



Simudémie – Livrable 4

Équipe 9

Génie logiciel orienté objet

GLO-2004

Travail réalisé par

Henri Bernard ; 111 285 905 ; henri.bernard-st-laurent.1@ulaval.ca
Frederick Hughes; 111 008 245; frederick.hughes.1@ulaval.ca
Jonathan Roy-Noel; 536 776 368; jonathan.roy-noel.1@ulaval.ca
Michael Vermette; 536 835 891; michael.vermette.1@ulaval.ca

Date de remise

26-04-2021

Table des matières

Vision	4
Contexte	5
Vision	5
Visuel	6
Modèle de domaine	13
Modèle des cas d'utilisation.....	15
Modèle de conception	17
Bilan.....	19
En bref	20
Propositions & améliorations.....	21
Équipe 9.....	22
Contribution - Livrable 1.....	23
Comptes rendus – Livrable 1.....	25
Suivi du développement – Livrable 1	27
Contribution – Livrable 2.....	28
Comptes rendus – Livrable 2	30
Suivi du développement – Livrable 2	32
Contribution - Livrable 3.....	33
Suivi du développement – Livrable 3	33
Contribution – Livrable 4.....	34
Suivi du développement – Livrable 4	34
Annexes	35
Esquisses.....	36
Interfaces – v3.1	36
Thème et design – v3.1	42
Interfaces – v2.0	44
Thèmes & design – v2.0	46
Interfaces - v1.0.....	47
Thèmes & design - v1.0	61
Diagrammes de Gantt	63
Diagramme – Livrable 1.....	63
Diagramme – Livrable 2.....	65

Pseudocode d'un algorithme – Livrable 2	67
Archives	70
Modèle de domaine	71
Modèle du domaine – Livrable 1.....	72
Diagramme de classes conceptuelles – Livrable 1	72
Description de classes conceptuelles – Livrable 1.....	73
Modèle du domaine – Livrable 3.....	74
Modèle des cas d'utilisation.....	75
Diagrammes des cas d'utilisation – Livrable 1	76
Description des cas d'utilisation – Livrable 1	80
Création d'une carte mondiale.....	80
Création d'une maladie	88
Création d'un scénario	91
Simulation.....	95
Diagramme de séquence système – Livrable 1.....	103
Diagramme des cas d'utilisation – Livrable 3	107
Modèle de conception	108
Diagramme de classe de conception – Livrable 2	109
Architecture logique – Livrable 2	112
Diagramme de séquence de conception – Livrable 2	114
Diagramme de classe de conception – Livrable 3	119

Vision

Contexte

En ce temps de pandémie et en prévision d'un futur incertain, notre équipe du cours de génie logiciel a été contactée par l'Organisation mondiale de la Santé pour développer un logiciel de simulation. Cette application devra être en mesure de générer des cartes du monde sur lesquelles des simulations pourront être effectuées. Cet outil sera utile aux utilisateurs pour observer l'efficacité des différentes mesures appliquées ainsi que la vitesse de propagation des maladies. L'OMS souhaite être en mesure d'utiliser l'application à l'échelle mondiale.

Vision

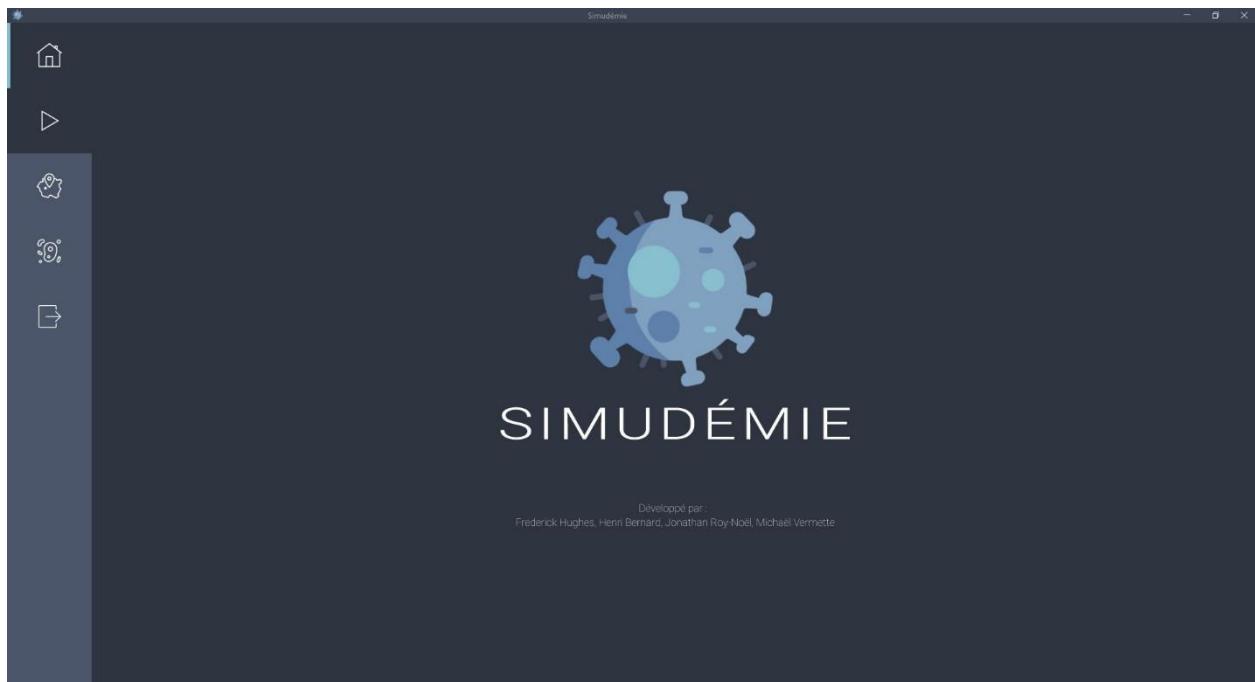
L'application sera codée en Java et dotée d'une interface graphique à partir de laquelle le client pourra effectuer ses opérations. L'application a deux fonctions principales séparées.

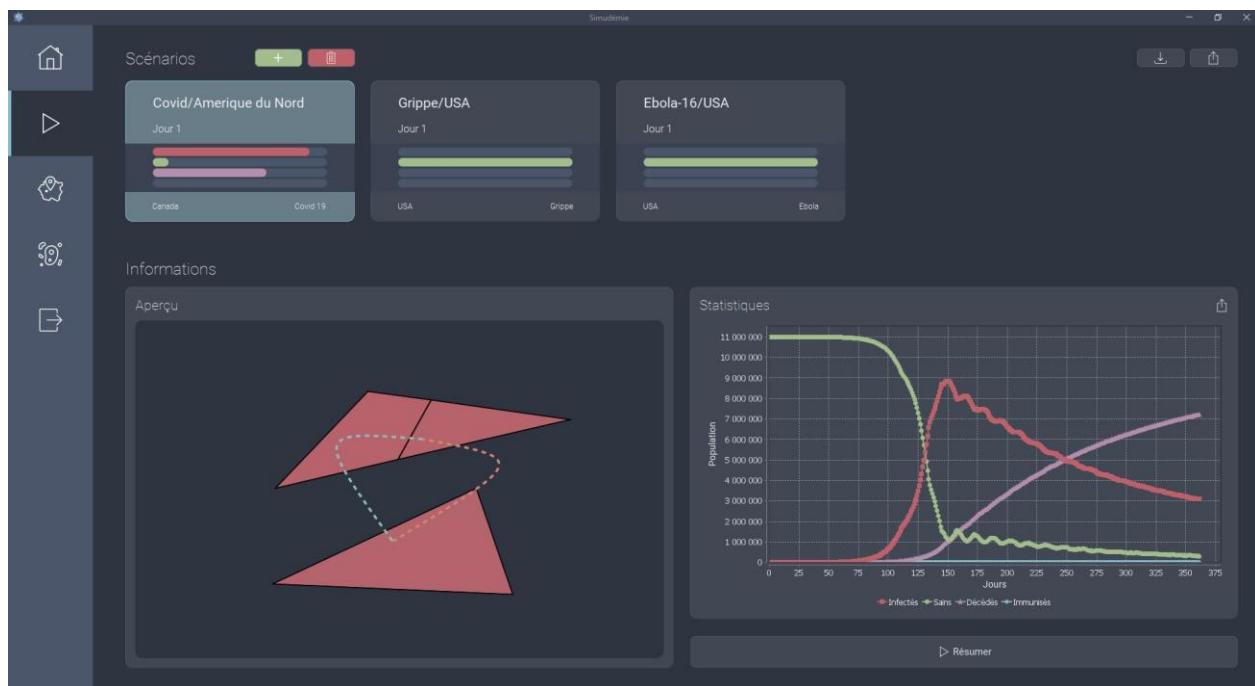
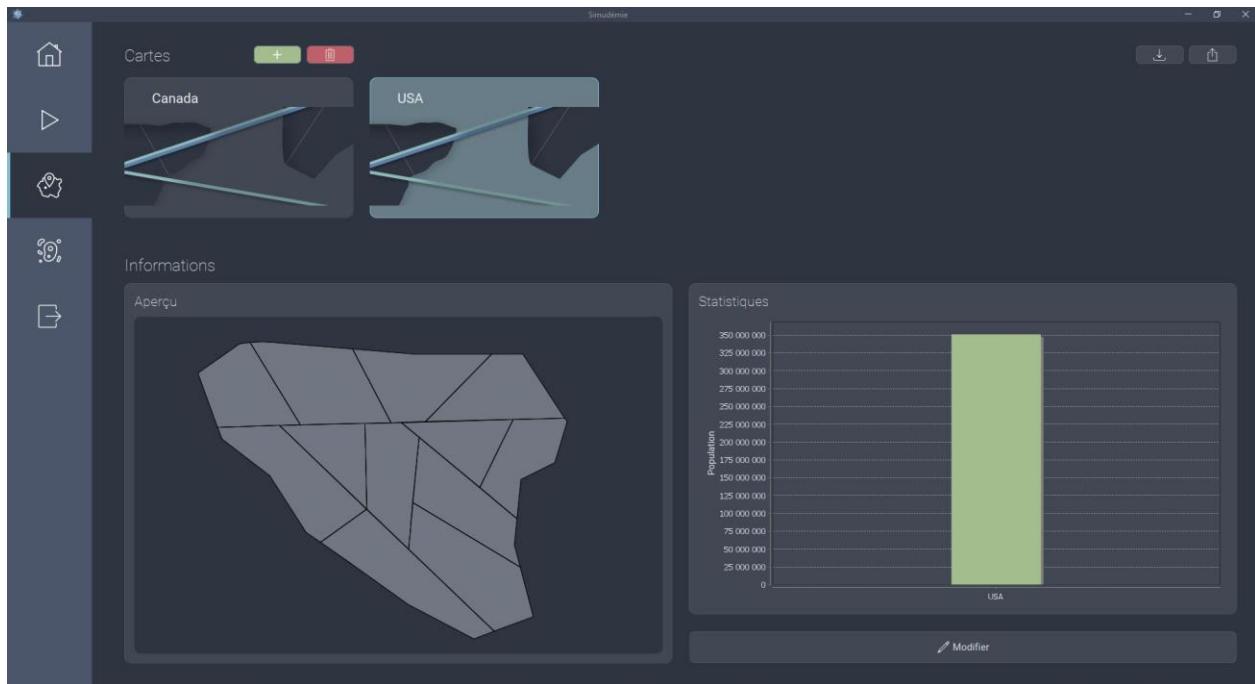
La première est de pouvoir créer une carte mondiale avec les pays désirés (et les interconnexions entre ceux-ci). Dans ce mode de création, l'utilisateur peut utiliser des formes géométriques ou créer sa propre forme pour représenter un pays. Il peut ensuite connecter les pays par des voies terrestres, aériennes et/ou maritimes. Certaines informations devront être entrées selon les pays comme le nombre de régions, la population et la densité de population (ou la superficie).

La deuxième fonction de l'application est un mode simulation. L'utilisateur entre donc les informations concernant la maladie. Ces données sont principalement le taux de propagation, le taux de guérison et le taux de mortalité (possiblement ajouter les possibilités de mutation). À partir de ces informations, la simulation peut débuter. Le programme va s'exécuter en continu pour chaque unité de temps (qu'il nous reste à déterminer) qui représentera un « pas » dans l'application (une journée). L'utilisateur peut ainsi analyser la progression de la maladie et ensuite mettre la simulation sur pause pour introduire de nouvelles mesures et l'effet de ces mesures sur la propagation de la maladie.

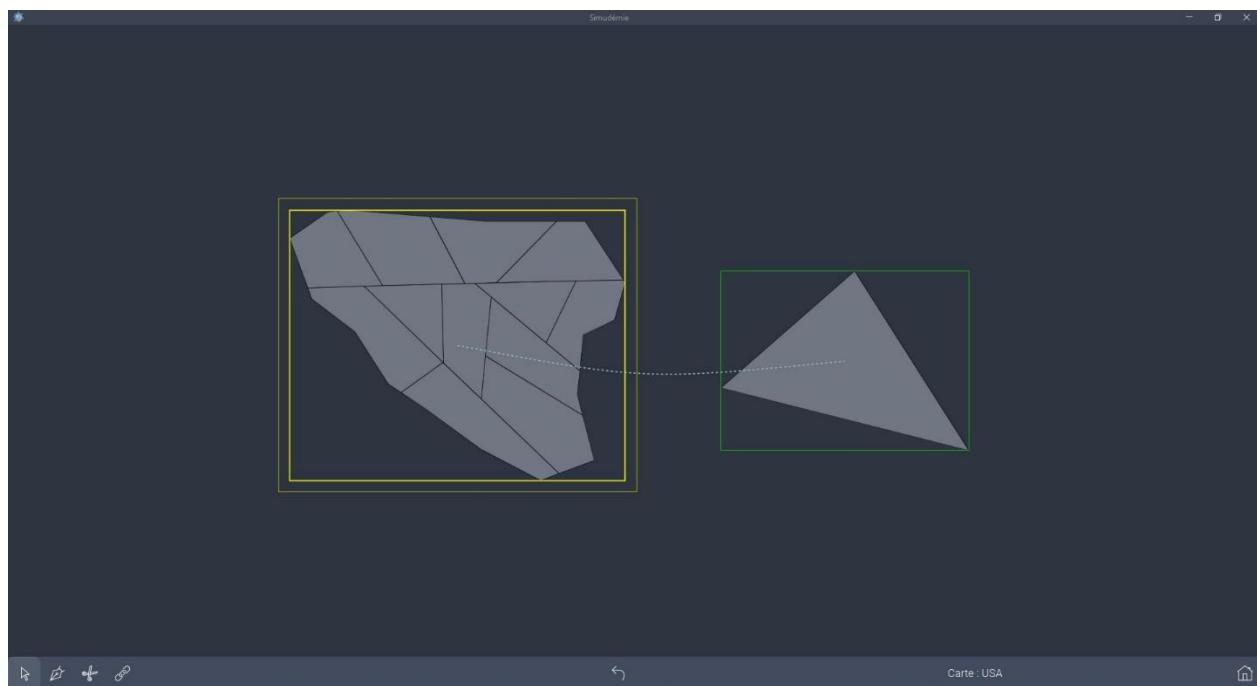
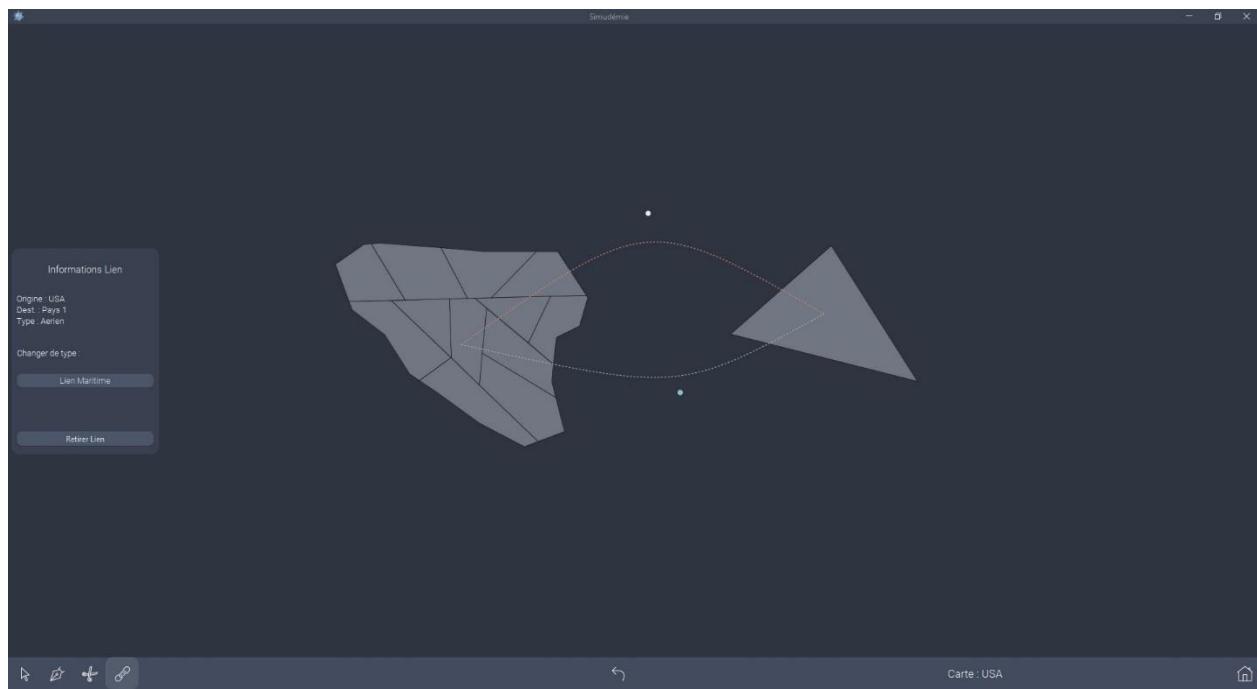
Visuel

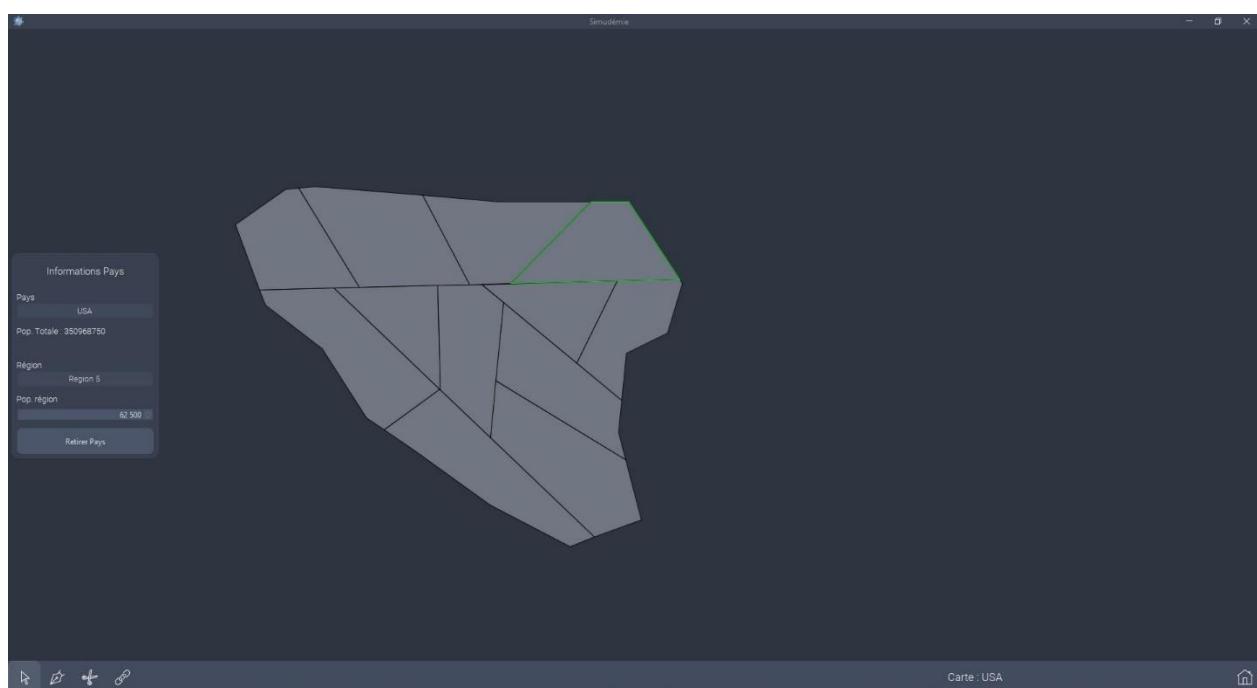
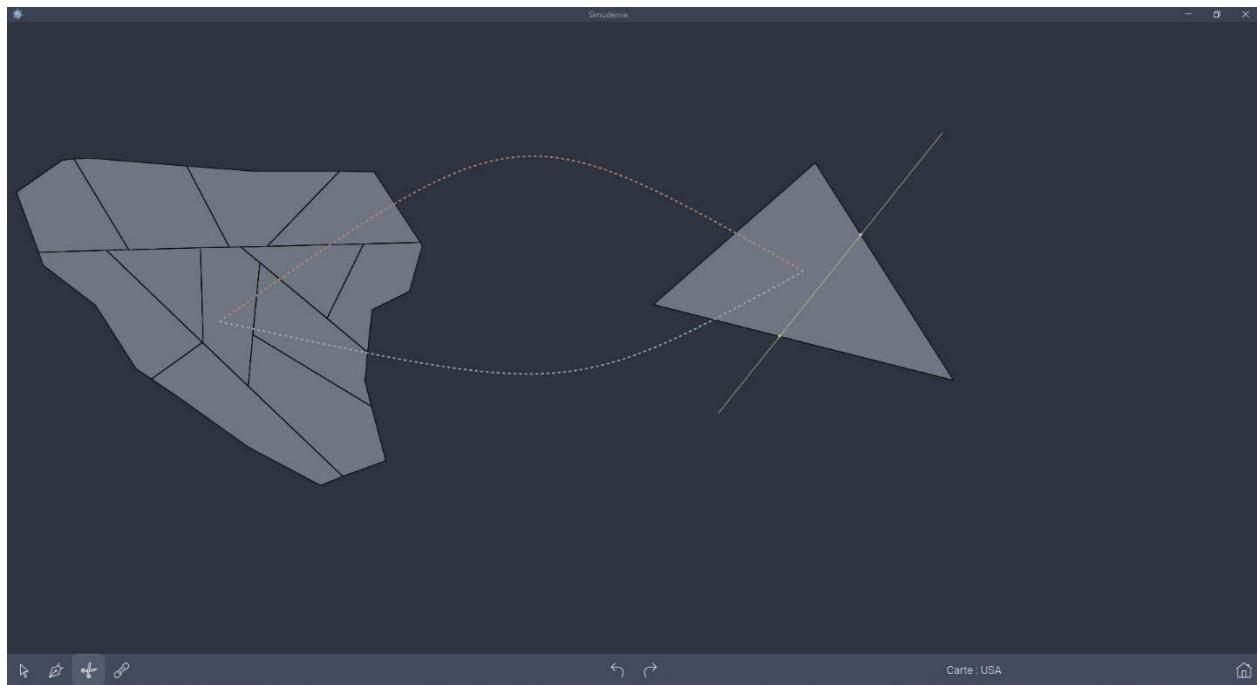
Menu Principal – v4.2



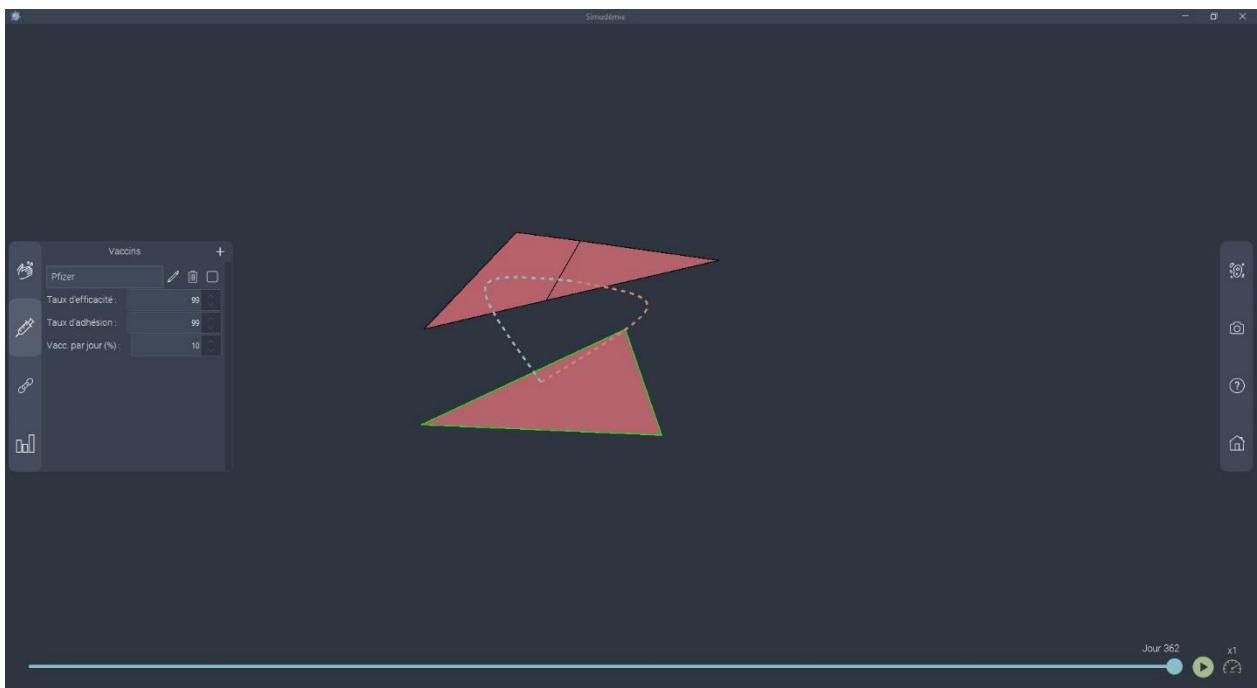
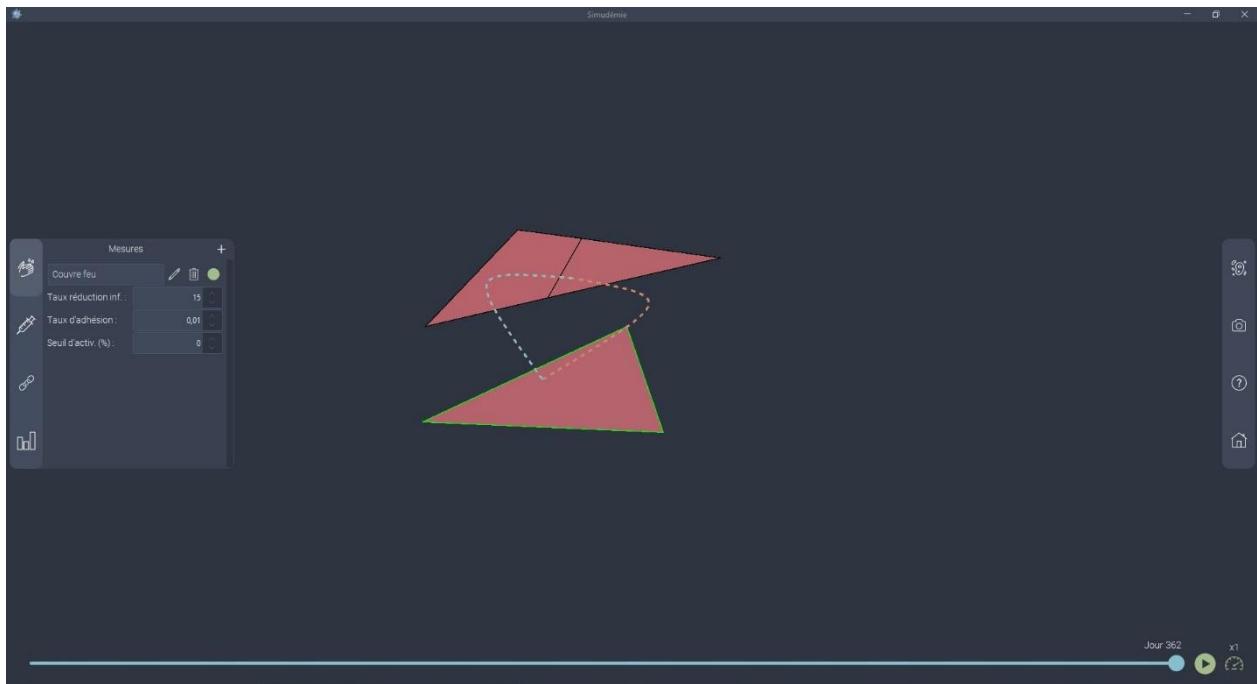


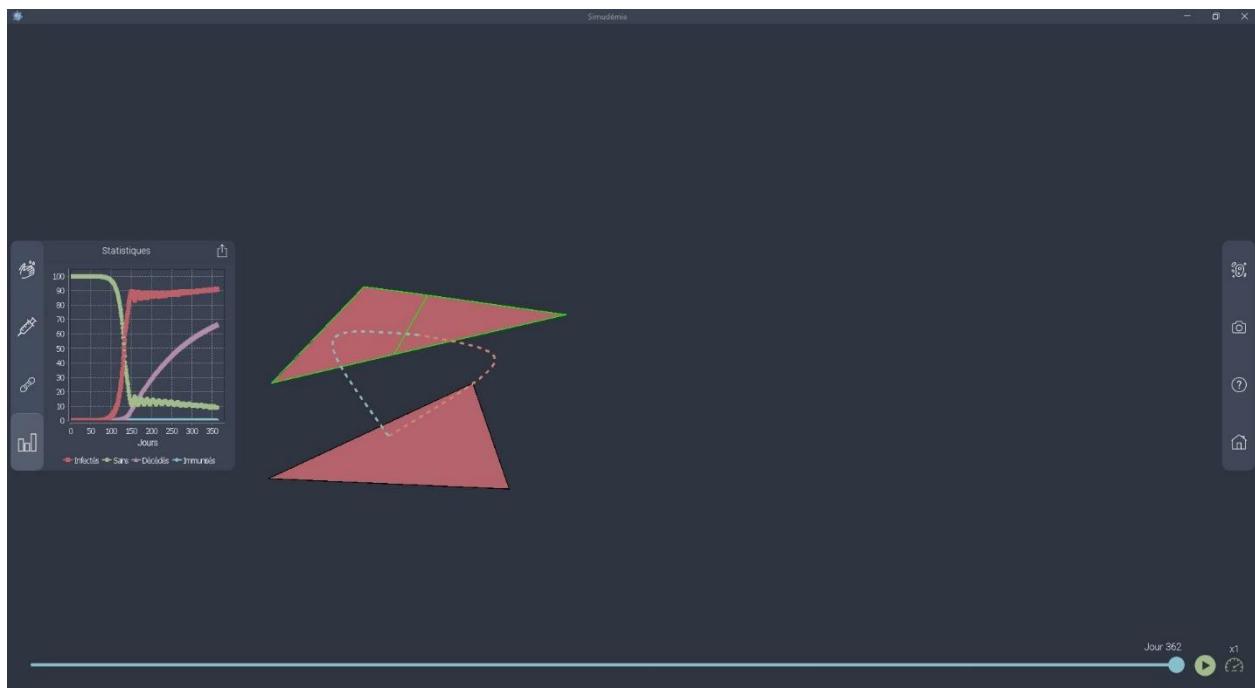
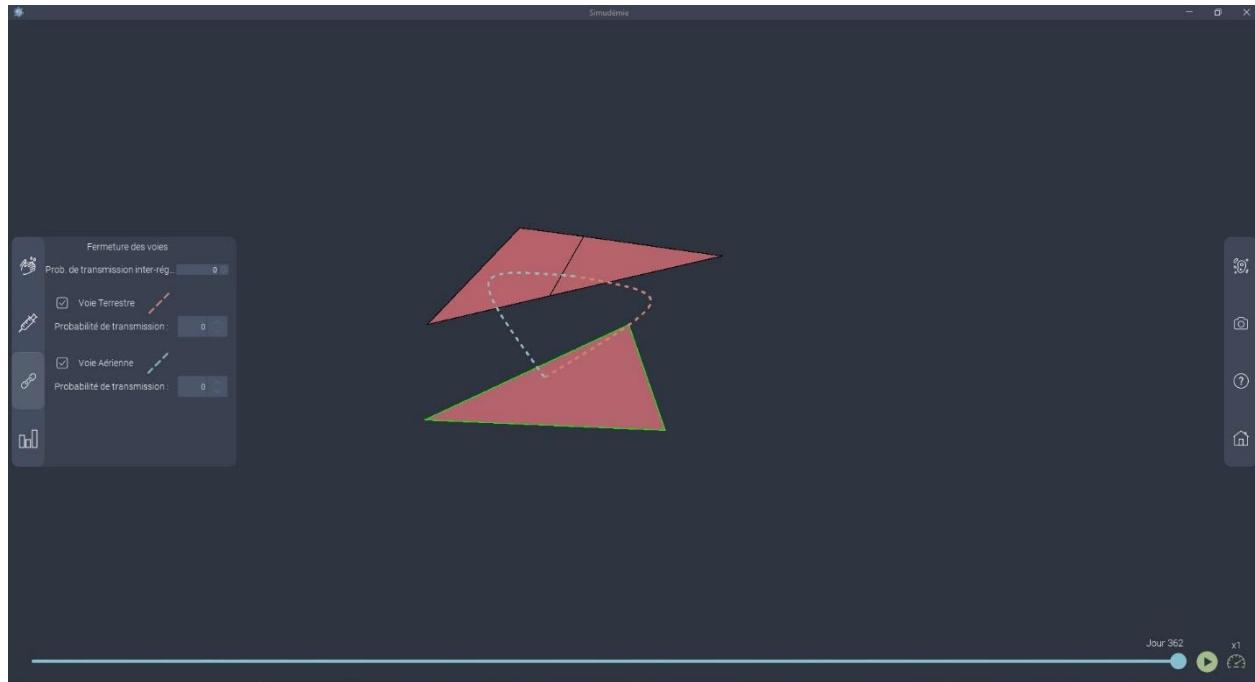
Création Carte – v4.2

