Einen neuen Wert dem EventType Enum hinzufügen.
2. In der containedByCity Methode festlegen, ob das Event zu einer bestimmten Stadt gehört.
3. In der isMultipleEventType festlegen, ob das Event mehrere Male in einer Stadt vorkommen kann.
4. Analog zu den anderen Events in der parseEvent Methode das Event einlesen.

6.5 Hinzufügen neuer Aktionen

Um eine neue Aktion hinzuzufügen, müssen folgende Schritte durchgeführt werden:

1. Einen neuen Wert in das ActionType Enum einfügen.
2. In der getCosts Methode die Kosten definieren.
3. In generateAllPossibleActions alle möglichen Aktionen dieses Typs hinzufügen.
4. In der getScore Funktion die Aktion bewerten.

Möchte man zusätzlich die Aktion in der Heuristik in Betracht ziehen, muss man in der Methode getScore in ActionHeuristik in der switch-Anweisung einen neuen case für den ActionType der hinzugefügten Action hinzufügen.

7 Validierung

Für die Validierung unsere Lösung wurde ein Wrapper (das Testskript) für den *G/Client* geschrieben, welcher beliebige Seeds spielt und die Winrate berechnet. Hierbei wird bei den ersten 1000 Seeds eine Winrate von etwa 63% ($\pm 1,5\%$) erreicht, was eine erhebliche Verbesserung zu den 21,79% ist, wenn man jede Runde beenden würde. Das eine deutliche Verbesserung vorliegt, nämlich ca. 41%, deckt sich mit unseren theoretischen Überlegungen aus dem Kapitel Lösungsansatz.

8 Reflexion

8.1 Software Architektur

Insgesamt ist es uns gelungen ein robustes und leicht erweiterbares Grundgerüst zu entwickeln (siehe dazu 6 Wartung und Weiterentwicklung). Nur das GUI ist sehr starr und unflexibel geworden. Da wir bisher erst einmal ein Projekt mit GUI umgesetzt haben, ist es dafür sehr ansehnlich geworden. Für eine gute Implementierung der Größenänderung brauchen wir vermutlich mehr Erfahrung. Auch, dass das komplette GUI nur in einer Klasse und der FXML Datei programmiert ist, ist vermutlich nicht die beste Art der Umsetzung.

8.2 Bibliotheks Nutzung

Uns ist erst sehr spät aufgefallen, dass wir eine sehr alte und nicht gepflegte Bibliothek zum Einlesen von JSONs benutzen. Die Entwicklung von JSON-Simple wurde 2012 eingestellt und durch das neue GSON ersetzt. Außerdem benutzen wir den HTTPServer von Oracle, weswegen unser Programm nur mit Oracle Java genutzt werden kann. Hier hätten man eine externe Bibliothek nehmen können und diese einfach mit Maven einbinden können. Da wir vor diesem Projekt noch nie selber Build Tools benutzt haben, war uns jedoch nicht bewusst, wie leicht externe Bibliotheken eingebunden werden können. Daher erschien der einfache Import von „com.sun.net.httpserver“ als bessere Option.

8.3 Programmierstil

Da wir uns an die Java Code Konventionen gehalten haben, sollte der Großteil unseres Codes leserlich sein. Allerdings ist die Benutzung von Streams teilweise ausgeartet, woher auch unser Name kommt. Hier hätte man leserlicher schreiben können.

Beispiel für einen zu komplexen Stream

```
1 cities.stream().filter(c -> !this.showDistinctCityCB.getValue().contains("Healthy") || !c.isInfected())
2   .filter(c -> !this.showDistinctCityCB.getValue().contains("Infected"))
3     || c.isInfected(selectedPathogen) && selectedPathogen != null
4     || selectedPathogen == null && c.isInfected())
5   .filter(c -> selectedCity == null || selectedCity == c
6     || (selectedCity.getConnections().contains(c) && connectionBox.isSelected()))
7   .forEach(c -> drawCity(c));
```

8.4 Heuristik

Insgesamt sind wir zufrieden mit dem, was unsere Heuristik leistet. Jedoch ist uns leider zu spät eine bessere Idee zur Bewertung der Aktionen eingefallen. In der von uns implementierten Version hat die Punktzahl, die jede Aktion bekommt, kaum Bedeutung. Gute Aktionen bekommen mehr, schlechte Aktionen weniger bis keine Punkte. Bei einer erneuten Implementierung würden wir differenzierter an die Bewertung gehen. Jeder Aktion wird eine Punktzahl zugeordnet, die repräsentiert wie viele Menschen durch sie gerettet werden könnten. Dadurch entsteht eine Struktur, die auch mit weiteren Aktionen intuitiv erweiterbar ist. Weiterhin kann mithilfe eines Algorithmus die beste Teilmenge von Aktionen ausgewählt werden, die die meisten Leute rettet und ausführbar ist (Rucksackproblem). Aus dieser Teilmenge wird die effektivste Aktion ausgewählt. In der folgenden Runde würde der Algorithmus im Idealfall wieder die gleiche Teilmenge, nur ohne die bereits ausgeführte Aktion, wieder auswählen.

8.5 Organisatorisch

Rückblickend liegt unser größtes Verbesserungspotential in der Organisation des Ablaufs des gesamten Projekts. Wir haben zunächst unseren theoretischen Lösungsansatz besprochen. Dann aber ohne Absprache parallel „drauf los programmiert“, was letztendlich dazu führte, dass Teilmodule wie der Knapsack und Bewertung von Mengen von Aktionen nicht mehr aufeinander abgestimmt waren. Dies führte letztendlich zum Entfernen vom Knapsack. Problematisch war auch, dass wir keine Aufgaben zugewiesen haben, es also oft unklar war, wer für was verantwortlich ist. Auch würden wir nächstes mal die Arbeit besser über den gesamten Zeitraum verteilen. Kurz nach Veröffentlichung der Aufgabenstellung herrschte eine Anfangseuphorie. In der Mitte des Semesters wurde dann kaum noch gearbeitet und zum Ende, just zu diesem Zeitpunkt, wurden wir vom Abgabetermin angetrieben, was die Produktivität nun nochmal deutlich steigerte.

9 FAQ

Q: Wieso habt ihr euch Team Stream genannt?

A: Der Name hat sich im Laufe des Projekts ergeben. Zunächst standen Namen wie Team OneLiner im Raum, weil wir es geschafft haben viele Funktionen auf eine Zeile Code zu kondensieren. Diese „OneLiner“ bestanden aber häufig aus einer Verkettung von `java.util.stream` Funktionen. Dies nahm dann schließlich so ein Ausmaß an, dass der Großteil aller `for`-Schleifen durch Streams ersetzt wurden und sich Team Stream durchsetzte.

Q: Wieso hat das Programm die Ausgabe **Could not create server**?

A: Der Server konnte nicht erstellt werden. Vermutlich läuft bereits ein anderer Server auf dem Port 50123.

Q: Wieso hat das Programm die Ausgabe **Event not implemented**?

A: In der von dem *GI Client* übermittelten Runde ist ein Event aufgetreten, welches von uns nicht implementiert wurde.

Q: Wieso hat das Programm die Ausgabe **Error while reading body**?

A: In der Theorie nicht erreichbare Fehlermeldung. Kann durch interne Fehler oder externe Manipulation entstehen.

Q: Wieso hat das Programm die Ausgabe **Could not load constants**?

A: Dies ist der Fall, wenn die Datei „constants.txt“ fehlerhaft ist. Es existiert also eine Zeile, die nicht dem gewünschten Format entspricht, also weder eine Leerzeile, noch eine Zeile beginnend mit „//“ ist oder nicht dem Format [Name Wert] entspricht. Name ist eine beliebige Zeichenkette und Wert eine Double-Zahl.

Q: Wieso hat das Programm die Ausgabe **Error while sending response**?

A: Beim Schicken der auszuführenden Aktion an den *GI Client* ist ein Fehler aufge-

treten. Meistens passiert dies, wenn der *GI Client* geschlossen wurde bevor ein Spiel zuende gespielt wurde.

Q: Wieso hat das Programm die Ausgabe **Blocking deque LOCK was executed?**

A: Werden mehrere Spiele gleichzeitig ausgeführt, wird das GUI deaktiviert und gesperrt. Kommt diese Ausgabe, wurde die Sperre ignoriert. Sollte dies passieren, gibt es vermutlich ein Problem bei der Synchronisation der Threads.

Q: Wieso hat das Programm die Ausgabe **Got a non POST request: Closing server!?**

A: Der Server beendet sich, sobald er eine Anfrage bekommt, die nicht vom Typ POST ist. Dies dient zum ausschalten, wenn der Server im Hintergrund läuft. Wird diese Nachricht ausgegeben, so hat der Server eine solche Anfrage bekommen, die den Server beendet hat.

Q: Wieso beendet sich die GUI, wenn ich den *GI Client* starte?

A: Die GUI soll sich schließen, wenn mehrere Spiele gleichzeitig laufen. Beim Starten des Testskriptes oder dem alleinigen starten mehrerer *GI Clients* schließt sich die GUI und der Webservice wechselt in den „Auto play“ Modus. Dies ist der beabsichtigte Fall.

Sollte man den *GI Client* beenden, während der Webservice auf eine Antwort wartet und anschließend einen neuen *GI Client* starten, so erkennt der Webservice mehrere zeitgleiche Spiele und wechselt seinen Modus.

Q: Ich habe eine große Menge an automatischen Zügen spielen lassen und der *GI Client* wurde beendet.

A: Das bedeutet, dass das Spiel zuende ist. In diesem Fall sieht man in der blauen Ausgabe des GUIs das Ergebnis des Spiels.

Q: Wie viel kostet es, wenn ich die Funktion bei AWS Lambda bereitstelle und benutze?

A: Das alleinige Bereitstellen kostet nichts. Man bezahlt pro Aufruf der Funktion.

Genaue Angaben zu den Kosten können wir nicht machen, aber um eine Größenordnung zu nennen: Bei allen unseren Tests haben wir insgesamt 1 Cent ausgegeben. Genaue Werte lassen sich leicht mit dem von AWS zur Verfügung gestellten *Kostenrechner* berechnen.

10 Anhang

10.1 Herausgefundene Daten über GI client

Unser Programm funktioniert für allgemeine Pathogene und Städte. Da dies jedoch nicht gefordert ist, ist im Anhang eine Liste der Informationen, die wir über die existierenden Pathogene und Städte gesammelt haben.

10.1.1 Pathogene

Name	Infectivity	Mobility	Duration	Lethality	Average duration in a city
Admiral Trips	++	+	-	++	7
Azmodeus	o	o	o	o	12
Coccus innocuus	o	--	o	--	12
Endoictus	-	o	+	--	24
Hexapox	o	+	o	o	15
Influenza iutiubensis	++	++	--	--	1
Methanobrevibacter co	o	++	--	-	1
Moricillus ☠	-	o	-	+	17
N5-10	+	++	o	+	9
Neurodermantotitis	+	o	o	o	12
Phagum vidiianum	o	o	+	+	18
Plorps	o	o	+	-	14
Procrastinalgia	-	--	++	--	30
Rhinonitis	o	-	-	--	9
Saccharomyces cerevisi	o	o	o	-	13
Shanty	+	-	o	-	10
Xenomonocythemia	-	--	o	++	33
Φthisis	o	o	o	o	17

