



Simudémie

Livrable 4

Travail remis à
Thierry Eude
dans le cadre du cours
GLO-2004 *Génie Logiciel Orienté Objet*

Par :

Maï-Anh Porlier (111 099 753)
Monique Lambert (111 264 598)
Charles Paquet (111 107 151)
Guillaume Vallerand (536 820 757)

Département d'informatique et de génie logiciel
Faculté des sciences et génie, Université Laval

Date de remise : 26 avril 2021

Table des matières

1 Énoncé de vision

1.1	Positionnement
1.1.1	Énoncé du problème
1.1.2	Position du produit
1.2	Parties prenantes et leur problèmes
1.2.1	Utilisateurs
1.2.2	Environnement utilisateurs

2 Capture d'écran de l'application

3 Modèle du domaine

3.1	Diagramme
3.2	Texte explicatif
3.2.1	Maladie
3.2.2	Carte du monde
3.2.3	Pays
3.2.4	Région
3.2.5	Mesure Sanitaire
3.2.6	Forme
3.2.7	Voie de Liaison
3.2.8	Irrégulière
3.2.9	Régulière

4 Diagramme de classe de conception

4.1	Diagramme
4.2	Texte explicatif
4.2.1	Contrôleur de Larman
4.2.2	Simulation Manager
4.2.3	Drawing
4.3	Dtos
4.3.1	Entités
4.3.2	Helpers

5 Diagramme de cas d'utilisation

6 Architecture Logique

6.1	Diagramme de packages
6.2	Texte explicatif
6.2.1	Couche présentation
6.2.2	Couche domaine

7 Conclusion

7.1	Points forts
7.2	Points à améliorer
7.3	Contribution des membres de l'équipe

1 Énoncé de vision

L'application que nous désirons développer servira comme support aux autorités pour guider leurs décisions dans un contexte épidémiologique. Il permettra de faire des simulations d'épidémie pour que ses utilisateurs puissent juger l'efficacité de leurs mesures sanitaires. L'application aura la fonctionnalité de pouvoir contrôler la force de la maladie et le taux de propagation.

1.1 Positionnement

1.1.1 Énoncé du problème

Les contextes de pandémie sont des situations difficiles à gérer pour les gouvernements et les autorités. Il est délicat pour ceux-ci de prendre des décisions en se basant uniquement sur des spéculations et des prévisions. Notre solution au problème est de donner un outil à ces autorités en leur offrant une application pouvant modéliser une pandémie et pouvoir tester les résultats selon différents paramètres.

1.1.2 Position du produit

L'application que nous voulons développer prend de plus en plus de place sur le marché compte tenu de la situation de pandémie qui touche la majorité des régions depuis la dernière année. Notre vision du produit permet une flexibilité hors du commun car elle permet aux différentes autorités gouvernementales de paramétriser la force de la maladie, le taux d'adhésion de la population et la force des mesures sanitaires appliquée aux différentes régions. Actuellement sur le marché, les applications de simulation ne permettent pas de paramétriser les différentes propriétés des maladies et d'ajouter des mesures sanitaires différentes pour chaque région.