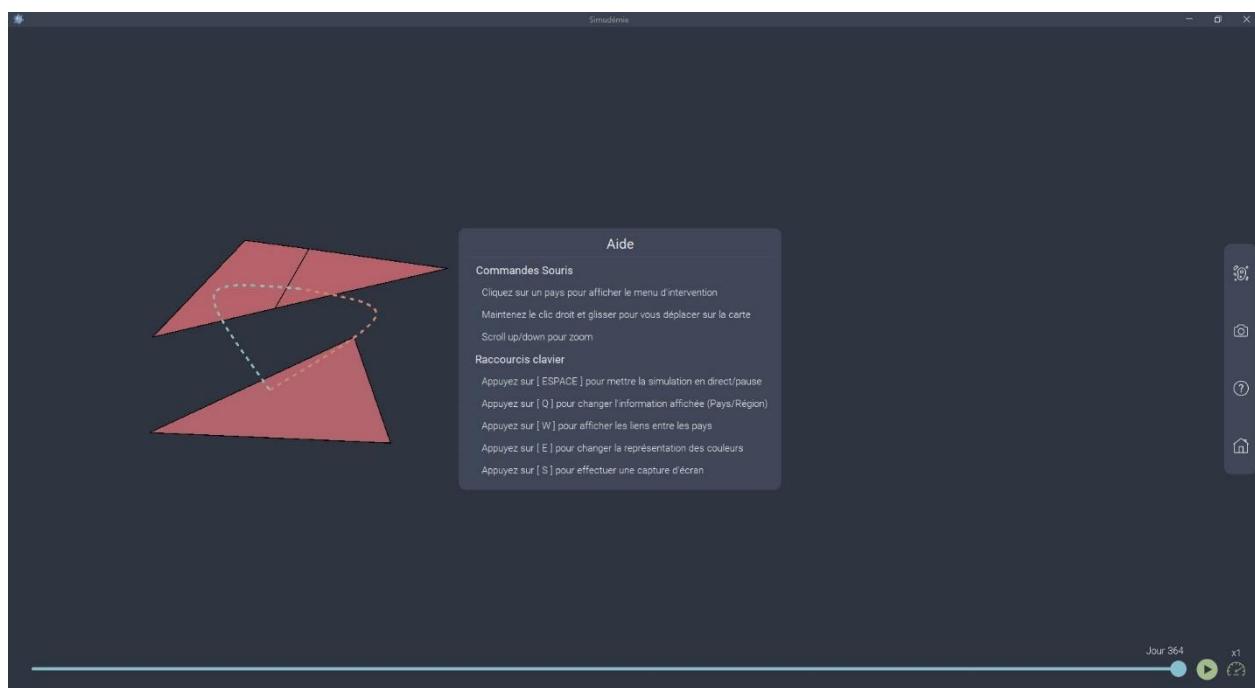
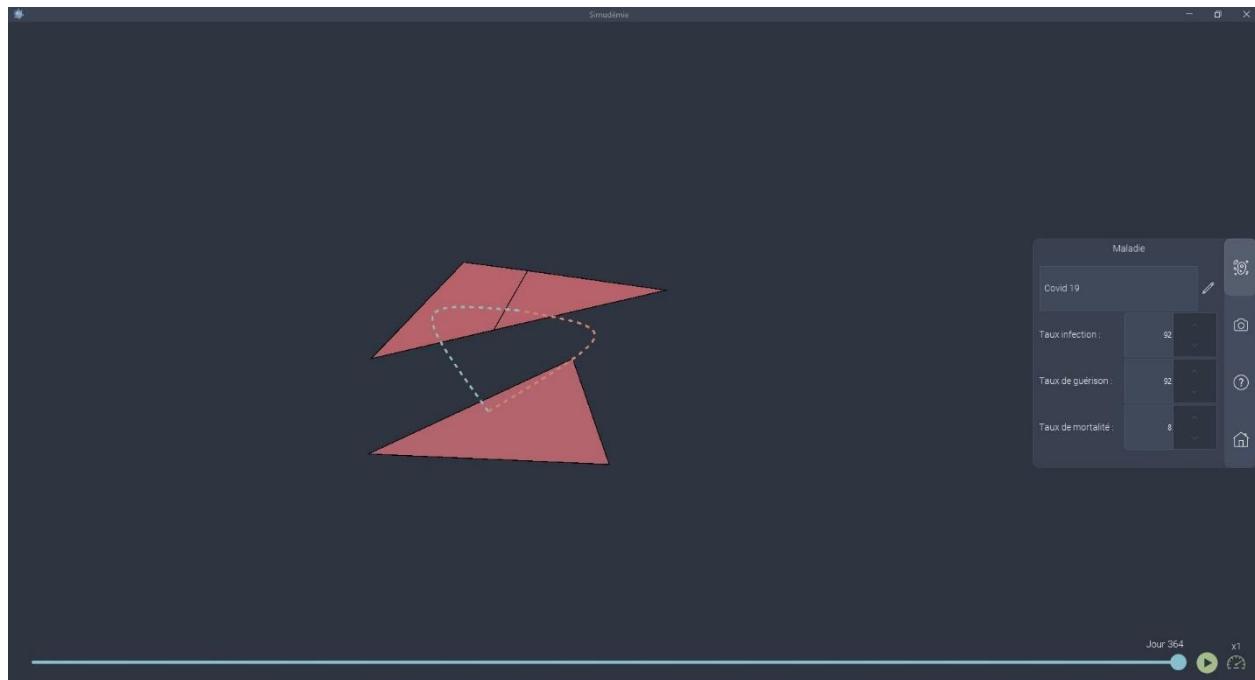




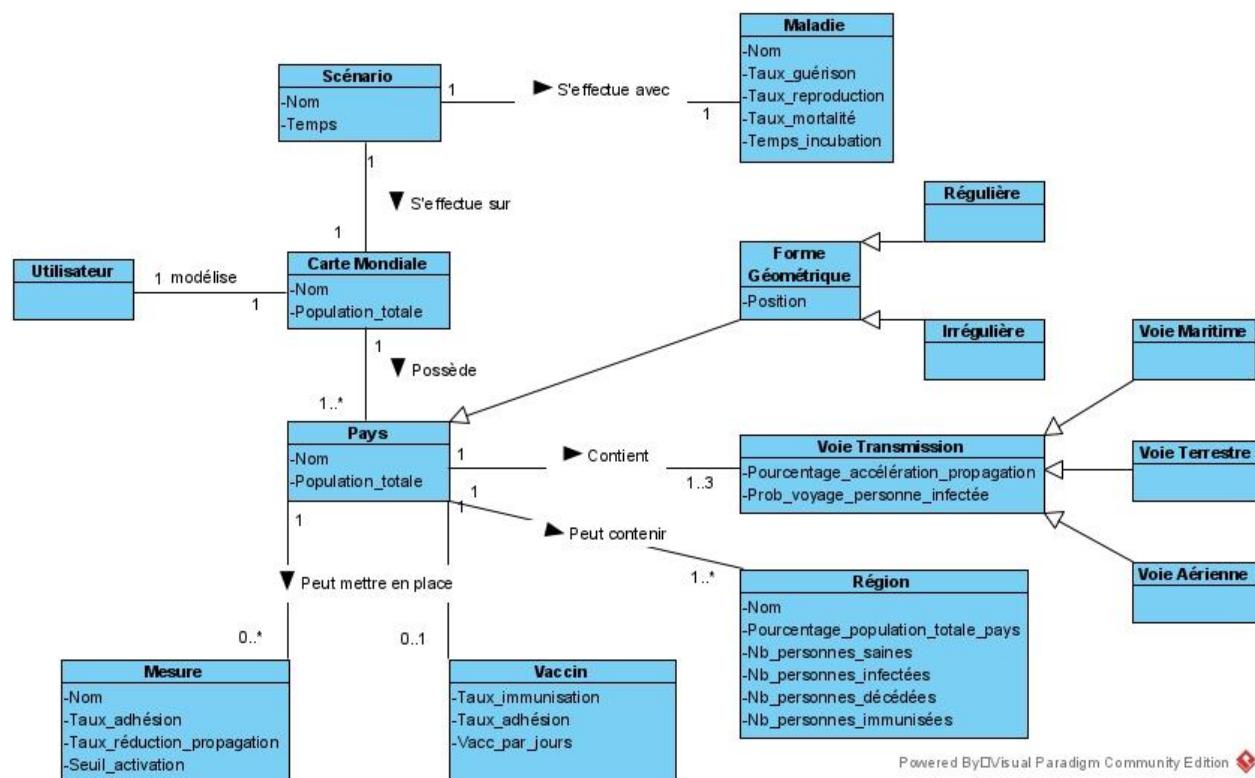
Simulation – v4.2





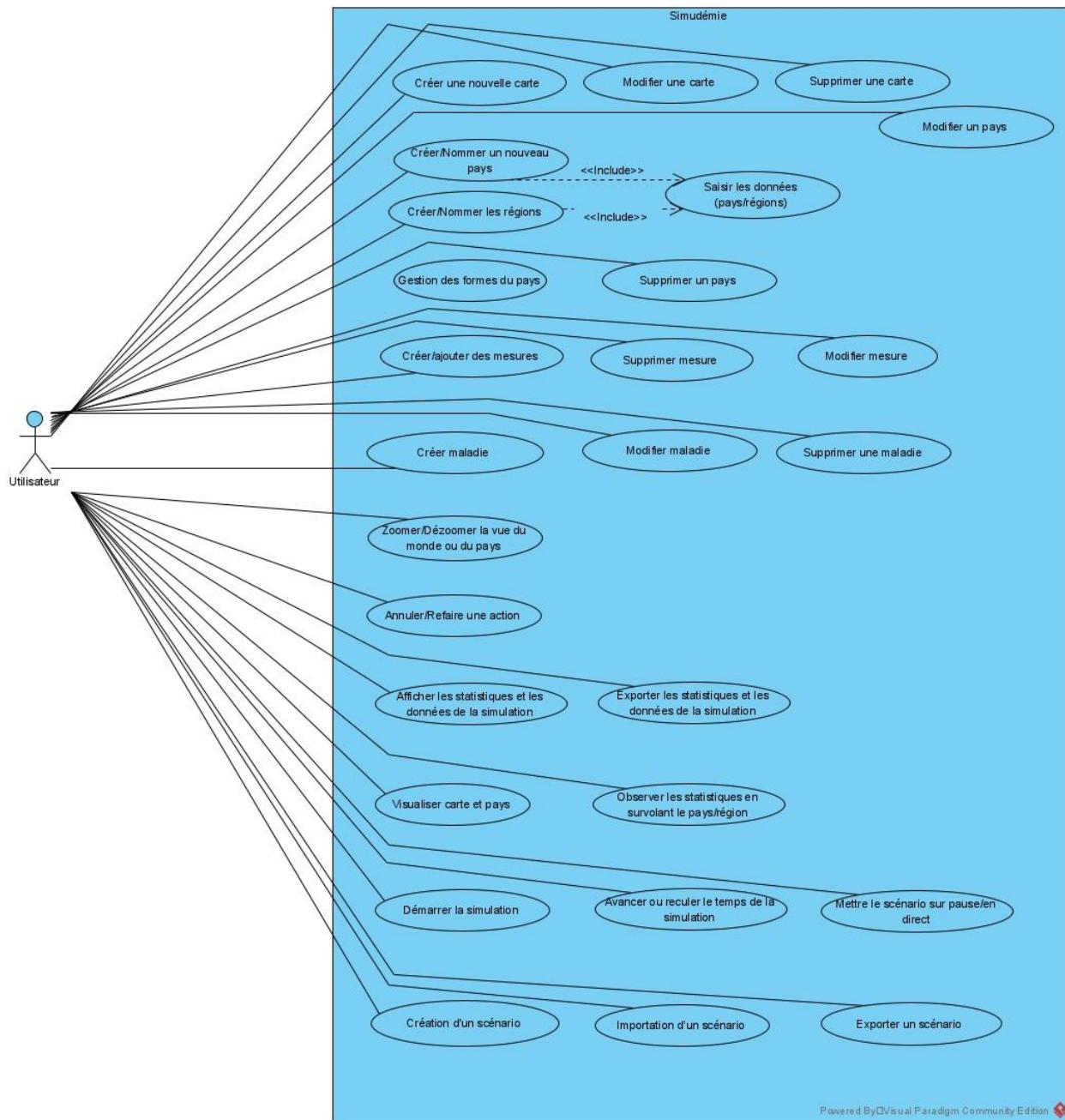


Modèle de domaine

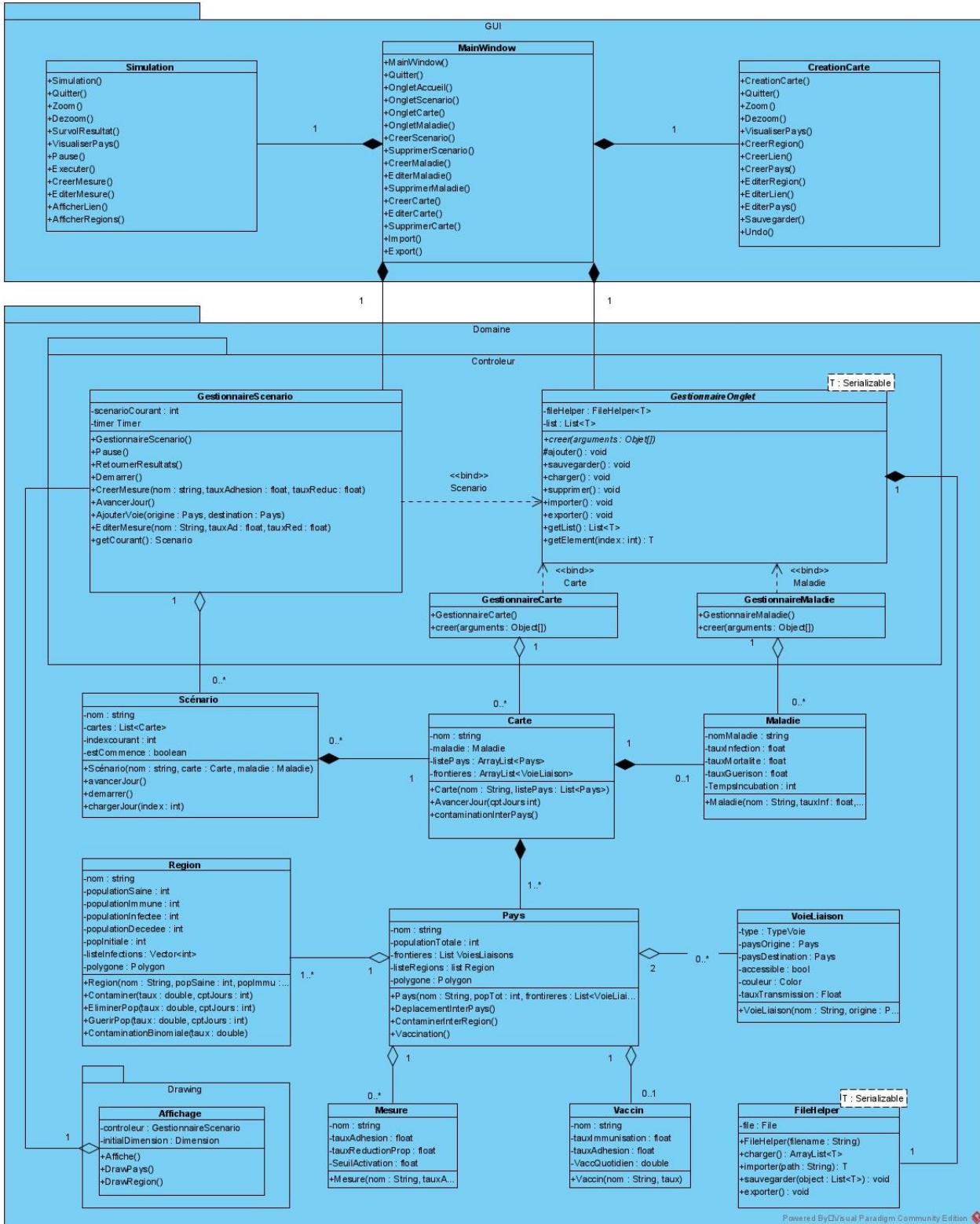


Powered By Visual Paradigm Community Edition

Modèle des cas d'utilisation



Modèle de conception



Bilan

En bref

Nous sommes incroyablement satisfaits du produit que nous sommes en mesure de livrer à notre client. Notre application représente bien tout le temps qui a été investi dans ce projet par notre équipe. À la production du livrable 3, nous avons passé en moyenne 5 heures par jour en équipe. Cela veut dire qu'au total, une vingtaine d'heures a été mise dans le développement du précédent livrable tous les jours, et ce, sur une période de deux semaines.

Il est évident que notre application est plus qu'agréable à l'œil. Nous avons utilisé un thème récurrent pour chaque différente fenêtre. L'attention a été mise sur un grand nombre de détails. Par exemple, nous avons exploré différents types de contours, d'opacité, de transparence, d'icônes, etc. C'est la culmination de chacun de ces choix qui confère à notre application un attrait indéniable. Nous avons aussi tenté de respecter ce qui est considéré comme de bonnes pratiques en termes de conception d'interfaces. Notre vision était d'obtenir une interface simple et conviviale, mais qui offrirait toutes les options demandées et même plus.

Propositions & améliorations

Notre application n'est toutefois pas sans failles. En l'absence de stratégie précise concernant la propagation de la maladie, nous avons tenté de reproduire un modèle de propagation qui se rapproche de la pandémie actuelle. Nous voulions éviter une propagation trop lente ainsi qu'une propagation exponentielle. Nous ne tenons évidemment pas compte des différentes caractéristiques des régions qui pourraient avoir une incidence sur la propagation de la maladie (exemple : concentration d'habitant au mètre carré).

Étant donné les contraintes temporelles, nous sommes heureux de pouvoir remettre un projet qui ne contient que très peu d'erreurs critiques. Ainsi, l'utilisateur ne devrait pas se retrouver avec une application inutilisable. Toutefois, si nous étions moins serrés dans le temps, nous effectuerions davantage de tests pour confirmer l'intégrité du logiciel.

Outre l'amélioration de la logique mathématique représentant la propagation de la maladie, il y a un grand nombre de fonctionnalités supplémentaires que nous souhaitions ajouter. Ces fonctionnalités étaient parfois superficielles, comme l'ajout d'un thème supplémentaire (palette de couleur moins sombre) et parfois plus "concret" comme l'ajout de saisons dans le calcul des infections (propagation plus importante en hiver lorsque la population se retrouve à l'intérieur).

Équipe 9

Contribution - Livrable 1

Répartition des tâches – Livrable 1

Tâches	Fait par	Révisé par
Énoncé de vision		
Contexte + énoncé de vision	Frederick Hughes	Michael Vermette
Modèle du domaine		
Diagramme des classes conceptuelles.	Jonathan Roy-Noel & Henri Bernard	Frederick Hughes & Michael Vermette
Texte explicatif	Jonathan Roy-Noel	Henri Bernard
Modèle des cas d'utilisation		
Diagramme des cas d'utilisation	Michael Vermette	Jonathan Roy-Noël & Henri Bernard & Frederick Hughes
Texte des cas d'utilisation	Frederick Hughes & Michael Vermette	Henri Bernard
Diagramme de séquence système (DSS)	Jonathan Roy-Noel & Henri Bernard	Frederick Hughes & Michael Vermette
Esquisses des interfaces utilisateur		
Menus Scénarios	Jonathan Roy-Noel & Henri Bernard	Frederick Hughes & Michael Vermette
Menus d'ajouts	Jonathan Roy-Noel & Henri Bernard	Frederick Hughes & Michael Vermette
Simulations	Jonathan Roy-Noel & Henri Bernard	Frederick Hughes & Michael Vermette

Diagramme de Gantt + Conclusion		
Diagramme de Gantt	Frederick Hughes	Henri Bernard
Conclusion	Frederick Hughes	Michael Vermette
Section contribution des membres de l'équipe + Compte rendu des réunions d'équipe		
Contribution	Henri Bernard	Jonathan Roy-Noel & Frederick Hughes & Michael Vermette
Comptes rendus	Henri Bernard	Jonathan Roy-Noel & Frederick Hughes & Michael Vermette
Mise en page & Correction + Remise		
Mise en page & Correction	Henri Bernard	Jonathan Roy-Noel & Frederick Hughes & Michael Vermette
Remise	Henri Bernard	Jonathan Roy-Noel & Frederick Hughes & Michael Vermette

Comptes rendus – Livrable 1

Compte rendu no.1

Réunion no. 1	
Titre	Introduction & présentations
Lieu	Discord — Serveur privé pour l'équipe
Date & Heure	Jeudi 21 janvier 2021 – 16 h 30
Participants	Henri Bernard, Jonathan Roy-Noel, Frederick Hughes & Michael Vermette
Contenu	<ul style="list-style-type: none">• Présentation des membres de l'équipe• Mise en commun des horaires et des disponibilités• Mise en place d'un objectif commun, d'avoir lu complètement l'énoncé du projet et celui du livrable 1 d'ici le 28 janvier.

Compte rendu no.2

Réunion no. 2	
Titre	Fusion de vision & Élaboration de l'échéancier
Lieu	Discord — Serveur privé pour l'équipe
Date & Heure	Jeudi 28 janvier 2021 – 16 h 30
Participants	Henri Bernard, Jonathan Roy-Noel, Frederick Hughes & Michael Vermette
Contenu	<ul style="list-style-type: none">• Conversation & «brainstorm» du projet de session• Premier brouillon du diagramme de classes conceptuelles• Initialiser les outils de travail commun (Discord, Teams/OneDrive & GitLab)• Élaboration de l'échéancier• Séparation de tâches

Compte rendu no.3

Réunion no. 3	
Titre	Solidification de la vision
Lieu	Discord — Serveur privé pour l'équipe
Date & Heure	Lundi 1er février 2021 – 16 h 30
Participants	Henri Bernard, Jonathan Roy-Noel, Frederick Hughes & Michael Vermette
Contenu	<ul style="list-style-type: none"> • Finaliser le modèle de domaine & les diagrammes de cas d'utilisation en équipe • Mise en accord de la vision et de l'ampleur du projet • Répondre en équipe aux questions de chacun des membres de l'équipe • Mise à jour de l'échéancier & distribution des tâches restantes

Compte rendu no.4

Réunion no. 4	
Titre	Uniformisation & finalisation
Lieu	Discord — Serveur privé pour l'équipe
Date & Heure	Jeudi 4 février 2021 – 16 h 30
Participants	Henri Bernard, Jonathan Roy-Noel, Frederick Hughes & Michael Vermette
Contenu	<ul style="list-style-type: none"> • Repasser en équipe sur chacune des parties du travail afin d'uniformiser les termes et la vision du projet • Répondre en équipe aux questions de chacun des membres de l'équipe • Finalisation des parties incomplètes (description des cas d'utilisation & diagramme de Gantt)

Compte rendu no.5

Réunion no. 5	
Titre	Dernières vérifications & petites retouches
Lieu	Discord — Serveur privé pour l'équipe
Date & Heure	Lundi 8 février 2021 – 11 h
Participants	Henri Bernard, Jonathan Roy-Noel, Frederick Hughes & Michael Vermette
Contenu	<ul style="list-style-type: none">• Visualisation du document final en équipe• Petites retouches• Remise

Suivi du développement – Livrable 1

Il est de notre avis que le projet est bien entamé. En effet, la production d'autant d'esquisses nous a permis d'effectuer un travail d'analyse beaucoup plus avancé et pointilleux concernant la vision et la faisabilité du projet. Nous avons songé à de multiples options supplémentaires pour compléter et embellir l'application et ainsi rendre le produit visuellement et techniquement plaisant. Nous avons débuté le squelette de notre application pour nous assurer d'être dans les délais et nous avons déterminé nos normes de programmation.

Nous sommes impatients d'avoir le retour du client pour ainsi valider que nous sommes bien en accord sur la vision du projet et s'il y a des modifications qu'ils souhaitent apporter.

Contribution – Livrable 2

Répartition des tâches – Livrable 2

Tâches	Fait par	Révisé par
Le diagramme de classe de conception		
Diagramme + Texte	Frederick	Michael & Henri & Jonathan
L'architecture logique		
Diagramme + Texte	Frederick	Michael & Henri & Jonathan
Diagrammes de séquence de conception (DSC)		
3.1 + Texte	Jonathan & Henri	Michael & Frederick
3.2 + Texte	Jonathan	Henri
3.3 + Texte	Jonathan & Henri	Michael & Frederick
3.4 .1 + Texte	Jonathan & Henri	Michael & Frederick
3.4.2 + Texte	Jonathan & Henri	Michael & Frederick
Le pseudocode d'un algorithme		
Pseudocode + Texte	Frederick	Michael & Jonathan & Henri
Plan de travail		
Diagramme de Gantt + Texte	Frederick	Michael

Finalisation		
Contribution & Comptes rendus	Henri	Frederick & Michael & Jonathan
Mise en page & Correction + Remise	Henri	Frederick & Michael & Jonathan
Version fonctionnelle		
Menu Principal	Jonathan & Henri	Michael
Création Carte	Henri	Jonathan
Simulation	Jonathan	Henri

Comptes rendus – Livrable 2

Compte rendu no.6

Réunion no. 6	
Titre	Mise en point du livrable 2
Lieu	Discord — Serveur privé pour l'équipe
Date & Heure	Lundi 15 février 2021 – 16 h 30
Participants	Henri Bernard, Jonathan Roy-Noel, Frederick Hughes & Michael Vermette
Contenu	<ul style="list-style-type: none">• Lecture et compréhension en groupe du livrable 2• Séparation des tâches• Avancement de la version fonctionnelle du projet

Compte rendu no.7

Réunion no. 7	
Titre	Remue-méninges d'un pseudocode et d'une architecture logique
Lieu	Discord — Serveur privé pour l'équipe
Date & Heure	Jeudi 18 février 2021 – 16 h 30
Participants	Henri Bernard, Jonathan Roy-Noel, Frederick Hughes & Michael Vermette
Contenu	<ul style="list-style-type: none">• Élaboration d'un pseudocode d'un algorithme• Élaboration de l'architecture logique du projet• Avancement de la version fonctionnelle du projet

Compte rendu no.8

Réunion no. 8	
Titre	Avancement du squelette
Lieu	Discord — Serveur privé pour l'équipe
Date & Heure	Lundi 22 février 2021 – 16 h 30
Participants	Henri Bernard, Jonathan Roy-Noel, Frederick Hughes & Michael Vermette
Contenu	<ul style="list-style-type: none">• Première implémentation des classes du domaine• Avancement de la version fonctionnelle du projet• Objectif d'avoir terminé tous les points du livrable 2 d'ici la prochaine réunion d'équipe, soit le 25 février 2021 à 16 h 30

Compte rendu no.9

Réunion no. 9	
Titre	Uniformisation & finalisation
Lieu	Discord — Serveur privé pour l'équipe
Date & Heure	Jeudi 25 février 2021 – 16 h 30
Participants	Henri Bernard, Jonathan Roy-Noel, Frederick Hughes & Michael Vermette
Contenu	<ul style="list-style-type: none">• Uniformisation des points du livrable 2• Finalisation des points• Avancement de la version fonctionnelle du projet

Compte rendu no.10

Réunion no. 10	
Titre	Dernières vérifications & petites retouches
Lieu	Discord — Serveur privé pour l'équipe
Date & Heure	Dimanche 28 février 2021 – 16 h 30
Participants	Henri Bernard, Jonathan Roy-Noel, Frederick Hughes & Michael Vermette
Contenu	<ul style="list-style-type: none">• Correction• Dernières vérifications• Dernières retouches• Préparation de la remise

Suivi du développement – Livrable 2

Malgré le léger retard précédemment mentionné, nous sommes très satisfaits du produit que nous pouvons présenter au client. En effet, notre squelette d'interface est sublime et nous avons commencé à dessiner les formes rectangulaires. Pour chaque élément du projet qui n'a pas encore été fait, nous avons des esquisses, des modèles ou des algorithmes auxquels nous avons songé. Si un quelconque problème venait à survenir, nous avons déjà pensé à de multiples solutions que nous avons classées en ordre de priorité et complexité. Ainsi, si le temps venait à manquer, nous sommes certains de pouvoir apporter des modifications dans le code qui vont simplifier nos méthodes et faciliter la production d'une itération du projet.

Il nous reste quelques questions à poser au client concernant l'utilisation de librairies externes pour l'embellissement de l'interface et certaines questions concernant le cœur de l'application. Par exemple, nous souhaitons valider l'impact qu'aura chaque voie de liaison ainsi que la méthodologie de séparation du pays. Si tout se déroule comme prévu, ces questions seront répondues le jeudi 25 février 2021 ou le jeudi suivant.

Contribution - Livrable 3

Répartition des tâches – Livrable 3

Tâches	Fait par	Révisé par
Code opérationnel et démo		
Interface utilisateur	Jonathan Roy-Noël	Henri Bernard
Domaine	Frederick Hughes	Michael Vermette & Henri Bernard
Contrôleurs & sérialisation	Michael Vermette & Henri Bernard	Jonathan Roy-Noël
Rapport		
Modèle du domaine mis à jour	Frederick Hughes	Michael Vermette & Henri Bernard
Modèle des cas d'utilisation mis à jour	Frederick Hughes	Jonathan Roy-Noël
Modèle de conception mis à jour	Frederick Hughes	Michael Vermette & Henri Bernard
Contribution, correction & remise	Henri Bernard	Michael Vermette & Frederick Hughes & Jonathan Roy-Noël

Suivi du développement – Livrable 3

Nous avons apporté des modifications à nos diagrammes des cas d'utilisation ainsi qu'à notre diagramme de classes conceptuelles. Nous avons conclu, suivant les correctifs demandés par le client et les différents diagrammes qu'il nous a remis, que nous avons tenté d'en faire trop. En effet, en voulant introduire des éléments/actions supplémentaires à l'application, nous avons complexifié notre tâche, tant au niveau des diagrammes qu'au niveau du code. Sans délaisser certaines de nos idées de départ, nous sommes arrivés à la conclusion qu'il serait mieux de faire le « minimum » recommandé pour la construction de nos diagrammes ainsi que pour notre première ébauche de code.