10.1.2 Städte

name	latitude	longitude	population	connection	economy	government	hygiene	awareness	sum population	percent of population
35.682386	139.753906	37555	[Reykjavik; '上海市': 'القاهرة'; '北京市': 'Jakarta'; 'New Orleans'; 'El Paso'; '西安市'; 'الرياض'; 'Köln']	+	+	+	+	+	37555	4.37
重庆市	29.43471	106.904271	30480	[Atlanta; 'تونس': 'Pristine'; 'Mocaa'; 'Quito'; 'Miami'; 'Tirana'; '香港'; 'Manaus'; '长春市': 'Houston']	0	+	++	+	69035	8.99
Jakarta	-6.207899	106.843429	29960	[Tijuana; 'Kebenhavn'; '東京'; '香港'; 'Nantes'; 'İstanbul'; 'Milano'; 'Los Angeles']	+	0	+	0	97995	12.96
开罗	19.074411	72.89079	28860	[Atlanta; 'Sevilla'; 'Los Angeles'; 'Washington D.C.'; 'Cairo-Terepdyr'; 'Nairobi'; 'دمشق'; 'Dallas'; 'نيويورك']	0	0	+	0	126855	16.77
上海市	31.228743	121.462987	24240	[الغردقة; 'Port-au-Prince'; '昆明市': 'السراة'; 'Nassau'; '武汉市': 'Chicago'; '東京'; 'Los Angeles'; 'بغداد'; '北京市': '开罗'; '広島市']	+	++	+	+	151095	19.98
开罗	28.615314	77.711043	21800	[Köln; 'Copenhag'; 'تونس': 'Pristine'; 'Hå Nø'; 'New York City'; 'Berlin'; '上海市': 'Dallas']	0	0	0	+	172895	22.86
北京市	39.900508	116.420775	21540	[上海市': 'Montevideo'; 'Houston'; 'Buenos Aires'; '开罗'; 'Georgetown (Guyana); 'Kebenhavn'; '武汉市': '東京']	+	+	++	+	194435	25.71
成都市	30.57868	104.072105	15710	[Rio de Janeiro; 'İstanbul'; 'Manaus'; 'Tirana'; 'Atlanta'; 'بغداد'; 'Vilnius'; 'Warszawa'; 'Jakarta']	+	+	++	0	210145	27.78
Los Angeles	34.062266	-118.270315	13311	[Jakarta; 'Bujumbura'; 'New York City'; '长春市': 'Monrovia'; 'Vancouver'; '开罗'; 'Paramaribo'; '上海市': 'Santiago de Chile']	++	+	+	+	223456	29.54
Mocaa	55.754797	37.622831	11920	[Kinshasa; 'بغداد'; '重庆市': 'Reykjavik'; '서울'; 'Tijana'; 'Tepaa'; 'Yaounde'; '临沂市': 'Kampala']	+	+	+	+	235376	31.12
Kinshasa	-4.398981	15.266437	11860	[Tijana; 'القاهرة'; 'Hamburg'; 'Mocaa'; 'Washington D.C.'; 'الرياض'; 'Riyko'; 'Ankara']	-	-	+	+	247238	32.69
武汉市	30.581732	114.303534	11060	[Köln; 'Copenhag'; 'Rome'; 'Monrovia'; '上海市': 'Santiago de Chile'; '北京市': 'Chicago']	0	+	0	0	265316	34.16
Lima	-12.046764	77.052347	10480	[Bogotá; 'Lima'; 'Paris'; 'Stockholm'; 'Gdańsk'; 'Conaky'; 'Miami'; 'Tirana'; '香港']	0	+	0	0	269779	35.54
临沂市	35.103643	118.349429	10440	[الغردقة; 'New York City': 'Brasilia'; 'Lima'; 'Hosoo'; 'Copenhag'; 'Mocaa'; '长春市': 'İstanbul'; 'Porto Novo']	++	+	+	+	270236	36.92
서울특별시	37.55987	126.981275	9767	[Bruxelles; 'Toulouse'; 'Terpooanooc-Kaa-arcooi'; 'Washington'; 'Seoul'; 'Chicago'; 'القاهرة'; 'Caracas'; 'Ankara'; '开罗']	++	++	+	+	286003	38.21
Chicago	41.870165	-87.65026	9555	[القاهرة; 'الكويت': '上海市': '서울'; 'بغداد'; 'Anchorage'; 'Diyaaide'; 'Ciudad de México'; 'Lundaa'; '広島市']	0	+	0	0	298558	39.47
开罗	23.813919	90.413144	8906	[Asunción; 'القاهرة'; 'Bogotá'; '开罗'; 'الرياض'; '北京市': 'Winnipeg'; '昆明市': 'Budapest'; '서울'; '临沂市']	+	0	0	0	307464	40.85
Ciudad de México	19.432127	-99.146388	8855	[Lima; 'Libreville'; 'Kigali'; 'Chicago'; 'Washington D.C.'; 'Ankara'; 'Conaky'; 'Dallas']	0	+	+	+	316319	41.82
تونس	35.694502	51.377963	8694	[Rio de Janeiro; '重庆市': 'Chicago'; 'Tirana'; 'Snagovcoot'; '武汉市': 'بغداد'; '2006'; 'Roma'; 'Jakarta']	+	0	+	+	325013	42.97
西安市	34.341878	108.942458	8628	[서울특별시; 'قطيف'; 'Tegucigalpa'; '東京'; 'Miami'; 'Dallas'; 'Buenos Aires'; 'Oslo'; '开罗'; 'Pot Mosb'; '서울특별시'; '成都市']	0	+	0	+	333641	44.1
New York City	40.71354	-74.04286	8623	[Bujumbura; 'Los Angeles'; 'Rio de Janeiro'; '开罗'; 'Pot Mosb'; 'München'; 'Harare'; '临沂市']	0	0	0	0	342264	45.25
الغردقة	15.504181	32.554741	8364	[Nashville; '東京'; '昆明市': 'Nairobi'; 'Paramaribo'; '上海市': '香港'; 'Snagovcoot']	0	-	-	0	350628	46.36
กรุงเทพมหานคร	13.759911	100.539718	8281	[Kigali; 'Mico'; 'Bujumbura'; '临沂市': '成都市': 'Jakarta'; 'Ciudad de Panamá'; '서울특별시'; 'قطيف']	+	0	+	++	358909	47.45
London	51.507194	-0.127526	8136	[Atlanta; 'Lima'; 'Paris'; 'Stockholm'; 'Gdańsk'; 'Conaky'; 'Miami'; 'Tirana'; '香港']	0	+	++	+	367045	48.53
科伦坡	3.138161	101.679091	8063	[Kinshasa; 'Bogotá'; 'Chicago'; 'Luanda'; 'Belfast'; '東京'; 'Ljubljana'; 'Maputo'; '开罗']	0	+	+	+	375108	49.59
长春市	43.814754	125.306995	7676	[München; '临沂市': 'Hå Nø'; 'Ljubljana'; 'Warszawa'; 'Los Angeles'; 'Libreville'; '重庆市']	++	+	+	+	382782	50.61
بغداد	33.16831	44.365516	7665	[서울특별시; 'Mocaa'; 'Mocaa'; 'Abiyoh'; 'القاهرة'; '上海市': '开罗'; 'Bogotá'; 'Hå Nø'; 'La Habana'; 'Washington D.C.'; 'Budapest'; 'Houston'; 'Mymaaok']	-	-	-	0	390447	51.62
香港	22.32477	114.15932	7392	[重庆市': 'Lima'; 'Ankara'; 'Jakarta'; 'Antananarivo'; 'Windhoek'; 'Terpooanooc-Kaa-arcooi'; 'Caracas'; 'الغردقة']	+	+	0	+	397839	52.6
Dallas	32.77903	-96.796782	7233	[开罗'; '开罗'; '开罗'; '开罗'; 'Bujumbura'; 'Ciudad de México'; 'Bergen'; 'بغداد'; 'Freetown']	0	+	++	+	405072	53.55
Bogotá	4.711245	-74.074171	7181	[Ljubljana; 'Bratslava'; 'Lima'; 'Dakar'; 'بغداد'; 'Rio de Janeiro'; 'Kebenhavn'; 'Hå Nø'; 'Ankara'; 'Brasilia'; 'Santiago de Chile'; '开罗']	0	0	0	+	412252	54.55
Houston	29.763251	-95.387592	6773	[الغردقة; 'Thopropuak'; 'Yamoussoukro'; 'Buenos Aires'; '北京市': 'Luanda'; '서울특별시; 'قطيف'; '重庆市': 'القاهرة']	0	+	0	0	419028	55.4
昆明市	24.883738	102.8255	6728	[القاهرة; '2006'; '上海市': 'القاهرة'; '开罗'; 'London'; 'Bucuresti'; 'Kigali']	0	+	+	++	423754	56.29
Hå Nø	21.02794	105.834989	6440	[El Paso; '长春市': 'Caracas'; 'القاهرة'; 'Libreville'; 'Bogotá'; '开罗'; '서울'; 'بغداد']	0	+	0	+	432023	57.14
Rio de Janeiro	-22.99001	-43.194678	6320	[Libreville; 'Bogotá'; 'Hyp-Cymrak'; 'New York City'; 'Amsterdam'; '成都市': 'تونس'; 'Brazzaville']	0	0	0	0	438263	57.98
Washington D.C	38.907165	-77.042004	6132	[Gaborone; 'Windhoek'; 'Paris'; 'Ciudad de México'; 'Recife'; '开罗'; 'Kinshasa'; 'بغداد']	0	0	0	0	444655	58.79
Atlanta	33.748233	-84.389707	5790	[New Orleans; 'London'; 'Oslo'; 'Atlanta'; 'Ciudad de Panamá'; '重庆市': 'Bujumbura'; '开罗'; '成都市']	0	0	0	0	450445	59.55
قطيف	1.366398	103.8455	5709	[İstanbul; 'Houston'; 'Lundaa'; 'بغداد'; 'Toukexet'; 'Snagovcoot'; '西安市': 'Philadelphia']	+	+	0	+	456154	60.31
Santiago de Chile	-33.454952	-70.675157	5614	[武汉市'; 'مدينة الجزائر'; 'Miami'; 'Edmonton'; 'Los Angeles'; 'القاهرة'; 'Bogotá'; 'بغداد']	0	+	+	+	461768	61.05
Miami	25.760749	-80.194785	5565	[Pittsburgh; 'London'; '西安市': '重庆市': 'Köln'; 'بغداد'; '2006'; 'Honolulu'; 'Belfast'; 'Santiago de Chile']	0	+	+	+	467333	61.79
Ankara	39.932257	32.865068	5445	[Kinshasa; 'Dakar'; 'Bogotá'; 'Belfast'; 'Sydney'; 'Göteborg'; '서울'; '서울특별시']	+	0	0	0	472778	62.51
الرياض	24.723684	46.682419	5188	[Mocaa; 'Houston'; '서울특별시'; '开罗'; 'قطيف'; 'Ciudad de Panamá'; 'Portland'; 'Mico']	++	+	+	++	477966	63.19
Melbourne	-37.807498	144.972085	4936	[Cincinnati; 'Dodoma'; 'Lungsoo ng Maynila'; 'Abuja'; 'Riga'; '乌鲁木齐'; 'Juba']	+	+	+	+	482902	63.84
Cairo-Terepdyr	59.936461	30.306277	4880	[Köln; 'Toulouse'; '开罗'; 'مادجاسكار'; 'Honolulu'; 'Ouagadougou'; 'Köln'; 'Oslo'; 'Niamyei']	+	+	+	+	487782	64.49
