

# Rapport PlagueINT

Modélisation de la propagation des épidémies

CHERRE Romain	COROLLER Stevan
PAMART Pierrick	PIPEREAU Yohan

Encadrant: Mr. Vincent Gauthier

22 mai 2017

# Table des matières

Pré-Rapport	2
Conception détaillé	8
Bilan	9
Manuel utilisateur	10

# Pré-Rapport

## Analyse des besoins

### Fonction du produit

- Mode de visualisation (écoulement du temps)
- Modélisation mondiale avec cellule de la taille d'un pays
- Possibilité d'exporter le résultat dans un fichier lisible
- Voix de transports prise en compte
- Possibilité d'ajouter des événements (blocage d'aéroports, gare, etc..) au début

### Contraintes techniques

- Utiliser Java8 avec Eclipse et éventuellement d'autres langages si nécessaires
- Possibilité d'exécution en mode terminal puis graphique
- Portabilité Windows, Linux, MAC OS (géré nativement par Java)

### Critères d'acceptabilité et de réception

- Application performante avec un temps d'exécution raisonnable
- L'interface de l'application doit être conforme à la maquette suivante :

—— Simulation de propagation de maladies ——

- (1) Lancer la simulation
- (2) Paramètres de simulation
- (3) Quitter la simulation

—— Paramètres de simulation ——

- (1) Quitter les options sans sauvegarder
- (2) Date de début et durée de la simulation
- (3) Pays infectés et nombre d'infectés de départ
- (4) Choisir les constantes de propagation
- (5) Gérer les évènements
- (6) Choisir une maladie pré-enregistrée
- (7) Quitter et sauvegarder les paramètres

—— Evènements ——

- (1) Créer un évènement
- (2) Voir les évènements
- (3) Supprimer un évènement
- (4) Revenir aux paramètres

## Extensions

- Interface - graphique
- Informations sur les cellules (petits graphiques, etc...)
- Modification de l'environnement (hygiène, température, etc...)

## Juridique

Creative Commons sans usage commerciale [BY NC SA]

## Spécification fonctionnelle générale

## Fonction du produit

Pour l'écoulement du temps, nous avons choisi de discrétiser le temps. Pour l'importation des données et leur traitement, Python est fortement envisagé en tant qu'outil plus performant que Java à l'aide de certaines bibliothèques précodées. Pour l'exportation des résultats dans un fichier lisible, on exporterait les données dans un fichier csv en utilisant des fonctionnalités de lecture/écriture de fichier.

Pour modéliser les voix de transports, on utiliserait un seul graphe avec comme noeuds du graphe les pays et sur les branches, le nombre de passagers par jour.

## Critères d'acceptabilité et de réception

Pour la résolution des équations différentielles, on utiliserait dans un premier temps une méthode d'Euler. Dans un second temps, on implémenterait une méthode de Runge Kutta qui nous permettrait de gagner en performance.

Pour l'interface utilisateur, on permettrait à l'utilisateur de définir l'échelle de temps afin de gérer la rapidité du programme. L'utilisateur pourrait également choisir les pays de lancement de la maladie. Il écrirait le nom du pays et on vérifierait si le nom correspond au nom d'un pays présent dans la liste d'une variable pays.

Enfin pour les coefficients des différentes maladies, il y a le mode manuelle ou l'utilisateur saisi les coefficients à la main et lance ensuite notre programme de modélisation. Mais il y a aussi le mode d'utilisation ou l'utilisateur rentre la maladie et où cela va chercher dans des données que l'on a généré à partir des statistiques mondiales sur les maladies.