Contribution – Livrable 4

Répartition des tâches – Livrable 4

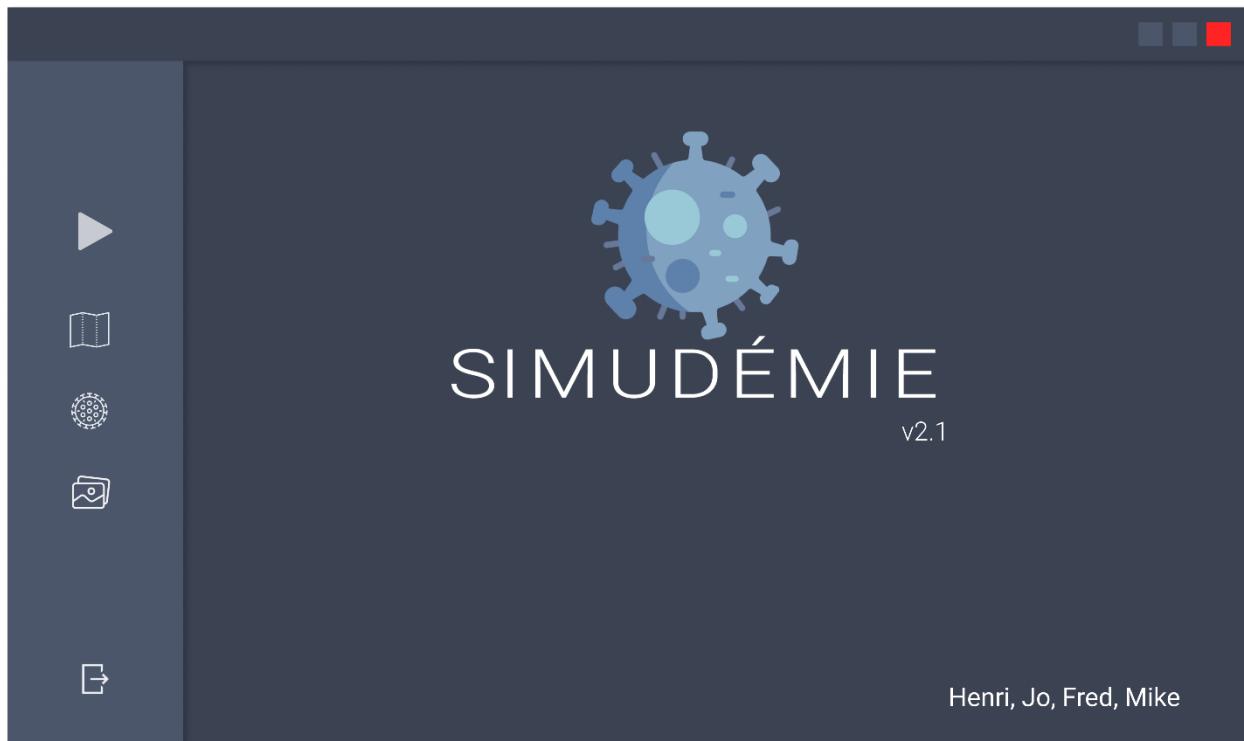
Tâches	Fait par	Révisé par
Undo/Redo	Jonathan Roy-Noël	Henri Bernard, Frederick Hughes
Drag/Drop pays & fusion d'un pays à un autre	Jonathan Roy-Noël	Henri Bernard, Frederick Hughes & Michael Vermette
Jeux de données (Mesures, Voies de transmission & Vaccin)	Henri Bernard, Frederick Hughes & Michael Vermette	Jonathan Roy-Noël
Export & affichage des données (Carte & Statistiques)	Michael Vermette & Jonathan Roy-Noël	Henri Bernard & Frederick Hughes
Fix bugs & finitions & UI	Henri Bernard & Michael Vermette	Frederick Hughes & Jonathan Roy-Noël
Rapport		
Mise à jour des artefacts	Henri & Frederick	Michael Vermette
Contribution, mise en page, correction & remise	Henri Bernard	Michael Vermette & Frederick Hughes & Jonathan Roy-Noël

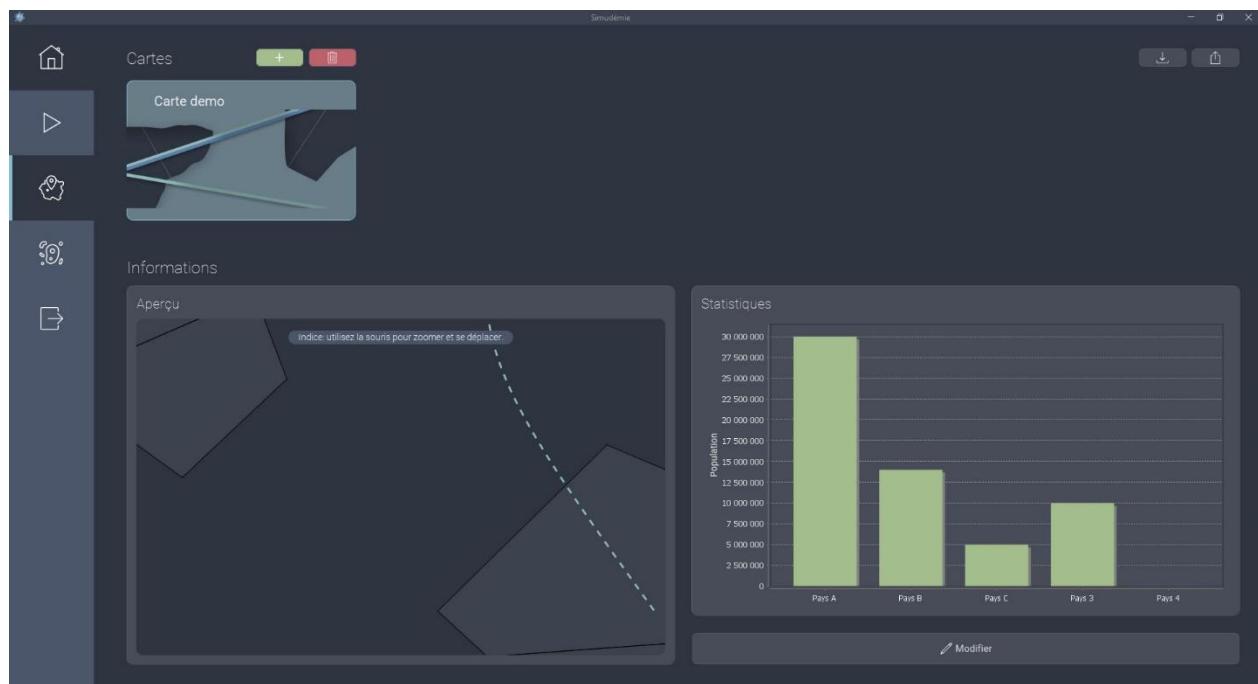
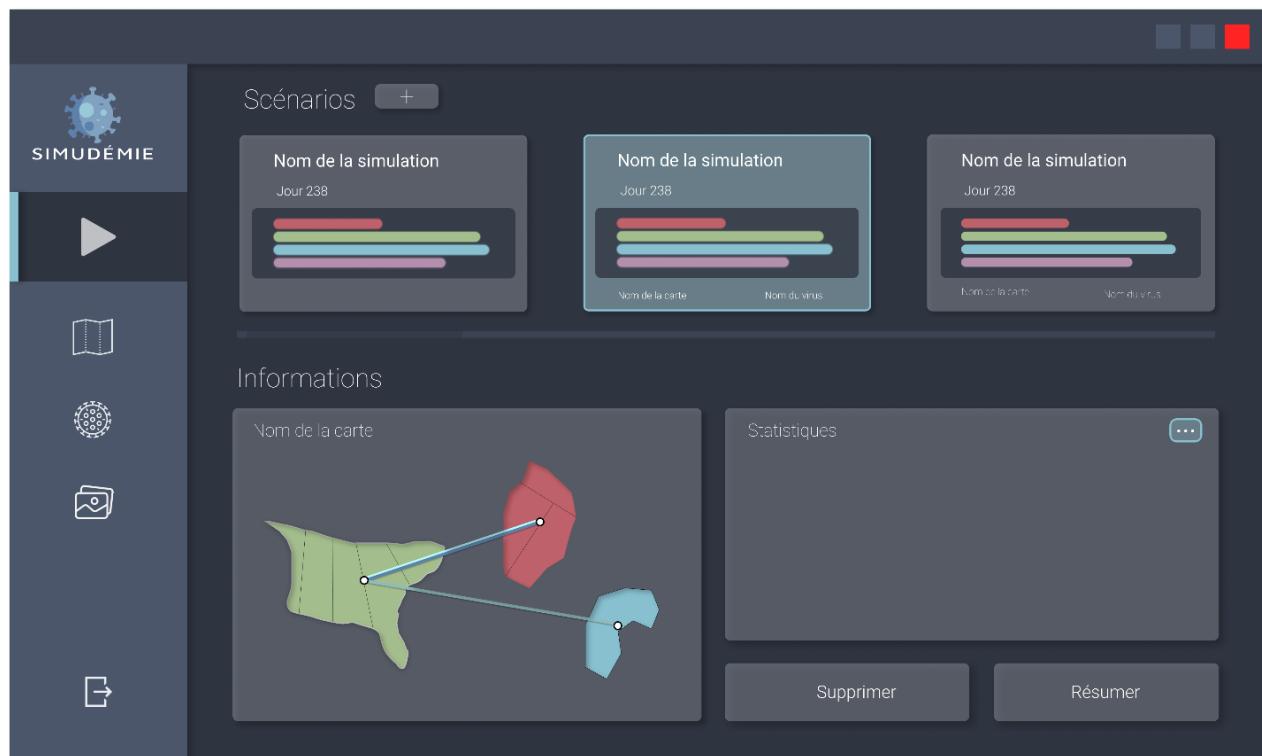
Suivi du développement – Livrable 4

Nous avons apporté des modifications à nos diagrammes du modèle de domaine, des cas d'utilisations et de conception afin de les mettre à jour par rapport à notre version final de l'application. L'ajout de certains attributs dans les classes comme vaccin, maladie, mesure. Nous n'avions pas énormément de changement à faire au niveau de nos artefacts alors nous avons décidé de nous concentrer sur la version finale de l'application.

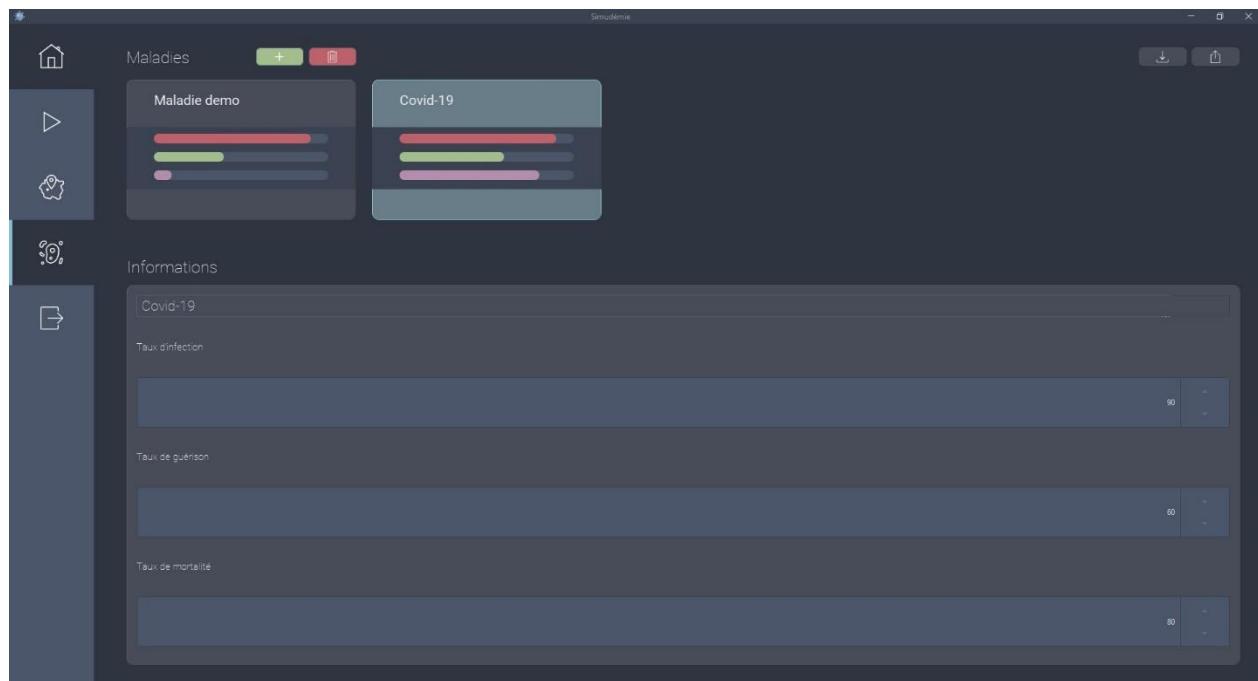
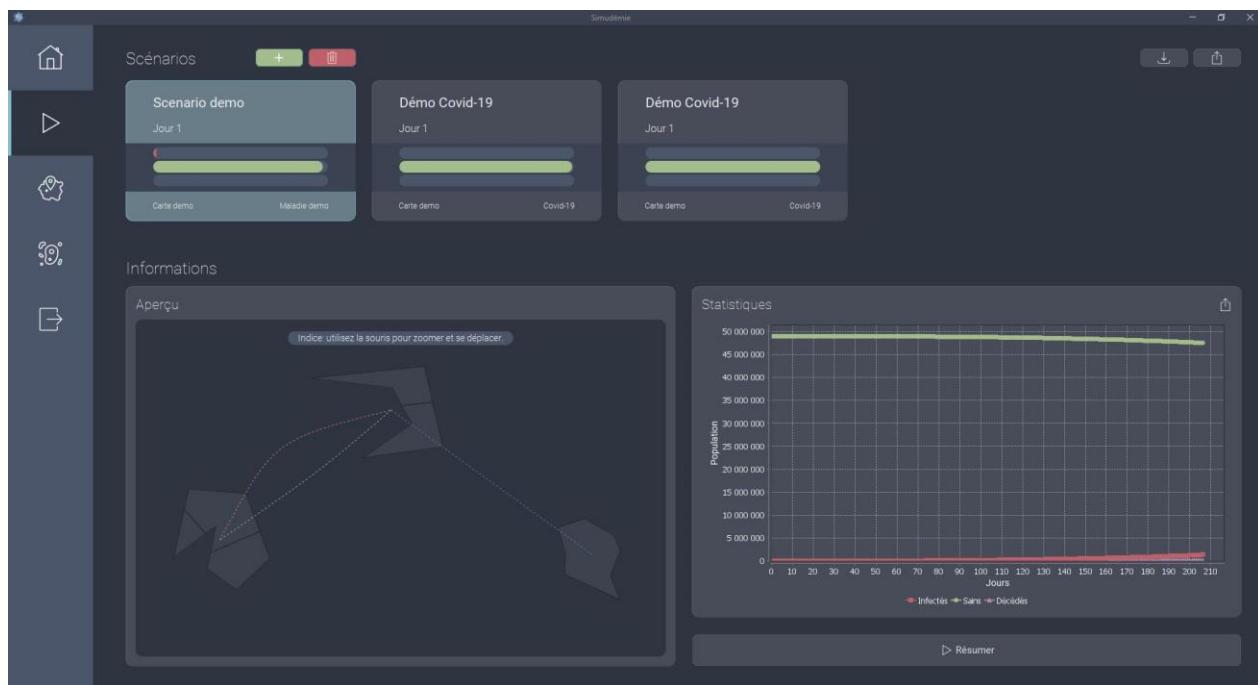
Annexes

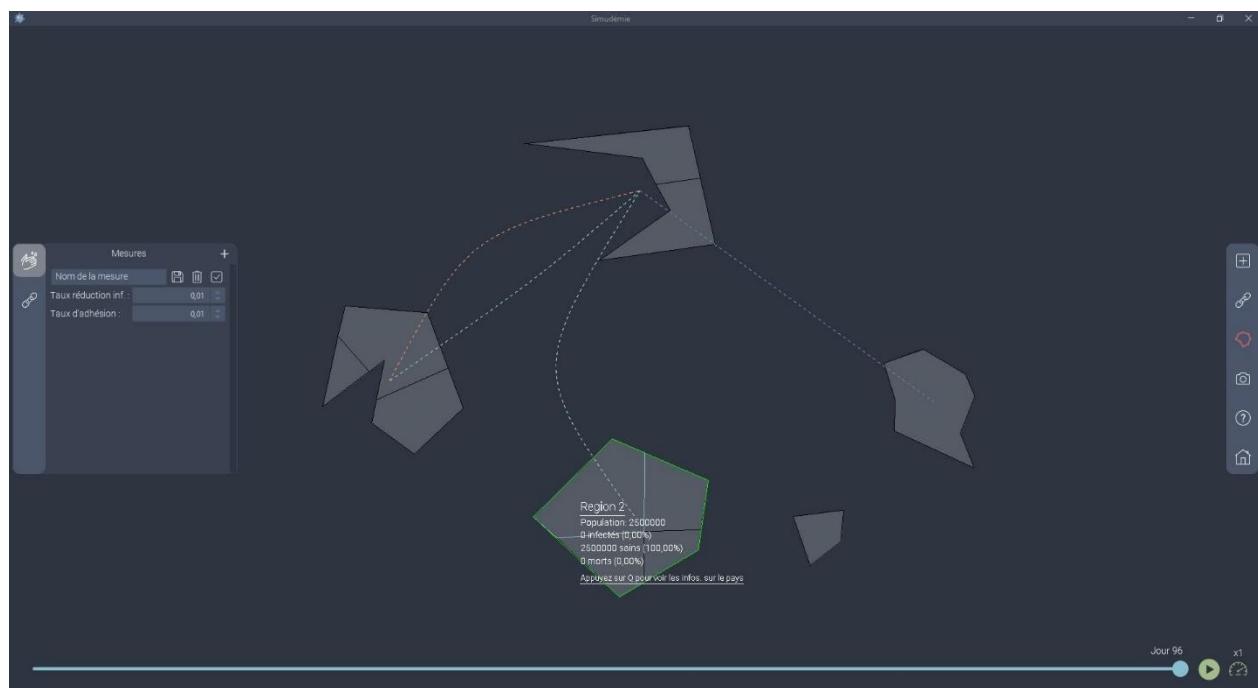
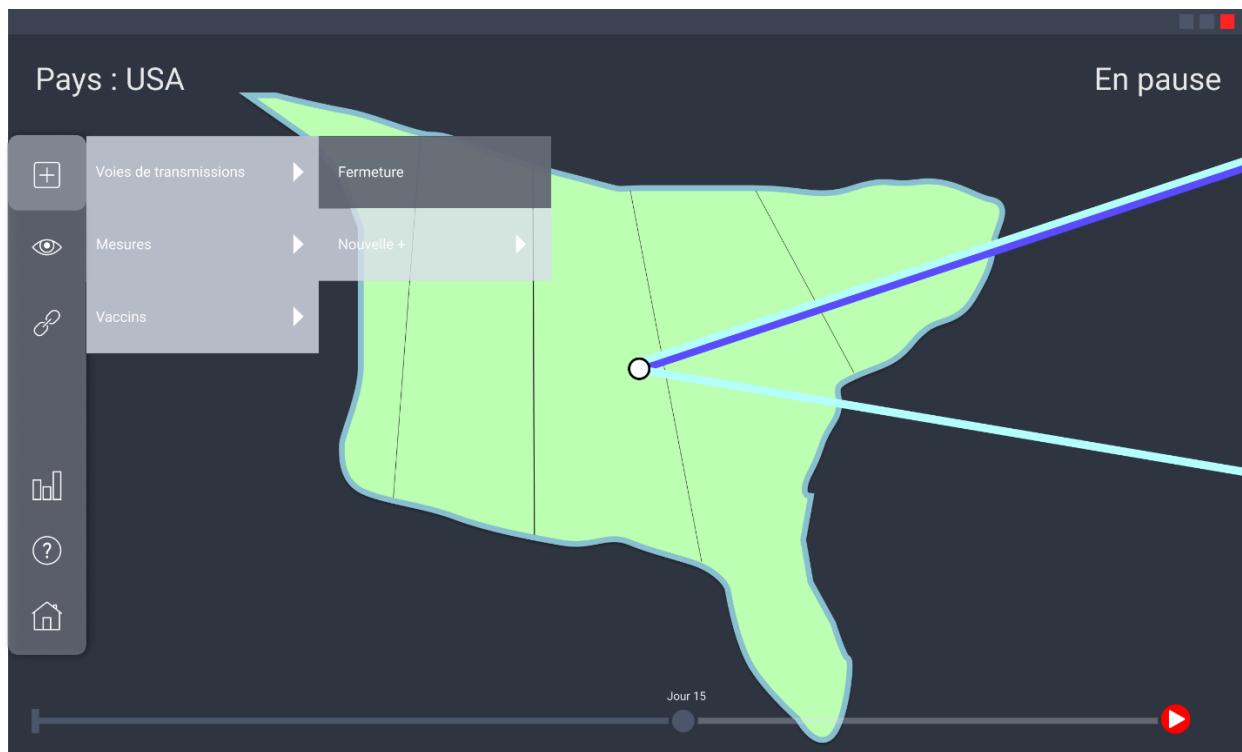
Esquisses
Interfaces – v3.1

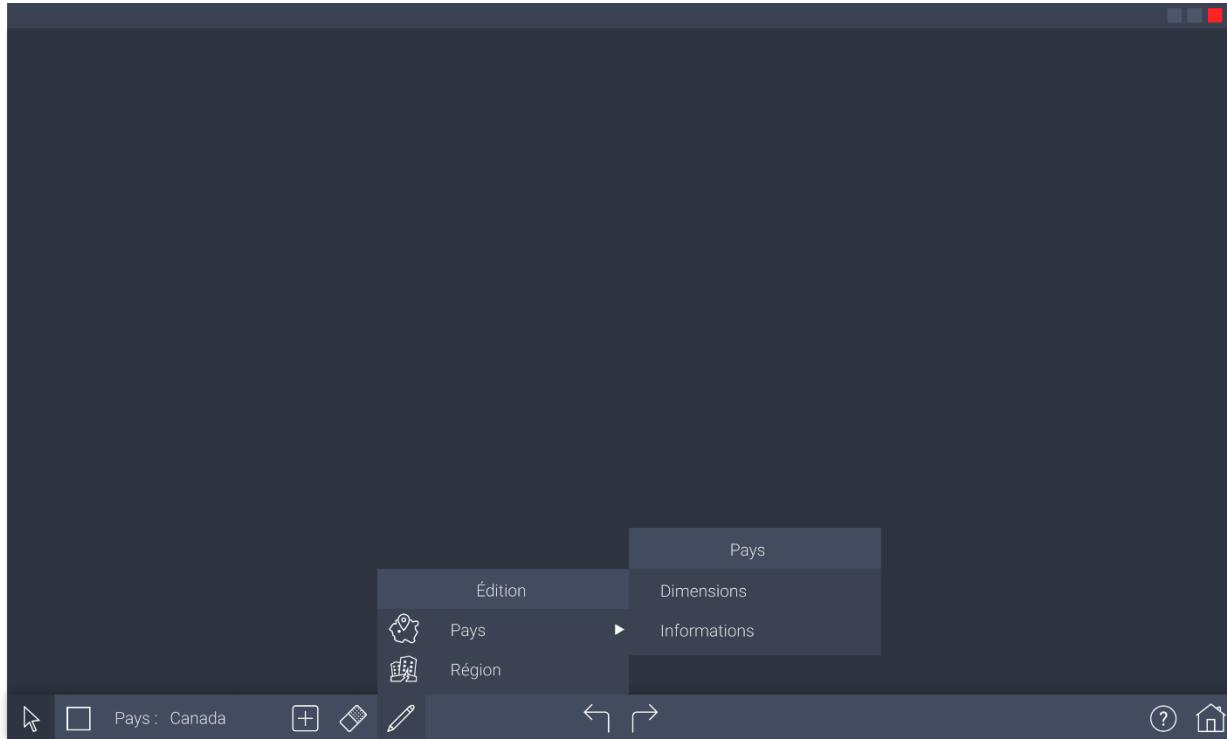
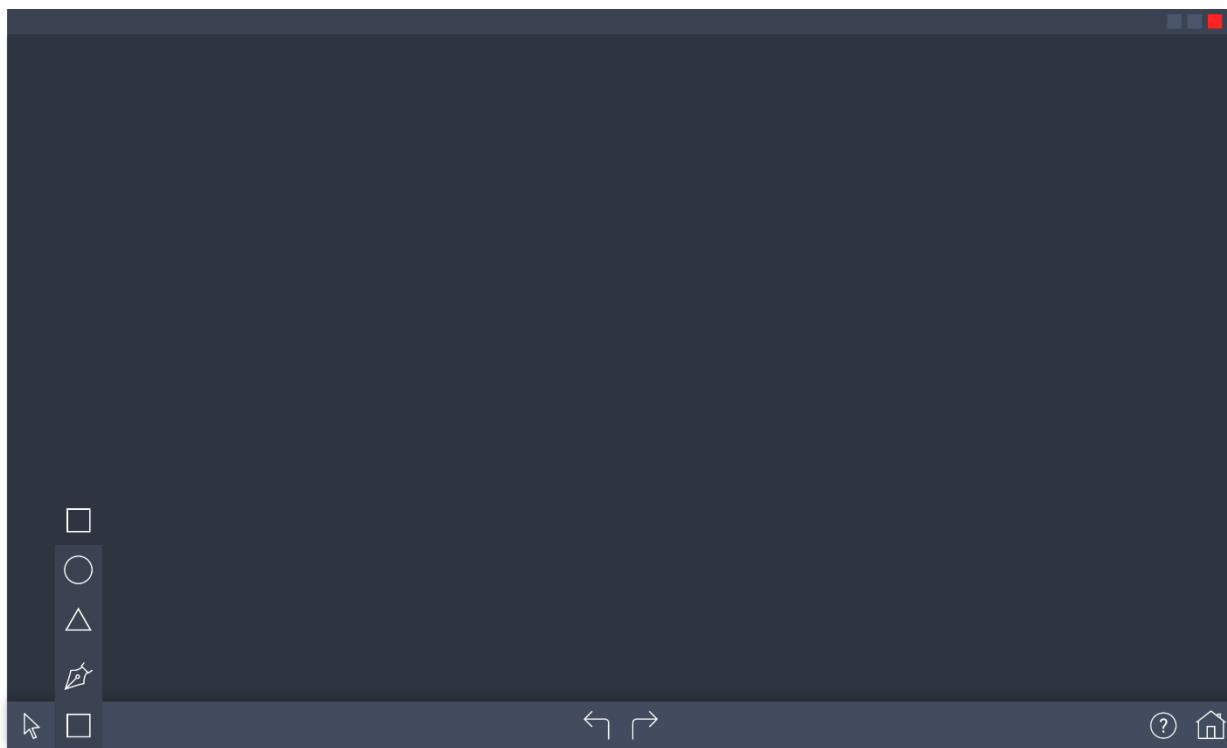