San Francisco	37.772894	-122.425584	4679	[Belfast; 'San Juan'; 'Lungsoo ng Maynila'; 'Chisinali'; 'Philadelphia'; 'Porto Alegre'; 'Niamyei'; 'Perth']	0	+	+	+	492461	65.11
Phoenix	33.451697	-112.074271	4682	[Bloemfontein; 'بغداد'; 'Toukexet'; 'Strasbourg'; 'Bak'; 'القاهرة'; 'Libreville'; 'Mico'; 'Harare']	0	+	+	++	497123	65.72
Sydney	-33.870735	151.194245	4627	[München; 'Honolulu'; 'Gdańsk'; 'Ankara'; 'القاهرة'; 'القاهرة']	+	++	+	+	501750	66.34
كابل	34.549668	69.207336	4600	[Bak'; 'Seattle'; 'Edmonton'; 'Milano'; 'Minneapolis'; 'مادجاسكار'; 'Antananarivo'; 'Bern']	0	-	0	-	506350	66.94
Detroit	42.33112	-83.044962	4298	[Snagovcoot; '开罗'; 'Freetown'; 'Manaus'; 'Tegucigalpa'; 'Cleveland'; 'Domeaok']	0	0	0	+	510648	67.51
Brasilia	-15.794443	-47.881742	4292	[Litzeburg; 'القاهرة'; '临沂市': 'القاهرة'; 'Denver'; 'Bogotá']	0	0	0	0	514940	68.08
عزل	31.951697	35.912604	4008	[Bloemfontein; 'Bratslava'; '広島市'; 'Montreal'; 'Brasilia']	-	0	0	0	518948	68.61
مدينة الجزائر	35.72441	44.191022	3937	[Terpooanooc-Kaa-arcooi; 'القاهرة'; '2006'; 'Köln'; 'Miami'; 'بغداد']	0	-	-	-	522885	69.1
Seattle	47.602296	-122.340698	3799	[Asunción; 'Brazzaville'; 'بغداد'; '澳門'; 'Praha'; '开罗']	0	+	+	++	526684	69.63
Berlin	52.520861	13.409419	3748	[Minneapolis; '开罗'; 'Luanda'; 'Bamako'; 'Sevilla'; '开罗']	+	+	+	+	530432	70.13
乌鲁木齐市	43.826398	88	3520	[القاهرة; '2006'; 'Kampala'; 'Melbourne'; 'Toukexet'; 'Bak'; 'Kebenhavn'; 'Ynaa-Gaarp'; 'Stockholm']	0	+	+	+	533952	70.59
مدينة الجزائر	36.730492	3.0878	3416	[El Paso; 'Ouagadougou'; 'Nairobi'; 'Ljubljana'; 'Santiago de Chile']	0	+	0	0	537368	71.05
اسلام آباد	8.985552	38.7572	3352	[Yaounde'; 'Caracas'; 'Asunción'; 'Vancouver'; 'New Orleans'; 'الرياض'; 'Terpooanooc-Kaa-arcooi']	0	+	+	+	540720	71.49
开罗	39.041651	125.761336	3271	[القاهرة; 'Recife'; 'Seattle'; 'Port-au-Prince'; 'Cape Town'; 'Nassau']	+	+	+	+	543991	71.92
Madrid	40.413302	-3.70289	3174	[Boston; 'Quito'; 'Accra'; 'Yamoussoukro'; 'Hosoo'; 'بغداد'; 'Charlotte'; 'Aggabaf']	+	+	++	+	547165	72.34
Nairobi	-1.292524	36.816152	3134	[Gdańsk; 'القاهرة'; 'الغردقة'; 'Freetown'; 'Manaus'; 'Tegucigalpa'; 'Cleveland'; 'Domeaok']	+	+	0	0	550299	72.76
Kia	50.451382	30.520325	2952	[Cairo-Terepdyr'; 'Mymaaok'; 'Brasiane'; '東京'; 'Conaky'; 'Pretoria']	+	+	+	0	555251	73.15
Toronto	43.651023	-79.383181	2830	[London; 'Winnipeg'; 'الرياض'; 'San Juan'; 'Ouagadougou'; 'Wien'; 'Milano'; 'Austin']	++	+	+	+	556181	73.53
Buenos Aires	-34.625654	-58.429857	2690	[Lundaa; 'Bogotá'; 'القاهرة'; '西安市'; 'Houston'; 'Lima']	0	+	+	+	562071	74.05
Roma	41.902337	12.453997	2872	[Atlanta; 'Winnipeg'; '武汉市'; 'Abuja'; 'Bloemfontein'; 'القاهرة']	0	0	0	0	561943	74.29
Denver	39.737331	-104.993821	2853	[开罗'; '台北'; 'Georpat'; 'La Habana'; 'Brasilia'; 'Luba'; 'Toulouse']	++	0	+	+	564796	74.67
Abuja	9.077381	7.401908	2750	[Libreville; 'Boston'; 'Roma'; 'Melbourne'; 'Warszawa']	0	-	-	-	567548	75.04
台北	25.040903	121.553061	2674	[Ciudad de Guatemala; '开罗'; 'Charlotte'; 'Denver'; 'Madrid'; 'El Paso'; 'Las Vegas'; 'Bamako'; 'Porto Alegre']	+	0	+	0	570220	75.39
Luanda	-8.168887	13.232604	2572	[Houston; 'Atlanta'; 'Hamburg'; 'Chicago'; 'الرياض'; 'Berlin']	0	0	0	0	572792	75.73
澳門	24.478244	118.08262	2520	[Brazzaville; 'القاهرة'; 'Bucuresti'; 'Kigali'; 'Salt Lake City'; 'Budapest']	+	+	0	0	575312	76.06
San Juan	18.46414	-66.10133	2509	[Accra; 'Dublin'; 'San Francisco'; 'Brisbane'; 'La Habana'; 'Bloemfontein']	0	+	+	0	577821	76.39
Charlotte	35.23056	-80.83915	2474	[Ciudad de Guatemala; 'Nantes'; '台北'; 'Vancouver'; 'Madrid'; 'Tallinn']	0	+	+	0	580295	76.72
Vancouver	49.282804	-123.121846	2463	[Los Angeles; 'Charlotte'; 'Wellington'; 'Gdańsk'; 'اسلام آباد']	+	+	+	+	582758	77.05
Ciudad de Guate	14.627538	-90.514192	2450	[Las Vegas; 'Managua'; 'Winnipeg'; '台北'; 'Ottawa'; 'Charlotte']	0	+	+	+	585208	77.37
Lusaka	-15.385705	28.320748	2404	[Yamoussoukro'; '开罗'; 'Mymaaok'; 'Niamyei'; 'Monrovia'; 'Montevideo']	0	-	-	-	587614	77.69
Toukexet	41.302633	69.244314	2393	[Aggabaf; 'Phoenix'; '乌鲁木齐'; '서울특별시; 'قطيف'; 'Stockholm']	0	0	0	0	590007	78.08
Brisbane	-27.46719	153.017885	2280	[San Juan; 'Nantes'; 'Köln'; 'Quito'; 'New Orleans']	++	++	+	+	592287	78.31
Bak	40.407905	49.864914	2236	[Montreal; 'Phoenix'; 'Wellington'; '乌鲁木齐'; 'El Paso'; 'كابل'; 'Ynaa-Gaarp']	0	+	+	+	594523	78.6
Ouagadougou	12.368182	-1.518315	2200	[Toronto; 'Cairo-Terepdyr'; 'Christchurch'; 'Nashville'; 'Albuquerque'; 'الغردقة'; 'بغداد']	-	-	-	0	596723	78.89
Cincinnati	39.102491	-84.505384	2165	[Cleveland; 'Barcelona'; 'Manaus'; 'Pristine'; 'Georpat']	0	+	+	+	598888	79.18
Paris	48.865679	2.351178	2141	[Barcelona; 'Reykjavik'; 'Washington D.C.'; 'Dublin'; 'London']	++	+	++	++	601029	79.46
La Habana	23.112576	-82.366307	2122	[Kingston; 'Denver'; 'Bamako'; 'بغداد'; 'Anchorage'; 'San Juan']	0	+	+	+	603151	79.74
Kansas City	39.093715	-94.574376	2105	[Tijana; 'Perth'; 'Bucuresti'; 'Manaus'; 'القاهرة'; 'Lisboa'; 'Nassau']	+	+	+	+	605256	80.02
Domeaok	-6.150345	35.781349	2084	[Montreal; 'Montevideo'; 'Tijana'; 'Diyaaide'; 'بغداد'; '2006'; 'Montreal']	0	0	0	0	607340	80.3
Caracas	10.481213	-66.904444	2082	[Hå Nø; 'Lisboa'; '香港'; '서울특별시'; 'اسلام آباد']	+	+	+	+	609422	80.57
Cleveland	41.497511	-81.692016	2077	[Las Vegas; 'Cincinnati'; '开罗'; 'Detroit'; 'Yaounde']	0	0	0	0	611499	80.85
بورتو	33.894418	35.501364	2060	[Toronto; 'Monrovia'; '香港'; 'Ljubljana'; 'Hyp-Cymrak'; 'Bujumbura'; '広島市']	+	-	0	-	613599	81.12
Austin	30.265588	-97.747916	2056	[Oslo; 'Freetown'; 'القاهرة'; 'München'; 'Toronto']	0	+	+	+	615615	81.39
Bamako	12.641059	-8.001008	2009	[Accra; '台北'; 'Berlin'; 'La Habana'; 'Cape Town'; 'Pretoria']	0	0	0	0	617624	81.66
Perth	31.945414	115.857137	1985	[Toronto; 'Maputo'; 'Kansas City'; 'San Francisco'; '开罗'; 'Portland']	++	++	++	+	619609	81.92
San José	9.929356	-84.090858	1979	[Aggabaf; 'Toronto'; 'Amsterdam'; 'بغداد'; 'Pittsburgh'; 'Harare']	0	+	+	0	621588	82.18
Mico	53.899187	27.558348	1975	[Salt Lake City; 'القاهرة'; 'İstanbul; 'Phoenix; 'Lima']	+	+	0	+	623563	82.44
Yaounde	3.8488	11.500151	1927	[Santo Domingo; '澳門'; 'Mocaa'; 'Cleveland'; 'اسلام آباد']	0	+	+	+	625490	82.7
Wien	48.207166	16.373535	1868	[Pot Mosb'; 'Praha'; 'Toronto; 'Conaky; 'Boston'; 'Hamburg'; 'Ottawa'; 'Ynaa-Gaarp'; 'القاهرة']	++	+	+	+	627358	82.94
Bucuresti	44.424811	26.098756	1836	[Kingston; '昆明市'; 'Malabo'; '澳門'; 'Kansas City']	0	+	+	+	629194	83.19
Brazzaville	-4.261914	15.243356	1827	[Riyko; 'القاهرة'; '澳門'; 'Rio de Janeiro'; 'Thopropuak'; 'Seattle']	0	-	0	-	631021	83.43
Hamburg	53.54845	9.978514	1822	[澳門; 'Wien'; 'Boston; 'Kinshasa'; 'Luanda']	++	+	+	+	632843	83.67
Antananarivo	-16.78757	-47.573762	1816	[香港; '开罗'; 'Recife'; 'مدينة الجزائر'; 'Tijana'; 'كابل'; 'Reykjavik']	0	0				