## Extensions

Pour l'interface graphique qui permettra d'afficher une carte du monde ainsi que de tracer des graphiques relatifs aux données de la cellule et leur

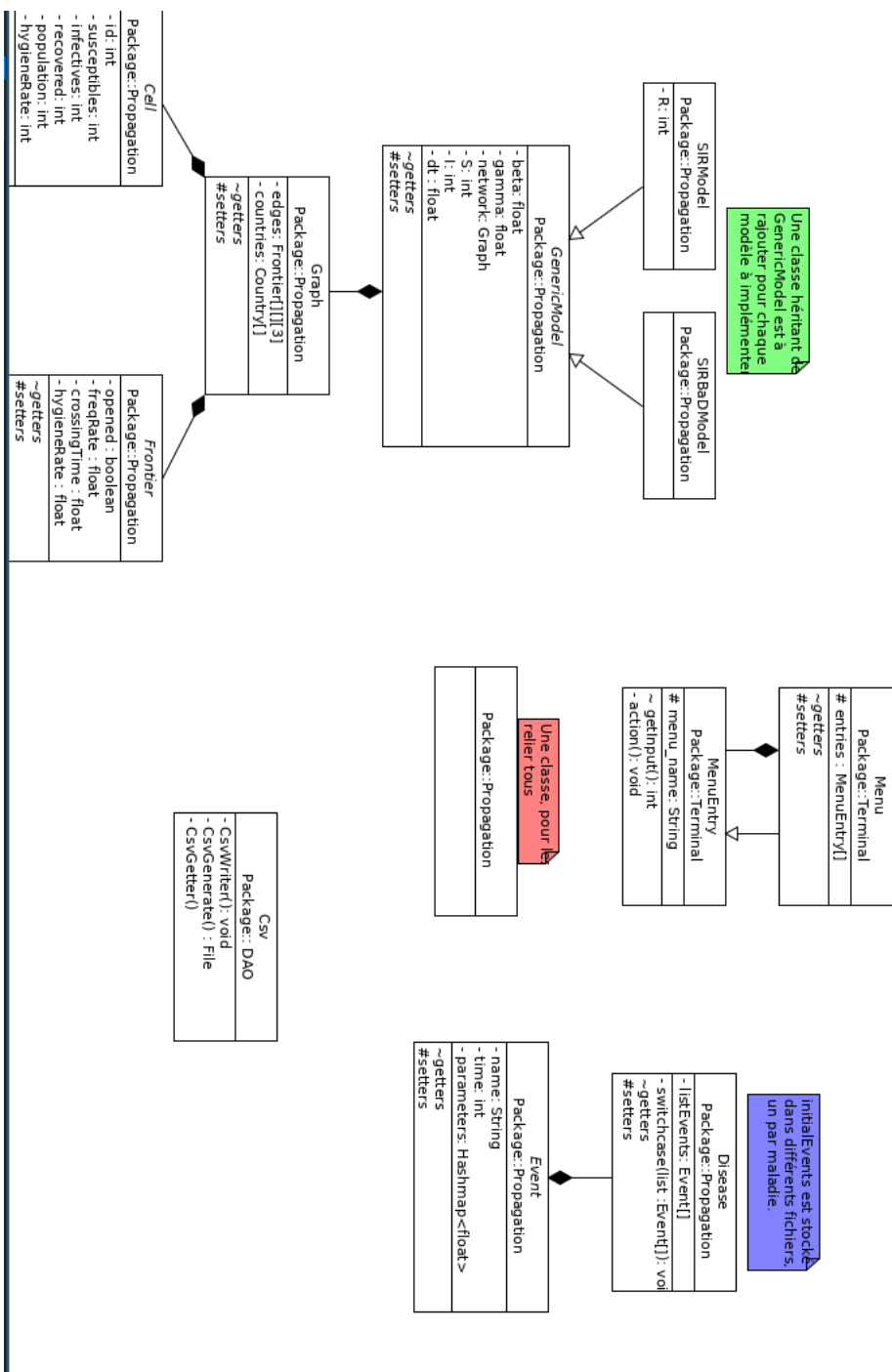
évolution dans le temps, nous utiliserons la librairie (toolkit) graphique JavaFX.

Pour la modification de l'environnement (hygiène, température, etc...), on modifie directement les coefficients de propagation de la maladie dans la cellule.

## Regroupement modulaire des fonctionnalités

- Visualisation
  - Terminal
  - Graphique
  - Exportation en CSV
- Évènements
  - Blocage de lieux de transports
  - Blocage des frontières
- Statistiques
  - Par pays : évolutions du nombre d'infectés, ...
  - Générales
- Calcul des évolutions temporelles

# Description du flux des données entre les modules



Le diagramme UML ci-dessus ne présente pas les relations entre toutes les classes car nous comptons voir comment regrouper certaines classes dans des phases ultérieures de notre développement.