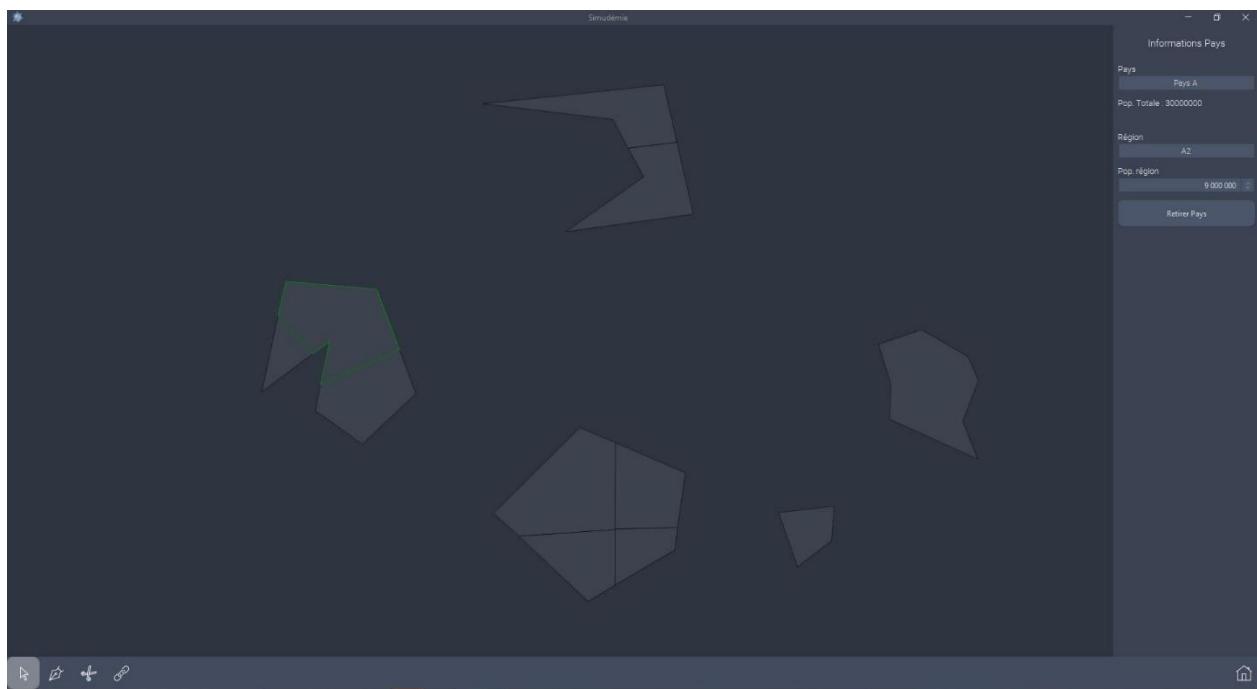


Simudémie — Livrable 4

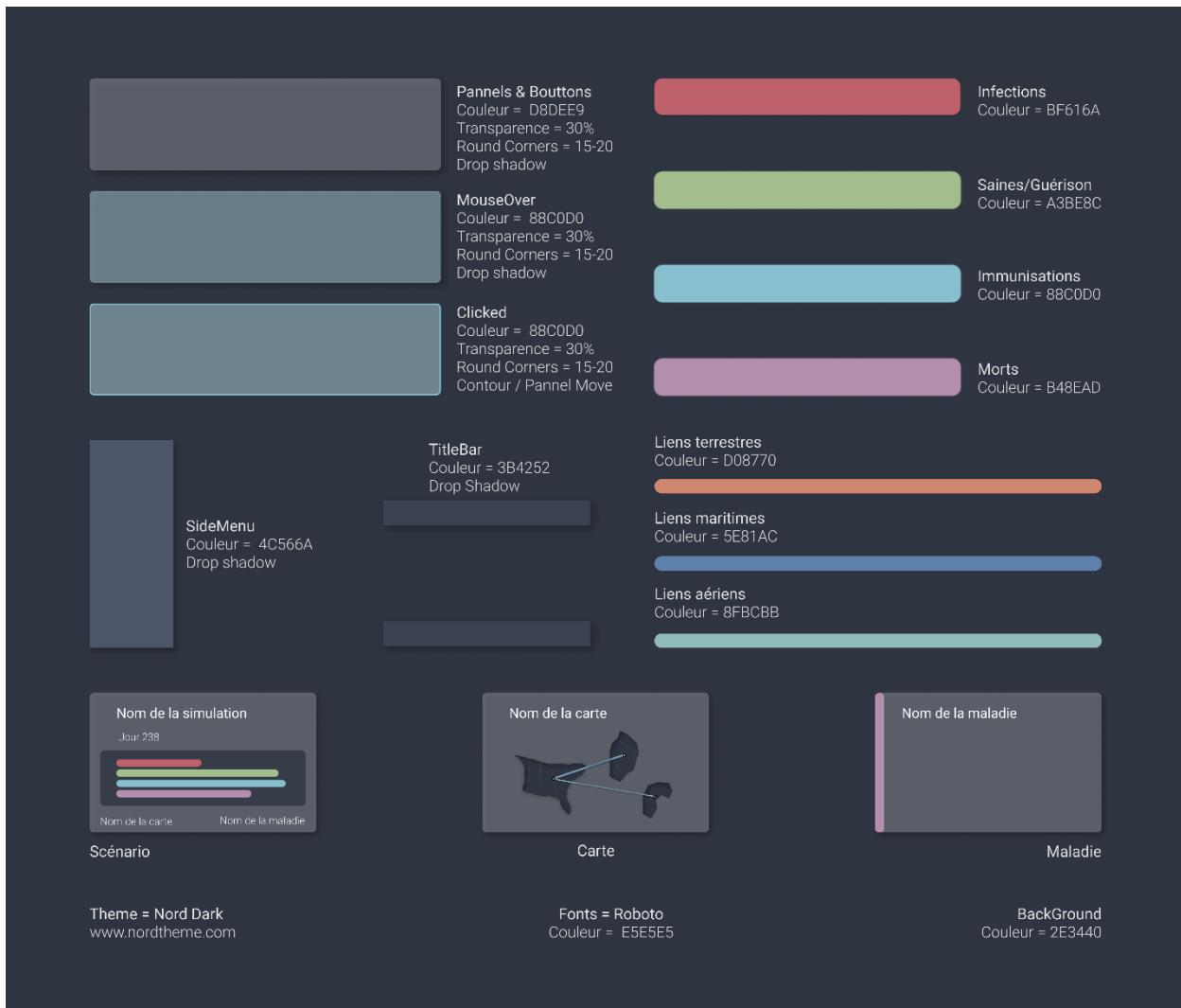


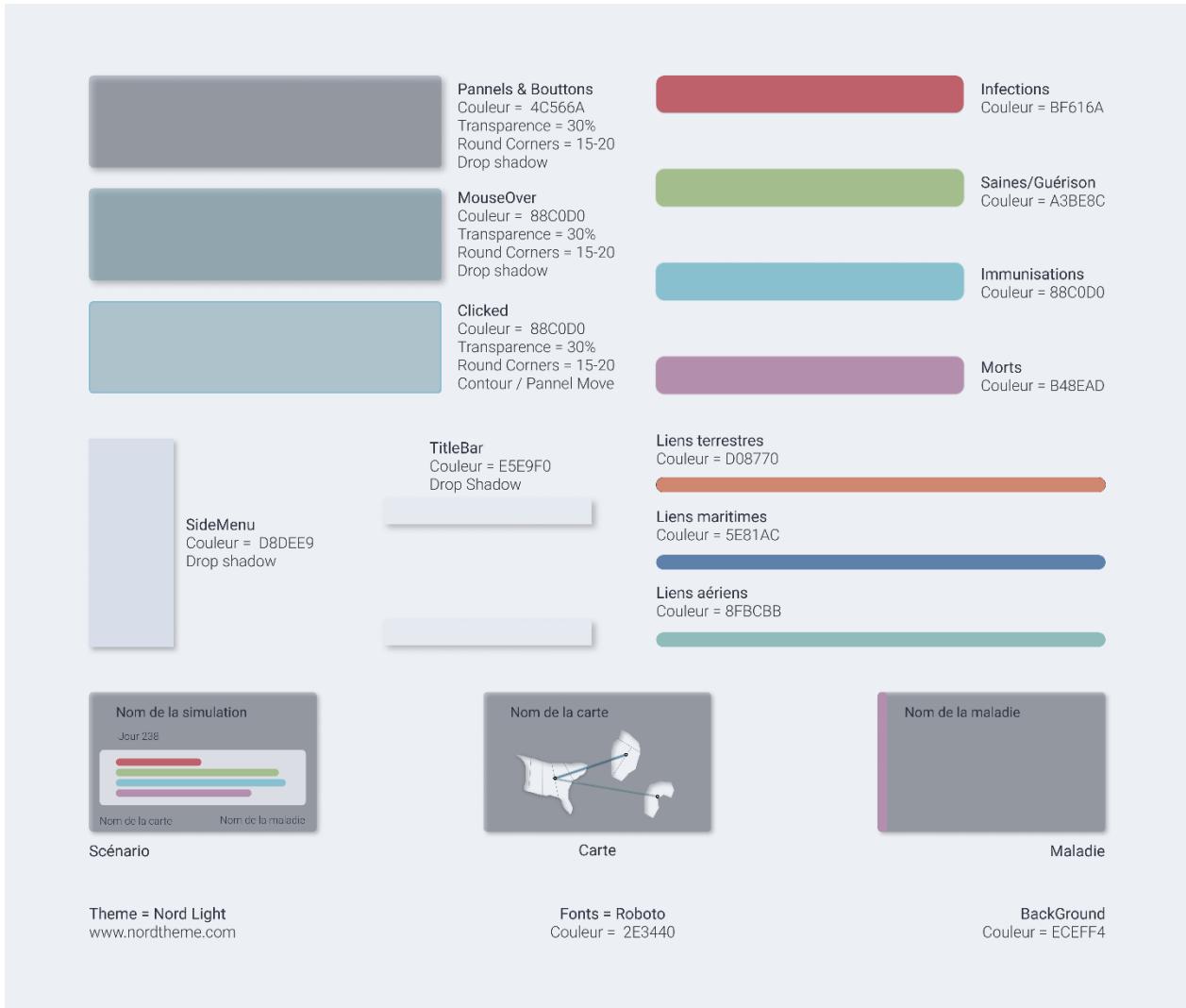




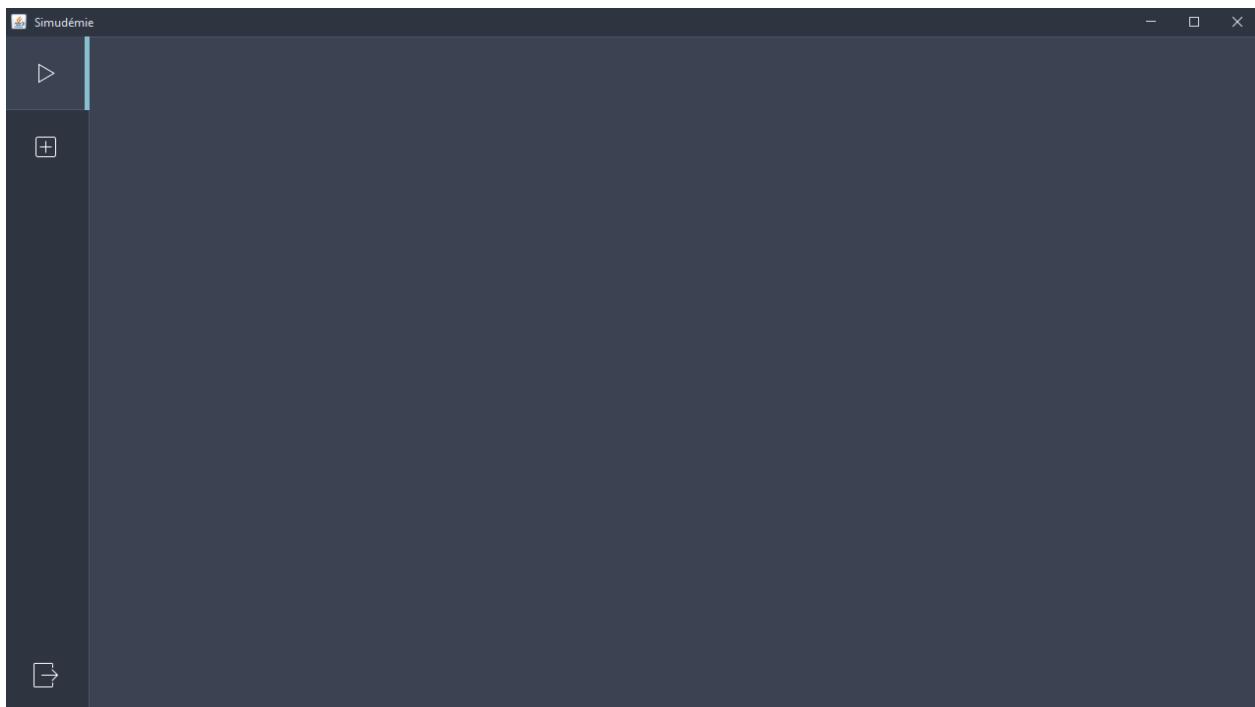


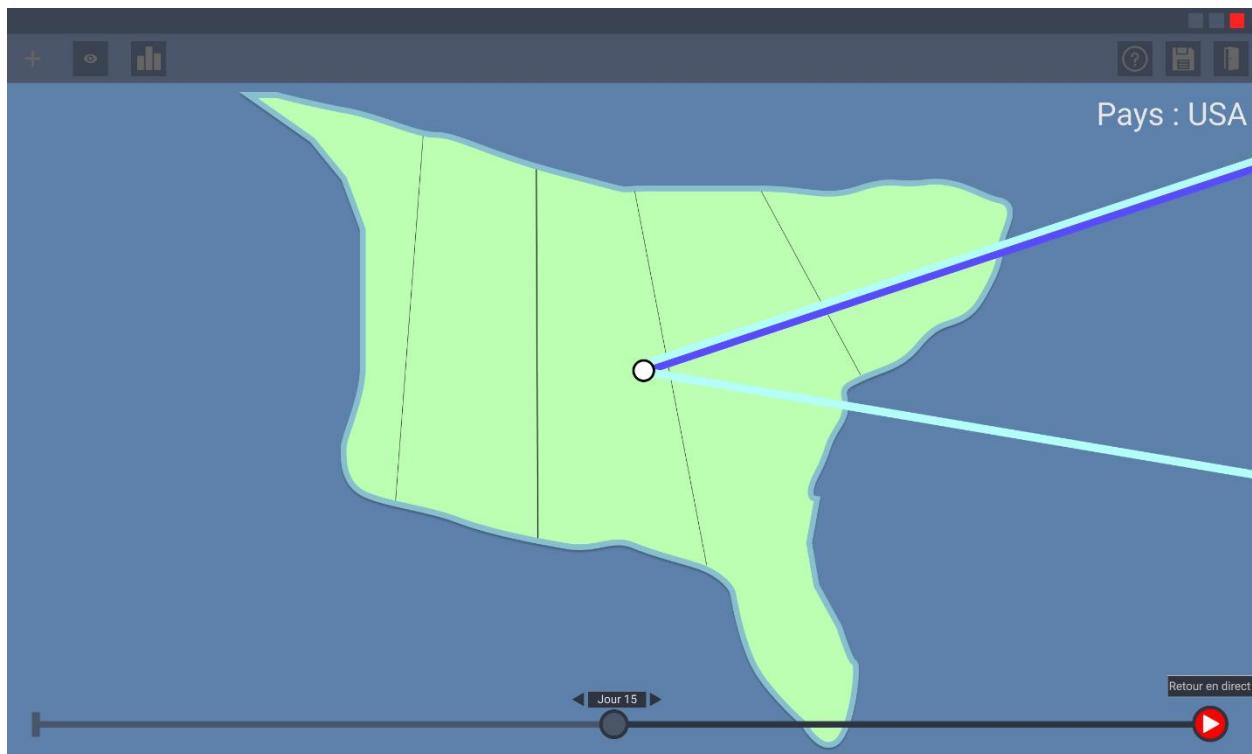
Thème et design – v3.1



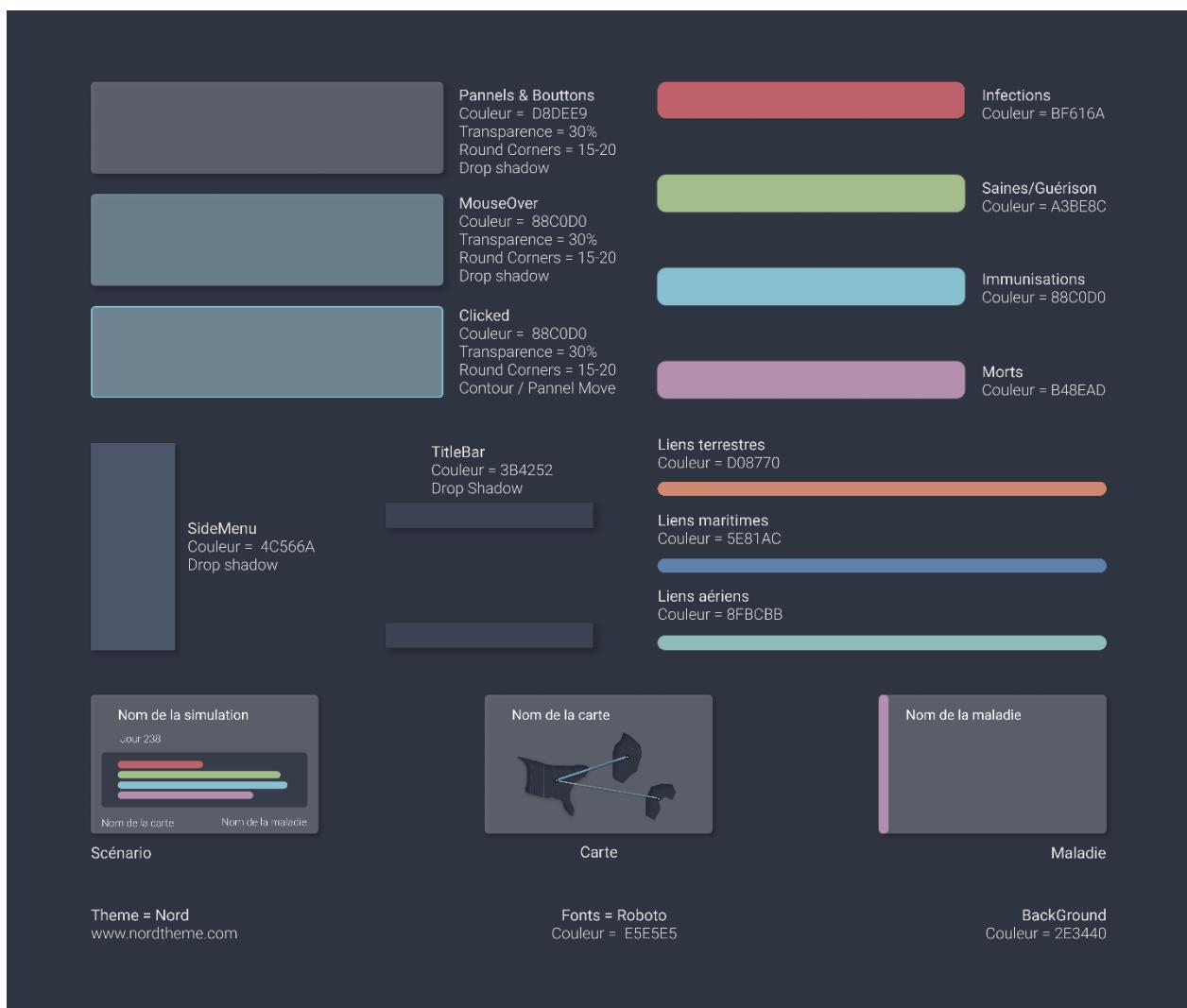
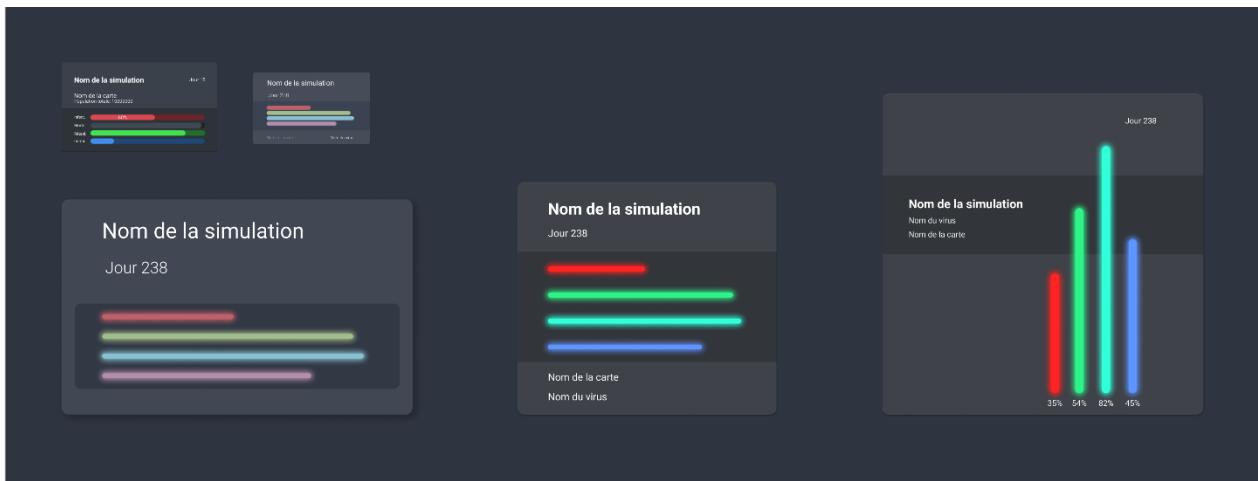


Interfaces – v2.0

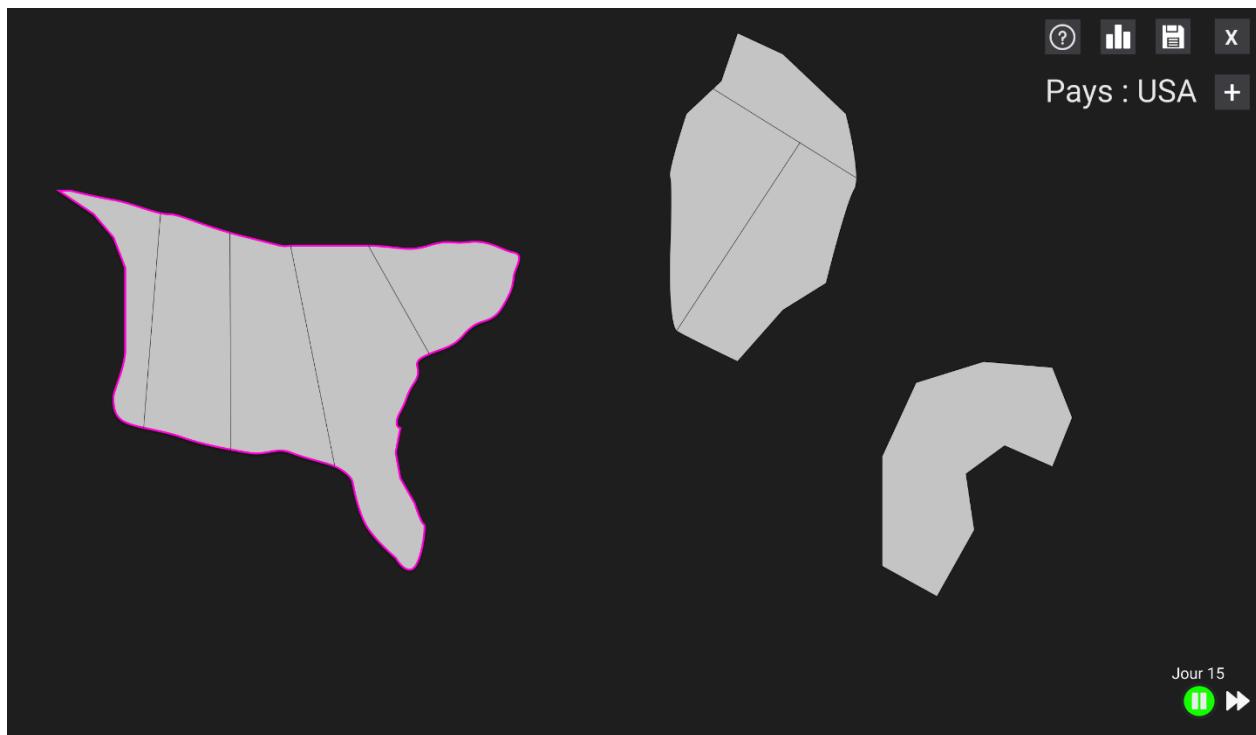




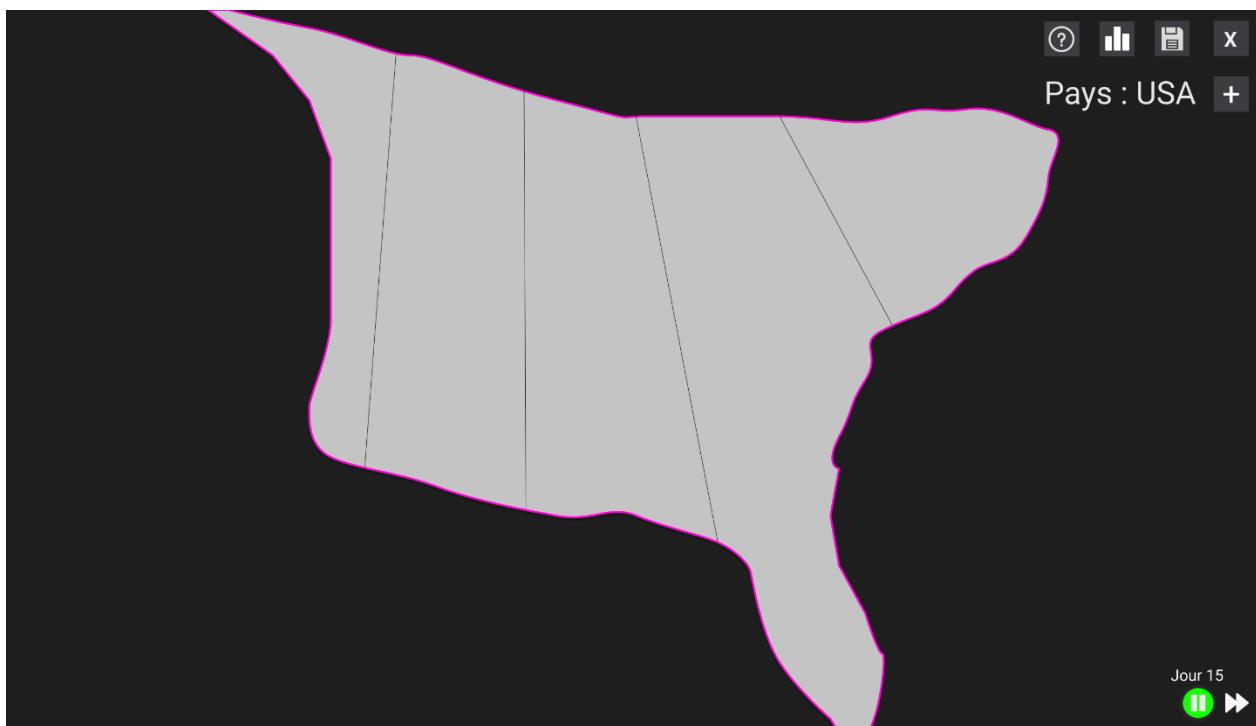
Thèmes & design – v2.0



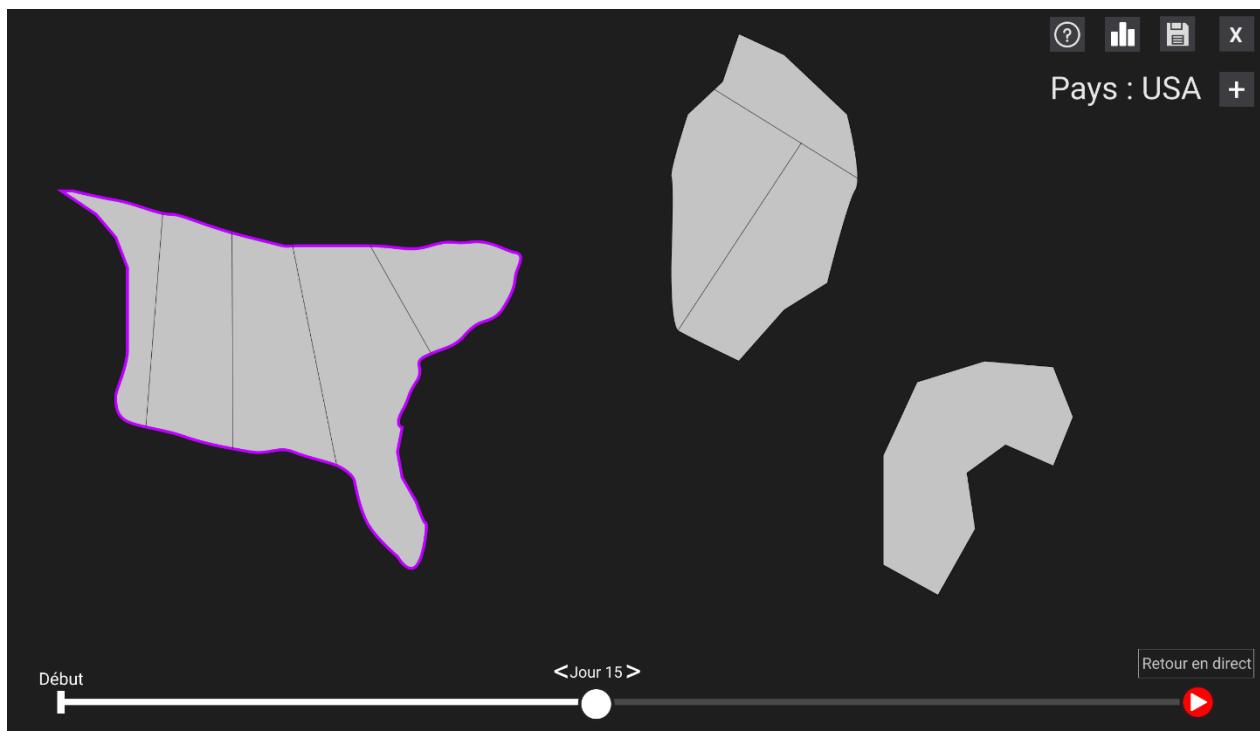
Interfaces - v1.0



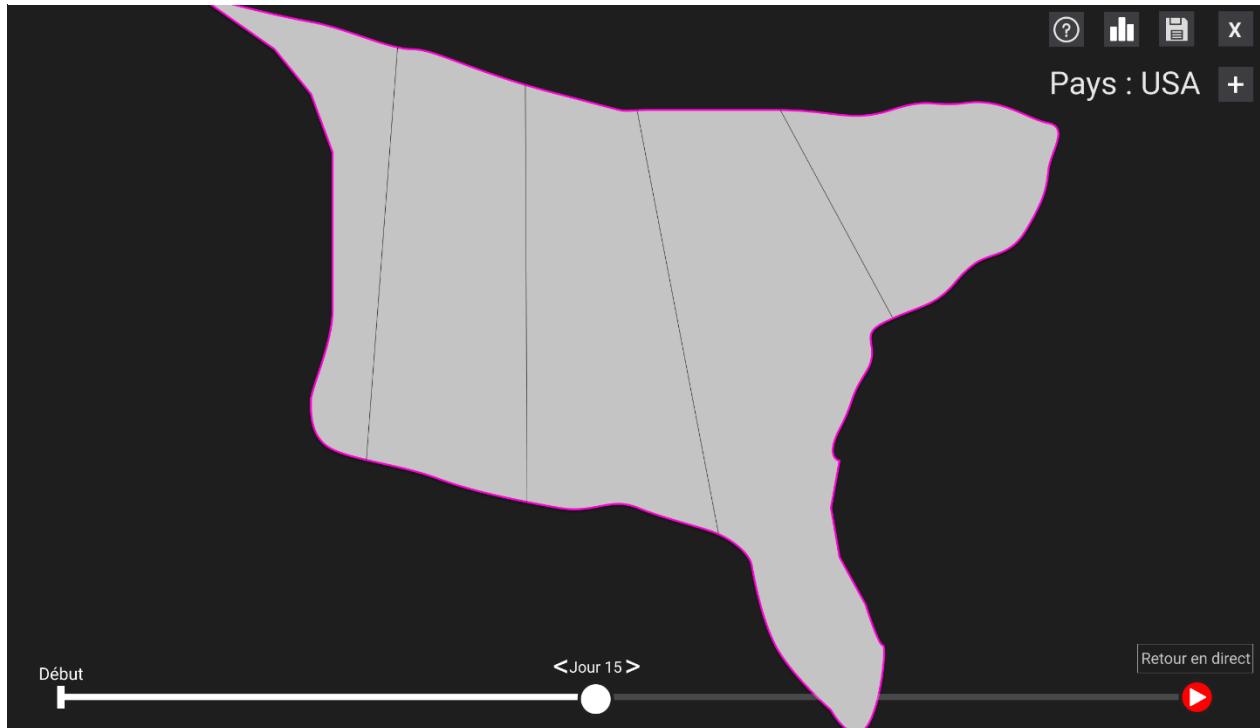
Esquisse 1.1 — Simulation — En Direct



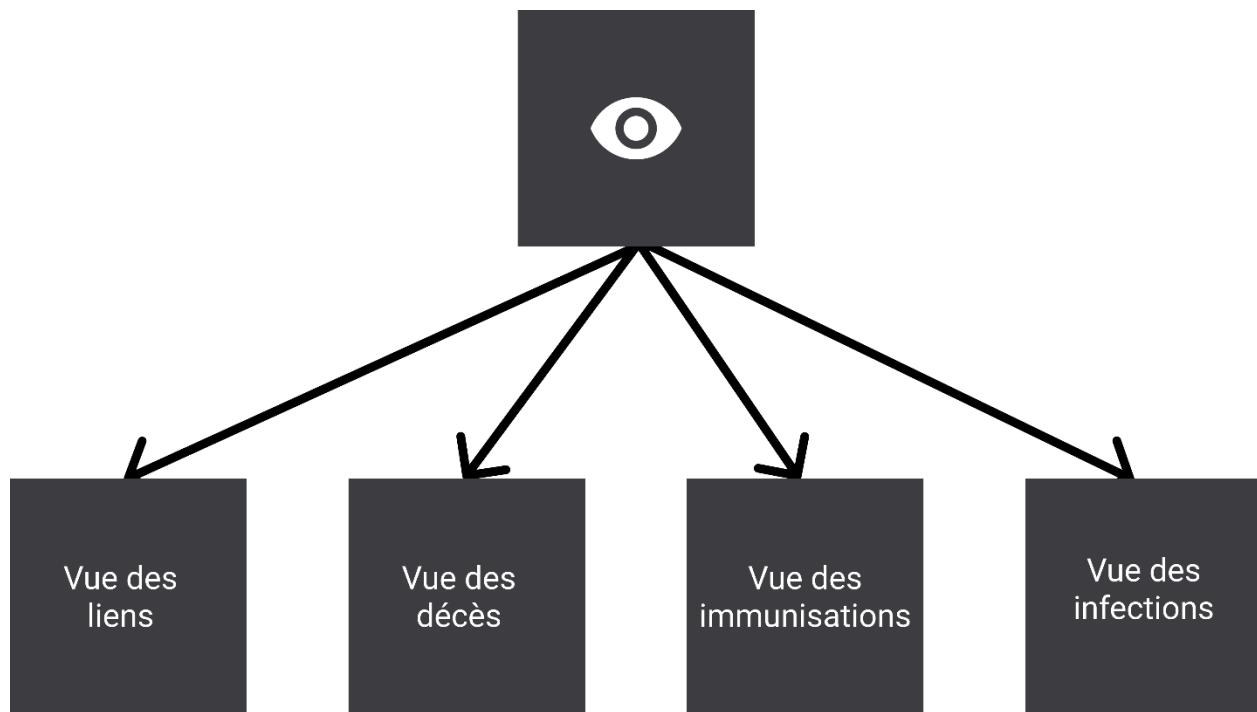
Esquisse 1.2 — Simulation — En Direct + Zoom



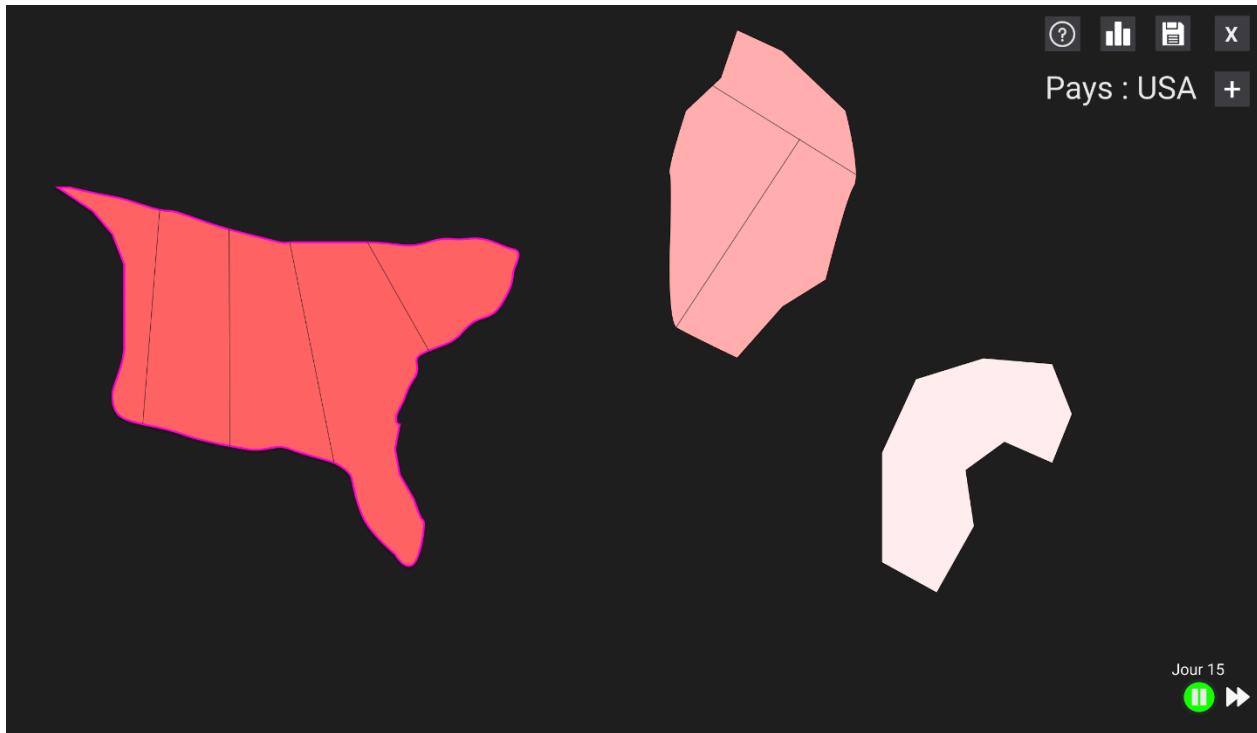
Esquisse 1.3 — Simulation — En Pause



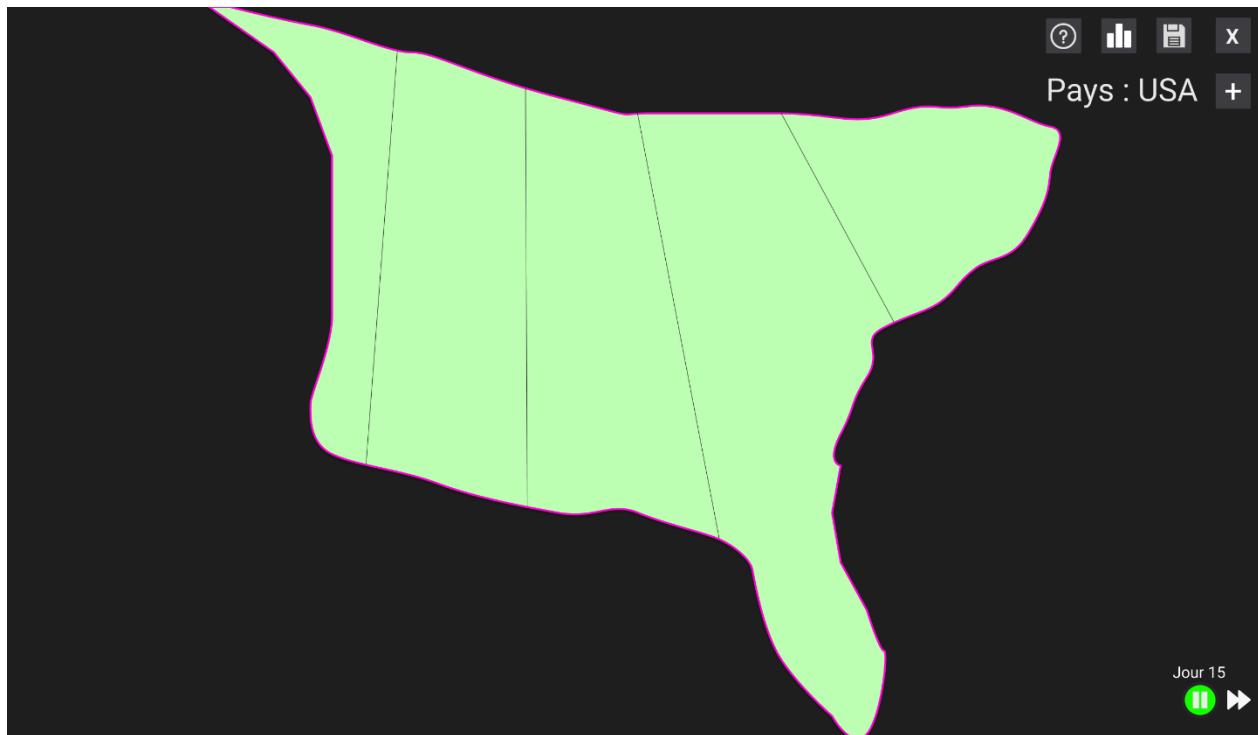
Esquisse 1.4 — Simulation — En Pause + Zoom