1.2 Parties prenantes et leur problèmes

1.2.1 Utilisateurs

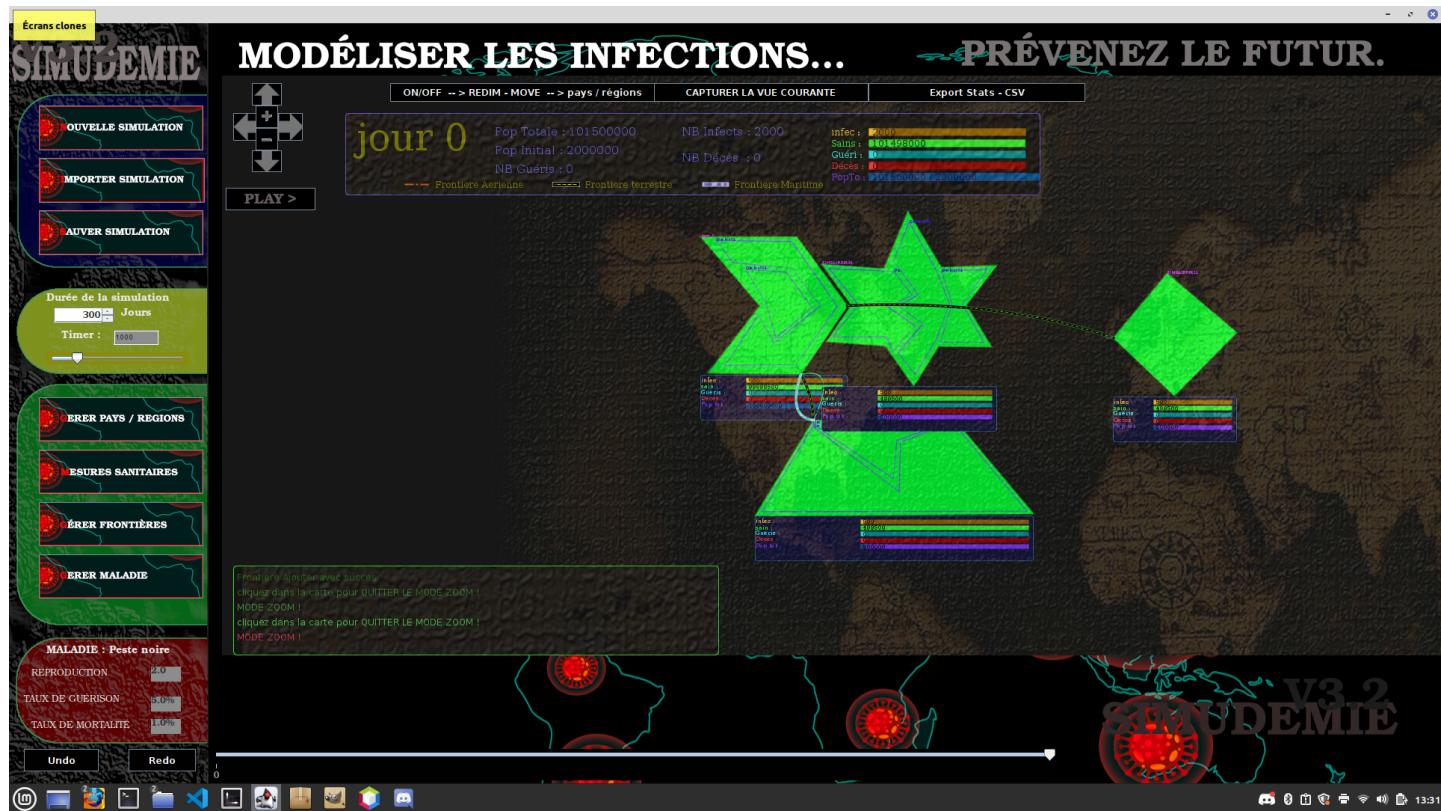
Simudémie vise à aider les autorités, les gouvernements, les chercheurs et l'organisation mondiale de la santé à réaliser des simulations à différentes échelles. Les intervenants souhaitent avoir des estimations précises en ce qui concerne différentes mesures pouvant être implémentées dans un contexte de pandémie.

1.2.2 Environnement utilisateurs

L'application simudémie sera utilisée par les autorités gouvernementales sur un ordinateur. La connexion internet n'est pas nécessaire pour l'utilisation de l'application.