# Conception détaillée

## Guide pour le développeur

L'arborescence des paquets est la suivante :

DAO : Data Access Object : Packet contenant les classes relatives aux données. Dans notre projet les données sont stockées dans fichier .txt avec séparateurs.

- Main : Contient la classe Context.java qui est itérée tous les dt et qui fait lance la création des objets et le déroulement du programme.
- propagation : Ce paquet contient les éléments relatifs aux pays, frontières, les modèles épidémiques et les événements. Nous voulions modélisé trois types de frontières : Air, Land, Maritime. Les appels pour les extractions de données du CSV se font dans la classe Graph.java.
- service : Ce paquet contient une première version de résolution des équations différentielles avant l'implémentation de utils.
- terminal : Ce paquet contient l'ensemble des fonctions donnant le terminal utilisateur.
- utils : Ce paquet contient les outils de résolution des équations différentielles en proposant une implémentation de la méthode d'Euler.

# Bilan

## **Comparaison entre l'objectif et la réalisation**

Nous avions initialement prévu de réaliser une interface graphique mais nous avons sous estimé la partie fonctionnelle du programme qui nous a pris plus de temps que prévu. En particulier, la modélisation initiale du problème a été complexe, il a fallu adopter une structure permettant de gérer plusieurs scénarios.

# Manuel utilisateur

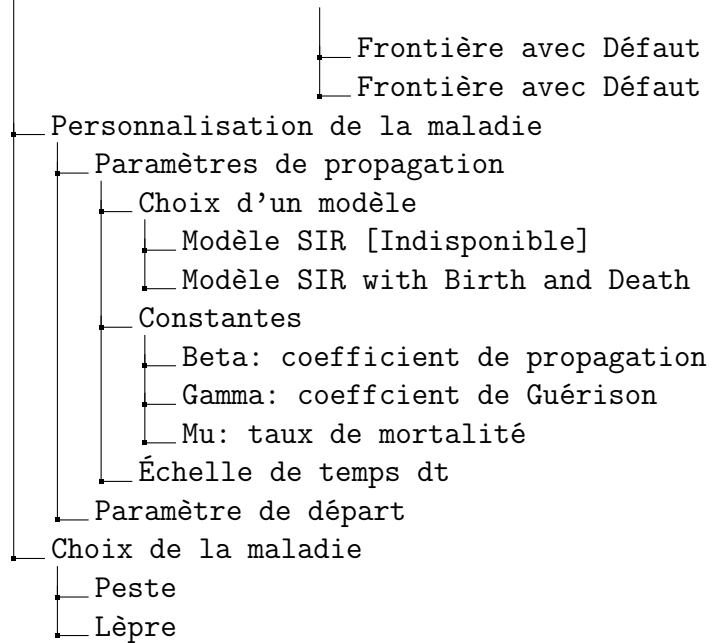
## Guide d'utilisation du programme PlagueINT

### Aborescence des Menus

L'interaction entre l'utilisateur et le programme se fait par l'utilisation d'un terminal divisés en Menus.

Gérer les événements

- └─ Lister et Supprimer l'événement choisi
  - └─ Créer un événement
    - └─ At which time does the event trigger?
      - └─ Constantes
        - └─ Beta: coefficient de propagation
        - └─ Gamma: coefficient de guérison
        - └─ Mu: taux de mortalité [Indisponible]
      - └─ Paramètres du graphe
        - └─ Default
          - └─ Population du pays
            - └─ Nombre de Sains
            - └─ Nombre d'infectés
            - └─ Nombre de guéris
          - └─ Etat des frontières du pays
            - └─ Frontière avec Défaut
            - └─ Frontière avec Défaut
            - └─ Frontière avec Défaut
        - └─ Default
          - └─ Population du pays
            - └─ Nombre de Sains
            - └─ Nombre d'infectés
            - └─ Nombre de guéris
          - └─ Etat des frontières du pays
            - └─ Frontière avec Défaut



Les événements sont des phénomènes modifiant les données du problème à un instant  $t$ . Par exemple : la fermeture d'une frontière.