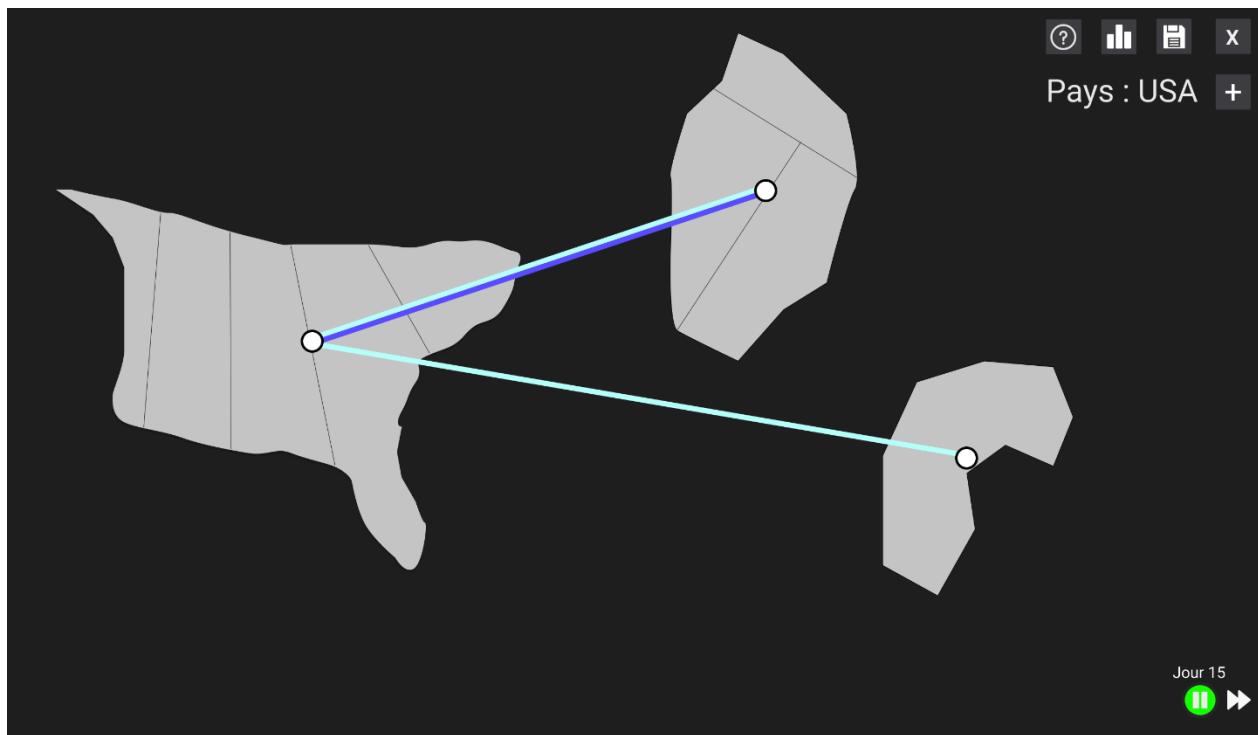
Esquisse 1,5 — Simulation — Bouton Vue



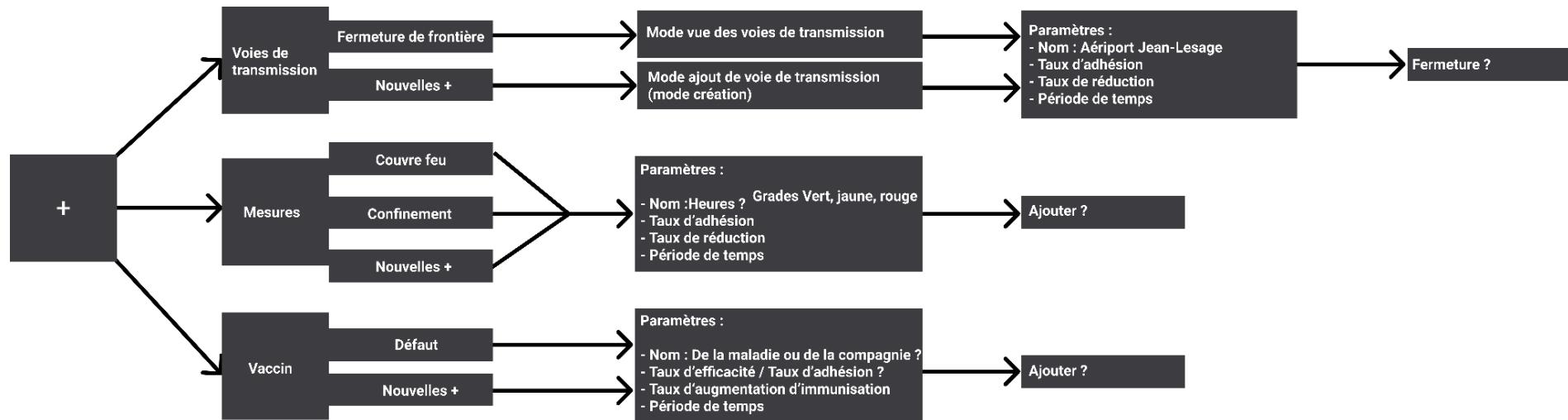
Esquisse 1,6 — Simulation — Vue Infection



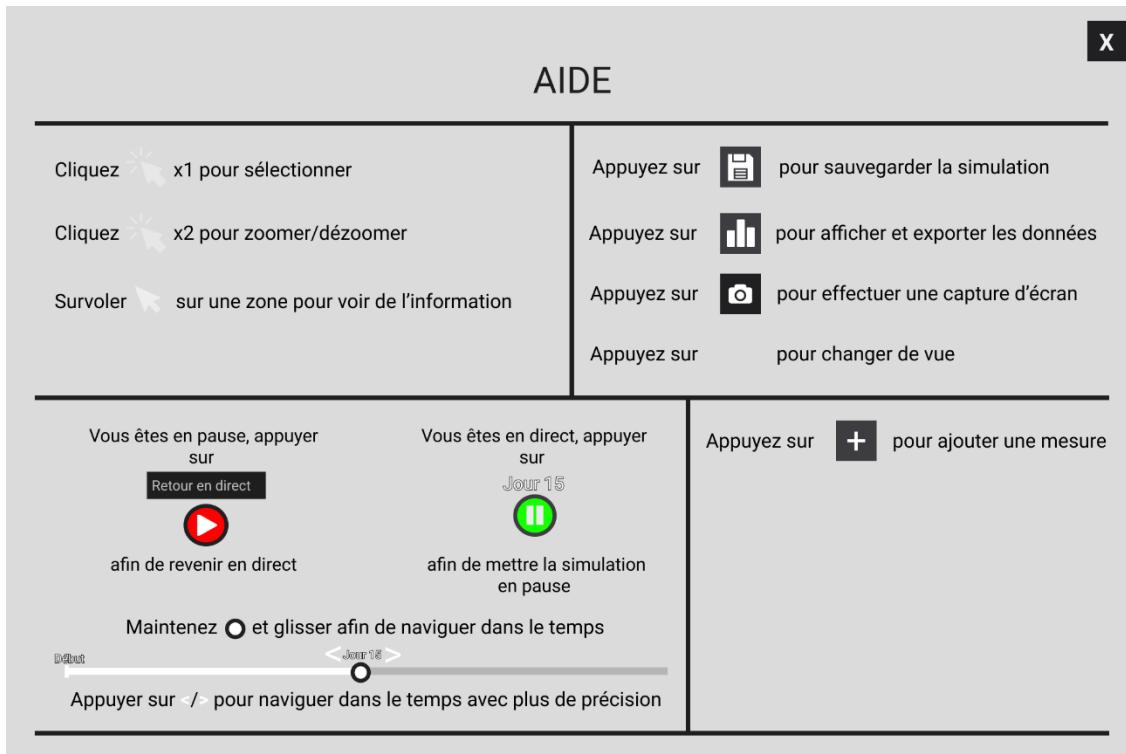
Esquisse 1,7 — Simulation — Vue Immunisation



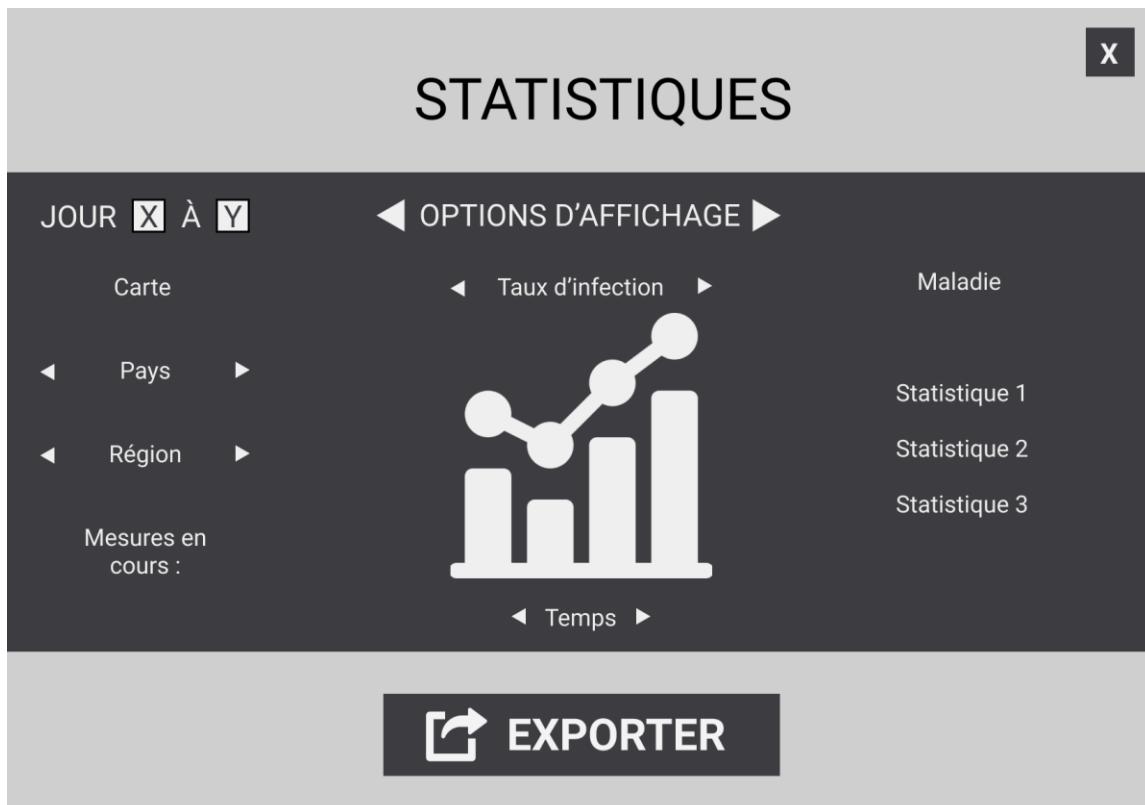
Esquisse 1,8 — Simulation — Vue Liens



Esquisse 1,9 — Simulation — Bouton Ajout



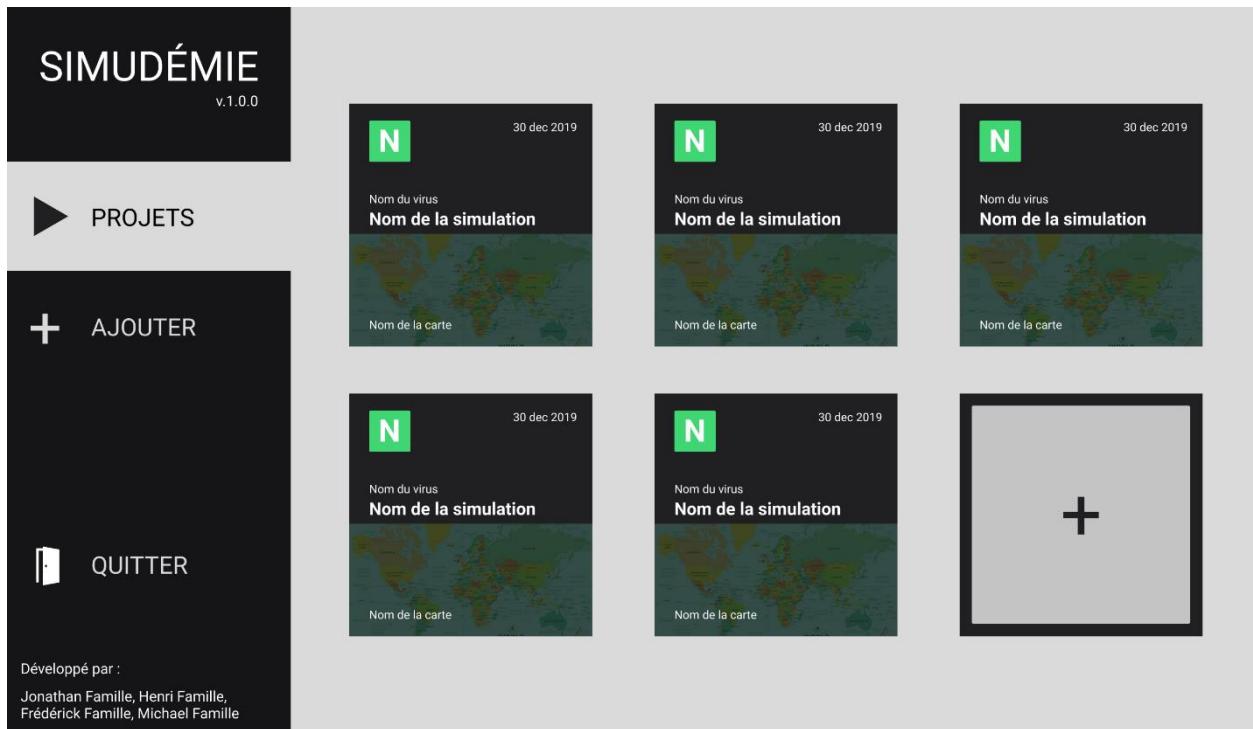
Esquisse 1,10 — Simulation — Aide



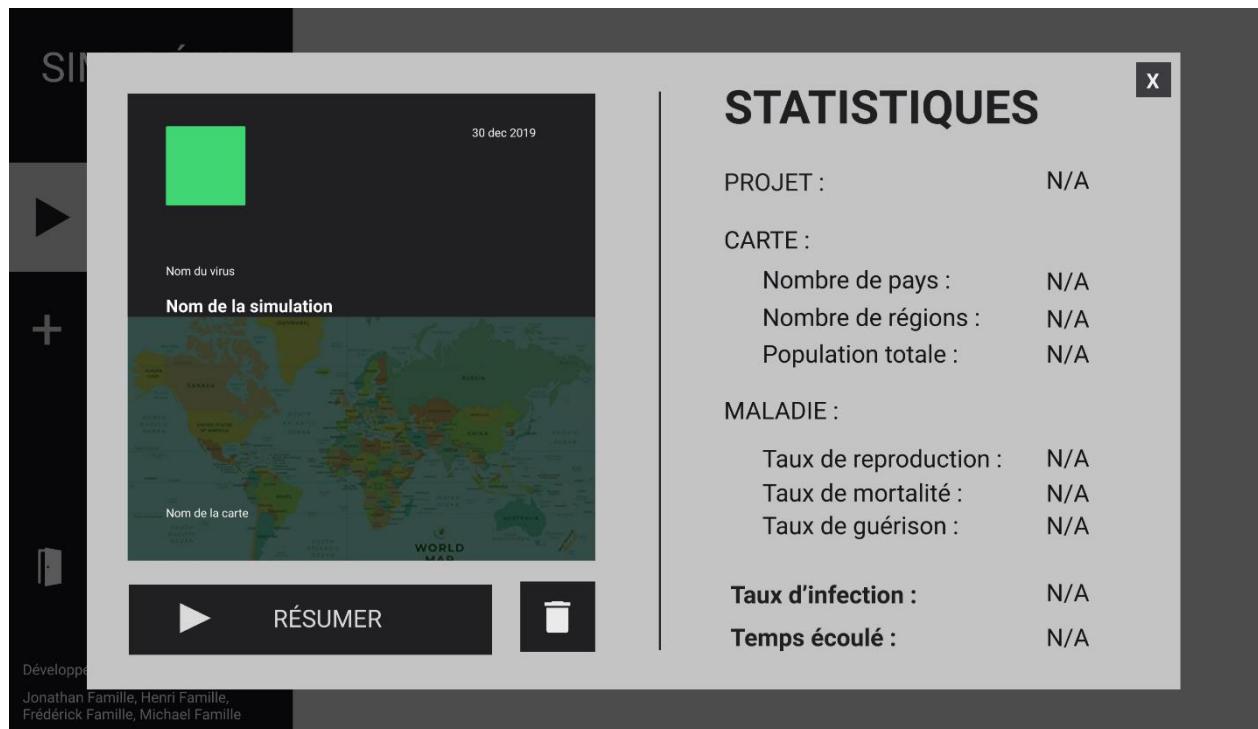
Esquisse 1,11 — Simulation — Statistiques



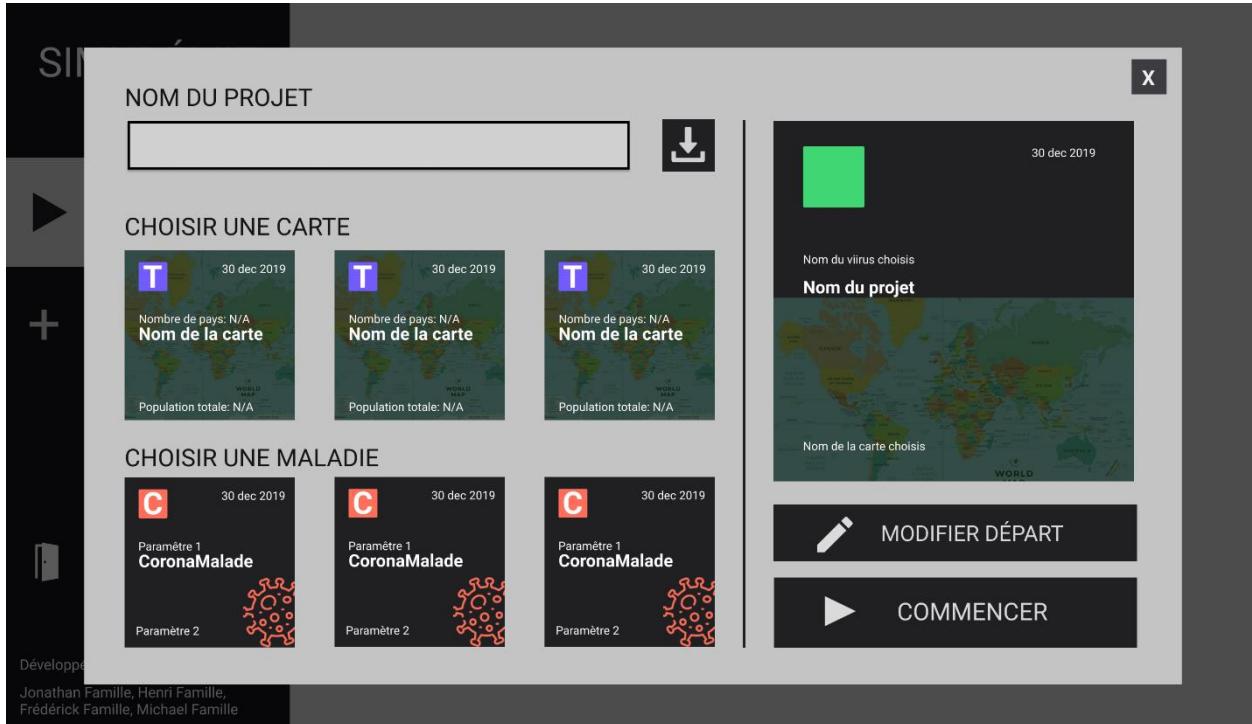
Esquisse 2.1 — Menus Scénarios — Logistique



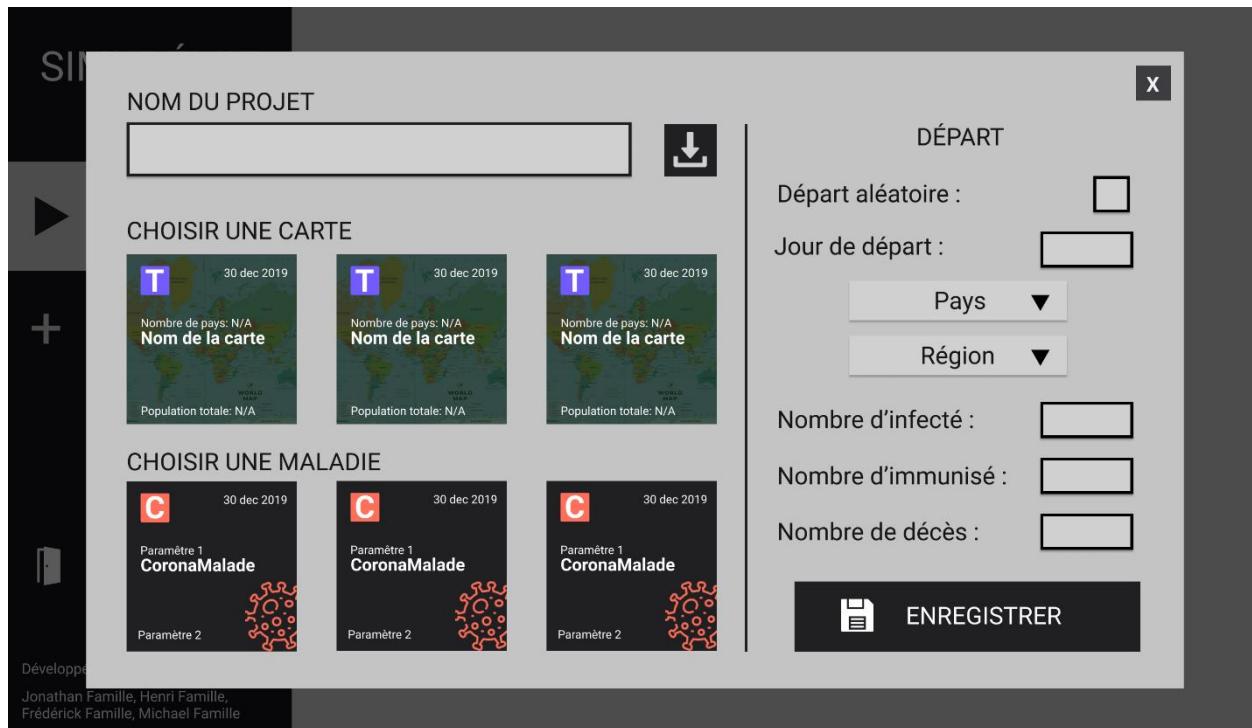
Esquisse 2.2 — Menus Scénarios — Choix d'un scénario



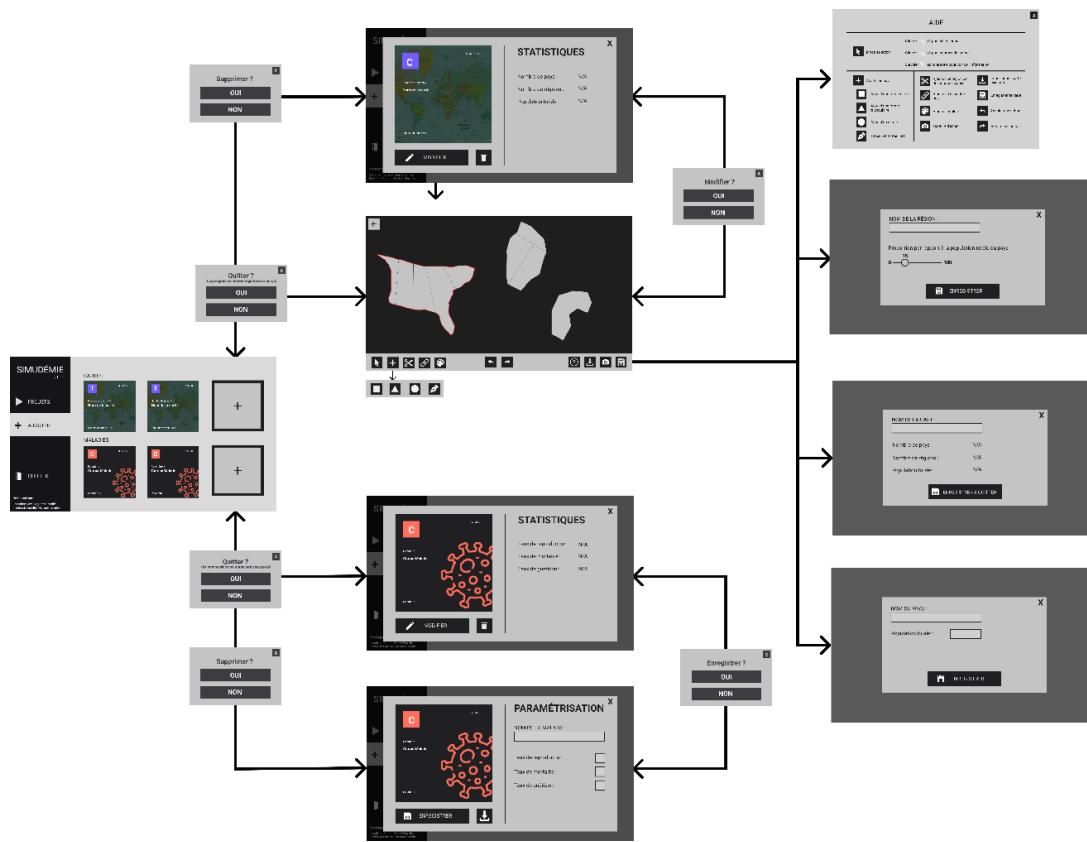
Esquisse 2.3 — Menus Scénarios - Affiche Scénario Existant



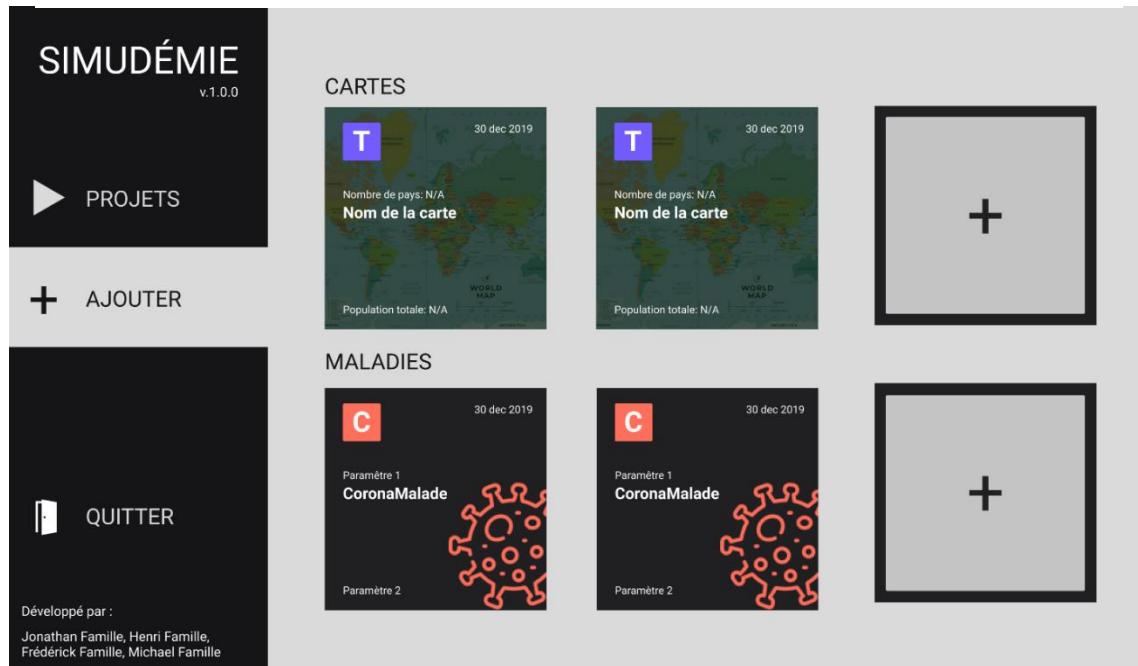
Esquisse 2.4 — Menus Scénarios — Création Nouveau Scénario



Esquisse 2.5 — Menus Scénarios — Création Nouveau Scénario — Modification Départ



Esquisse 3.1 — Menus Création — Logistique



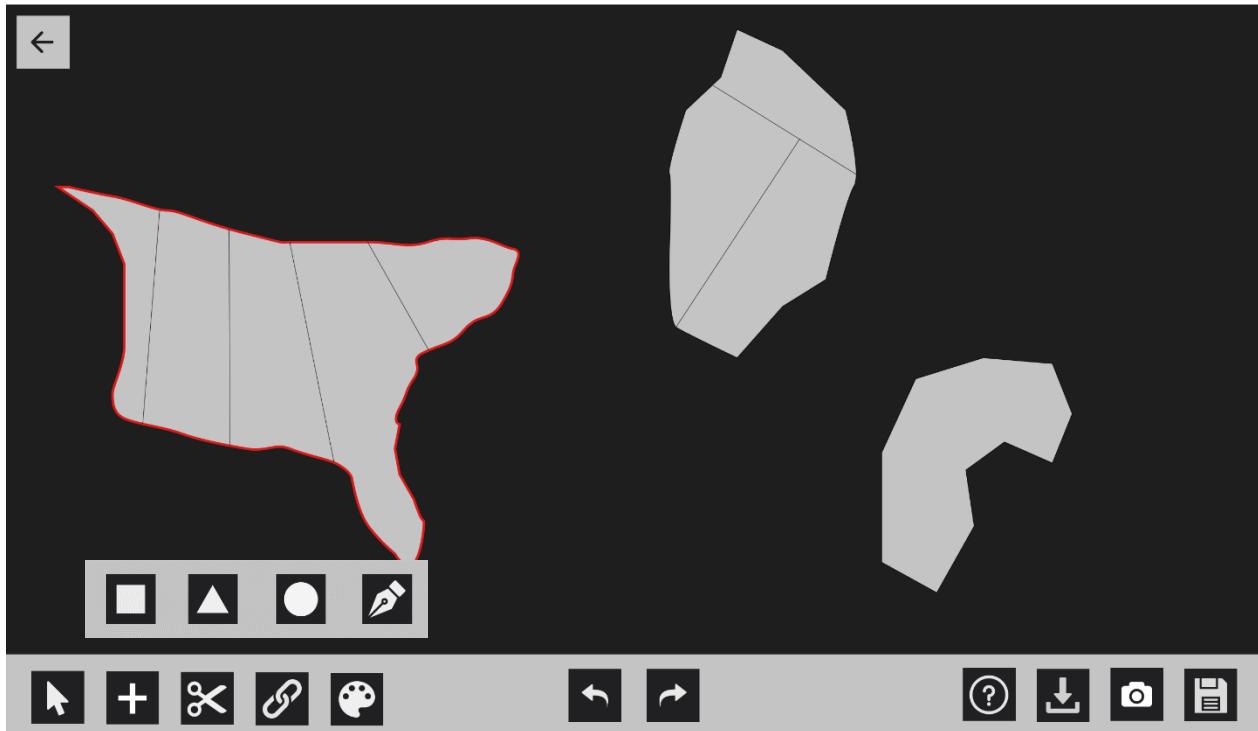
Esquisse 3.2 — Menus Création - Choix Carte & Maladie