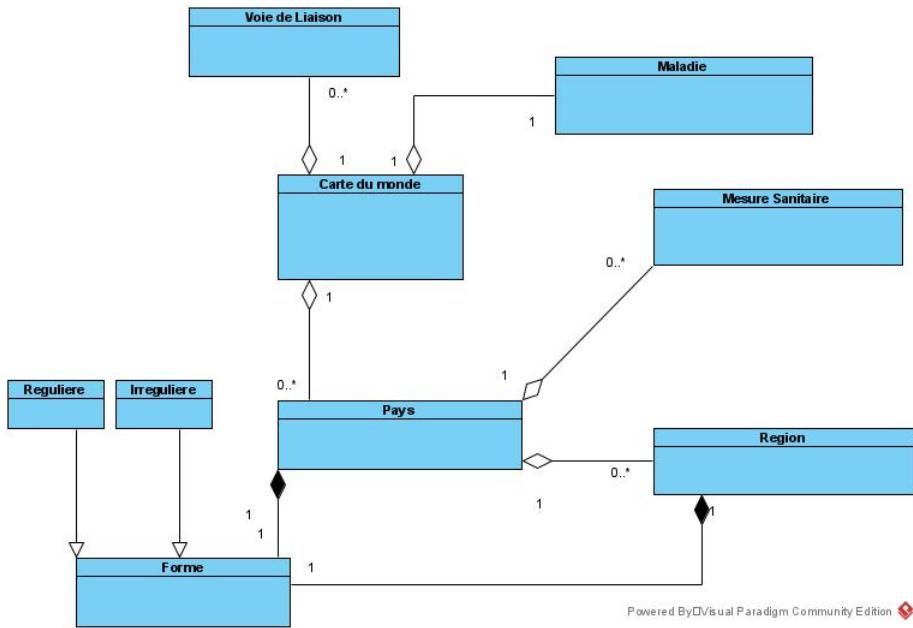
L'application se présente sous une allure dynamique mettant de l'avant une simplicité dans la navigation. la majorité des fonctionnalités sont paramétrées directement dans la fenêtre principale.

2 Capture d'écran de l'application



3 Modèle du domaine

3.1 Diagramme



3.2 Texte explicatif

3.2.1 Maladie

La classe conceptuelle **Maladie** représente la maladie dont les effets sur la population seront représentés dans une simulation. Chaque instrument possède un Taux de reproduction, un Taux de guérison, un Taux de mortalité et un Taux de transmission ainsi qu'un Nom. Il est possible de modifier les paramètres de la maladie. Cependant on ne peut pas inclure plus de 1 maladie par simulation.

3.2.2 Carte du monde

La classe conceptuelle **Carte du monde** correspond à une représentation d'une carte du monde qui peut contenir 0 ou plusieurs pays et qui a une certaine Taille. Elle est initialement vide et elle est personnalisée par l'utilisateur. Il y a seulement une carte du monde par simulation.

3.2.3 Pays

La classe conceptuelle **Pays** représente un pays de la carte du monde qui pourrait être influencé par une maladie et dont les statistiques et les comportements pourraient avoir une influence sur d'autres pays. Chaque **Pays** contient une Superficie, une Taille de Population, un Nom, une Couleur selon le niveau d'infection et de danger que la maladie a avec les autres pays, un Nombre de personnes infectées et un Nombre de personnes décédées de la maladie.

3.2.4 Région

La classe conceptuelle **Région** correspond à une section à l'intérieur d'un pays. Les pays peuvent être divisés en 0 ou plusieurs régions. Les régions ressemblent beaucoup au pays et il est possible de modifier leurs propriétés aussi qui sont le Pourcentage de population qui est associée à une région et la Superficie d'une région ainsi qu'un Nom.

3.2.5 Mesure Sanitaire

La classe conceptuelle **Mesure Sanitaire** correspond à une loi que le gouvernement impose pour contrer une maladie. Il peut y avoir plusieurs mesures sanitaires appliquer sur un pays ou sur une région en particulier. L'utilisateur peut changer les paramètres d'une **mesure sanitaire**, ajouter une mesure ou ôter une mesure durant une simulation. Il y a une liste de mesures sanitaires de base déjà inclus dans le logiciel, mais l'utilisateur peut en créer s'il le désire. Chaque Mesure Sanitaire contient un Moment de mise en vigueur qui détermine à partir de quelle elle entre en vigueur, une Durée de mise en vigueur qui détermine à partir de quel moment la mesure ne sera plus appliquée, un Taux d'adhésion de la population qui montre à quel point la population du pays ou de la région trouve celle-ci populaire et la Force avec laquelle la mesure sera appliquée.