name	latitude	longitude	population	connections	economy	government	hygiene	awareness	sum population	percent of population
New Orleans	29.95854	-90.074832	1269	[Brisbane; 'Winnipeg'; '塞维利亚'; 'Atlanta'; 'بغداد'; 'اسطنبول']	+	+	+	+	683473	90.36
Ottawa	45.423575	-75.70278	1236	[San Salvador; 'Lima'; 'بغداد'; 'Ciudad de Guatemala']	++	+	+	+	684709	90.53
Содбуя	42.72738	23.313681	1236	[Dakar; '呼和浩特'; 'بغداد'; 'Kampala'; 'Sukowati']	++	+	+	+	685945	90.69
اروحي	24.461165	54.380084	1206	[Phoenix; 'San José'; 'Amsterdam'; 'Rabat'; 'Capajevo']	++	++	++	++	687151	90.85
広島市	34.391566	132.456186	1194	[Hàng; 'Lungsod ng Maynila'; 'Chicago'; 'بيروت'; '上海市']	0	+	+	++	688345	91.01
Kigali	-1.970411	30.103516	1169	[Ciudad de México; '昆明市'; '厦門市'; 'Georgetown (Guyana)'; 'դղժառաքաւորս']	0	--	-	-	689514	91.16
كوردوبا	19.763475	96.078624	1160	[Detroit; 'اسطنبول'; 'Nassau'; 'Донецк'; 'Ciudad de Córdoba']	+	0	+	0	690674	91.31
Tegucigalpa	14.063599	-87.172358	1158	[Detroit; 'Ahtylor'; 'Amsterdam'; 'Managua'; 'Ciudad de México']	+	0	+	0	691832	91.47
مراكش	32.887837	13.190653	1158	[Hà Nội; 'Roma'; 'Juba'; 'Brazzaville'; 'Helsinki']	0	0	0	0	692990	91.62
Dakar	14.716335	-17.467371	1146	[Bogotá; 'Barcelona'; 'Содбуя'; 'Ankara'; 'الرياض']	-	-	-	-	694136	91.77
اسطنبول	41.717832	44.826491	1114	[Kingston; 'Yamoussoukro'; 'Maputo'; 'Санкт-Петербург']	0	0	0	0	695250	91.92
Maputo	-25.968124	32.572927	1101	[-]	0	0	--	0	696351	92.06
البحرين	12.133307	15.053682	1092	[Antananarivo; 'القاهرة'; 'ဂန္တဝါ'; 'بغداد'; 'اسطنبول'; 'Minneapolis']	0	-	-	0	697443	92.21
Köln	50.941441	6.958324	1085	[Miami; 'Quito'; 'اسطنبول'; 'Санкт-Петербург'; 'Ciudad de México']	+	+	+	+	698528	92.35
لوسايس	40.188261	44.516031	1075	[Albuquerque; 'Ottawa'; 'Wien'; 'Wellington'; 'Kansas City']	0	0	0	0	699603	92.49
Freetown	8.463847	-13.229604	1056	[Praha; 'Dallas'; 'ဂန္တဝါ'; 'Wien'; 'Austin'; 'Detroit']	0	-	-	-	700659	92.63
Managua	12.113763	-86.23752	1034	[اسطنبول]; 'Sevilla'; 'Ciudad de Guatemala'; 'Barcelona'; 'Tegucigalpa']	0	+	0	+	701693	92.77
Aggabat	37.959876	58.338167	1032	[El Paso; 'Toukenn'; 'Anchorage'; 'San José'; 'Madrid']	+	0	0	0	702725	92.91
Niamey	13.51043	2.125212	1027	[اسطنبول]; 'Санкт-Петербург'; 'La Paz'; 'San Francisco'; 'Lusaka']	0	0	0	0	703752	93.04
اسلام اباد	33.6812	73.039408	1015	[Phoenix; 'Yamoussoukro'; 'Malabo'; 'الرياض'; 'Sydney']	0	0	-	0	704767	93.18
Monrovia	6.315329	-10.80607	1011	[Lusaka; '武汉市'; 'بيروت'; 'Edmonton'; 'Los Angeles']	-	0	0	0	705778	93.33
कोलकाता	27.178226	85.320639	1003	[Vilnius; 'Lusaka'; '台北'; 'Berlin'; 'Göteborg']	0	0	+	+	706781	93.44
Нгп-Сүмтэн	51.158199	11.47801	1002	[Rio de Janeiro; 'Bnaqueros'; 'اسطنبول'; 'Quito'; 'Amsterdam']	0	0	++	++	707783	93.58
Honolulu	21.309511	-157.859207	993	[Miami; 'Санкт-Петербург'; 'Sydney']	++	++	++	++	708716	93.74
Nepes	50.02691	56.25899	991	[Milwaukee; 'Moscow'; 'Nairobi'; 'Edinburgh']	++	+	+	+	709767	93.83
Llongwe	13.966186	33.769217	986	[Anchorage; 'Port-au-Prince']	0	0	-	0	710753	93.97
Suwaes	42.875596	74.56611	977	[اسطنبول]; 'Detroit'; 'Содбуя']	0	-	0	0	711730	94.1
Santo Domingo	18.484924	-69.931906	965	[Milano; 'Riga'; 'Yaounde']	+	0	0	+	712695	94.23
Stockholm	59.333422	18.067878	965	[Toukenn; 'Barcelona'; '乌鲁木齐市'; 'London']	+	++	++	++	713660	94.35
نواكشوط	18.073033	-15.960086	958	[Praha; 'Maputo'; 'Buenos Aires'; 'Austin'; 'Kinshasa']	0	0	0	0	714618	94.48
Донецк	48.013298	37.800314	919	[Lomé; 'Detroit'; 'اسطنبول']	0	0	0	+	715537	94.6
Albuquerque	35.086263	-106.646962	910	[Ouagadougou; 'لوسايس']	0	0	0	0	716447	94.72
اسمره	15.323331	38.922393	896	[Cincinnati; '上海市'; 'Ottawa']	0	0	0	-	717343	94.84
С'порт	31.76945	35.211749	874	[Bergen; 'اسطنبول']	0	+	+	+	718217	94.96
Amsterdam	52.367178	4.891943	864	[Rio de Janeiro; 'Нгп-Сүмтэн'; 'Tegucigalpa'; 'San José'; 'Montevideo']	++	+	++	++	719081	95.07
El Paso	31.761445	-106.480144	840	[الرياض]; 'Aggabat'; 'Baki'; '台北'; 'Hà Nội'; '塞维利亚']	-	0	0	-	719921	95.19
Lomé	6.127019	1.228492	840	[Донецк; 'Sevilla'; 'الرياض']	0	-	-	0	720761	95.28
Bangi	4.394482	18.566256	833	[-]	-	0	-	-	721594	95.4
الربيع	25.279646	51.530877	797	[Hosoobiporo; 'Lomé']	++	+	+	+	722391	95.51
Zagreb	45.814816	15.983556	793	[Gdansk; 'بيروت'; 'Barcelona']	0	0	0	0	723184	95.61
Душанбе	38.561035	68.786042	770	[Chicago; 'Warszawa'; 'Dodoma']	-	0	0	0	723954	95.71
La Paz	-16.490746	-68.11687	757	[Namey; 'Budapest']	0	0	0	0	724711	95.81
نواكشوط	17.980348	103	754	[Nashville; 'Perth'; 'Cleveland'; 'Denver'; 'ဂန္တဝါ'; 'بغداد'; 'اسطنبول']	0	0	0	0	725465	95.91
اسطنبول	6.925927	79.864788	753	[厦門市]; 'Bruxelles']	+	+	+	0	726218	96.01
Pretoria	-25.745595	28.232962	742	[Bamako; 'Kin'; 'الرياض']	0	0	0	0	726960	96.11
Winnipeg	49.896219	-97.143992	730	[Ciudad de Guatemala; 'Springfield (Missouri)'; 'Toronto'; 'Roma'; 'اسطنبول'; 'New Orleans']	0	+	++	+	727690	96.21
Llbreville	0.421112	9.467307	704	[Abuja; 'Rio de Janeiro'; 'Hà Nội'; '长春市'; 'Ciudad de México']	0	-	-	-	728394	96.3
Sevilla	37.389006	-5.984241	690	[اسطنبول]; 'Lomé'; 'München'; 'Berlin'; 'Managua']	++	+	0	+	729084	96.39
Boston	42.361278	-71.05749	685	[Abuja; 'Madrid'; 'Wien'; 'Hamburg']	0	+	0	+	729769	96.48
Chigandu	47.009915	28.861951	670	[San Francisco; 'Pittsburgh']	+	0	++	+	730439	96.57
Ahtylvo	38.065338	23.719113	664	[Ciudad de México; 'الرياض'; 'Tegucigalpa']	0	0	0	0	731103	96.66
Kingston	18.017443	-76.811815	649	[Bouacore; 'الرياض'; 'La Habana']	+	0	+	++	731752	96.75
Portland	45.501254	-122.67582	648	[اسطنبول]; 'Seacrest'; 'Perth']	0	0	0	0	732400	96.83
Las Vegas	36.171493	-115.137929	642	[台北; 'Cleveland'; 'Oulu'; 'Hosoobiporo'; 'Ciudad de Guatemala']	++	0	++	+	733042	96.92
Riga	56.949308	24.107184	641	[Edinburgh; 'Santo Domingo'; 'Melbourne']	+	+	++	+	733583	97
تونس	36.808333	10.183444	639	[Cape Town; 'Zagreb']	+	+	+	0	734322	97.08
Olo	59.912831	10.731481	634	[Санкт-Петербург; 'Hosoobiporo'; 'اسطنبول'; 'Austin'; '西安市']	++	++	++	+	734956	97.17
Helinki	60.167987	24.941757	632	[Harare; 'Kampala'; 'الرياض']	++	++	++	+	735588	97.25
Kabenhavn	55.675313	12.572536	623	[乌鲁木齐市; 'Bogotá'; '北京市'; 'Jakarta']	+	++	+	+	736211	97.33
澳門	22.198593	113.543892	623	[Hamburg; 'Yaounde'; 'Seattle'; 'Toukenn']	0	+	0	0	736834	97.42
Внагуосокто	43.120517	131.870348	599	[Hosoobiporo'; 'اسطنبول'; 'Нгп-Сүмтэн'; 'الرياض']	+	0	+	++	737433	97.5
Gdansk	54.351072	18.648802	582	[Nairobi; 'Sydney'; 'Kampala'; 'Vancouver'; 'Zagreb'; 'London'; 'Lungsod ng Maynila']	+	0	+	+	738015	97.57
الرياض	33.973755	-6.851085	578	[Dodoma; 'Lëtzebuerg'; 'Содбуя'; 'اسطنبول200e'; '昆明市'; 'Quito']	0	0	0	0	738593	97.65
Göteborg	59.173938	11.965736	573	[Bissau; 'Ankara'; 'कोलकाता']	+	++	++	++	739166	97.73
Cionje	42.00508	21.436219	547	[Maseru]; 'Georpaq'; 'Bern'; 'Πορτοφουα']	+	+	+	0	739713	97.8
Nantes	47.220016	-1.552848	545	[Jakarta]; 'القاهرة'; 'Brisbane'; 'Charlotte']	+	+	0	+	740258	97.87
Dublin	53.349658	-6.260816	544	[Paris; 'San Juan']	+	+	+	+	740802	97.94
Vilnius	54.677467	25.266659	544	[कोलकाता]; '成都市']	+	0	+	+	741346	98.01
Asunción	-25.264861	-57.574745	525	[اسطنبول]; 'Улаанбаатар'; 'Seattle']	0	0	0	0	741871	98.08
Lisboa	38.727332	-9.140634	505	[Caracas; 'Kansas City'; 'Montevideo']	+	+	0	+	742376	98.15
Juba	4.859835	31.571876	500	[Melbourne; 'Edmonton'; 'الرياض'; 'Denver']	0	0	0	0	742876	98.22
Bujumbura	-3.362274	29.36568	497	[Atlanta; 'Los Angeles'; 'New York City'; 'դղժառաքաւորս'; 'Philadelphia'; 'Dallas']	0	0	0	-	743373	98.28
Edinburgh	55.952531	-3.189204	482	[Riga; 'Περμα']	++	+	++	++	743855	98.35
Ciudad de Panar	8.953738	-79.519064	477	[الرياض]; 'Atlanta'; 'Lima'; 'Nashville'; 'դղժառաքաւորս']	+	+	0	+	744332	98.41
Springfield (Miss	39.783079	-89.65092	459	[Windhoek; 'Winnipeg']	++	+	++	+	744791	98.47
Cape Town	-33.923514	18.432142	434	[Asmara; 'Bamako'; '香港']	0	0	0	0	745225	98.53
Tallinn	59.439721	24.751283	427	[Charotte; 'ဂန္တဝါ'; 'بغداد'; 'اسطنبول']	+	0	+	+	745652	98.58
Brasília	48.14716	17.102178	424	[Acara; 'Bogotá'; 'الرياض']	0	0	0	0	746076	98.64
Minneapolis	44.975776	-93.269377	422	[-]	0	0	0	0	746498	98.69
Tirana	41.327368	19.816449	419	[重庆市; 'Lima'; '成都市'; 'La Habana'; 'Ciudad de Córdoba'; 'Toulouse']	0	0	+	0	746917	98.75
Anchorage	61.216336	-149.90229	403	[Chicago; 'Llongwe'; 'Aggabat'; 'La Habana'; 'Ciudad de Córdoba'; 'Toulouse']	+	++	+	+	747320	98.8
Bissau	11.862489	-15.583656	388	[San Salvador; 'Göteborg']	0	0	0	0	747708	98.85
Christchurch	-43.53957	172.633404	382	[Ouagadougou; 'Recife'; 'Montreal']	+	++	++	+	748090	98.91
Pot Moabi	-9.443109	147.179444	364	[Wien; 'New York City'; '西安市']	++	0	+	0	748454	98.95
Maseru	-29.315503	27.485807	331	[Ryeyor; 'Cionje']	0	0	0	0	748785	99
Windhoek	-22.561205	17.064971	326	[Washington; D.C.; '香港'; 'Strasbourg'; 'Springfield (Missouri)'; 'Harare']	0	0	0	0	749111	99.04
Мурмакк	68.956676	33.078235	307	[الرياض]; 'Kin'; 'Lusaka']	0	+	0	0	749418	99.08
Pittsburgh	40.441277	-79.996355	302	[Conaky; 'Miami'; 'Chigandu'; 'San José']	+	+	+	0	749720	99.12
Malabo	3.759131	8.785107	297	[-]	+	0	0	0	750017	99.16
Capajevo	43.855689	18.417655	291	[-]	0	0	0	0	750308	99.2
Belfast	54.597843	-5.93041	280	[武汉市; 'Maputo'; 'اسطنبول']	0	+	+	+	750588	99.24
Ljubljana	46.056805	14.502228	280	[-]	0	0	0	0	750868	99.27
Strasbourg	48.575232	7.756599	277	[Windhoek; 'Phoenix']	0	0	0	++	751145	99.31
Alexandria	35.188093	33.383325	276	[Salt Lake City; 'اسطنبول']	+	0	+	+	751421	99.35
Bergen	60.406287	5.323408	272	[Porto Alegre; 'Praha'; 'Dallas'; 'اسطنبول']	+	++	+	++	751693	99.38
Ryeytk	62.034779	129.675975	270	[Kinshasa; 'Brazzaville'; 'Montreal'; 'Ciudad de Córdoba'; 'Maseru']	0	+	0	+	751963	99.42
Nassau	25.044986	-77.353104	266	[Kansas City; 'Lungsod ng Maynila'; '上海市'; '명 도시'; 'اسطنبول']	+	+	+	0	752229	99.45
Porto-Novo	6.497144	2.628993	264	[临沂市; 'Yamoussoukro']	+	+	-	0	752493	99.49
Bloemfontein	-29.082368	26.161928	256	[Roma; 'Phoenix'; 'الرياض'; 'San Juan']	0	0	0	0	752749	99.52
Paramaribo	5.85191	-55.202738	241	[Los Angeles; 'اسطنبول']	0	0	0	0	752990	99.55
San Salvador	13.695812	-89.217141	238	[Bissau; 'Ottawa']	0	0	+	0	753228	99.58
Gaborone	-24.629578	25.927756	232	[Washington; D.C.; 'Quito']	0	0	+	0	753460	99.62
العون	27.151611	-13.200184	218	[اسطنبول]; 'Port-au-Prince'; 'Porto Alegre'; 'Santiago de Chile'; 'اسطنبول']	0	0	+	0	753678	99.64
Yamoussoukro	6.828338	-5.286286	213	[Lusaka]; 'اسطنبول'; 'Houston'; 'Acara'; 'Porto-Novo'; 'Madrid'; 'الرياض']	0	0	0	0	753891	99.67
Salt Lake City	40.762026	-111.895081	201	[Alexandria; '厦門市'; 'Toulouse'; 'Mikoi']	0	+	+	+	754092	99.7
Oulu	65.044998	25.459239	200	[Las Vegas; 'Brisbane']	++	+	++	++	754292	99.73
Wellington	-41.284544	174.780388	191	[Toronto; 'Baki'; 'Vancouver'; 'اسطنبول']	++	+	++	++	754483	99.75
Πορτοφουα	42.433871	19.258892	186	[Brazzaville; 'Cionje'; '临沂市; 'Houston']	0	0	0	0	754669	99.77
Περτοναουεκ-1	53.043017	158.681761	181	[香港; 'اسطنبول200e'; 'Warszawa'; '서울특별시'; 'اسطنبول']	0	+	+	+	754850	99.8
Bruxelles	50.849102	4.349639	179	[서울특별시]; 'Milwaukee'; 'اسطنبول']	0	+	++	++	755029	99.82
القاهرة										