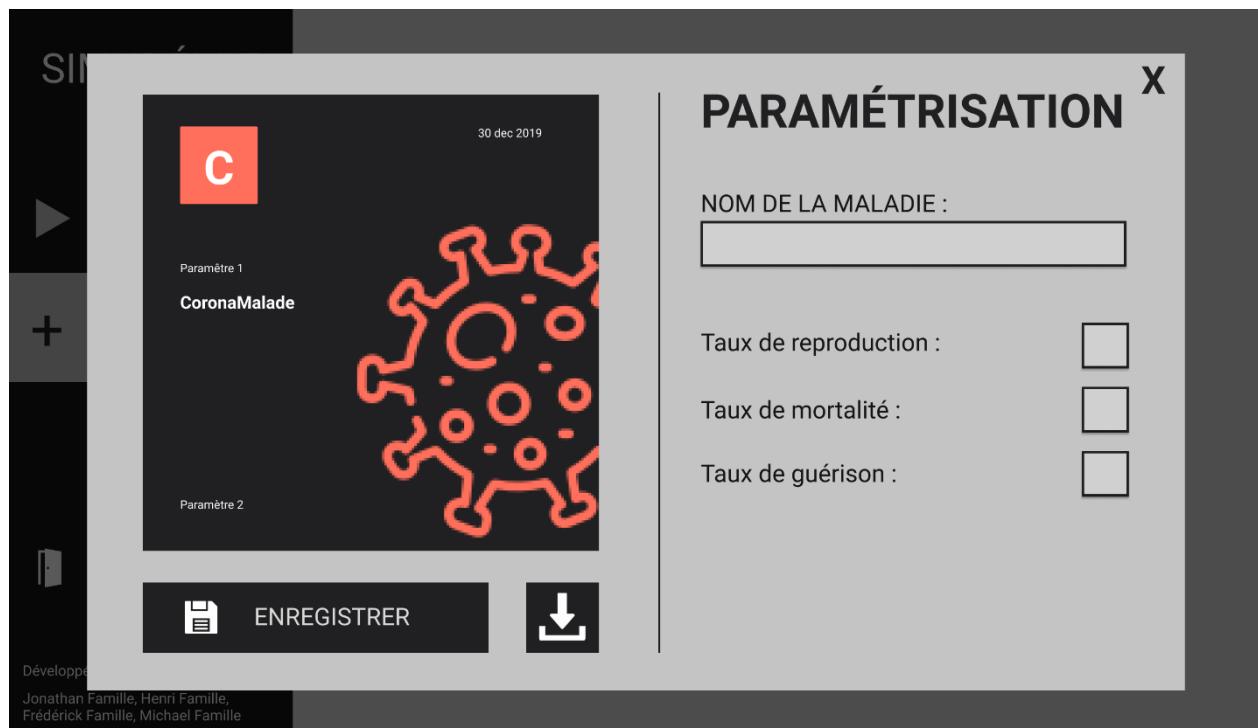
Esquisse 3.3 — Menus Création — Affiche Carte Existante



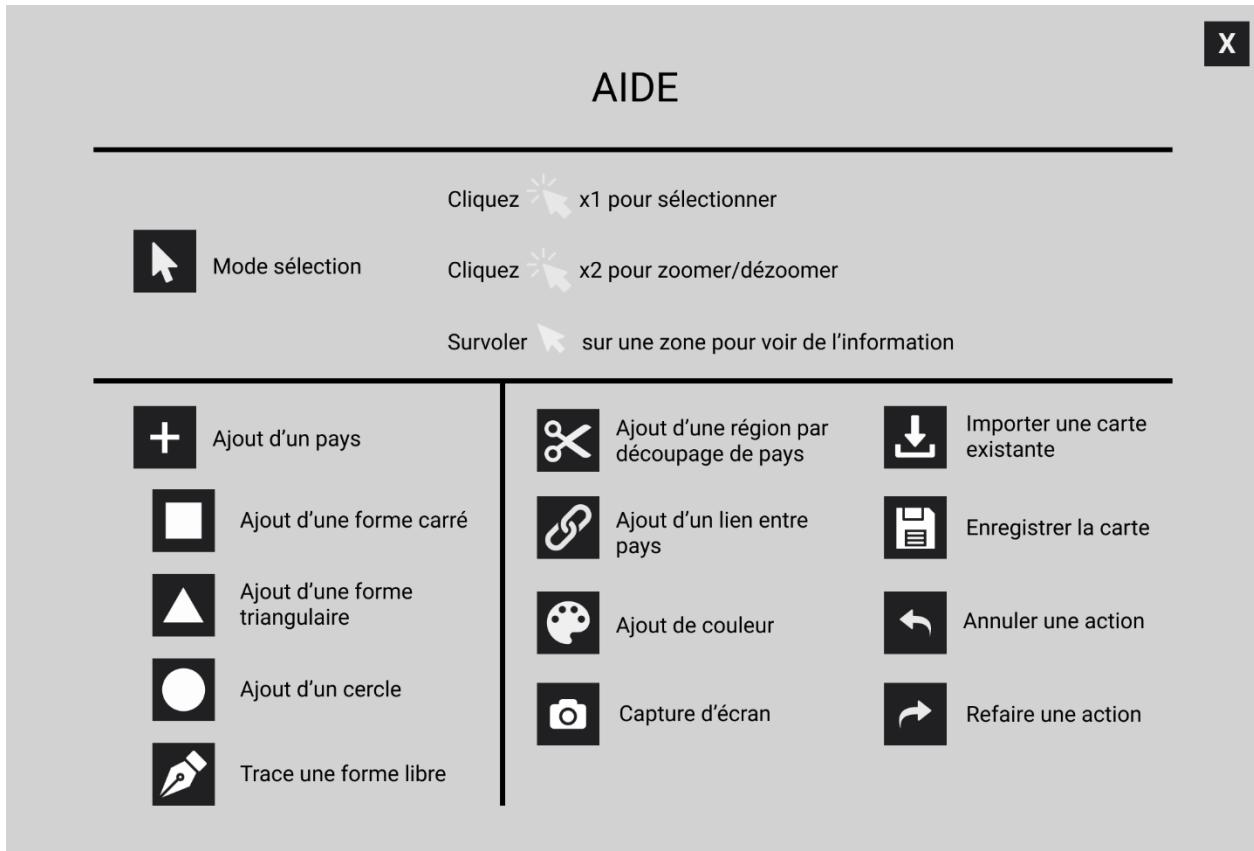
Esquisse 3.4 — Menus Création — Affiche Maladie Existante



Esquisse 3.5 — Menus Création — Crédit/Modification Carte Mondiale



Esquisse 3.6 — Menus Création — Crédit/Modification Maladie



Esquisse 3.7 — Menus Création — Crédit Carte Mondiale — Aide

NOM DU PAYS :

Population totale :

X

 ENREGISTRER

Esquisse 3.8 – Menus Création – Crédation Carte Mondiale – Crédation Pays

NOM DE LA RÉGION :

Proportion par rapport à la population totale du pays :

15
0 ————— 100

X

 ENREGISTRER

Esquisse 3.9 – Menus Création – Crédation Carte Mondiale – Crédation Région

NOM DE LA CARTE :

Nombre de pays : N/A

Nombre de régions : N/A

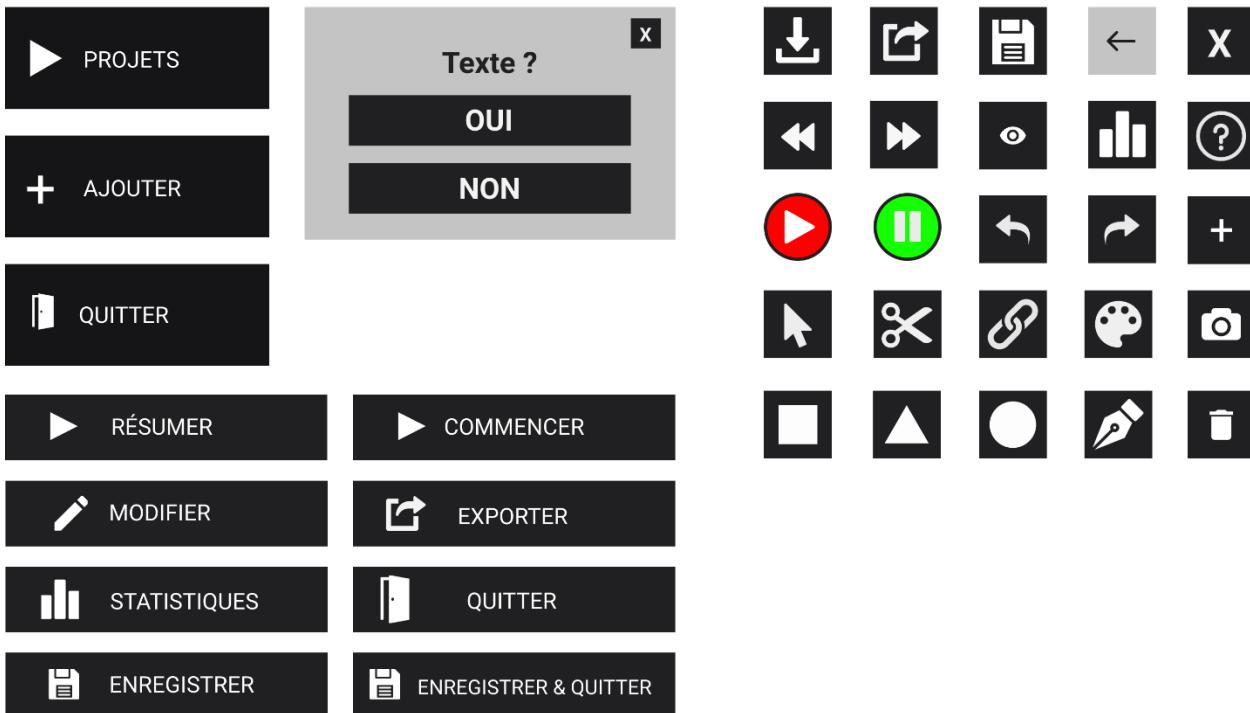
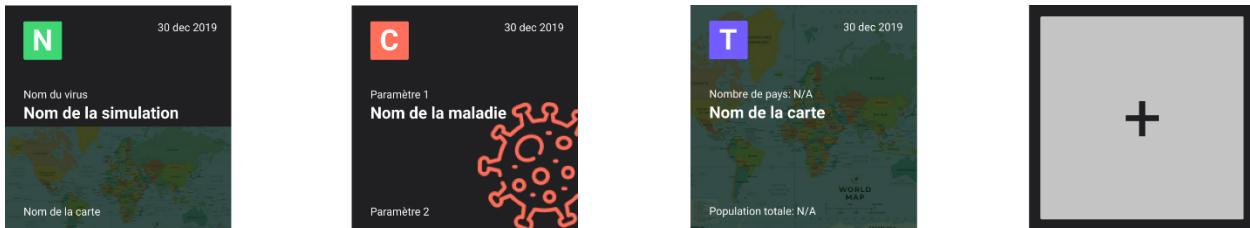
Population totale : N/A

X

 ENREGISTRER & QUITTER

Esquisse 3.10 – Menus Création – Crédation Carte Mondiale – Enregistrer

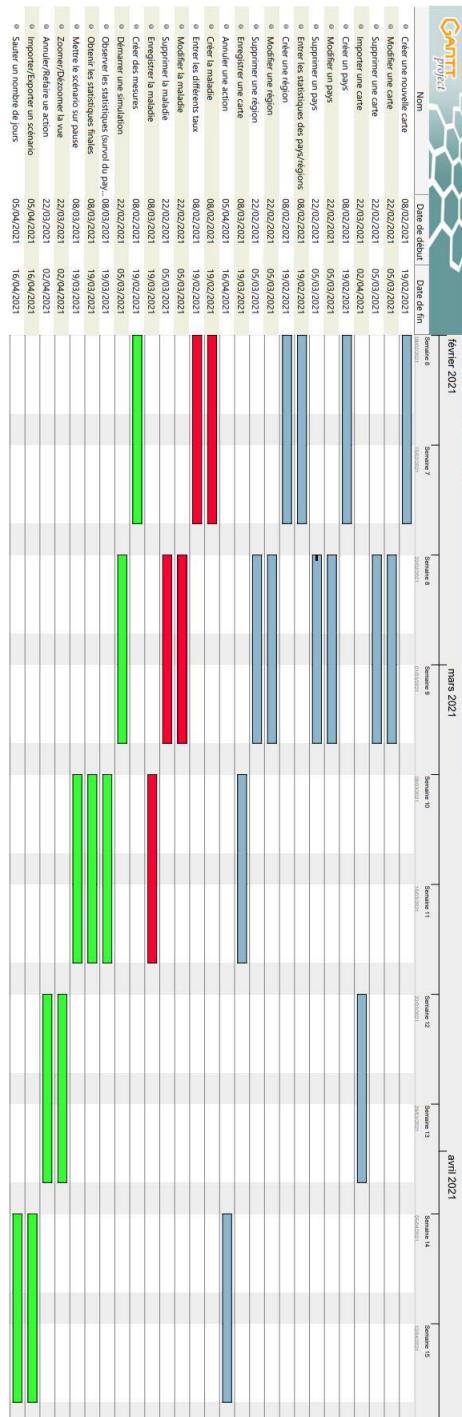
Thèmes & design - v1.0





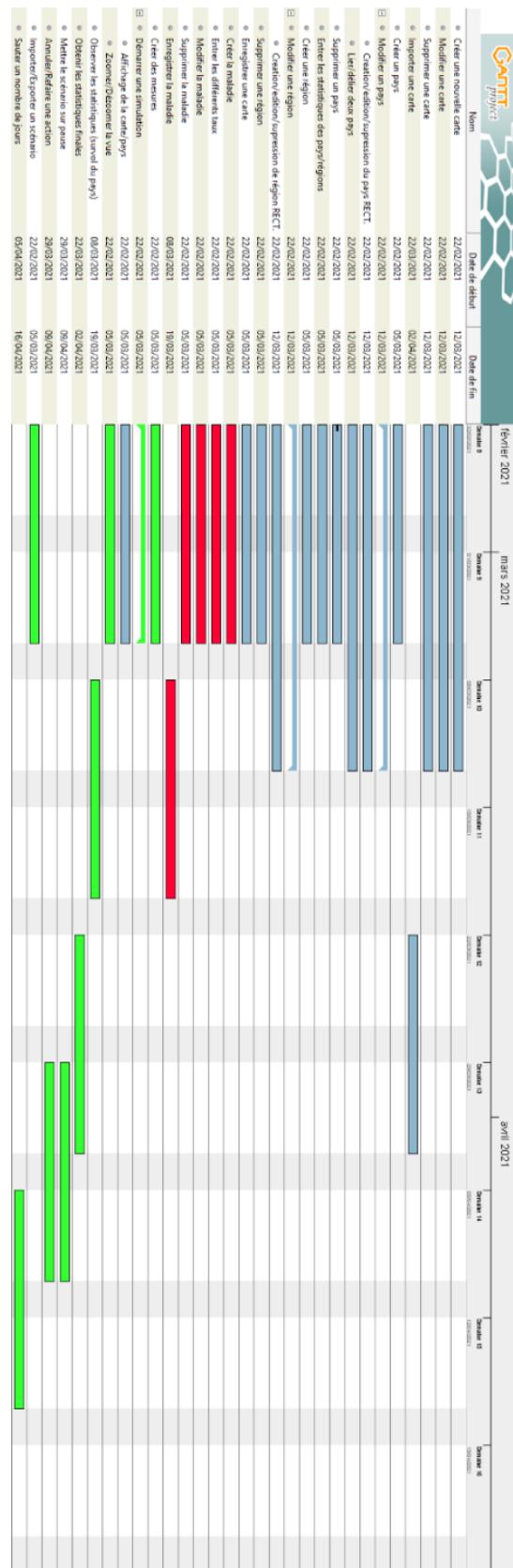
Diagrammes de Gantt

Diagramme – Livrable 1



Tâches			
Nom		Date de début	Date de fin
Créer une nouvelle carte		08/02/2021	19/02/2021
Modifier une carte		22/02/2021	05/03/2021
Supprimer une carte		22/02/2021	05/03/2021
Importer une carte		22/03/2021	02/04/2021
Créer un pays		08/02/2021	19/02/2021
Modifier un pays		22/02/2021	05/03/2021
Supprimer un pays		22/02/2021	05/03/2021
Entrer les statistiques des pays/régions		08/02/2021	19/02/2021
Créer une région		08/02/2021	19/02/2021
Modifier une région		22/02/2021	05/03/2021
Supprimer une région		22/02/2021	05/03/2021
Enregistrer une carte		08/03/2021	19/03/2021
Annuler une action		05/04/2021	16/04/2021
Créer la maladie		08/02/2021	19/02/2021
Entrer les différents taux		08/02/2021	19/02/2021
Modifier la maladie		22/02/2021	05/03/2021
Supprimer la maladie		22/02/2021	05/03/2021
Enregistrer la maladie		08/03/2021	19/03/2021
Créer des mesures		08/02/2021	19/02/2021
Démarrer une simulation		22/02/2021	05/03/2021
Observer les statistiques (survol du pays)		08/03/2021	19/03/2021
Obtenir les statistiques finales		08/03/2021	19/03/2021
Mettre le scénario sur pause		08/03/2021	19/03/2021
Zoomer/Dézoomer la vue		22/03/2021	02/04/2021
Annuler/Réfaire une action		22/03/2021	02/04/2021
Importer/Exporter un scénario		05/04/2021	16/04/2021
Sauter un nombre de jours		05/04/2021	16/04/2021

Diagramme – Livrable 2

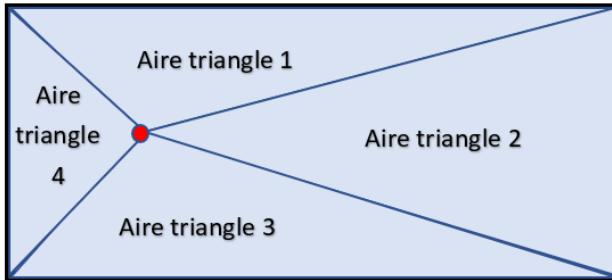


Tâches

Nom	Date de début	Date de fin
Créer une nouvelle carte	22/02/2021	12/03/2021
Modifier une carte	22/02/2021	12/03/2021
Supprimer une carte	22/02/2021	12/03/2021
Importer une carte	22/03/2021	02/04/2021
Créer un pays	22/02/2021	05/03/2021
Modifier un pays	22/02/2021	12/03/2021
Creation/editon/supression du pays RECT.	22/02/2021	12/03/2021
Lier/délier deux pays	22/02/2021	12/03/2021
Supprimer un pays	22/02/2021	05/03/2021
Entrer les statistiques des pays/régions	22/02/2021	05/03/2021
Créer une région	22/02/2021	05/03/2021
Modifier une région	22/02/2021	12/03/2021
Creation/editon/supression de région RECT.	22/02/2021	12/03/2021
Supprimer une région	22/02/2021	05/03/2021
Enregistrer une carte	08/03/2021	19/03/2021
Annuler une action	05/04/2021	16/04/2021
Créer la maladie	22/02/2021	05/03/2021
Entrer les différents taux	22/02/2021	05/03/2021
Modifier la maladie	22/02/2021	05/03/2021
Supprimer la maladie	22/02/2021	05/03/2021
Enregistrer la maladie	08/03/2021	19/03/2021
Créer des mesures	22/02/2021	05/03/2021
Démarrer une simulation	22/02/2021	05/03/2021
Affichage de la carte/pays	22/02/2021	05/03/2021
Zoomer/Dézoomer la vue	22/02/2021	05/03/2021
Observer les statistiques (survol du pays)	08/03/2021	19/03/2021
Obtenir les statistiques finales	08/03/2021	19/03/2021
Mettre le scénario sur pause	08/03/2021	19/03/2021
Annuler/Refaire une action	22/03/2021	02/04/2021
Importer/Exporter un scénario	05/04/2021	16/04/2021
Sauter un nombre de jours	05/04/2021	16/04/2021

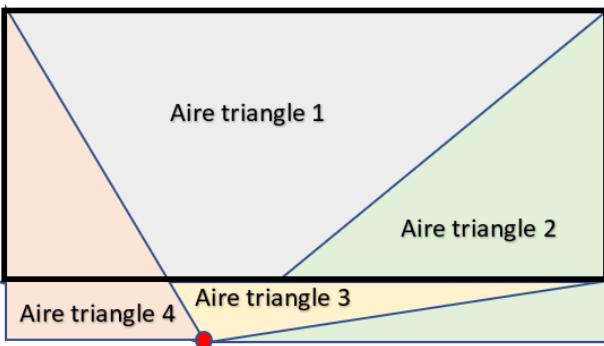
Pseudocode d'un algorithme – Livrable 2

Le client nous a posé le problème suivant : comment saurons-nous si l'utilisateur a cliqué sur une forme irrégulière (pouvant être convexe par exemple) ? Nous avions déjà solutionné l'énigme des formes rectangulaires en calculant les aires des quatre triangles formés par le point et les sommets. En effet, la somme de l'aire des quatre triangles produits à partir du point et des sommets doit égaler l'aire totale du rectangle.



$$\text{Aire rectangle} = A_1 + A_2 + A_3 + A_4$$

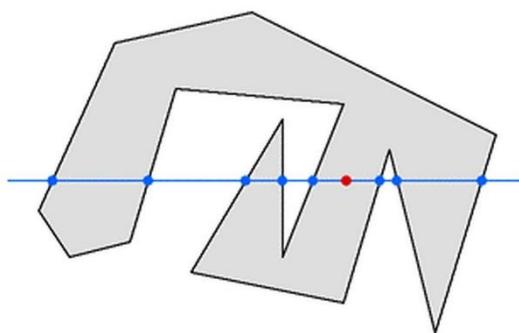
Donc le point est à l'intérieur de la forme.



$$\text{Aire rectangle} < A_1 + A_2 + A_3 + A_4$$

Donc le point est à l'extérieur de la forme.

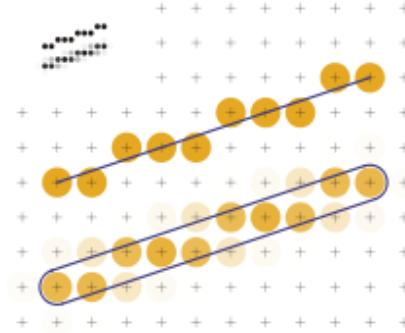
Pour la situation d'une forme irrégulière, les polygones convexes deviennent problématiques. La solution est donc de tracer une ligne sur l'axe des X à l'endroit Y où passe le point. Il suffit ensuite de regarder le nombre d'intersections entre les arrêtes et la droite. Si le nombre d'intersections est impair de chaque côté du point, le point est bel et bien à l'intérieur de la forme irrégulière. Voici une image qui illustre bien le principe (source ; alienryderflex.com/polygon/) :



Ce qui nous a demandé une certaine réflexion est la méthode permettant de trouver les points contenus dans les arrêtes. La solution que nous proposons ici n'est pas optimale, mais était suffisamment intrigante pour l'essayer. Puisqu'une droite est constituée de deux sommets, voici comment les points seront dessinés (source wiki) :

```
dx = x2 - x1
dy = y2 - y1
listePoints [ ]

for x from x1 to x2 do
    y = y1 + dy × (x - x1) / dx
    listePoints.add (x, y)
```



Nous obtenons ainsi une liste contenant tous les points qui font le contour de la forme. Nous pourrons donc implémenter notre algorithme décrit précédemment et effectuer la gestion d'exception (par exemple si un point se trouve sur un sommet).

```
lePoint = (x, y)
listeIntersectionsDroite []
for (i = x; i < extremiteTableau; i++)
    if (listePoints. contains (lePoint (x + i, y)))
        listeIntersectionsDroite.add (lePoint)
```

Nous avons donc ajouté toutes les intersections à la droite du point d'origine. Nous allons ensuite effectuer la même opération pour les intersections de gauche.

```
lePoint = (x, y)
listeIntersectionsGauche []
for (i = 0; i < x; i++)
    if (listePoints. contains (lePoint (x - i, y)))
        listeIntersectionsGauche.add (lePoint)
```

Il ne nous reste plus qu'à nous assurer que le nombre d'intersections contenues dans chaque liste est impair ou que le point est dans la liste.

```
If (listeIntersectionsGauche.contains(lePoint)
or listeIntersectionsDroite.contains(lePoint)
    Return True //À l'intérieur de la forme
Else if ((listeIntersectionsGauche.size Mod 2
== 1) and (listeIntersectionsDroite.size Mod 2 == 1))
    Return True
Else
    Return False //La pièce est à l'extérieur
```

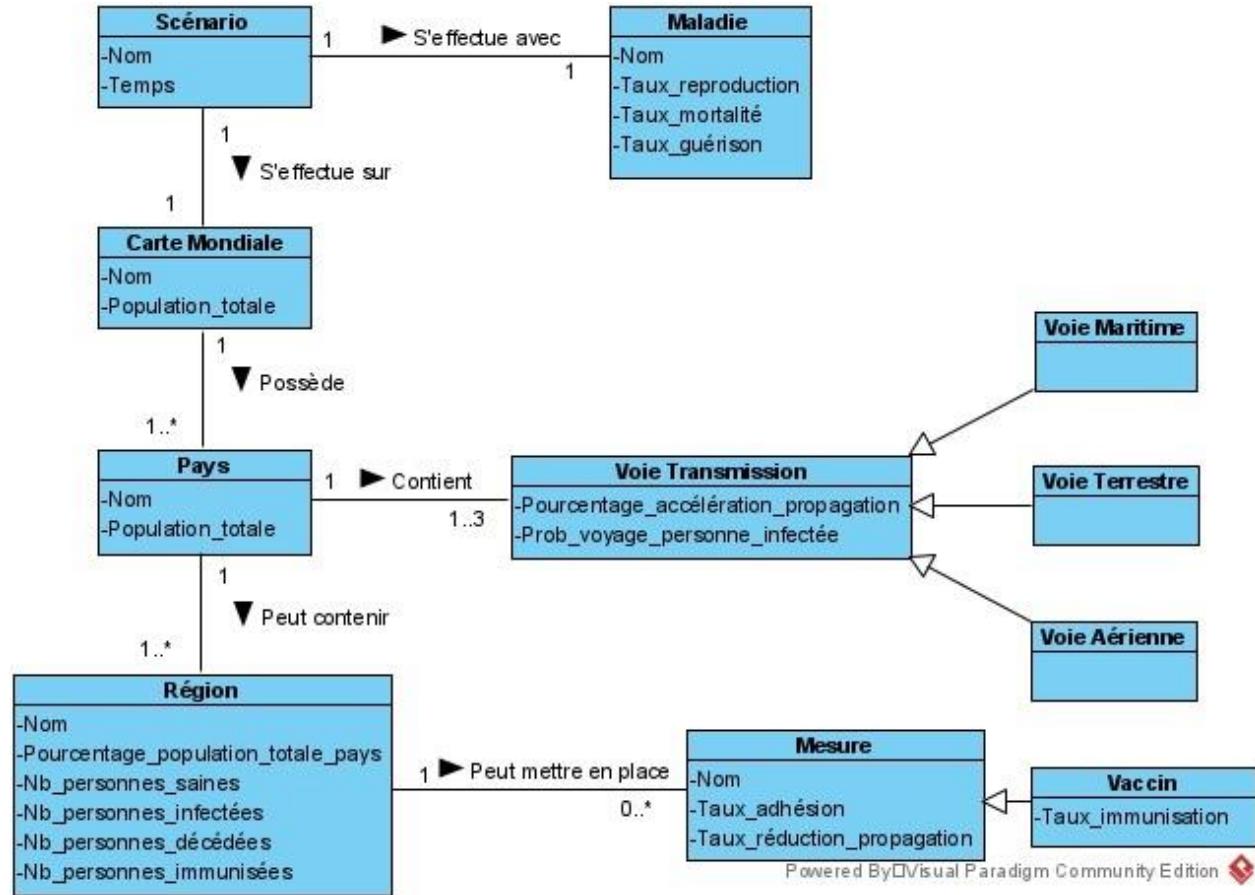
Nous obtenons finalement un booléen représentant la présence de notre point à l'intérieur de la forme irrégulière. Évidemment que ce processus n'est pas optimal, car il doit être effectué pour chaque pièce et à chaque clic. La solution simple est de « remplir » chaque point de l'objet Polygon créé et faire une simple validation sur le polygone pour savoir si le point en fait partie à l'aide de Polygon.contains (int x, int y).

Archives

Modèle de domaine

Modèle du domaine – Livrable 1

Diagramme de classes conceptuelles – Livrable 1



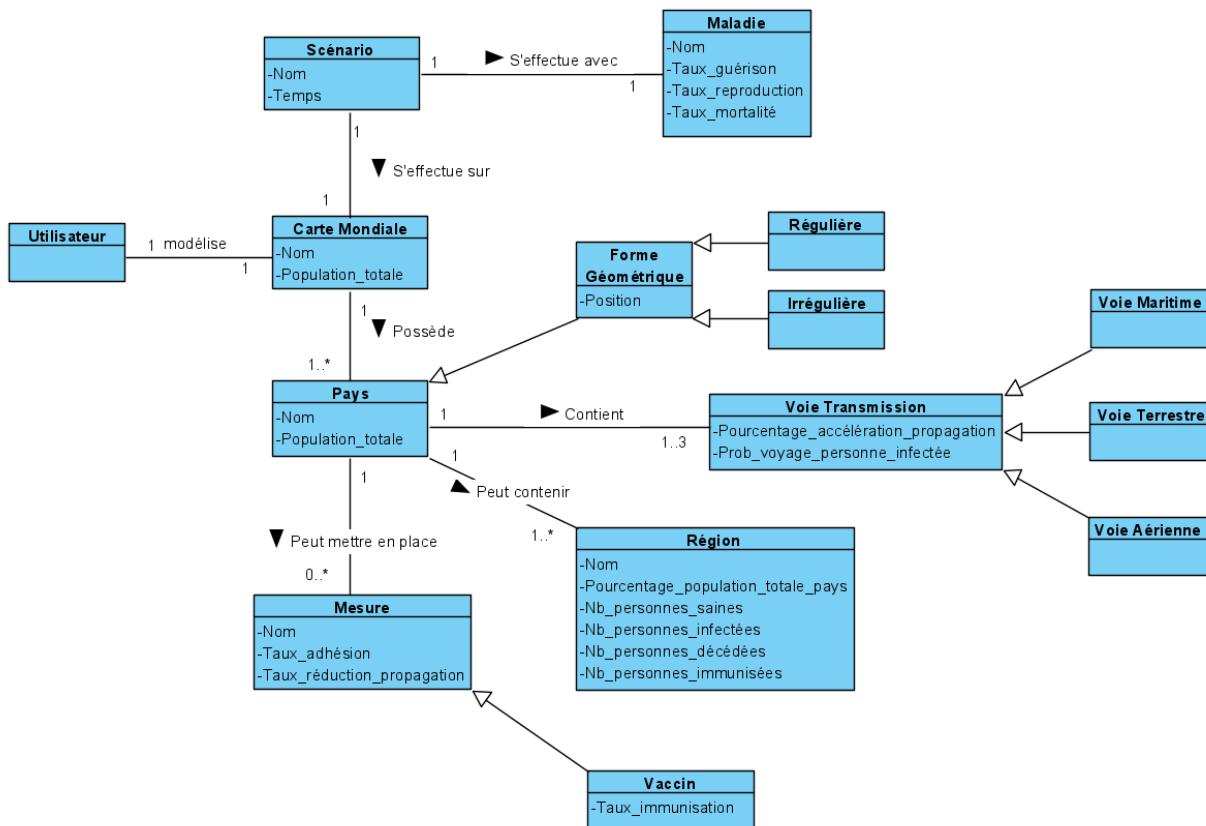
Powered By Visual Paradigm Community Edition

Description de classes conceptuelles – Livrable 1

Classe	Description
Scénario	Représente le lien entre la Maladie et la Carte Mondiale à utiliser pour la simulation. Possède un nom ainsi que le temps présent en jours de la simulation.
Maladie	Représente le virus et ses paramètres qui seront utilisés pendant la simulation, par le Scénario .
Carte Mondiale	Représente la carte préconfigurée qui sera utilisée par le Scénario et qui comporte plusieurs Pays . Compte la population totale à partir de ces derniers.
Pays	Représente une forme dans la Carte Mondiale qui peut être séparée en plusieurs Régions . Un Pays contient également une ou plusieurs voies de transmissions — nombre configuré par l'utilisateur.
Voie Transmission	Représente les voies de transmission de la Maladie pour la simulation. Une voie peut être maritime , terrestre ou aérienne . Chaque voie de transmission contient également son pourcentage d'accélération de propagation de la Maladie et la probabilité qu'une personne infectée voyage que l'utilisateur pourra configurer.
Région	Représente une subdivision d'un Pays . Une Région connaît son pourcentage de population selon la population totale de son Pays , le nombre de personnes saines, le nombre de personnes infectées, le nombre de personnes décédées et le nombre de personnes immunisées. Finalement, une Région peut aussi mettre en place diverses Mesures pour ralentir la transmission de la Maladie .
Mesure	Représente des opérations mises en place par l'utilisateur lors de la configuration du Scénario qui auront un impact sur la transmission de la Maladie .