3.2.6 Forme

Chaque objet **Forme** correspond à la forme géométrique qui représentera un pays ou une région sur la carte du monde. Cette forme peut être une forme standard pré-faite comme

un carré ou un triangle ou bien une forme peut être dessiner par l'utilisateur pour être irrégulière. Chaque objet **Forme** a une Couleur, un Grosseur, un Nom, ainsi qu'un attribut Sommet qui correspond au nombre de sommets qui constitue la forme

3.2.7 Voie de Liaison

Un objet **Voie de Liaison** correspond à la sorte de division qui sépare un pays d'une autre. Durant une simulation ou lors de la création d'un pays, l'utilisateur peut décider d'ajouter une frontière qui reliera un pays en question avec un autre. Chaque objet **Frontière** a un attribut Type qui sera un choix entre maritime, terrestre ou aérienne et un Nom qui représentera quels sont les deux pays reliés ensemble par cette frontière.

3.2.8 Irrégulière

Un objet **Irrégulière** correspond à une forme géométrique basique que l'utilisateur peut choisir pour représenter un pays ou une région.

3.2.9 Régulière

Un objet **Régulière** correspond à une forme géométrique basique que l'utilisateur peut choisir pour représenter un pays ou une région.

4 Diagramme de classe de conception

4.1 Diagramme



4.2 Texte explicatif

4.2.1 Contrôleur de Larman

Le contrôleur de Larman sert d'intermédiaire entre l'interface de l'utilisateur et le domaine de Simudemie. Lorsqu'un bouton ou une action sera faite dans l'inteface, celle-ci sera gérée par l'interface mais l'action à entreprendre et la logique sera faite par le contrôleur. Cela permet l'indépendance entre les deux entités. Le contrôleur créera une carte du monde et s'occupera de faire la gestion de celle-ci. Il fera aussi l'appel à la classe Serializable pour qui s'occupe de faire la sauvegarde des simulations et du CarteDrawer qui dessine la carte du monde sur le Jpanel.

4.2.2 Simulation Manager

Le package Simulation Manager contient une seule classe Serializable qui s'occupe de faire la conversion des données binaires pour la sauvegarde, le chargement et la création d'une nouvelle simulation dans l'application. Il possède comme attribut une liste d'état. Cette liste

d'état est une implémentation de la structure de donnée d'une liste doublement chaînée. Elle sera surtout utilisé pour avoir un contrôle temporel si l'utilisateur souhaite revenir en arrière dans la simulation. Chaque noeud possédera un objet Carte du monde. Cette structure de donnée s'avèrera très utile pour avoir accès aux données des itérations précédentes puisqu'on sauvegardera l'état de la carte du monde dans un objet noeud en créant une copie de celle-ci à chaque itération. Si le contrôleur souhaite retourné en arrière, l'attribut carte du monde de contrôleur sera donc changer pour cette objet carte du monde.

4.2.3 Drawing

Le package Drawing contient une classe CarteDrawer qui s'occupe de dessiner la carte du monde en mémoire pour que le JPanel le dessine sur l'interface.

4.3 Dtos

Si l'interface veut avoir accès aux attributs des classes Maladie, Pays, Région, Frontière et Mesure Sanitaire, elle doit passer par des dtos pour pouvoir faire le pont entre l'interface et le domaine. Les dtos peuvent être accédé grâce à des accesseurs pour chacun et ils possèdent tous un constructeur qui permet de prendre l'objet du domaine et le transformé en dto pour ensuite le donner à l'interface.

4.3.1 Entités

Les entités sont essentiellement les classes du domaine qui seront nécessaires pour afficher l'information à l'utilisateur de l'application. Elles représentent les concepts réels du problème et implémentent les méthodes pour le déroulement de la simulation. Elles ont tous une connexion étroites entre elles et elles seront gérées par le contrôleur de Larman.