name	latitude	longitude	population	connections	economy	government	hygiene	awareness	sum population	rcent of populat
Toluca	-54.510552	-87.193003	3		o	o	+	o	756367	100
Xaravá	71.964	102.440398	3		o	+	o	o	756370	100
Ciudad del Vaticano	41.904207	12.453601	1		+	++	++	++	756371	100

10.1.3 Seeds

Die folgenden Seeds wurden nur mit der Aktion „Runde beenden“ gespielt.

Seed	Outcome	Rounds	Startpathogen	Otherpathogen	
	1 loss		15 Moricillus 𐌆;N5-10		2
	2 loss		27 Procrastinalgia;Phagum vidianum;Moricillus 𐌆		3
	3 win		25 Methanobrevibacter colferi;Rhinonitis;N5-10		3
	4 loss		5 Admiral Trips		1
	5 loss		33 Moricillus 𐌆;Endoictus;Neurodermantotitis		3
	6 loss		32 Phagum vidianum;Shanty	Azmodeus=26	2
	7 loss		51 Procrastinalgia;N5-10	Φthisis=17	2
	8 loss		28 Phagum vidianum;Plorps		2
	9 loss		5 Admiral Trips;Xenomonocythemia;Procrastinalgia		3
	10 loss		29 Rhinonitis;Moricillus 𐌆;Saccharomyces cerevisiae mutans		3
	11 loss		10 Admiral Trips;Azmodeus;N5-10		3
	12 loss		18 Azmodeus;N5-10		2
	13 loss		53 Endoictus;Shanty;Phagum vidianum	Influenza iutiubensis=40;Neurodermantotitis=44	3
	14 win		41 Shanty;Admiral Trips	Endoictus=7	2
	15 loss		27 Plorps;Admiral Trips;Φthisis	Neurodermantotitis=24	3
	16 loss		8 Admiral Trips;Influenza iutiubensis;Shanty		3
	17 loss		48 Xenomonocythemia;Saccharomyces cerevisiae mutans;Methanobrevibacter colferi		3
	18 win		31 Neurodermantotitis;N5-10;Saccharomyces cerevisiae mutans		3
	19 loss		28 Phagum vidianum;Hexapox;Coccus innocuus		3
	20 loss		5 Xenomonocythemia;Admiral Trips;Procrastinalgia		3
	21 loss		48 Endoictus;Xenomonocythemia;Neurodermantotitis	Moricillus 𐌆=31	3
	22 loss		27 Moricillus 𐌆;Neurodermantotitis;Methanobrevibacter colferi		3
	23 loss		22 Hexapox;Rhinonitis;Moricillus 𐌆	N5-10=17	3
	24 loss		49 Azmodeus;Xenomonocythemia;N5-10	Rhinonitis=15	3
	25 win		33 Plorps;Rhinonitis;Admiral Trips		3
	26 win		23 N5-10;Hexapox		2
	27 win		29 Saccharomyces cerevisiae mutans;Moricillus 𐌆		2
	28 loss		29 Methanobrevibacter colferi;Moricillus 𐌆;Phagum vidianum		3
	29 loss		28 Hexapox;Methanobrevibacter colferi;Phagum vidianum		3
	30 loss		14 Admiral Trips;Phagum vidianum;Φthisis		3
	31 win		38 Admiral Trips;Shanty;Endoictus		3
	32 loss		21 Neurodermantotitis;Moricillus 𐌆		2
	33 loss		15 Influenza iutiubensis;Admiral Trips;Azmodeus		3
	34 loss		29 Methanobrevibacter colferi;Phagum vidianum;Azmodeus	Admiral Trips=3;Neurodermantotitis=23;Plorps=25	3
	35 loss		57 Coccus innocuus;Hexapox;Xenomonocythemia	Rhinonitis=27;Shanty=34;Procrastinalgia=47;Influenza iutiubensis=	3
	36 loss		26 Hexapox;Coccus innocuus;Xenomonocythemia	N5-10=16	3
	37 loss		21 Phagum vidianum;N5-10;Φthisis		3
	38 loss		27 Phagum vidianum;Coccus innocuus;Plorps	Shanty=17	3
	39 win		42 Hexapox;Φthisis;N5-10		3
	40 win		26 Moricillus 𐌆;Procrastinalgia		2
	41 loss		27 Coccus innocuus;Phagum vidianum;Saccharomyces cerevisiae mutans		3
	42 loss		28 Moricillus 𐌆;Coccus innocuus;Φthisis		3
	43 loss		65 Procrastinalgia;Moricillus 𐌆;Methanobrevibacter colferi	Saccharomyces cerevisiae mutans=44	3
	44 loss		38 Phagum vidianum;Neurodermantotitis;Φthisis	Shanty=38	3
	45 win		33 Moricillus 𐌆;Methanobrevibacter colferi;Saccharomyces cerevisiae mutans		3
	46 loss		33 Methanobrevibacter colferi;Phagum vidianum;Plorps		3
	47 loss		33 Admiral Trips;Saccharomyces cerevisiae mutans;Xenomonocythemia	Endoictus=26	3
	48 loss		69 Influenza iutiubensis;Phagum vidianum;Procrastinalgia	Coccus innocuus=55;Neurodermantotitis=60;Azmodeus=60	3
	49 loss		38 Methanobrevibacter colferi;Xenomonocythemia;Phagum vidianum		3
	50 loss		5 Admiral Trips;Procrastinalgia		2
	51 loss		71 Xenomonocythemia;Procrastinalgia;Methanobrevibacter colferi		3
	52 win		29 Rhinonitis;Admiral Trips;Azmodeus		3
	53 loss		40 Neurodermantotitis;Xenomonocythemia;Azmodeus	Hexapox=37	3
	54 loss		24 Phagum vidianum;Moricillus 𐌆		2
	55 win		11 N5-10;Influenza iutiubensis		2
	56 loss		46 Xenomonocythemia;Plorps;Rhinonitis		3
	57 win		17 Moricillus 𐌆		1
	58 loss		27 Phagum vidianum;Φthisis;Admiral Trips		3
	59 win		24 Shanty;Rhinonitis;N5-10	Methanobrevibacter colferi=12	3
	60 win		41 Shanty;Coccus innocuus;Moricillus 𐌆	Rhinonitis=18	3
	61 win		28 Coccus innocuus;Phagum vidianum		2
	62 loss		62 Coccus innocuus;Rhinonitis;Xenomonocythemia		3
	63 win		34 Coccus innocuus;Influenza iutiubensis;N5-10		3
	64 loss		47 Saccharomyces cerevisiae mutans;Xenomonocythemia		2
	65 loss		37 Influenza iutiubensis;Xenomonocythemia;Moricillus 𐌆		3
	66 win		58 Endoictus;Moricillus 𐌆		2
	67 loss		42 Endoictus;Xenomonocythemia	Moricillus 𐌆=29;Saccharomyces cerevisiae mutans=29	2
	68 loss		19 Neurodermantotitis;Moricillus 𐌆		2
	69 loss		27 Rhinonitis;Phagum vidianum;Methanobrevibacter colferi	Hexapox=17	3
	70 loss		44 Xenomonocythemia;Moricillus 𐌆	Endoictus=36	2
	71 loss		21 Admiral Trips;Saccharomyces cerevisiae mutans;Moricillus 𐌆		3
	72 loss		24 Neurodermantotitis;Moricillus 𐌆;Shanty		3
	73 loss		29 Xenomonocythemia;Azmodeus;Hexapox	Φthisis=25	3
	74 loss		30 Φthisis;Phagum vidianum;Methanobrevibacter colferi		3
	75 win		35 N5-10;Endoictus		2
	76 loss		41 Influenza iutiubensis;Xenomonocythemia;Neurodermantotitis		3
	77 loss		19 N5-10;Phagum vidianum;Admiral Trips		3
	78 loss		43 Procrastinalgia;Xenomonocythemia;Coccus innocuus	Rhinonitis=36	3
	79 loss		56 Methanobrevibacter colferi;Rhinonitis;Xenomonocythemia	Plorps=19;Phagum vidianum=41	3
	80 loss		29 N5-10;Xenomonocythemia		2
	81 loss		20 Plorps;Admiral Trips;Coccus innocuus		3
	82 loss		37 Rhinonitis;Moricillus 𐌆;Plorps	Shanty=18	3
	83 loss		42 Saccharomyces cerevisiae mutans;Xenomonocythemia;Φthisis	N5-10=37	3
	84 loss		17 Rhinonitis;Influenza iutiubensis;Admiral Trips		3
	85 loss		14 Admiral Trips;Φthisis;Hexapox		3
	86 loss		31 Φthisis;Azmodeus;Moricillus 𐌆		3
	87 loss		54 Rhinonitis;Moricillus 𐌆	Coccus innocuus=19;Endoictus=19;Methanobrevibacter colferi=49	2
	88 loss		28 Moricillus 𐌆;Neurodermantotitis;Plorps		3