Modèle du domaine – Livrable 3

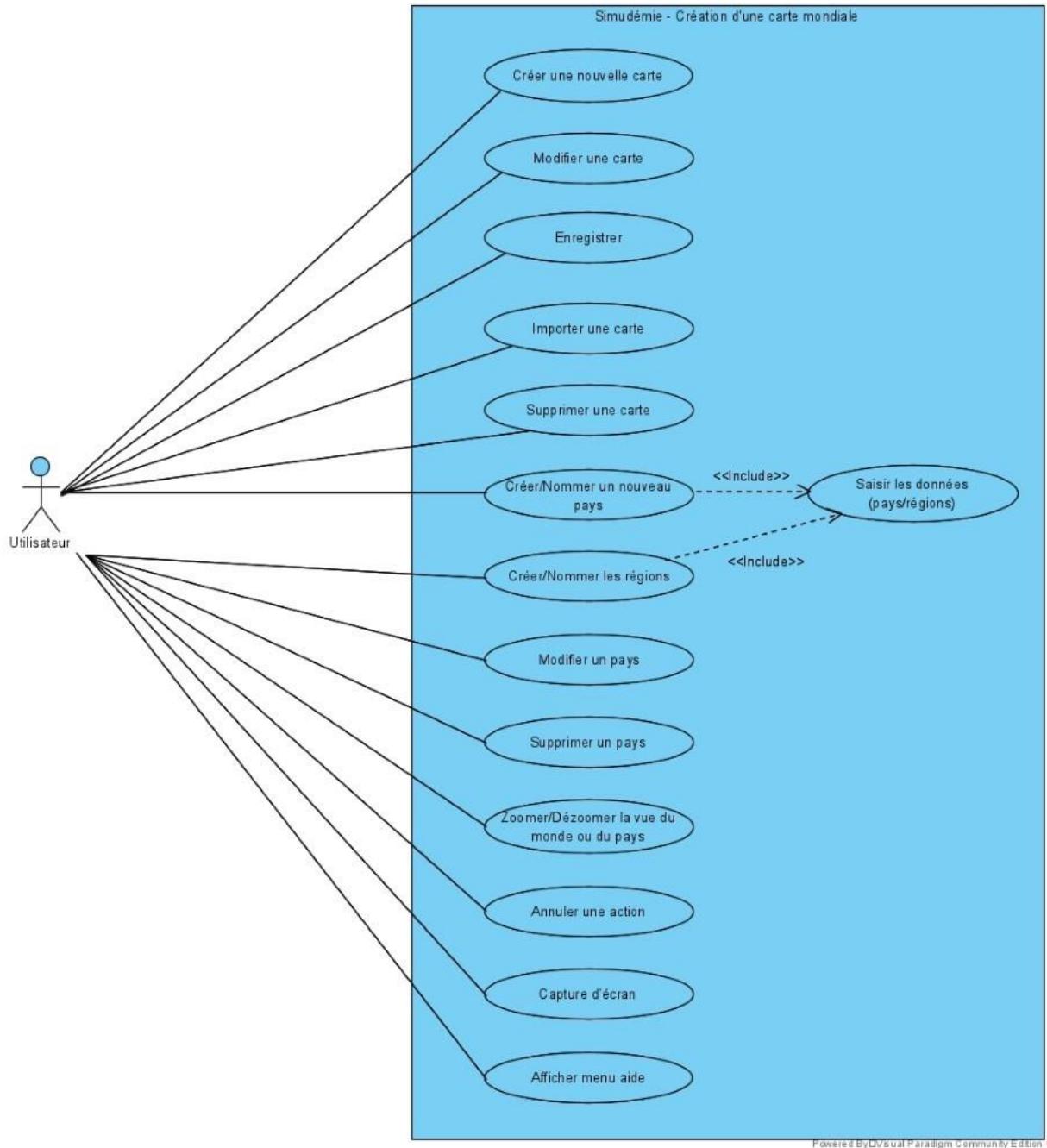
Notre révision du modèle du domaine suit l'un des exemples qui nous ont été fournis par le client. Nous n'avons pas eu à effectuer beaucoup de modifications, mais certaines informations étaient manquantes telles que l'utilisateur et les formes géométriques représentant les pays.



Modèle des cas d'utilisation

Diagrammes des cas d'utilisation – Livrable 1

Diagramme des cas d'utilisation - Crédit à la création d'une carte mondiale



Powered By UML Visual Paradigm Community Edition

Diagramme des cas d'utilisation - Crédation d'une maladie

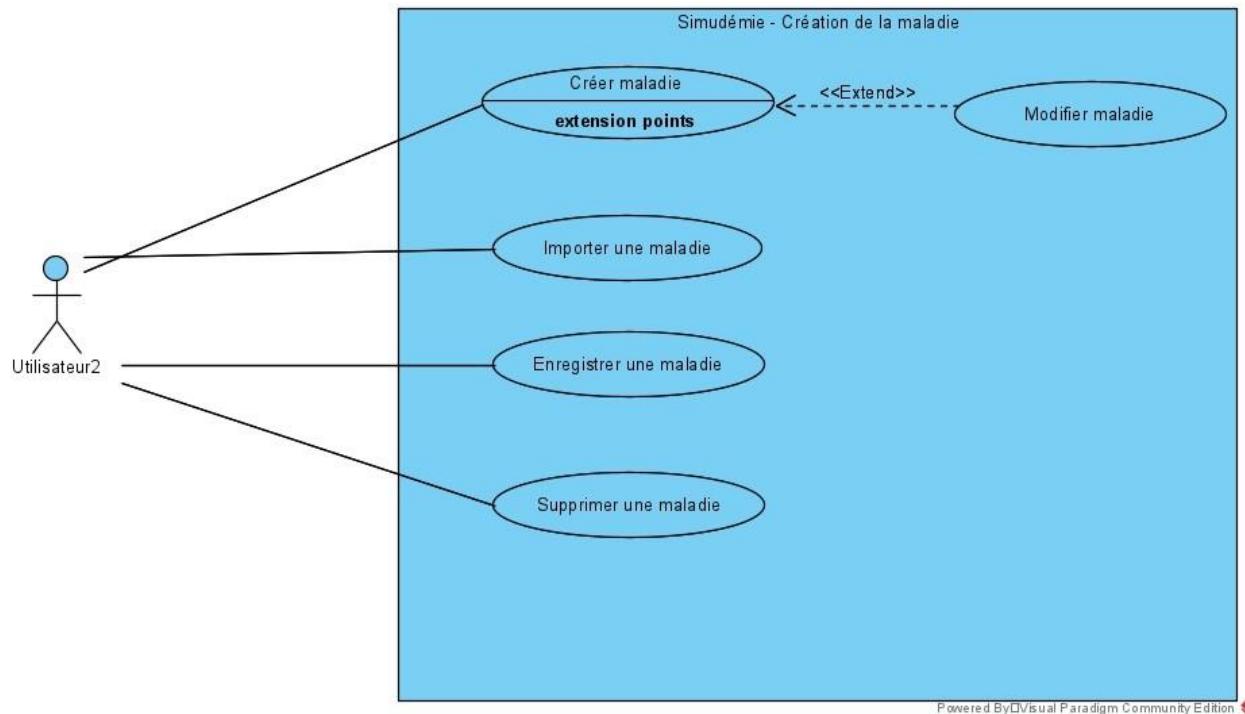


Diagramme des cas d'utilisation - Crédit à l'origine de la modélisation

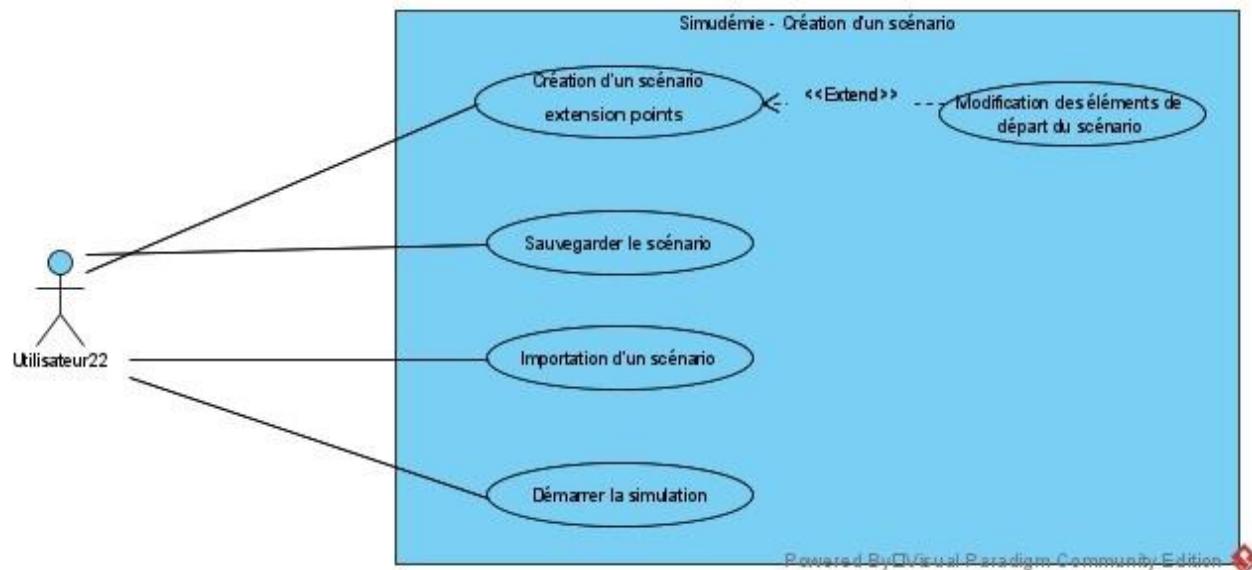
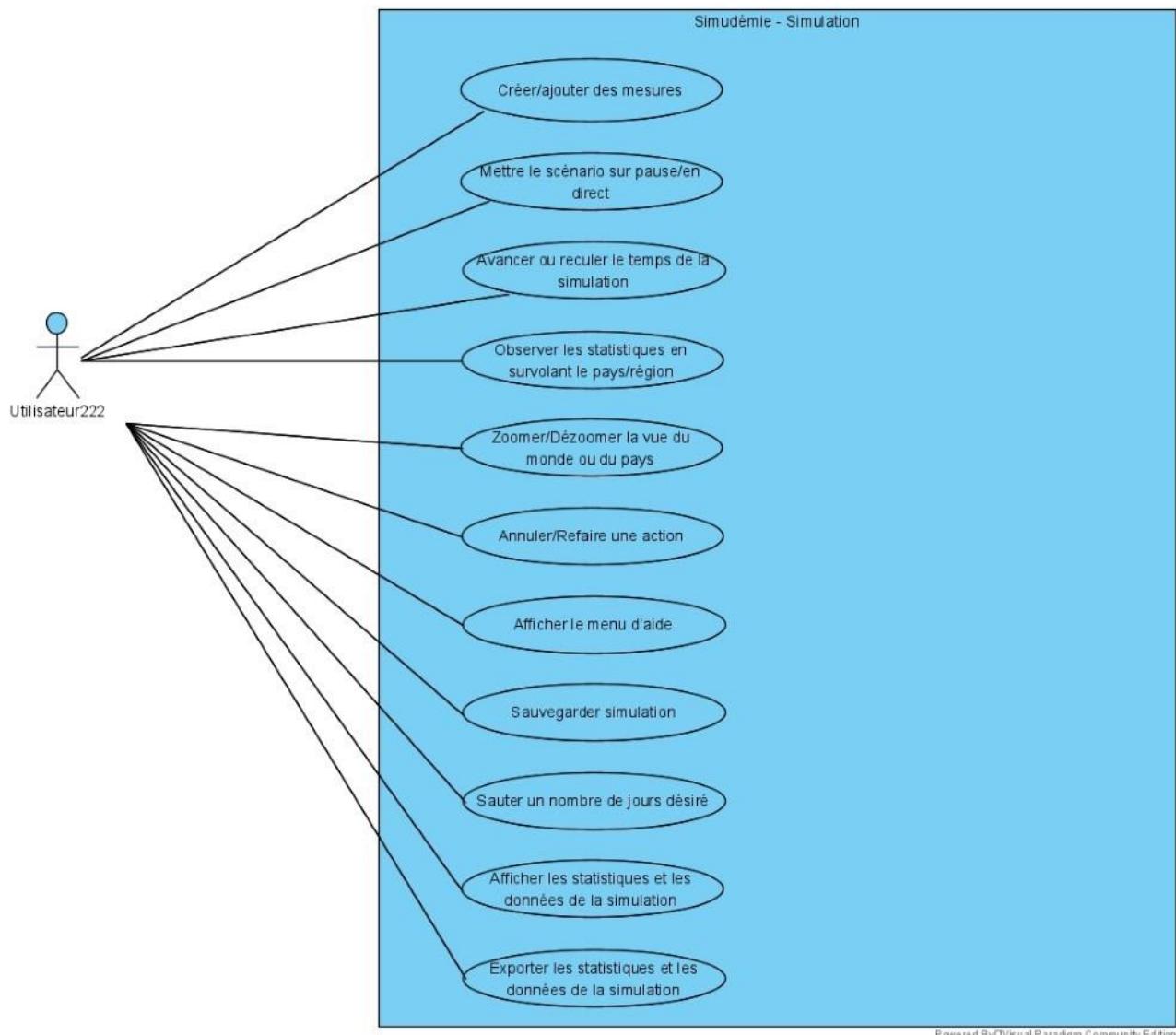


Diagramme des cas d'utilisation - Simulation



Powered ByVisual Paradigm Community Edition

Description des cas d'utilisation – Livrable 1

Création d'une carte mondiale

Description des cas d'utilisation - Crédit Carte Mondiale

Cas d'utilisation	Créer une nouvelle carte
Système	Simudémie — Création de cartes
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Désire créer une nouvelle carte Programmeur : Offre une interface adéquate et stimulante à l'utilisateur
Préconditions	Aucune
Garantie de succès	L'utilisateur peut accéder aux interfaces suivantes (création de pays ou de régions)
Scénario principal	
Utilisateur	Système Simudémie
L'utilisateur sélectionne l'option de créer une nouvelle carte.	Ouvre le menu de création d'une carte
Saisis le nom de la nouvelle carte & enregistre	Génère un fichier de carte
Scénario alternatif	
Aucun	

Description des cas d'utilisation — Modifier Carte Mondiale

Cas d'utilisation	Modifier une carte existante
Système	Simudémie — Création de cartes
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Veut modifier une carte Programmeur : Veut permettre la modification de la carte sans corrompre celle-ci
Préconditions	Il existe des cartes précédemment créées
Garantie de succès	La carte est modifiée sans erreur
Scénario principal	
Utilisateur	Système Simudémie
L'utilisateur sélectionne la carte existante voulant être modifiée	Ouvre le menu de création d'une carte et génère le contenu du fichier de la carte existante
L'utilisateur effectue les modifications désirées	Enregistre les modifications du fichier de la carte
Scénario alternatif	
Aucun	

Description des cas d'utilisation — Saisir les données (Pays/Régions)

Cas d'utilisation	Saisir les données (pays/régions)
Système	Simudémie — Création de cartes
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Souhaite entrer les statistiques aux pays et régions Programmeur : Souhaite que les statistiques soient associées aux bons objets
Préconditions	Pays ou région sélectionné(e)
Garantie de succès	Les statistiques sont enregistrées aux bons endroits
Scénario principal	
Utilisateur	Système Simudémie
L'utilisateur sélectionne pays/région	Affiche la fenêtre de saisie de données du pays/région sélectionné
L'utilisateur saisit les données	Enregistre les données
Scénario alternatif	
L'utilisateur saisit de mauvais paramètres	Message d'erreur demandant une nouvelle saisie

Description des cas d'utilisation — Importer une carte

Cas d'utilisation	Importer une carte
Système	Simudémie — Création de cartes
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Veut reprendre une carte précédemment sauvegarder Programmeur : Veut que le système reprenne exactement carte sauvegarder précédente en conservant les informations et les formes exactes.
Préconditions	Sauvegarde existante
Garantie de succès	La simulation est importée et lancée.
Scénario principal	
Utilisateur	Système Simudémie
L'utilisateur appuie sur le bouton [+]	Ouvre le menu de création d'une carte
L'utilisateur appuie sur le bouton [importer]	Ouvre le répertoire de fichiers
L'utilisateur choisit la carte désirée	Génère le contenu du fichier de la carte sélectionné dans l'interface de création
L'utilisateur enregistre la carte	Enregistre le fichier de la carte dans le répertoire du système
Scénario alternatif	
L'utilisateur choisit un fichier corrompu	Message d'erreur demandant de choisir un fichier différent

Description des cas d'utilisation — Zoom/Dézoom de la carte

Cas d'utilisation	Zoomer/Dézoomer la vue du monde ou du pays
Acteur(s)	Utilisateur
Type	Secondaire
Description	L'utilisateur peut agrandir/réduire la vue sur la carte. En double-cliquant sur le pays désiré, la caméra effectue un zoom sur ce dernier.

Description des cas d'utilisation — Crée pays

Cas d'utilisation	Créer/Nommer un nouveau pays
Acteur(s)	Utilisateur
Type	Primaire
Description	L'utilisateur veut créer/nommer un nouveau pays. Il choisit l'option d'ajout d'une forme au choix qui deviendra un pays et entre les informations nécessaires.

Description des cas d'utilisation — Crée région

Cas d'utilisation	Créer/Nommer une nouvelle région
Acteur(s)	Utilisateur
Type	Primaire
Description	Le client veut créer/nommer la nouvelle région. Il choisit l'option de découpage [bouton ciseaux] afin de découper le pays. Ce découpage deviendra une région et l'utilisateur entre les informations nécessaires.

Description des cas d'utilisation — Modifier pays/région

Cas d'utilisation	Modifier un pays ou une région
Acteur(s)	Utilisateur
Type	Secondaire
Description	L'utilisateur veut modifier un pays ou une région existante. Il sélectionne le pays ou la région à l'aide de l'option de sélection.

Description des cas d'utilisation — Supprimer pays/région

Cas d'utilisation	Supprimer un pays ou une région
Acteur(s)	Utilisateur
Type	Secondaire
Description	L'utilisateur veut supprimer un pays ou une région existante. Il sélectionne le pays ou la région à l'aide de l'option suppression [bouton de la gomme à effacer].

Description des cas d'utilisation — Crée une voie de transmission

Cas d'utilisation	Créer une voie de transmission
Acteur(s)	Utilisateur
Type	Primaire
Description	Le client veut créer une nouvelle voie de transmission. Il sélectionne l'option des liens et ajoute la voie au choix entre les pays désirés.

Description des cas d'utilisation — Ajouter une couleur

Cas d'utilisation	Ajouter une couleur
Acteur(s)	Utilisateur
Type	Secondaire
Description	L'utilisateur veut ajouter une couleur à un pays. Il prend l'option de peinture et ajoute une couleur au pays.

Description des cas d'utilisation — Affiche Aide

Cas d'utilisation	Afficher menu aide
Acteur(s)	Utilisateur
Type	Secondaire
Description	L'utilisateur peut obtenir de l'aide en appuyant sur le bouton correspondant.

Description des cas d'utilisation — Capture d'écran

Cas d'utilisation	Capture d'écran
Acteur(s)	Utilisateur
Type	Secondaire
Description	L'utilisateur peut faire une capture d'écran des éléments présents sur la carte.

Description des cas d'utilisation — Enregistrer Carte

Cas d'utilisation	Enregistrer
Acteur(s)	Utilisateur
Type	Secondaire
Description	L'utilisateur veut enregistrer sa carte. Lorsque la carte est à son goût, il choisit l'option d'enregistrement.

Création d'une maladie

Description des cas d'utilisation - Création Maladie

Cas d'utilisation	Créer une nouvelle maladie
Système	Simudémie — Création de maladies
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Désire créer une nouvelle maladie Programmeur : Offre une interface adéquate et stimulante à l'utilisateur
Préconditions	Aucune
Garantie de succès	La maladie est créée avec succès et s'affiche maintenant dans l'interface
Scénario principal	
Utilisateur	Système Simudémie
L'utilisateur clique sur l'onglet [ajouter] dans le menu	Change d'onglet dans le menu
L'utilisateur ajoute une maladie avec le [+] dans la section respective	Ouvre la fenêtre de saisie des paramètres
L'utilisateur saisit les paramètres souhaités	Affiche la saisie des paramètres
L'utilisateur sauvegarde	Message de confirmation
Scénario alternatif	
L'utilisateur saisit de mauvais paramètres	Message d'erreur demandant une nouvelle saisie

Description des cas d'utilisation — Modifier Maladie

Cas d'utilisation	Modifier une maladie existante
Système	Simudémie — Création de maladies
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Désire modifier une maladie Programmeur : Veut permettre la modification de tous les attributs de la maladie
Préconditions	Une maladie doit déjà exister
Garantie de succès	La maladie est belle et bien modifiée
Scénario principal	
Utilisateur	Système Simudémie
L'utilisateur clique sur l'onglet [ajouter] dans le menu	Change d'onglet dans le menu
L'utilisateur clique sur la maladie à modifier	Affiche la maladie
Il clique ensuite sur le bouton de modification	Ouvre la fenêtre de saisie des paramètres
Il modifie les champs souhaités et enregistre le tout après	Affiche la saisie des paramètres & Message de confirmation d'enregistrement
Scénario alternatif	
L'utilisateur saisit de mauvais paramètres	Message d'erreur demandant une nouvelle saisie

Description des cas d'utilisation — Importer maladie

Cas d'utilisation	Importer une maladie
Acteur(s)	Utilisateur
Type	Secondaire
Description	L'utilisateur veut importer une maladie déjà existante avec les informations exactes qu'elle avait.

Description des cas d'utilisation — Enregistrer maladie

Cas d'utilisation	Enregistrer une maladie
Acteur(s)	Utilisateur
Type	Secondaire
Description	L'utilisateur veut enregistrer sa maladie. Lorsque la maladie convient, il choisit l'option d'enregistrement.

Description des cas d'utilisation — Supprimer maladie

Cas d'utilisation	Supprimer une maladie
Acteur(s)	Utilisateur
Type	Secondaire
Description	L'utilisateur veut supprimer une maladie existante. Il sélectionne la maladie et clique sur la poubelle pour supprimer celle-ci.

Création d'un scénario

Description des cas d'utilisation - Création Scénario

Cas d'utilisation	Création d'un nouveau scénario
Système	Simudémie - Crédit scénario
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Désire créer un scénario Programmeur : Fournit une interface afin de choisir la carte, la maladie et le nom de ce scénario.
Préconditions	L'utilisateur devra avoir choisi une carte et une maladie.
Garantie de succès	L'utilisateur a maintenant un scénario prêt à rouler.
Scénario principal	
Utilisateur	Système Simudémie
L'utilisateur ajoute un scénario avec le [+]	Affiche la fenêtre de création de scénarios
L'utilisateur entre le nom du scénario, il choisit la carte à utiliser ainsi que la maladie.	Affichage visuel des éléments sélectionnés
Commencer le scénario pour générer la sauvegarde.	Génère la sauvegarde du scénario
Scénario alternatif	
L'utilisateur n'a pas choisi de carte ou de maladie	Message d'erreur indiquant les éléments manquants.

Description des cas d'utilisation - Modification des éléments de départ

Cas d'utilisation	Modification des éléments de départ d'un scénario
Système	Simudémie - Création scénario
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Désire créer un scénario & modifier les éléments de départ du scénario Programmeur : Fournit une interface afin de choisir la carte, la maladie, le nom de ce scénario & les éléments de départ à modifier.
Préconditions	L'utilisateur devra avoir choisi une carte et une maladie afin de modifier ces éléments.
Garantie de succès	L'utilisateur a maintenant un scénario prêt à rouler avec un départ modifiable.
Scénario principal	
Utilisateur	Système Simudémie
L'utilisateur ajoute un scénario avec le [+]	Affiche la fenêtre de création de scénarios
L'utilisateur entre le nom du scénario, il choisit la carte à utiliser ainsi que la maladie.	Affichage visuel des éléments sélectionnés
L'utilisateur clique sur [modifier départ]	Affiche la fenêtre de modification de départ
L'utilisateur saisit les éléments souhaités	Affichage des éléments saisis par l'utilisateur
Enregistrer le scénario pour générer la sauvegarde.	Génère la sauvegarde du scénario
Scénario alternatif	
L'utilisateur saisit de mauvais paramètres	Message d'erreur demandant une nouvelle saisie

Description des cas d'utilisation – Importation d'un scénario

Cas d'utilisation	Modification des éléments de départ d'un scénario
Système	Simudémie — Création d'un scénario
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Désire reprendre le scénario qu'un collègue lui a envoyé Programmeur : Fournis l'option d'intégrer un scénario externe dans la plage locale de l'utilisateur
Préconditions	Le scénario à importer doit être un scénario valide créé par la même application
Garantie de succès	L'utilisateur peut utiliser le scénario comme s'il était le sien
Scénario principal	
Utilisateur	Système Simudémie
L'utilisateur ajoute un scénario avec le [+]	Affiche la fenêtre de création de scénarios
L'utilisateur clique sur l'icône d'importation	Affiche les répertoires de fichiers de l'utilisateur
L'utilisateur choisit le fichier de scénario désiré	Importe et génère les informations contenues dans le fichier du scénario
Scénario alternatif	
L'utilisateur choisit un fichier corrompu	Message d'erreur demandant de choisir un nouveau fichier

Description des cas d'utilisation — Supprimer Scénario

Cas d'utilisation	Supprimer un scénario
Acteur(s)	Utilisateur
Type	Secondaire
Description	L'utilisateur veut supprimer un scénario existant. Il sélectionne le scénario et clique sur la poubelle afin de le supprimer.

Simulation

Description des cas d'utilisation — Crer/Ajouter Mesures

Cas d'utilisation	Crer/ajouter des mesures
Systme	Simudmie — Simulation
Acteur	Utilisateur
Parties prenantes et intrts	Utilisateur : Dsire ajouter une mesure  la simulation Programmeur : Va reprsenter les effets de la mesure dans la simulation (ajuster les taux, les liens, un vaccin, etc.)
Prconditions	La simulation doit tre en cours
Garantie de succs	Les mesures seront prsentes et prtent  l'utilisation.
Scnario principal	
Utilisateur	Systme Simudmie
Appuie sur le bouton [+]	Affiche la liste des mesures disponibles  l'ajout dans la simulation
L'utilisateur choisit de crer une nouvelle mesure en appuyant sur [nouvelle +]	Affiche la fentre de configuration d'une mesure
Il saisit les paramtres de la mesure et appuie sur le bouton appliquer	Affiche la saisie des paramtres, enregistre ces derniers et applique ceux-ci  la simulation
Scnario alternatif	
Appuie sur le bouton [+]	Affiche la liste des mesures disponibles  l'ajout dans la simulation
L'utilisateur choisit la mesure dsire tablie par dfaut	Applique cette mesure  la simulation prsente
Scnario alternatif	
L'utilisateur saisit de mauvais paramtres	Affiche un message d'erreur demandant de recommencer la saisie

Description des cas d'utilisation — Simulation en pause/en direct

Cas d'utilisation	Mettre le scénario sur pause/en direct
Système	Simudémie — Simulation
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Peut mettre la simulation sur pause pour effectuer des actions la tête tranquille
Préconditions	La simulation doit exister
Garantie de succès	L'utilisateur arrive à mettre sur pause la simulation sans rien et arrive à la redémarrer sans rien briser
Scénario principal	
Utilisateur	Système Simudémie
Lorsque l'utilisateur est dans la simulation, il peut cliquer sur le bouton rouge/vert pour arrêter ou continuer la simulation	Mets sur pause ou en direct la simulation
Scénario alternatif	
Aucun	

Description des cas d'utilisation — Avancer/Reculer Temps Simulation

Cas d'utilisation	Avancer ou reculer le temps de la simulation
Système	Simudémie — Simulation
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Désire avancer ou reculer la simulation Programmeur : Offre la possibilité d'avancer ou de reculer la simulation à l'aide d'une bande glissante
Préconditions	La simulation doit exister
Garantie de succès	La simulation a bel et bien avancé ou reculé dans le temps
Scénario principal	
Utilisateur	Système Simudémie
Lorsque l'utilisateur est dans la simulation en pause en cliquant sur le bouton vert.	Mets la simulation sur pause
Il glisse la barre de jours à gauche et à droite afin de reculer ou avancer, respectivement.	Affiche et génère les jours où la barre de jour atterrit
Glisse à gauche complètement, il retournera au départ de la simulation.	Affiche le jour de départ de la simulation
Scénario alternatif	
Aucun	

Description des cas d'utilisation — Sauter Temps

Cas d'utilisation	Sauter le temps de la simulation
Système	Simudémie — Simulation
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Désire sauter le temps de la simulation d'un certain nombre de jours, afin de voir les effets de ses mesures ou vaccins. Programmeur : Permet à l'utilisateur de sauter un nombre de jours désirés et observer les effets de ce saut rapidement
Préconditions	La simulation doit exister
Garantie de succès	La simulation a bel et bien sauté les jours demandés et généré les changements de ce saut temporel
Scénario principal	
Utilisateur	Système Simudémie
Lorsque l'utilisateur est dans la simulation en pause en cliquant sur le bouton vert.	Mets la simulation sur pause
L'utilisateur appuie sur le bouton de saut dans le temps	Affiche la fenêtre de configuration de saut temporel
Saisis le nombre de jours souhaité et applique le saut	Applique le saut et génère les changements de ce saut temporel
Scénario alternatif	
L'utilisateur saisit un nombre de jours trop élevé	Affiche un message d'erreur et demande de saisir un nombre moins élevé

Description des cas d'utilisation – Affiche Statistiques Pays/Région

Cas d'utilisation	Observer les statistiques en survolant le pays/région
Système	Simudémie — Simulation
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : Désire avoir de l'information sur une zone Programmeur : Offre la possibilité de consulter les statistiques de ces zones
Préconditions	La simulation doit être en cours
Garantie de succès	Les informations d'un pays/région sont bien affichées
Scénario principal	
Utilisateur	Système Simudémie
L'utilisateur survole avec le curseur sur un pays/région	Affiche les statistiques du pays/région survolé sur une fenêtre contextuelle
L'utilisateur clique sur le bouton des vues [bouton de l'œil] et prend la vue désirée	Change la vue des statistiques sur la carte et ajuste les taux et les couleurs en fonction de ce qui est demandé par l'utilisateur
Scénario alternatif	
Aucun	

Description des cas d'utilisation — Enregistrer Simulation

Cas d'utilisation	Enregistrer simulation
Acteur(s)	Utilisateur
Type	Secondaire
Description	L'utilisateur peut sauvegarder toutes les données relatives à la simulation pour la reprendre plus tard.