Carte du Monde

La carte du monde est la classe la plus importante des entités, c'est elle qui est reliée directement au contrôleur de Larman et elle est directement ou indirectement lié avec les autres classes. Elle possède comme attribut une ArrayList de pays, une maladie, le pays qui sera sélectionné par l'utilisateur ainsi que la région sélectionné s'il y en a. La classe pourra ajouter des pays dans sa list et en supprimer si nécessaire. De plus, elle aura une méthode qui permet de faire un retour en arrière de la simulation ainsi qu'un incrémentation du temps.

Liste d'état

Cette liste d'état est une implémentation de la structure de donnée d'une liste doublement chaînée. Elle sera surtout utilisé pour avoir un contrôle temporel si l'utilisateur souhaite revenir en arrière dans la simulation. Chaque noeud possédera un objet Carte du monde, un pointeur vers son noeud précédent et un pointeur vers son noeud suivant. Cette structure de donnée s'avèrera très utile pour avoir accès aux données des itérations précédentes puisqu'on sauvegardera l'état de la carte du monde dans un objet noeud en créant une copie de celle-ci à chaque itération. Si le contrôleur souhaite retourné en arrière, l'attribut carte du monde de contrôleur sera donc changer pour cette objet carte du monde.

Pays

La classe pays représente l'état d'un pays sur la carte du monde entre autre par ses différentes statistiques. Il possède comme attribut une liste de région, une taille de population, un nom, une forme (irrégulière ou non), une liste de mesures sanitaires, un ou des frontières, une couleur qui indiquera si le pays est en état critique ou non selon le nombre de personne infecté, un nombre de personne guéris, un pourcentage de personne malade et une position qui sera des point cardinaux important pour le JPanel. L'utilisateur a aussi le choix de diviser son pays en région et celui-ci possèdera donc une liste de région comme attribut. Chaque attribut pourra être modifié avec des assignateurs et mutateurs. La classe possède aussi des méthodes qui mettront à jour le nombre d'infecté et le nombre de personne guéris à l'aide de statistique et des mesures sanitaire implémenté dans ce pays.

Région

La classe région est une subdivision d'un pays, elle a comme attribut le pourcentage du pays en question, un nom et des coordonnées. Chaque attributs ont également des mutateurs et assignateurs. On peut aussi modifier les statistique d'une région. Cette méthode appellera des méthodes statiques dans le StatistiqueHelper pour quantifier et calculer le nombre de personne infectées et celles qui seront guéris après les itérations.

Maladie

La classe maladie représente la maladie qui sera la cause de la pandémie lors de la simulation. Elle possède comme attribut un taux de reproduction, un taux de mortalité, un taux de transmission et un nom. Chaque attributs possèdent un mutateur et un assignateur. Ces attributs seront important lors du calcul pour le nombre de personne infecté dans les pays à l'aide du StatistiqueHelper.

Mesure Sanitaire

La classe de mesure sanitaire représente entre autre les mesures qui seront implémenté par les différents pays pour contrer la maladie de la pandémie. Cette classe possède comme attribut un nom, un taux d'adhésion de la population, une efficacité qui est calculé selon le taux d'adhésion de la mesure, un début et une durée qui représente un instant temporel. Tous les attributs possèdent des mutateur et des assignateur pour être en mesure de les modifier et en avoir accès. La classe mesure sanitaire sera entre autre utilisé dans la simulation pour calculé le nombre de personne guéris dans un pays après un itération par l'entremise du StatistiqueHelper.

Voie de Liaison

Cette classe est une classe abstraite, elle défini une voie de liaison entre pays et le polymorphisme sera utilisé pour changer le comportement de cette voie selon sa classe enfant (Aerienne, terrestre ou maritime). Elle a pour attribut un booleen qui represente si la frontière est ouverte ou non et un taux d'achalandage. Tous les attributs ont des mutateurs et assignateurs puisque ceux-ci pourraient changer si le pays déciderait de fermer ses frontières lors de la simulation. Les attributs des classes enfants seront entre autre utilisé dans la simulation pour calculé le nombre de personne infectées dans un pays après un itération par l'intermédiaire du StatistiqueHelper.