Seed	Outcome	Rounds	Startpathogen	Otherpathogen	
	89	loss	27 Moricillus 𐌆;Φthisis		2
	90	win	82 Hexapox;Admiral Trips;Procrastinalgia	Coccus innocuus=19	3
	91	win	49 N5-10;Φthisis;Endoictus		3
	92	loss	12 Phagum vidianum;Admiral Trips;Moricillus 𐌆		3
	93	loss	14 Procrastinalgia;Admiral Trips;Φthisis		3
	94	loss	37 Azmodeus;Φthisis;Phagum vidianum		3
	95	loss	27 Φthisis;Rhinonitis;Moricillus 𐌆		3
	96	loss	22 Rhinonitis;Admiral Trips;Shanty		3
	97	loss	17 Neurodermantotitis;Admiral Trips;Rhinonitis		3
	98	loss	76 Procrastinalgia;Xenomonocythemia;Moricillus 𐌆	Shanty=48	3
	99	loss	38 Hexapox;Rhinonitis;Admiral Trips	Plorps=18;Influenza iutiubensis=38	3
	100	loss	24 Moricillus 𐌆;Hexapox;Procrastinalgia		3
	101	win	34 Shanty;Moricillus 𐌆;Influenza iutiubensis	Methanobrevibacter colferi=18;Coccus innocuus=20	3
	102	win	52 Xenomonocythemia;Coccus innocuus;Endoictus	Azmodeus=35	3
	103	loss	46 Xenomonocythemia	Azmodeus=30	1
	104	loss	43 Procrastinalgia;Plorps;Admiral Trips	Neurodermantotitis=21;Rhinonitis=34	3
	105	loss	37 Influenza iutiubensis;Plorps;Moricillus 𐌆		3
	106	win	43 Φthisis;N5-10;Saccharomyces cerevisiae mutans		3
	107	loss	14 Admiral Trips;Phagum vidianum;Methanobrevibacter colferi		3
	108	loss	7 Admiral Trips;Methanobrevibacter colferi;Procrastinalgia	Moricillus 𐌆=6	3
	109	win	45 N5-10;Endoictus;Azmodeus		3
	110	loss	28 Procrastinalgia;Phagum vidianum;Moricillus 𐌆		3
	111	loss	38 Xenomonocythemia;Hexapox	Phagum vidianum=26	2
	112	loss	29 Phagum vidianum;Φthisis		2
	113	loss	41 Shanty;Phagum vidianum;Xenomonocythemia		3
	114	loss	32 Procrastinalgia;Plorps;Phagum vidianum	Endoictus=22;Moricillus 𐌆=29	3
	115	loss	27 Phagum vidianum;Neurodermantotitis;Azmodeus		3
	116	loss	45 Coccus innocuus;N5-10;Procrastinalgia	Plorps=9;Rhinonitis=32;Saccharomyces cerevisiae mutans=38;Pha	3
	117	win	62 Saccharomyces cerevisiae mutans;Influenza iutiubensis;Admiral Trips	Endoictus=18;Φthisis=45	3
	118	loss	42 Xenomonocythemia	Azmodeus=7;Rhinonitis=19	1
	119	loss	45 Rhinonitis;Xenomonocythemia;Hexapox	Phagum vidianum=33	3
	120	win	109 Xenomonocythemia;Procrastinalgia;Coccus innocuus	Endoictus=53	3
	121	win	26 Phagum vidianum;Rhinonitis		2
	122	loss	26 Phagum vidianum;Azmodeus;Influenza iutiubensis		3
	123	win	36 Saccharomyces cerevisiae mutans;N5-10;Azmodeus		3
	124	win	39 N5-10;Endoictus;Shanty		3
	125	win	77 N5-10;Shanty;Neurodermantotitis	Procrastinalgia=18	3
	126	loss	41 N5-10;Endoictus;Admiral Trips	Shanty=33;Influenza iutiubensis=34;Coccus innocuus=36	3
	127	loss	22 Neurodermantotitis;Azmodeus;Moricillus 𐌆		3
	128	loss	22 Hexapox;N5-10;Methanobrevibacter colferi		3
	129	win	28 N5-10;Saccharomyces cerevisiae mutans;Coccus innocuus		3
	130	loss	41 Rhinonitis;N5-10	Xenomonocythemia=16	2
	131	loss	27 Φthisis;N5-10;Phagum vidianum		3
	132	loss	58 Endoictus;Plorps;Admiral Trips	Φthisis=43	3
	133	loss	27 Saccharomyces cerevisiae mutans;Admiral Trips;Methanobrevibacter colferi	Hexapox=17	3
	134	loss	42 Xenomonocythemia;Rhinonitis;Azmodeus	Hexapox=31	3
	135	win	24 N5-10;Rhinonitis		2
	136	loss	40 Hexapox;Xenomonocythemia		2
	137	win	57 Procrastinalgia;N5-10		2
	138	loss	12 Shanty;Xenomonocythemia;Admiral Trips		3
	139	loss	26 Hexapox;Phagum vidianum		2
	140	loss	28 Hexapox;Φthisis;Admiral Trips		3
	141	loss	25 Moricillus 𐌆;Phagum vidianum;Methanobrevibacter colferi		3
	142	loss	54 Φthisis;Xenomonocythemia;Rhinonitis		3
	143	loss	43 Endoictus;Admiral Trips;Rhinonitis	Phagum vidianum=5	3
	144	loss	44 Phagum vidianum;Shanty;Endoictus		3
	145	loss	51 Endoictus;Admiral Trips;Plorps	Hexapox=43	3
	146	win	24 Influenza iutiubensis;Phagum vidianum		2
	147	loss	32 Plorps;Moricillus 𐌆;Influenza iutiubensis		3
	148	loss	50 Xenomonocythemia;Φthisis;Saccharomyces cerevisiae mutans		3
	149	loss	42 Admiral Trips;Endoictus;Azmodeus	Influenza iutiubensis=28;N5-10=35	3
	150	loss	26 Phagum vidianum;Neurodermantotitis;Shanty		3
	151	loss	50 Coccus innocuus;Methanobrevibacter colferi;N5-10	Hexapox=42	3
	152	loss	15 Shanty;Azmodeus;Admiral Trips		3
	153	loss	73 Xenomonocythemia;Influenza iutiubensis;Coccus innocuus		3
	154	loss	46 Xenomonocythemia;Methanobrevibacter colferi		2
	155	loss	33 Methanobrevibacter colferi;Influenza iutiubensis;Phagum vidianum	Coccus innocuus=4;Plorps=5	3
	156	win	10 N5-10		1
	157	loss	12 Procrastinalgia;Azmodeus;Admiral Trips		3
	158	loss	27 Phagum vidianum;Xenomonocythemia;Plorps		3
	159	loss	29 Phagum vidianum;Methanobrevibacter colferi;Φthisis		3
	160	loss	39 Influenza iutiubensis;Xenomonocythemia;N5-10		3
	161	loss	41 Procrastinalgia;Influenza iutiubensis;Moricillus 𐌆	Rhinonitis=25;Neurodermantotitis=33	3
	162	loss	20 N5-10;Φthisis;Moricillus 𐌆		3
	163	loss	68 Shanty;Rhinonitis;Xenomonocythemia	Hexapox=19;Influenza iutiubensis=41;Neurodermantotitis=60	3
	164	loss	22 Moricillus 𐌆;Hexapox		2
	165	loss	27 Moricillus 𐌆;Phagum vidianum;Shanty		3
	166	loss	28 Phagum vidianum;Plorps;N5-10		3
	167	loss	32 Φthisis;Moricillus 𐌆		2
	168	loss	24 Moricillus 𐌆;Plorps		2
	169	loss	47 Coccus innocuus;Neurodermantotitis;Xenomonocythemia	N5-10=41	3
	170	loss	31 Coccus innocuus;Φthisis;Phagum vidianum		3
	171	loss	14 Admiral Trips;Plorps		2
	172	loss	24 N5-10;Saccharomyces cerevisiae mutans;Procrastinalgia	Admiral Trips=19	3
	173	win	45 N5-10;Endoictus;Saccharomyces cerevisiae mutans		3
	174	loss	10 Φthisis;N5-10;Admiral Trips		3
	175	loss	62 Endoictus;Admiral Trips;Azmodeus	Xenomonocythemia=45;Neurodermantotitis=52	3
	176	loss	24 Methanobrevibacter colferi;Moricillus 𐌆;Saccharomyces cerevisiae mutans		3

Seed	Outcome	Rounds	Startpathogen	Otherpathogen	
177	loss	35	Φthisis;Phagum vidianum;Shanty	Endoictus=35	3
178	loss	25	Phagum vidianum;Neurodermantotitis;Procrastinalgia		3
179	loss	21	Saccharomyces cerevisiae mutans;Moricillus 𐄂;N5-10		3
180	win	23	Moricillus 𐄂;Influenza iutiubensis		2
181	loss	9	Admiral Trips;Moricillus 𐄂;Influenza iutiubensis	Endoictus=7	3
182	win	54	Endoictus;Neurodermantotitis;Phagum vidianum		3
183	loss	25	Neurodermantotitis;Xenomonocythemia;Phagum vidianum		3
184	win	21	Azmodeus;N5-10		2
185	loss	52	Saccharomyces cerevisiae mutans;Xenomonocythemia;Methanobrevibacter colferi		3
186	loss	5	Admiral Trips		1
187	loss	6	Admiral Trips;Influenza iutiubensis		2
188	loss	50	N5-10;Endoictus;Xenomonocythemia	Neurodermantotitis=41;Plorps=41	3
189	loss	48	Φthisis;Influenza iutiubensis;Xenomonocythemia	Coccus innocuus=36	3
190	loss	20	Moricillus 𐄂;Plorps;N5-10		3
191	loss	17	Coccus innocuus;N5-10;Azmodeus		3
192	loss	24	Moricillus 𐄂;Neurodermantotitis;Phagum vidianum		3
193	loss	29	Plorps;Neurodermantotitis;Moricillus 𐄂		3
194	loss	17	Admiral Trips;Coccus innocuus;Phagum vidianum		3
195	loss	34	N5-10;Methanobrevibacter colferi;Shanty	Neurodermantotitis=24	3
196	loss	12	Admiral Trips;Influenza iutiubensis;Azmodeus		3
197	loss	14	Neurodermantotitis;Admiral Trips;Xenomonocythemia		3
198	loss	20	N5-10;Phagum vidianum		2
199	win	24	Influenza iutiubensis;Phagum vidianum		2
200	loss	51	Xenomonocythemia;Saccharomyces cerevisiae mutans;Plorps	Azmodeus=22;Hexapox=42	3
201	loss	50	Xenomonocythemia;Azmodeus;Saccharomyces cerevisiae mutans	Shanty=24;Neurodermantotitis=42	3
202	loss	50	Rhinonitis;Saccharomyces cerevisiae mutans;Xenomonocythemia		3
203	win	32	Neurodermantotitis;Admiral Trips;Shanty		3
204	loss	27	Phagum vidianum;Procrastinalgia;Hexapox		3
205	loss	62	Endoictus;Admiral Trips;Procrastinalgia	Coccus innocuus=33;Hexapox=53	3
206	loss	33	Rhinonitis;Admiral Trips;Xenomonocythemia		3
207	loss	73	Endoictus;Xenomonocythemia;Methanobrevibacter colferi	Procrastinalgia=33;Moricillus 𐄂=48	3
208	win	45	N5-10;Endoictus;Plorps		3
209	loss	8	Coccus innocuus;Influenza iutiubensis;Admiral Trips	Xenomonocythemia=4	3
210	loss	27	Moricillus 𐄂;Coccus innocuus;Φthisis		3
211	loss	63	Xenomonocythemia;Rhinonitis	Coccus innocuus=29	2
212	loss	6	Influenza iutiubensis;Xenomonocythemia;Admiral Trips		3
213	loss	30	Methanobrevibacter colferi;Φthisis;Phagum vidianum		3
214	loss	65	N5-10;Xenomonocythemia;Endoictus		3
215	loss	28	Φthisis;Admiral Trips;Hexapox		3
216	loss	28	Phagum vidianum;Moricillus 𐄂	Xenomonocythemia=17	2
217	win	91	Moricillus 𐄂;Methanobrevibacter colferi;Procrastinalgia	Rhinonitis=23	3
218	loss	24	Moricillus 𐄂;Shanty;Hexapox		3
219	loss	20	Phagum vidianum;N5-10;Shanty		3
220	loss	27	Phagum vidianum;Coccus innocuus;Moricillus 𐄂		3
221	loss	23	Xenomonocythemia;Hexapox;Moricillus 𐄂		3
222	win	34	N5-10;Φthisis;Methanobrevibacter colferi	Rhinonitis=20	3
223	loss	26	Phagum vidianum;Azmodeus		2
224	loss	40	Moricillus 𐄂;Methanobrevibacter colferi;Coccus innocuus	Plorps=31	3
225	loss	25	Xenomonocythemia;Moricillus 𐄂;Rhinonitis	Azmodeus=16	3
226	loss	31	Phagum vidianum;Φthisis;Influenza iutiubensis		3
227	loss	5	Xenomonocythemia;Admiral Trips;Procrastinalgia		3
228	loss	14	Shanty;Admiral Trips;Φthisis		3
229	win	51	Methanobrevibacter colferi;Admiral Trips;Endoictus	Plorps=31	3
230	loss	42	Xenomonocythemia;Influenza iutiubensis	Saccharomyces cerevisiae mutans=20	2
231	win	34	N5-10;Endoictus		2
232	loss	46	Methanobrevibacter colferi;Xenomonocythemia;Moricillus 𐄂	Endoictus=30	3
233	loss	41	Xenomonocythemia;Neurodermantotitis;Methanobrevibacter colferi		3
234	loss	36	Xenomonocythemia;Phagum vidianum		2
235	loss	24	Rhinonitis;Neurodermantotitis;Moricillus 𐄂		3
236	win	21	Methanobrevibacter colferi;Moricillus 𐄂;Influenza iutiubensis		3
237	loss	17	Methanobrevibacter colferi;Moricillus 𐄂;N5-10		3
238	win	74	Moricillus 𐄂;Shanty;Procrastinalgia		3
239	loss	61	N5-10;Azmodeus;Endoictus	Xenomonocythemia=39	3
240	loss	23	Neurodermantotitis;Moricillus 𐄂;Procrastinalgia		3
241	loss	30	Admiral Trips;Influenza iutiubensis;Hexapox	Procrastinalgia=17;Saccharomyces cerevisiae mutans=20;Neurode	3
242	loss	20	Phagum vidianum;N5-10;Φthisis		3
243	win	24	N5-10;Influenza iutiubensis;Hexapox	Shanty=3	3
244	loss	28	Φthisis;N5-10;Neurodermantotitis		3
245	win	15	N5-10;Rhinonitis		2
246	loss	28	Moricillus 𐄂;Procrastinalgia;Phagum vidianum		3
247	win	30	Methanobrevibacter colferi;Shanty;Phagum vidianum		3
248	loss	28	Xenomonocythemia;Moricillus 𐄂;Phagum vidianum		3
249	loss	46	Saccharomyces cerevisiae mutans;Xenomonocythemia;Influenza iutiubensis	Hexapox=27	3
250	loss	76	Saccharomyces cerevisiae mutans;Xenomonocythemia	Coccus innocuus=23;Neurodermantotitis=42;Methanobrevibacter c	2
251	win	66	Influenza iutiubensis;Moricillus 𐄂;Procrastinalgia		3
252	win	54	Phagum vidianum;Procrastinalgia;Endoictus	Neurodermantotitis=38	3
253	loss	25	Neurodermantotitis;Phagum vidianum;Xenomonocythemia	Shanty=25	3
254	loss	23	Neurodermantotitis;Moricillus 𐄂;Plorps		3
255	loss	27	Neurodermantotitis;Admiral Trips;Phagum vidianum		3
256	loss	27	Phagum vidianum;Φthisis;Moricillus 𐄂		3
257	loss	28	Methanobrevibacter colferi;Φthisis;Moricillus 𐄂	Xenomonocythemia=3	3
258	loss	53	Hexapox;Rhinonitis;Xenomonocythemia		3
259	win	34	N5-10;Endoictus		2
260	loss	51	Φthisis;Endoictus;Phagum vidianum		3
261	loss	27	Saccharomyces cerevisiae mutans;Moricillus 𐄂;Procrastinalgia		3
262	loss	32	Moricillus 𐄂;Coccus innocuus	Plorps=18	2
263	loss	33	Phagum vidianum;Neurodermantotitis;Saccharomyces cerevisiae mutans		3
264	win	50	N5-10;Endoictus;Shanty	Plorps=32	3

