Université de Paris 2020/2021

« Corona Bounce »

Projet de

Aliche Zahra, Bragina natalia, Golikova Anna, SaHI KENZA, RAVENEAU EMILIEN

Sommaire:

[Cahier des charges 2](#_Toc69656694)

[Les objets principaux 2](#_Toc69656695)

[Réglages possibles 2](#_Toc69656696)

[Coté GUI 3](#_Toc69656697)

[Schéma des classes 3](#_Toc69656698)

[Parties traitées 3](#_Toc69656699)

[Chapitre 1 3](#_Toc69656700)

[Chapitre 2 3](#_Toc69656701)

[Gestion des threads 3](#_Toc69656702)

[Difficultés rencontrées 3](#_Toc69656703)

[Problèmes connus 3](#_Toc69656704)

[Pistes d’extensions 4](#_Toc69656705)

Notre projet est une simulation d’une épidémie, permettant observer son développement sur deux populations en variant des paramètres. Cela permettra de visualiser tous les spectres des cas.

# Cahier des charges

## Les objets principaux

Les points qui peuvent prendre 4 états : sain, malade, contagieux et guéri.

Le virus passe d’un point à l’autre si un point sain arrive dans un « nuage viral » entourant un point contagieux (rayon de contamination).

La contamination se produit immédiatement. La guérison prend quelque temps, qui peut être réglé comme paramètre dans « Settings ». Après la guérison un point obtient l’immunité pendant un certain temps et ne peut être infecté à nouveau qu’à partir de ce temps écoulé.

Tous les points se bougent selon le scenario.

Score de nombre des points sains, malades/contagieux et guéris est affiché en temps réel, ainsi que les graphs représentants ses proportions dans population.

En appuyant sur les boutons de menu principale on peut :

1. Lancer le mouvement (« Start »)
2. Arrêter le mouvement et le reprendre (« Pause/Resume »)
3. Changer les réglages (« Settings »)
4. Recharger les populations à nouveau tout en gardant les réglages ainsi que scenarios (« Reset »)
5. Montrer la légende pour chaque élément de la scène (« ? »).

## Réglages possibles

Notre programme permet à utilisateur d’effectuer les changements tant les paramètres de populations et de maladie, que de choisir les scenarios appliques par gouvernement.

Manettes de réglages à partir de « Settings » permettent varier les valeurs de paramètres suivantes :

1. Maladie

* Durée de la guérison
* Durée de l’immunité
* Rayon de contamination

1. Population

* Quantité de population

1. Espace

* Quantité de murs

Outre cela nous proposons choisir parmi les scenarios suivants :

* « No scénario » (déplacement aléatoire et non limité)
* « Lockdown » (déplacement s’effectue dans une zone de quelque rayon de position initial d’individu)
* « Bounds » (déplacement aléatoire mais limité par les frontières)
* « Movement limited » ().
* « Lockdown + bounds »

## Coté GUI

Fenêtre s’adapte au dimensions de l’écran.

# Schéma des classes

# Méthodologie et déroulement du projet

## Chapitre 1

## Chapitre 2

## Vue – Modèle - Controller

## Gestion des threads

Dans notre projet nous avons implémenté le multi-thread architecture.

Etant donné nous avons 5 threads : l’un qui est principale (celui de App.java) et deux pour chaque une des populations (un pour gérer le déplacement des points, l’autre pour gérer les cycles de maladie et sauvegarder l’historique).

A part de main thread, les threads se créent dans deux classes : classe Population et classe Zone.

Le thread de timer crée dans la Population partagent les classes suivantes :

* Classe Incubating – pour gérer la durée d’incubation de maladie, temps écoulé - Incubating devient Sick. Durée = durée de non contamination.
* Classe Sick - pour gérer la durée de maladie après le contact – dès que le temps s’écoule l’individu Sick change la classe pour Recovered. Durée = Durée de maladie.
* Classe Recovered – pour gérer la durée de non contamination, temps écoulé - recovered devient healthy. Durée = durée de non contamination.
* Classe Population – pour sauvegarder la statistique afin de la représenter en graph.

Le thread de timer crée dans la Zone partagent les classes suivantes :

* Classe Zone – pour effectuer le déplacement des individus.

Ou est Mur ?

L’animation dans GUI est attachée au thread principal.

Reentrantlock

# Difficultés rencontrées

# Problèmes connus

# Pistes d’extensions

* Frontières fermée, ouverture s’effectue après.
* Réglage de la vitesse des points.