Forme Geometrique

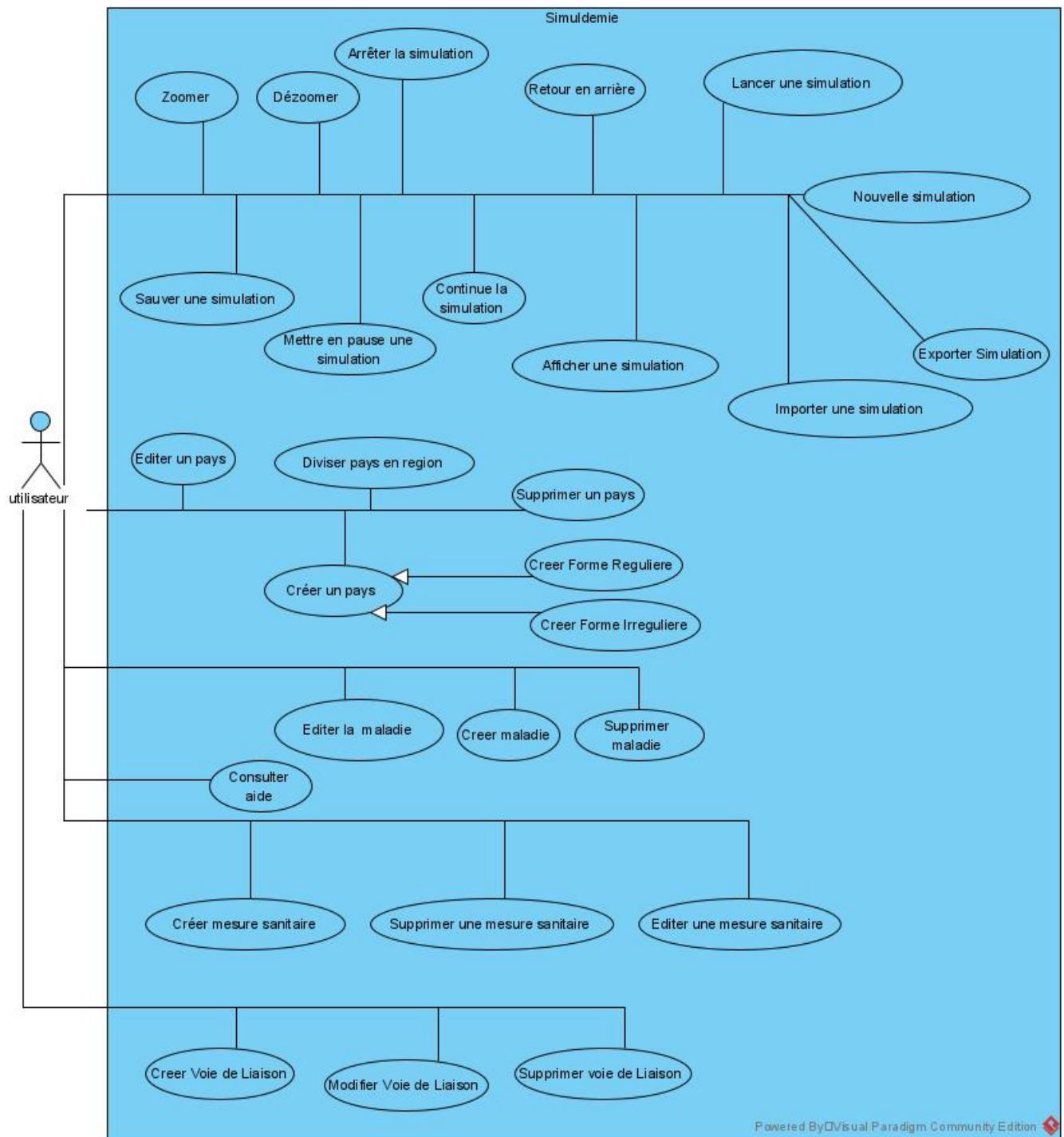
Les formes géométrique sont sous forme d'un package, il contient la classe Forme, une classe abstraite qui a pour attribut une couleur et une liste de point avec des mutateurs et assignateurs. Cette classe est utilisé pour représenté une forme d'un pays. Le pays peut avoir une forme irrégulière ou une forme régulière d'où les deux classes enfants. Selon ce que l'utilisateur décide de choisir la forme de son pays, le constructeur de Regulière ou Irreguliere sera appelé pour pouvoir créer le bon type de forme. Il aura aussi le choix d'ajouter des sommets à cette forme avec la méthode addPoint(). La classe sera aussi utilisé pour notre algorithme qui détermine si le clique d'un utilisateur est dans une forme régulière ou non.