Seed	Outcome	Rounds	Startpathogen	Otherpathogen	
	265 loss		27 Saccharomyces cerevisiae mutans;N5-10;Phagum vidianum		3
	266 loss		5 Xenomonocythemia;Admiral Trips	Procrastinalgia=5	2
	267 loss		15 Hexapox;Plorps;Admiral Trips		3
	268 win		20 Neurodermantotitis;N5-10		2
	269 win		40 Saccharomyces cerevisiae mutans;Coccus innocuus;Admiral Trips		3
	270 loss		32 Phagum vidianum;Saccharomyces cerevisiae mutans;Rhinitis	Methanobrevibacter colferi=10	3
	271 win		31 Rhinitis;Moricillus 𐀇;Influenza iutiubensis		3
	272 loss		9 Admiral Trips;Moricillus 𐀇		2
	273 loss		64 Influenza iutiubensis;Φthisis;Xenomonocythemia	Rhinitis=3;Shanty=31	3
	274 loss		10 Xenomonocythemia;Methanobrevibacter colferi;Admiral Trips		3
	275 loss		50 Coccus innocuus;Xenomonocythemia		2
	276 loss		49 Xenomonocythemia;Influenza iutiubensis;Neurodermantotitis	Procrastinalgia=42	3
	277 loss		28 N5-10;Φthisis;Phagum vidianum		3
	278 loss		27 Hexapox;Saccharomyces cerevisiae mutans;Moricillus 𐀇		3
	279 loss		26 Admiral Trips;Methanobrevibacter colferi;Neurodermantotitis	Hexapox=16	3
	280 loss		60 Shanty;Phagum vidianum;Procrastinalgia	Coccus innocuus=21;N5-10=45	3
	281 win		42 Endoictus;Admiral Trips;Saccharomyces cerevisiae mutans		3
	282 loss		35 Saccharomyces cerevisiae mutans;Xenomonocythemia;Neurodermantotitis	Azmodeus=22	3
	283 loss		17 Admiral Trips;Phagum vidianum;Hexapox		3
	284 win		29 Rhinitis;Shanty;Admiral Trips		3
	285 loss		35 Φthisis;Plorps;Xenomonocythemia	Moricillus 𐀇=27	3
	286 win		22 Coccus innocuus;N5-10;Neurodermantotitis		3
	287 loss		27 Moricillus 𐀇;Phagum vidianum;Procrastinalgia		3
	288 loss		24 N5-10;Azmodeus;Φthisis	Saccharomyces cerevisiae mutans=20	3
	289 loss		28 Phagum vidianum;Saccharomyces cerevisiae mutans		2
	290 win		27 Admiral Trips;Plorps	Rhinitis=18	2
	291 loss		20 N5-10;Phagum vidianum;Coccus innocuus		3
	292 loss		22 Phagum vidianum;N5-10;Azmodeus		3
	293 loss		29 Phagum vidianum;Plorps;Coccus innocuus		3
	294 win		36 Admiral Trips;Saccharomyces cerevisiae mutans;Φthisis		3
	295 loss		42 Saccharomyces cerevisiae mutans;Shanty;Admiral Trips	Methanobrevibacter colferi=30	3
	296 loss		49 Methanobrevibacter colferi;Phagum vidianum;Procrastinalgia	Saccharomyces cerevisiae mutans=42	3
	297 loss		11 N5-10;Moricillus 𐀇;Admiral Trips		3
	298 loss		14 Phagum vidianum;Admiral Trips	Rhinitis=5	2
	299 win		25 Rhinitis;Phagum vidianum		2
	300 win		36 Admiral Trips;Shanty;Endoictus		3
	301 loss		15 N5-10;Phagum vidianum;Moricillus 𐀇		3
	302 win		19 Admiral Trips;Saccharomyces cerevisiae mutans		2
	303 loss		5 Admiral Trips;Procrastinalgia		2
	304 win		26 Hexapox;Influenza iutiubensis;N5-10		3
	305 loss		30 Rhinitis;Moricillus 𐀇;Φthisis		3
	306 win		33 Endoictus;N5-10		2
	307 loss		48 Procrastinalgia;Rhinitis;Xenomonocythemia	N5-10=21;Admiral Trips=41	3
	308 loss		35 Xenomonocythemia;Azmodeus;Neurodermantotitis	Phagum vidianum=13	3
	309 win		32 Shanty;Xenomonocythemia;Φthisis		3
	310 win		22 Influenza iutiubensis;Xenomonocythemia;Rhinitis		3
	311 loss		24 Moricillus 𐀇;Influenza iutiubensis;Hexapox	Phagum vidianum=16	3
	312 loss		26 Neurodermantotitis;N5-10;Azmodeus		3
	313 loss		74 Xenomonocythemia;Procrastinalgia;Rhinitis		3
	314 loss		26 Neurodermantotitis;Phagum vidianum;Xenomonocythemia		3
	315 loss		52 Endoictus;Moricillus 𐀇;Phagum vidianum	Influenza iutiubensis=41	3
	316 loss		61 Xenomonocythemia;Rhinitis;Endoictus	Azmodeus=24;Admiral Trips=57	3
	317 loss		53 Coccus innocuus;Phagum vidianum;Endoictus	Shanty=41;Azmodeus=43	3
	318 loss		21 Hexapox;Moricillus 𐀇		2
	319 loss		93 Rhinitis;Endoictus;Xenomonocythemia	Procrastinalgia=36	3
	320 loss		11 Procrastinalgia;N5-10;Admiral Trips		3
	321 loss		58 Plorps;Xenomonocythemia;Coccus innocuus		3
	322 loss		46 Saccharomyces cerevisiae mutans;Xenomonocythemia;Influenza iutiubensis		3
	323 loss		55 Xenomonocythemia;Plorps;Rhinitis		3
	324 loss		60 Xenomonocythemia;Coccus innocuus;Endoictus		3
	325 loss		33 Coccus innocuus;Moricillus 𐀇	Saccharomyces cerevisiae mutans=18;Influenza iutiubensis=20;Azmodeus=11	2
	326 loss		18 Xenomonocythemia;Coccus innocuus;Procrastinalgia	Admiral Trips=13;Shanty=17	3
	327 loss		31 Moricillus 𐀇;Rhinitis;Methanobrevibacter colferi	Neurodermantotitis=20	3
	328 loss		24 Moricillus 𐀇;Xenomonocythemia;Hexapox		3
	329 loss		6 Coccus innocuus;Admiral Trips		2
	330 loss		28 Shanty;Phagum vidianum;Hexapox		3
	331 loss		32 Phagum vidianum;Influenza iutiubensis;Φthisis		3
	332 loss		36 N5-10;Endoictus;Moricillus 𐀇	Admiral Trips=27	3
	333 loss		20 N5-10;Influenza iutiubensis;Φthisis		3
	334 loss		28 Plorps;Azmodeus;Phagum vidianum		3
	335 loss		44 Neurodermantotitis;Xenomonocythemia;Endoictus	Φthisis=34;Admiral Trips=38	3
	336 loss		66 Xenomonocythemia;Endoictus		2
	337 loss		8 Xenomonocythemia;Methanobrevibacter colferi;Admiral Trips	Hexapox=3	3
	338 loss		16 Admiral Trips;Phagum vidianum	Rhinitis=6	2
	339 loss		20 Phagum vidianum;N5-10;Φthisis		3
	340 loss		42 Endoictus;Moricillus 𐀇;Hexapox		3
	341 win		19 Admiral Trips;Saccharomyces cerevisiae mutans		2
	342 win		29 Φthisis;N5-10		2
	343 loss		27 Phagum vidianum;Φthisis;Moricillus 𐀇		3
	344 loss		49 Neurodermantotitis;Xenomonocythemia;Coccus innocuus		3
	345 loss		28 Moricillus 𐀇;Φthisis	Azmodeus=27	2
	346 win		42 Moricillus 𐀇;Endoictus		2
	347 loss		57 Plorps;Hexapox;Xenomonocythemia	Φthisis=49	3
	348 win		87 Neurodermantotitis;Endoictus;Xenomonocythemia	Rhinitis=47;Saccharomyces cerevisiae mutans=53	3
	349 loss		42 Plorps;Neurodermantotitis;Admiral Trips	N5-10=37	3
	350 loss		30 Shanty;N5-10;Procrastinalgia	Azmodeus=11	3
	351 win		37 N5-10;Plorps;Saccharomyces cerevisiae mutans		3
	352 loss		35 Admiral Trips;Plorps;Procrastinalgia	Moricillus 𐀇=29	3