4.3.2 Helpers

Les helpers sont des classes provenant du package helper. Celui-ci contient entre autre StatistiqueHelper et GeometrieHelper. La première classe possède des méthodes statiques qui seront fortement utiles pour calculer les statistiques des pays et des maladies de la simulation. Ces méthodes ne prennent qu'en paramètres des ints et n'ont pas de relation directe avec les

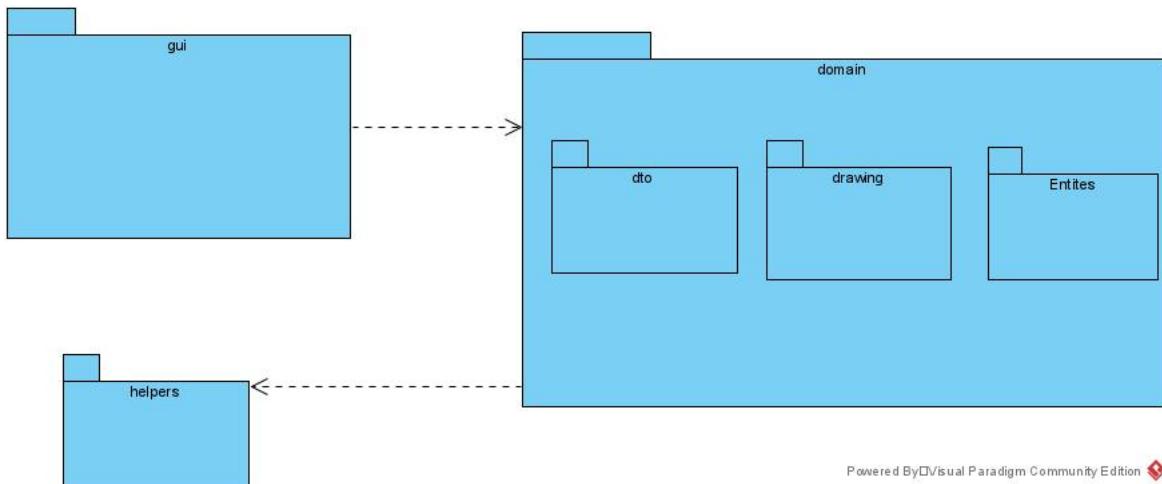
objects donc elles sont statiques. Pour ce qui est de la classe GeometrieHelper, elle nous sera utile pour faire le calcul de coordonnée de points et déterminé la dimension d'un pays et si celui-ci a une forme régulière ou non.

5 Diagramme de cas d'utilisation



6 Architecture Logique

6.1 Diagramme de packages



6.2 Texte explicatif

L'architecture est divisé en deux couches tel que le modèle-vue le demande. Nous avons la couche présentation qui est le GUI et la couche du domaine.

6.2.1 Couche présentation

La couche de présentation contient tout ce qui est en lien avec l'interface de l'utilisateur. Comme le demande l'architecture modele-vue, elle ne fait aucun calcul logique et ne sert que d'affichage. Elle contient un JPanel, un drawing panel, une main window. Elle fait appel au contrôleur lorsque de la logique doit être appliquée aux évènements. La couche présentation utilise la librairie Java Swing pour ce qui est des méthodes et gestion d'évènement effectué par l'utilisation.

6.2.2 Couche domaine

La couche de domaine comprend l'ensemble de la logique de l'application. Elle commence par le contrôleur de Larman dont le but est de déléguer les tâches selon ce que l'utilisateur entre comme commande ou action. Il s'occupe de faire les modifications à l'interne des classes et des calculs de statistiques pour retourner l'information nécessaire à l'interface.