Seed	Outcome	Rounds	Startpathogen	Otherpathogen	
	353	loss	45 Endoictus;Methanobrevibacter colferi;Moricillus 鼠		3
	354	loss	44 Xenomonocythemia;Saccharomyces cerevisiae mutans		2
	355	loss	52 Azmodeus;Hexapox;Xenomonocythemia	Neurodermantotitis=44	3
	356	loss	37 Admiral Trips;Shanty;Hexapox	Xenomonocythemia=5;N5-10=32	3
	357	loss	46 Hexapox;Xenomonocythemia;Saccharomyces cerevisiae mutans	Phagum vidianum=35	3
	358	loss	52 Rhinonitis;Endoictus;Xenomonocythemia	N5-10=31;Procrastinalgia=39	3
	359	win	24 Admiral Trips;Plorps;Methanobrevibacter colferi		3
	360	loss	22 Moricillus 鼠;Procrastinalgia;Hexapox		3
	361	loss	53 Rhinonitis;Xenomonocythemia;Coccus innocuus		3
	362	loss	24 Phagum vidianum;Admiral Trips;Neurodermantotitis		3
	363	loss	15 Phagum vidianum;Admiral Trips;Φthisis	Rhinonitis=5	3
	364	loss	28 Saccharomyces cerevisiae mutans;Phagum vidianum;Admiral Trips	Endoictus=23	3
	365	loss	31 Phagum vidianum;Φthisis		2
	366	loss	20 Phagum vidianum;N5-10;Admiral Trips		3
	367	win	19 Moricillus 鼠;Influenza iutiubensis		2
	368	loss	41 Xenomonocythemia;Phagum vidianum;Influenza iutiubensis		3
	369	loss	28 Coccus innocuus;Moricillus 鼠;Plorps		3
	370	loss	32 N5-10;Φthisis;Plorps	Saccharomyces cerevisiae mutans=22	3
	371	win	9 N5-10		1
	372	loss	46 Procrastinalgia;Moricillus 鼠;Azmodeus	Coccus innocuus=17;N5-10=37	3
	373	loss	11 Admiral Trips;Methanobrevibacter colferi;N5-10		3
	374	loss	26 Phagum vidianum;N5-10;Neurodermantotitis		3
	375	win	77 Admiral Trips;Endoictus;Neurodermantotitis	Saccharomyces cerevisiae mutans=36;Coccus innocuus=49	3
	376	loss	32 Shanty;Phagum vidianum;Plorps		3
	377	loss	23 Moricillus 鼠;Shanty;Azmodeus		3
	378	loss	25 Φthisis;N5-10;Hexapox		3
	379	loss	23 Moricillus 鼠;Rhinonitis;Saccharomyces cerevisiae mutans		3
	380	loss	80 Coccus innocuus;Hexapox;Xenomonocythemia	Procrastinalgia=14;Influenza iutiubensis=73	3
	381	loss	5 Procrastinalgia;Admiral Trips;Xenomonocythemia		3
	382	win	26 Φthisis;N5-10		2
	383	win	33 Endoictus;Admiral Trips;Methanobrevibacter colferi		3
	384	loss	29 Hexapox;N5-10;Phagum vidianum		3
	385	loss	8 Influenza iutiubensis;Admiral Trips		2
	386	win	72 Phagum vidianum;Coccus innocuus;Procrastinalgia		3
	387	loss	43 Rhinonitis;Xenomonocythemia	Shanty=42	2
	388	loss	22 Phagum vidianum;Neurodermantotitis;Admiral Trips		3
	389	loss	11 Admiral Trips;Azmodeus;Procrastinalgia		3
	390	win	29 Moricillus 鼠;Saccharomyces cerevisiae mutans		2
	391	loss	22 Admiral Trips;Moricillus 鼠;Hexapox		3
	392	loss	11 Methanobrevibacter colferi;Admiral Trips;Moricillus 鼠		3
	393	loss	11 Admiral Trips;Azmodeus		2
	394	loss	58 Endoictus;Phagum vidianum;Admiral Trips		3
	395	loss	39 Moricillus 鼠;Xenomonocythemia;Influenza iutiubensis		3
	396	loss	10 Moricillus 鼠;Admiral Trips;Rhinonitis		3
	397	loss	20 Procrastinalgia;Rhinonitis;Admiral Trips	N5-10=14	3
	398	loss	42 Neurodermantotitis;Xenomonocythemia;Azmodeus		3
	399	loss	60 Procrastinalgia;Phagum vidianum;Xenomonocythemia		3
	400	loss	36 Methanobrevibacter colferi;Moricillus 鼠	Phagum vidianum=24	2
	401	win	28 Moricillus 鼠;Shanty;Influenza iutiubensis		3
	402	win	21 Hexapox;Admiral Trips		2
	403	loss	54 Xenomonocythemia;Azmodeus;Endoictus	Admiral Trips=49	3
	404	win	36 N5-10;Endoictus;Methanobrevibacter colferi		3
	405	loss	66 Shanty;Xenomonocythemia;Procrastinalgia	Moricillus 鼠=53	3
	406	loss	16 Rhinonitis;N5-10;Moricillus 鼠		3
	407	loss	57 Endoictus;N5-10;Φthisis	Hexapox=48	3
	408	loss	20 Moricillus 鼠;Hexapox;N5-10		3
	409	loss	23 Hexapox;Moricillus 鼠;Shanty		3
	410	loss	74 Xenomonocythemia;Φthisis;Plorps	Rhinonitis=14;Saccharomyces cerevisiae mutans=16;Coccus innoc	3
	411	loss	28 Plorps;Φthisis;Moricillus 鼠		3
	412	loss	71 Neurodermantotitis;N5-10;Procrastinalgia	Hexapox=64	3
	413	loss	5 Procrastinalgia;Admiral Trips		2
	414	loss	71 Shanty;Moricillus 鼠	Procrastinalgia=25;Neurodermantotitis=63	2
	415	win	32 Coccus innocuus;Phagum vidianum;Influenza iutiubensis		3
	416	loss	28 Azmodeus;Phagum vidianum;Influenza iutiubensis		3
	417	win	24 Moricillus 鼠;Rhinonitis		2
	418	win	17 Azmodeus;Influenza iutiubensis		2
	419	loss	33 Moricillus 鼠;Procrastinalgia;Methanobrevibacter colferi	Phagum vidianum=20	3
	420	loss	30 Phagum vidianum;Coccus innocuus	Moricillus 鼠=23	2
	421	loss	20 N5-10;Phagum vidianum		2
	422	loss	24 Rhinonitis;N5-10;Azmodeus		3
	423	loss	26 Shanty;Phagum vidianum;Neurodermantotitis		3
	424	win	58 Procrastinalgia;Neurodermantotitis;Admiral Trips		3
	425	win	57 Endoictus;Xenomonocythemia;N5-10		3
	426	loss	14 Methanobrevibacter colferi;Rhinonitis;Admiral Trips		3
	427	win	43 Coccus innocuus;Azmodeus;N5-10		3
	428	loss	68 Saccharomyces cerevisiae mutans;Xenomonocythemia;Coccus innocuus	Influenza iutiubensis=16;Azmodeus=33;Phagum vidianum=55	3
	429	win	66 Phagum vidianum;Methanobrevibacter colferi;Procrastinalgia		3
	430	loss	28 Influenza iutiubensis;Azmodeus;Phagum vidianum	Procrastinalgia=3	3
	431	loss	47 Shanty;Xenomonocythemia		2
	432	win	22 N5-10;Plorps		2
	433	loss	35 Phagum vidianum;Plorps	Coccus innocuus=21	2
	434	loss	12 Procrastinalgia;Admiral Trips;Azmodeus		3
	435	loss	17 N5-10;Moricillus 鼠;Neurodermantotitis	Coccus innocuus=8	3
	436	loss	74 Neurodermantotitis;Xenomonocythemia;Influenza iutiubensis	Procrastinalgia=6;Shanty=22;Plorps=65	3

10.1.4 Seeds mit Starteigenschaften

10.2 Wichtige Codeauszüge

In diesem Kapitel werden essentielle Methoden, Codeauszüge und Klassen vorgestellt.

Methode die entscheidet ob es sich bei einem Pathogen um ein *ignoriertes Pathogen* handelt oder nicht.

```
1 public boolean ignorePathogenThisRound(Pathogen pathogen) {
2
3     ...
4
5     // Alle boolschen Werte zum Pruefen ob ein Pathogen ignoriert werden soll.
6     boolean isOld = this.getRound() - encounter.get().getRound() >= 10;
7     boolean hasLessAverage = this.getOutbreakEvents().stream().filter(e -> e.getPathogen() == pathogen)
8         .mapToDouble(e -> e.getPrevalence()).average().orElseGet(() -> 0) <= 0.10;
9     long numberOfCities = this.getOutbreakEvents().stream().filter(e -> e.getPathogen() == pathogen).count();
10    boolean hasFewCities = numberOfCities <= 10;
11    boolean hasNoCity = numberOfCities <= 0;
12
13    // Sollten diese beiden boolschen Werte wahr sein, werden keine Pathogene ignoriert
14    boolean enoughPoints = this.points <= 200;
15    boolean hasVeryFewCities = this.getOutbreakEvents().stream().count() <= 5;
16    if (enoughPoints && hasVeryFewCities) {
17        return false;
18    }
19
20    // Pruefe ob ein das aktuelle Pathogen alle Eigenschaften zum ignorieren erfuehlt.
21    boolean result = (isOld && (hasLessAverage || hasFewCities)) || hasNoCity;
22    this.ignoredPathogens.put(pathogen, result);
23    return result;
24 }
```

Numerische Repräsentation der Werte von z.B. „Infectivity“ eines Pathogens

```
1 public int getValue() {
2     switch (this) {
3         case MM: // Entspricht --
4             return 1;
5         case M: // Entspricht -
6             return 2;
7         case N: // Entspricht o
8             return 3;
9         case P: // Entspricht +
10            return 4;
11        case PP: // Entspricht ++
12            return 5;
13    }
14
15    return 0;
16 }
```

Methode zum Entscheiden ob Pathogen unter Quarantäne gesetzt werden soll

```

1 private static boolean doQuarantine(Pathogen pathogen) {
2
3     int infectivity = pathogen.getInfectivity().getNumericRepresentation();
4     int lethality = pathogen.getLethality().getNumericRepresentation();
5     int mobility = pathogen.getMobility().getNumericRepresentation();
6     int duration = pathogen.getDuration().getNumericRepresentation();
7
8     // Factors that contribute to a dangerous pathogen killing lots of people in a
9     // short amount of time
10    // get multiplied. High duration would lead to a long and therefore expensive
11    // Quarantine. Therefore we reverse the
12    // scale of the duration.
13    int score = infectivity * lethality * mobility * (6 - duration);
14
15    return score >= constants.get("QUARANTINE.THRESHOLD");
16 }

```

Methode zum Entscheiden, ob Impfstoff entwickelt werden soll

```

1 private static boolean doDevVaccine(Pathogen pathogen) {
2
3     int infectivity = pathogen.getInfectivity().getNumericRepresentation();
4     int mobility = pathogen.getMobility().getNumericRepresentation();
5
6     int score = mobility * infectivity;
7
8     if (pathogen.getDuration() == Scale.MM) {
9         return false;
10    }
11
12    if (doQuarantine(pathogen)) {
13        return false;
14    }
15
16    // If a pathogen expands slowly, vaccines should be developed.
17    return score <= constants.get("MAX.SLOW.PRODUCT");
18 }

```

Methode zum Entscheiden, ob Medikament entwickelt werden soll

```

1 private static boolean doDevMedication(Pathogen pathogen) {
2
3     int infectivity = pathogen.getInfectivity().getNumericRepresentation();
4     int mobility = pathogen.getMobility().getNumericRepresentation();
5
6     int score = mobility * infectivity;
7
8     if (pathogen.getDuration() == Scale.MM) {
9         return false;
10    }
11
12    // Because quarantine only contains a pathogen within one city it is not worth
13    // the
14    // points to develop a medication
15    if (doQuarantine(pathogen)) {
16        return false;
17    }
18
19    // If a pathogen expands fast, medication should be developed.
20    return score > constants.get("MIN.FAST.PRODUCT");
21 }

```

Teil der Quarantäne Evaluation, welche bestimmt, ob die größte, uninfizierte Stadt vor mindestens zwei Ausbrüchen von starken Pathogenen geschützt werden muss

```

1 *
2 * If a very strong pathogen for instance Admiral Trips breaks out in two or
3 * more cities protect the biggest city.
4 */
5 if (!city.isInfected()) {
6     // Count outbreaks that would normally require quarantine.
7     long strongPathogenAmount = game.getOutbreakEvents().stream().filter(e -> doQuarantine(e.getPathogen()))
8         .count();
9
10    // If the counter is two or greater put the biggest city under quarantine
11    if (strongPathogenAmount >= 2) {
12        score += constants.get("QUARANTINE.FACTOR") * action.getRounds() * city.getPopulation();
13        break;
14    }
15 }

```

Teil der Quarantäne Evaluation, welche bestimmt, ob nur noch ein aktives Pathogen in einer Stadt existiert, welches sich nicht ausbreiten soll.

```

1  /*
2  * Check if there is only one active one pathogen and if there is only one
3  * outbreak event for this pathogen put it under quarantine (enclose a pathogen
4  * inside a city).
5  */
6  if (game.getPathEncounterEvents().stream().filter(e -> !game.ignorePathogenThisRound(e.getPathogen()))
7      .count() == 1) {
8
9      if (!city.isInfected()) {
10         break;
11     }
12
13     // Safe the only pathogen as a variable as we will need it later two times.
14     Pathogen onlyActivePathogen = game.getPathEncounterEvents().stream()
15         .filter(e -> !game.ignorePathogenThisRound(e.getPathogen()))
16         .findAny().get().getPathogen();
17
18     // Check that there is only one outbreak event for the only active pathogen and
19     // that it is in the city for the current action.
20     if ((game.getOutbreakEvents().stream().filter(e -> e.getPathogen() == onlyActivePathogen).count() == 1)
21         && city.isInfected(onlyActivePathogen)) {
22
23         score += constants.get("QUARANTINE_FACTOR") * action.getRounds();
24         break;
25     }
26 }

```

Teil der Quarantäne Evaluation, welche bestimmt, ob der Rest der Welt vor dem Pathogen innerhalb einer Stadt geschützt werden muss.

```

1  // In a regular situation with no outbreak event a city does not require
2  // quarantine
3  if (!city.isInfected()) {
4      break;
5  }
6
7  // Check if the pathogen in the city of the action needs to be quarantined
8  if (!doQuarantine(city.getPathogen())) {
9      break;
10 }
11
12 // Check whether every city is infected and the quarantined pathogen can not
13 // spread any further. In this case there is no quarantine required.
14 if (game.getCities().stream().allMatch((City c) -> c.isInfected())) {
15     break;
16 }
17
18 // City should be quarantine
19 score += constants.get("QUARANTINE_FACTOR") * action.getType().getCosts(action.getRounds());
20 break;

```

Glossar

aktives Pathogen Siehe 4.4.1 Aktives Pathogen. 16

GI Client Kommandozeilenwerkzeug der Gesellschaft für Informatik, zu finden unter <https://github.com/informatiCup/informatiCup2020/releases>. 3, 11–14, 20–22, 26, 28, 29

ignoriertes Pathogen Siehe 4.4.5 Ignoriertes Pathogen. 16, 19, 21, 42

langsames Pathogen Siehe 4.4.3 Langsames Pathogen. 18

schnelles Pathogen Siehe 4.4.4 Schnelles Pathogen. 18

starkes Pathogen Siehe 4.4.2 Starkes Pathogen. 17, 18

Eigenständigkeitserklärung

Hiermit bestätigen wir, dass wir die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt haben. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen) entnommen sind, wurden unter Angabe der Quelle kenntlich gemacht.

Schull, J. Lorenz, R. Stank 15.01.20

(Unterschriften, Datum)