7 Conclusion

7.1 Points forts

À la suite de l'élaboration de notre application voici les points forts de celle-ci.

L'application présente une belle interface, au allure moderne. Les tuiles disposées sur la gauche de la fenêtre permet un accès rapide à tous les fonctionnalités de l'application. Ces dernières sont regroupées en différentes sections pour en faciliter l'accès rapide. Ces tuiles ont été conçus afin de fonctionner autant avec des cliques de souris, qu'avec des cliques de doigts pour de futur utilisateur d'écran tactile.

L'interface dynamique, c'est-à-dire la fluidité pour modifier un paramètre et la mise à jour rapide et en temps réel de ceux-ci dans l'interface facilite grandement l'utilisation et la compréhension. L'interface communique également en temps réel des consignes à l'utilisateur à l'aide d'un encadré discret dans le coin inférieur gauche. En effet des messages, discret et non-harassant, c'est-à-dire sans fenêtre de type pop-up, communiquent les étapes à suivre afin de paramétriser la carte du monde, la maladies, les mesures, etc.

La gestion des régions, que ce soit pour l'ajout, la redimension, le déplacement et la suppression est faite de façon fluide au sein même de la carte du monde. En seulement quelques cliques conviviales, l'utilisateur arrive à créer une carte du monde.

7.2 Points à améliorer

Dans notre application, quelques points sont améliorer au point de vue de la qualité du code.

En effet, le contrôleur ne s'en tient pas seulement à commander les appels de fonction dans d'autre classe, mais fait certaine logique. Par exemple, les méthodes redimensionner, bouger, etc, devrait être sortie du contrôleur et que celui-ci les appels.

Afin de pouvoir commercialisé notre application, l'application devrait utiliser des statistiques d'avantage réaliste. En effet, ceux-ci devrait prendre en considération les naissances, les décès qui ne sont pas liés à ceux de la maladie modélisée.

7.3 Contribution des membres de l'équipe

Monique Lambert :

- Refaire l'interface et code en profondeur de Mesure Sanitaire avec des nouveaux attributs
- Refaire l'interface et code en profondeur de Frontière avec des nouveaux attributs
- Refaire l'interface et code en profondeur de Maladie avec des nouveaux attributs
- Graphe de statistiques de Carte du monde avec array d'états sauver
- Graphe de statistiques de Pays avec array d'états sauver
- Indicateur de l'activation et désactivation des mesures lors de l'exécution de la simulation
- Mise a jour des diagrammes architecture, de cas, de domaine, de conceptuelle pour le livrable

Maï-Anh Porlier :

- Application des seuils sur les frontières
- Ajustement du statistique helpers sur certains calculs
- Calcul des probabilités des infectés inter-région
- Calcul des probabilités des infectés inter-frontière avec mesure sanitaires
- Calcul des probabilités sur les pays et régions avec mesures sanitaires
- Calcul des taux effectifs de mesures sanitaires
- Ajustement du déroulement de la simulation selon les nouvelles consignes
- Ajustement dans le rapport du livrable

Guillaume Vallerand :

- Ajout Pays Irrégulier/ Régions Irr. (CTRL addPoint)
- Redim Pays/Région Irréguliere (hover points - Cible de redim)
- Move Pays/Région Irr.
- Interface Graphique : Guillaume et Monique
- TalkToUSer(msg util refacto) + animation couleurs : Guillaume
- GLobalStatDrawer (affichage jours, légendes, statsBar Globales.)
- Affichage Stats : Guillaume
- Affichage Stats Bars Hover et Global + Responsiveness redim Pays
- Courbes paramétriques animées pour frontières
- Changement de couleur et stop animation des frontières.
- TIMER SImulation de FPS ++ Tous les animations basées sur FPS
- FullModeScreen

- Code Général + design

Charles Paquet :

- Enregistrement de la simulation (sérialisation)
- Overture de la simulation (désérialisation)
- Capture d'écran du JPanel
- Recherche / remue-méninge pour le undo / redo
- Diagramme de classe de conception mise à jour
- Mise à jour du latex