Description des cas d'utilisation – Affiche Statistiques Simulation

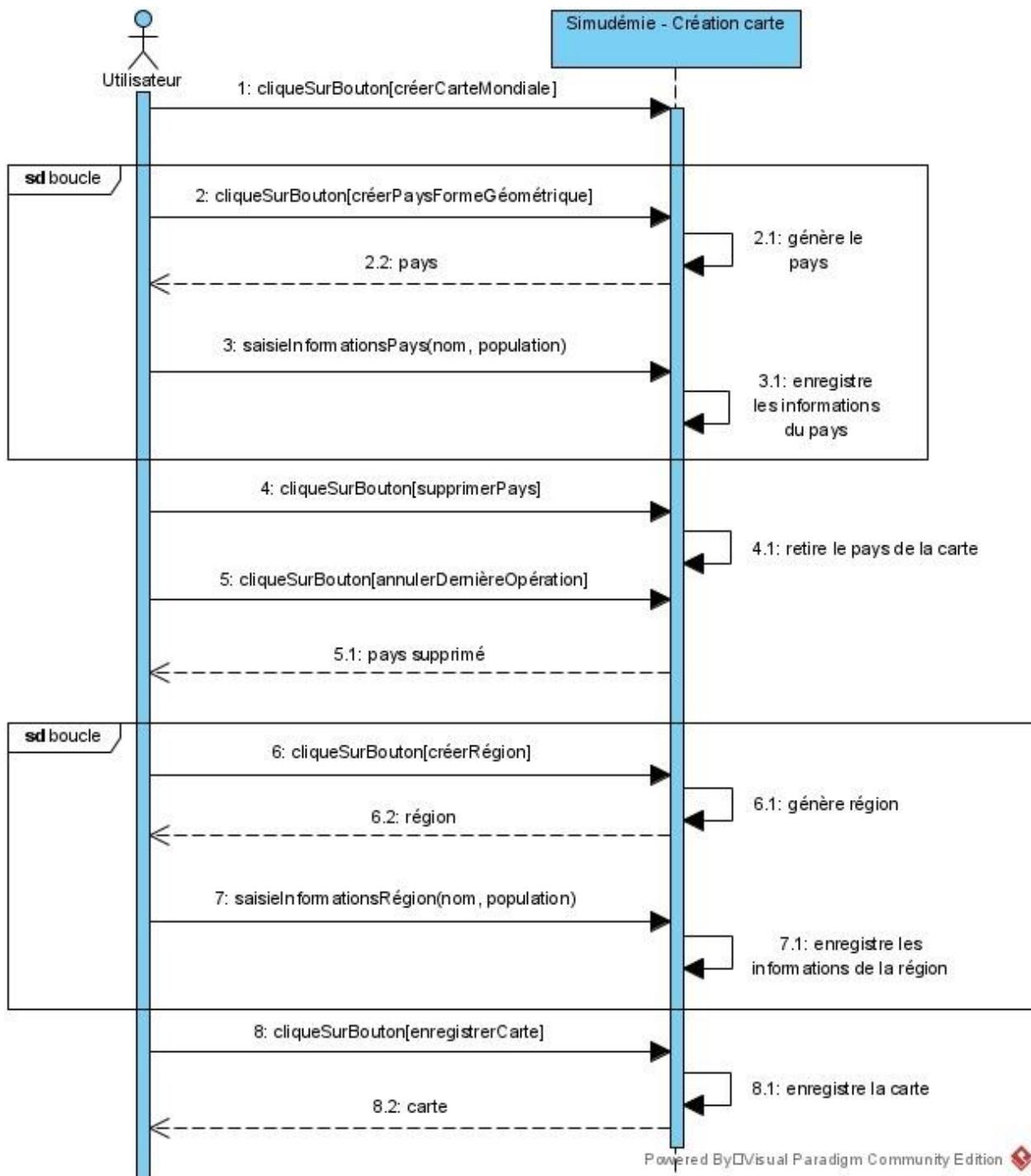
Cas d'utilisation	Afficher les statistiques et les données de la simulation
Système	Simudémie — Simulation
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : veut visionner les statistiques de la simulation Programmeur : offre une interface pour visionner les statistiques pour les jours voulus
Préconditions	La simulation doit en cours
Garantie de succès	L'interface de statistiques s'affiche bien et montre les bonnes données
Scénario principal	
Utilisateur	Système Simudémie
L'utilisateur clique sur l'icône statistique	Affiche la fenêtre des statistiques de la simulation
L'utilisateur saisit les informations désirant être affichées sur le graphique	Ajuste les informations saisies par l'utilisateur sur le graphique
Scénario alternatif	
Aucun	

Description des cas d'utilisation — Exporter Statistiques Simulation

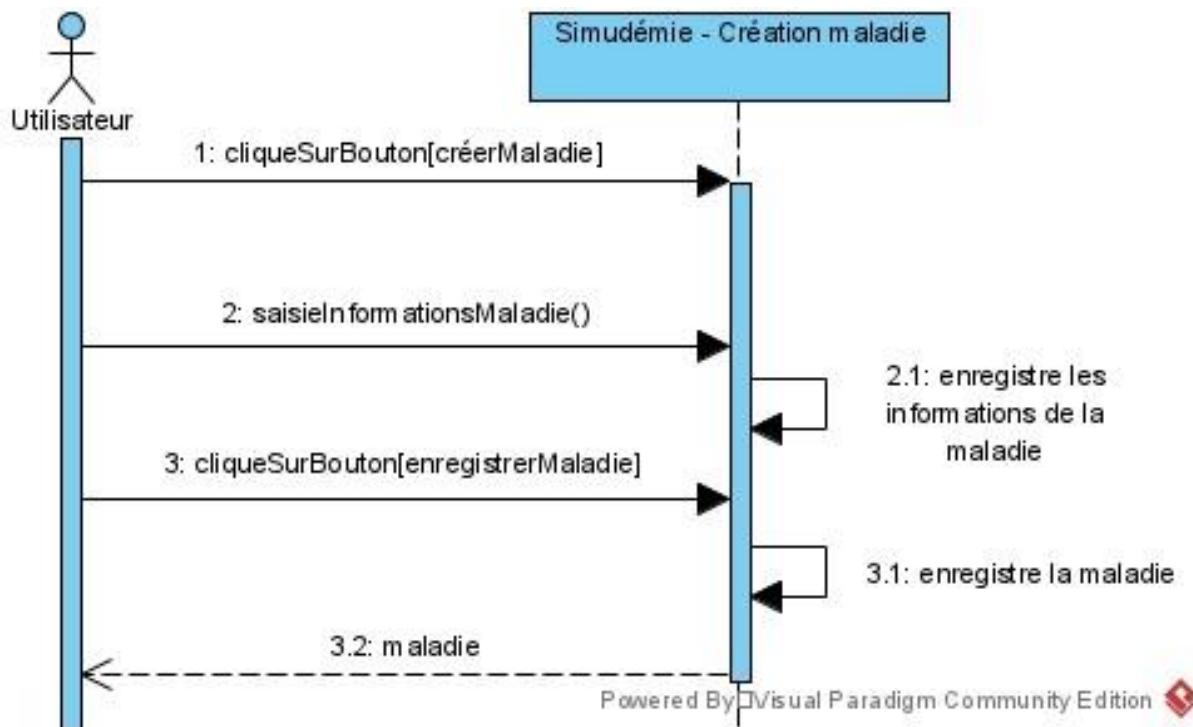
Cas d'utilisation	Exporter les statistiques et les données de la simulation
Système	Simudémie — Simulation
Acteur	Utilisateur
Parties prenantes et intérêts	Utilisateur : veut pouvoir exporter les statistiques de la simulation Programmeur : donne la possibilité de le faire avec un clic et une interface pour choisir où déposer le fichier
Préconditions	La simulation doit être en cours
Garantie de succès	Un fichier de statistiques est créé et déposé dans un lieu au choix de l'utilisateur
Scénario principal	
Utilisateur	Système Simudémie
L'utilisateur clique sur l'icône statistique	Affiche la fenêtre des statistiques de la simulation
L'utilisateur clique sur le bouton exporter	Ouvre la fenêtre de répertoire de fichiers
L'utilisateur choisit et confirme la destination d'export	Génère un fichier statistique à la destination désirée
Scénario alternatif	
Aucun	

Diagramme de séquence système – Livrable 1

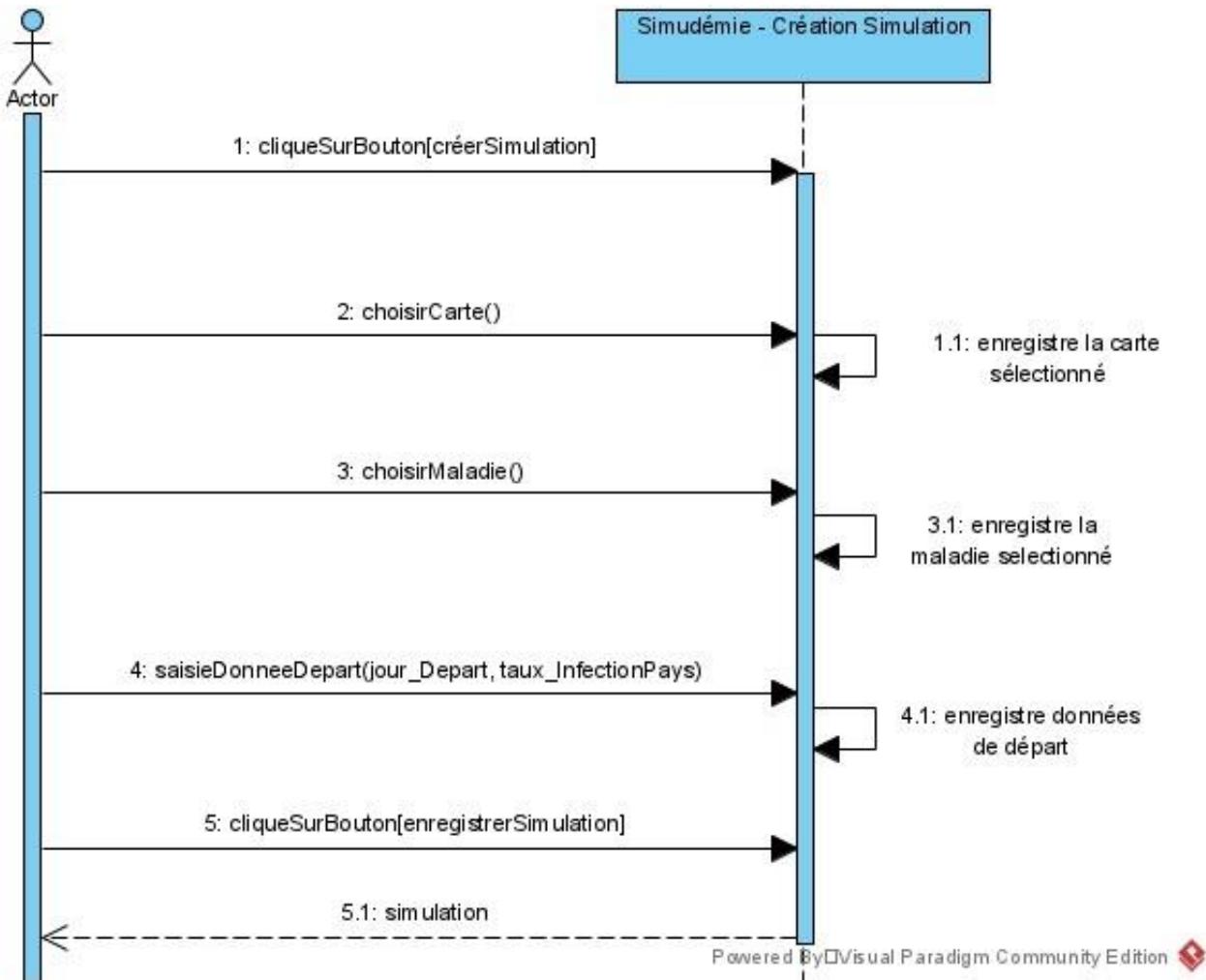
DSS - Création d'une carte mondiale



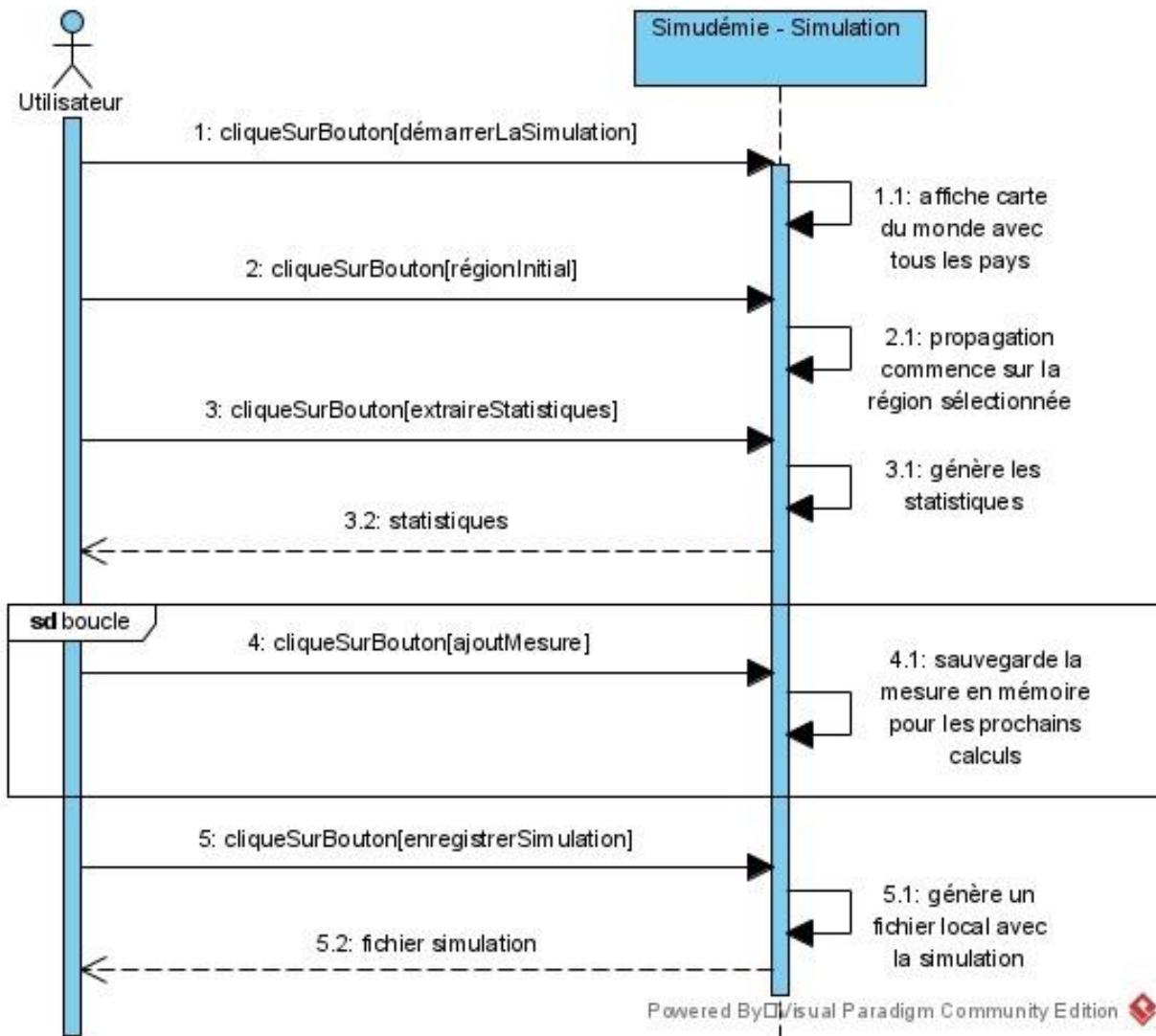
DSS - Création d'une maladie



DSS - Création d'un scénario



DSS – Simulation



Powered By Visual Paradigm Community Edition

Diagramme des cas d'utilisation – Livrable 3

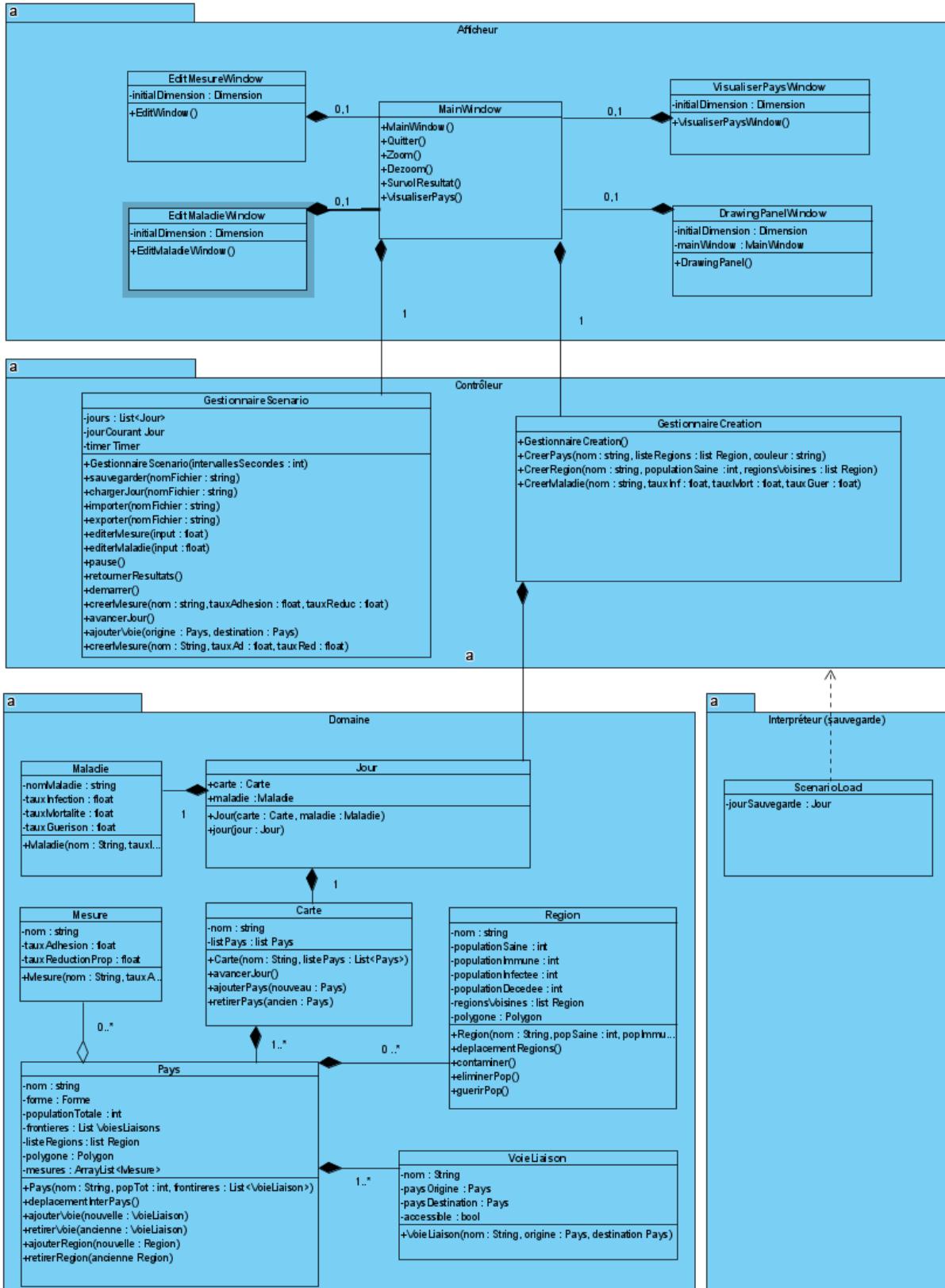
Au niveau du diagramme des cas d'utilisation, nous avons simplifié nos diagrammes. En effet, notre première ébauche contenait quatre différentes boîtes (équivalent à 4 systèmes). Nous avions opté pour ce format, car nous avions ajouté de multiples cas d'utilisations supplémentaires. Toutefois, il était évident que cela ne représentait en aucun cas la notation UML désirée par notre client.

En ajustant nos cas d'utilisation à ce que le client désire uniquement, nous avons été en mesure de construire un diagramme approprié au projet qui respecte la notation requise.



Modèle de conception

Diagramme de classe de conception – Livrable 2



Avant de détailler chacune des classes du diagramme de classe de conception, nous souhaitons mentionner que, par souci de lisibilité, nous avons omis tous les **gets** et les **sets** ainsi que les constructeurs avec paramètre(s). Nous avons simplement laissé le constructeur par défaut.

Dans le package afficheur, on retrouve toutes les classes de type fenêtre. **EditMesureWindow** et **EditMaladieWindow** sont les fenêtres qui servent, comme leur nom l'indique, à modifier les mesures et la maladie. La classe **VisualiserPaysWindow** permet de voir une version plus détaillée du pays accompagné des informations pertinentes. Les deux dernières classes sont les plus importantes, car il s'agit de la fenêtre principale (**MainWindow**) dans laquelle on peut suivre l'évolution de la maladie « en temps réel » et l'interface de création des pays (**DrawingPanelWindow**). Toutes les fenêtres communiquent avec les classes du domaine par l'entremise du contrôleur (séparé en deux contrôleurs pour faciliter la lisibilité et éviter une classe interminable en termes de fonctions). Nous avons choisi de ne pas inclure les méthodes présentes dans les fenêtres d'interface, car celles-ci sont très nombreuses et se retrouvent majoritairement sous la forme d'**event** (événement sur un clic ou un survol, par exemple).

La première classe du contrôleur est le **GestionnaireCreation** qui fait le pont entre les interfaces accessibles par l'utilisateur et le domaine. Dans cette classe, nous avons regroupé toutes les fonctions de « départ » ou de création (à l'exception des mesures qui peuvent être créées tout au long de la simulation). Ce gestionnaire ne devrait plus être utilisé lorsque la simulation a été démarrée par l'utilisateur.

Le second contrôleur est le **GestionnaireScenario** et celui-ci contient toutes les méthodes pouvant être complétées par le biais des interfaces. Cela inclut le démarrage du scénario, la mise sur pause, toutes les éditions de valeurs possibles, ainsi que les sauvegardes et chargements. Autrement dit, cette classe gère tout ce qui peut se faire suivant le démarrage. Comme il sera mentionné dans la section suivante du rapport, c'est ici que se déroulera l'avancement journalier. Nous avons inclus un attribut *timer* qui va exécuter l'avancement de journée après chaque intervalle de secondes «x». Au même moment, la méthode chargerJour() va être appelée pour ajouter l'objet de type Jour à notre liste de jours. C'est ainsi que nous pourrons gérer les jours précédents (revenir plusieurs jours en arrière).

Dans le package interpréteur, nous avons mis la classe **ScenarioLoad**. Il nous manque encore certaines informations quant au bon fonctionnement de celle-ci, mais c'est la classe qui sera responsable des sauvegardes/exportations/importations/chargements. Nous n'avons pas encore effectué des tests, mais nous devrions « sérialiser » un scénario pour être en mesure de le récupérer ultérieurement.

L'objet pays est au centre de tout ce qui tourne dans l'application. Chaque pays est constitué de voies de liaisons qui les relient à d'autres pays, d'une liste de région qui le constitue et finalement

d'une forme de type Polygon qui pourra être sauvegardé et chargé plus facilement sur l'interface. Le pays dispose de quelques méthodes. La première (*DeplacementInterPays()*) permet de déplacer une partie de sa population de pays en pays par le biais des voies de liaison. Les suivantes serviront principalement à la gestion des différentes listes du pays. Cela inclut ajouter et retirer des éléments à ces listes. Les **VoieLiaison** sont les objets qui connectent les pays et ces derniers ont un nom, un pays d'origine et un pays de destination et une valeur booléenne pour déterminer si cette voie est accessible ou si elle a été fermée.

La classe **Region** est la classe qui va rassembler toutes les informations les plus importantes, car c'est dans celle-ci que va se faire le traitement statistique. Comme dans la classe **Pays**, on retrouve une méthode qui déplace une portion de la population entre les régions (d'où la présence d'un attribut liste *regionsVoisines*). Toutes les autres méthodes de la classe région, servent à appliquer les différents taux dictés par les mesures ainsi que la maladie. Ainsi, la fonction contaminée va faire appel au taux d'infection de la maladie et s'appliquer sur la population saine et transférer les personnes infectées dans la catégorie population infectée. De la même manière, la fonction *éliminerPop*, s'applique sur la population infectée et déplace les gens décédés dans la catégorie population décédée, etc. La classe Région possède aussi un attribut de type Polygon (comme la classe Pays).

Les deux classes, **Mesure** et **Maladie**, possèdent des attributs multiplicateurs qui pourront être employés sur la population des différentes régions. Outre leurs attributs (et les *gets/sets*), ces deux classes ne font rien d'autre pendant la simulation, à moins que l'utilisateur effectue des changements.

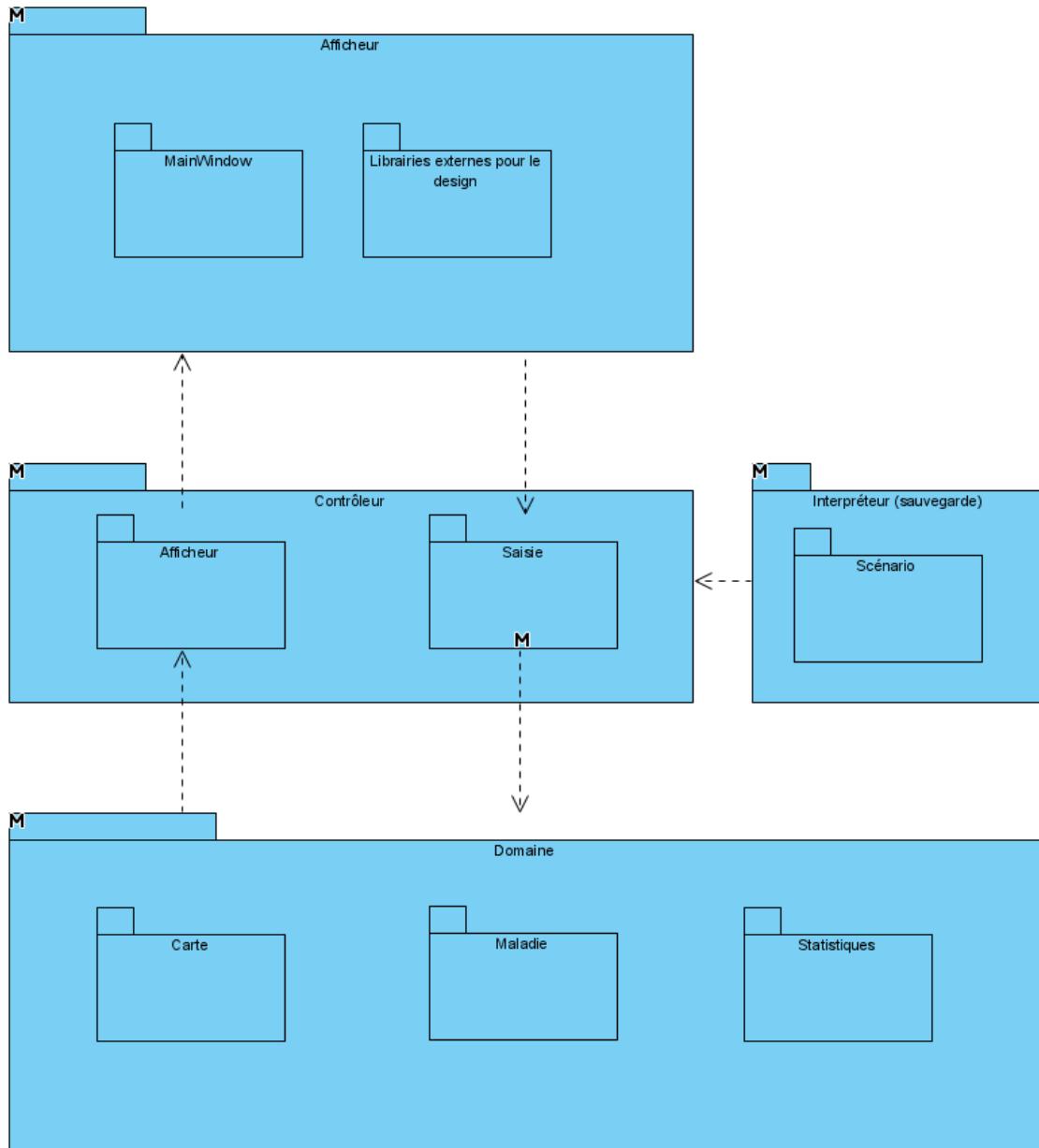
La classe **Carte** comporte tous les pays et les classes dont Pays hérite. Cette classe sert principalement de structure pour regrouper les différents éléments qui doivent être présents sur la carte. Ainsi, les seules méthodes présentes servent à faire des ajouts ou retraits de pays à la liste.

Notre dernière classe est **Jour**. C'est probablement la classe la plus importante du package Domaine, car elle regroupe tous ses éléments. Elle englobe donc la carte et ses pays, mais aussi la liste des mesures mises en place par l'utilisateur et la maladie choisie. Cela nous permet d'avoir une capture d'écran de la simulation au jour désiré. Tout comme la classe Carte, la classe Jour agit comme une structure qui sert de conteneur.

Nous croyons avoir fait le tour des classes et méthodes que nous allons employer, mais il est fort probable que cela soit appelé à changer. Si par exemple, le client souhaitait que l'on produise un rapport Excel, il faudrait créer une nouvelle classe Excel pour permettre la création d'objets Excel et ainsi faire le traitement souhaité. Il est évidemment possible qu'un aspect de l'application nous ait échappé et qu'il faille le rectifier ou alors que l'ajout d'un attribut ou d'une méthode permette d'alléger le code ou le simplifier.

Architecture logique – Livrable 2

Comme on pouvait le constater sur le diagramme de classe de conception, les classes se retrouvent à l'intérieur de package qui représente des systèmes globaux. Le plus important dans l'architecture que nous proposons est l'utilisation du contrôleur. Le contrôleur sert de pont entre les classes du domaine et l'interface de l'utilisateur. En effet, il serait problématique que l'utilisateur puisse avoir un accès direct aux classes à travers la vue.



Le premier package est l'afficheur (qu'on pourrait appeler interface utilisateur aussi) et contient toutes les fenêtres auxquels l'utilisateur a accès. Le second package est le contrôleur qui gère les requêtes de saisies et d'affichage du bas vers le haut et vice-versa. Comme mentionné plus haut, il sert d'intermédiaire ou de pont avec le package domaine.

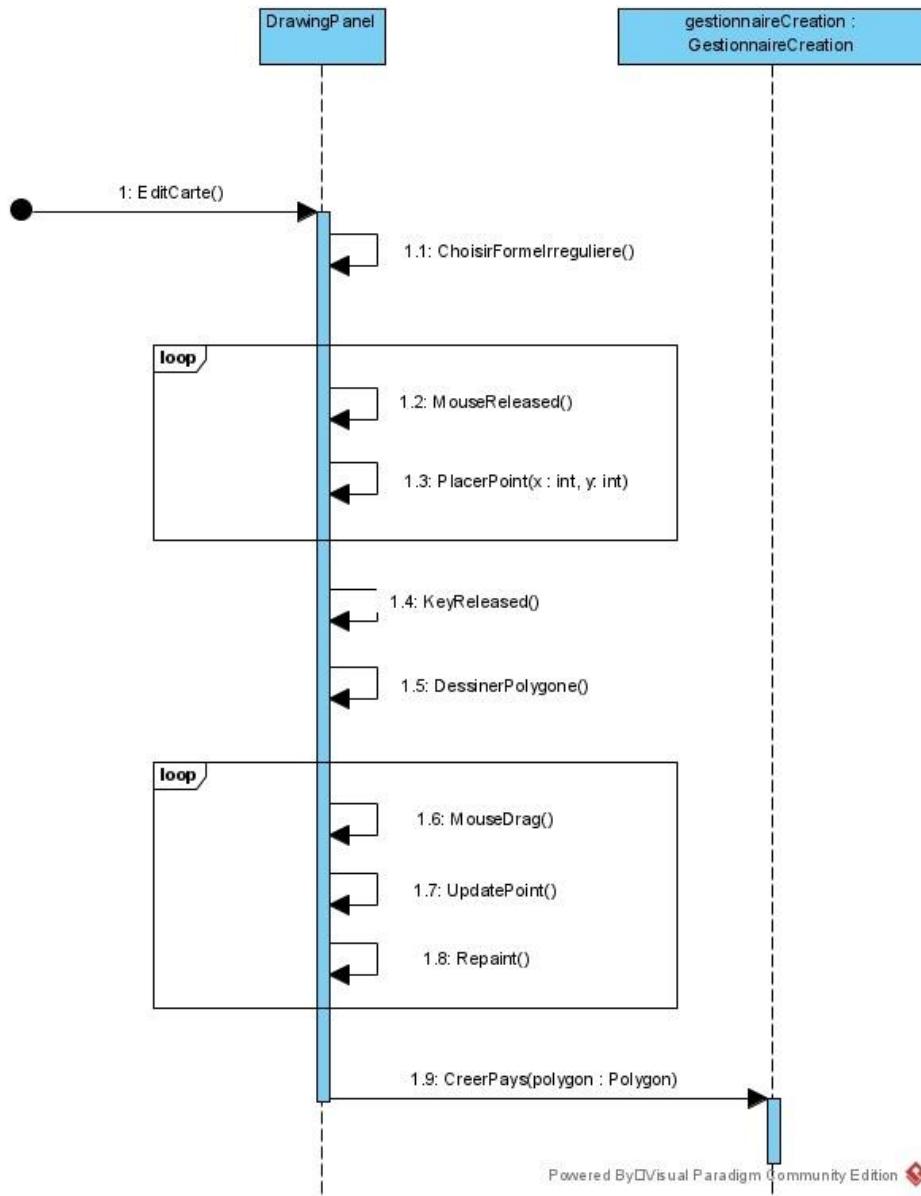
Les derniers packages sont le domaine qui contient le « moteur » de notre application et toutes les classes qui interagissent avec la carte, la maladie et les mesures. Finalement, nous avons ajouté le package interpréteur pour représenter les fichiers *serialized* qui nous serviront de fichier de sauvegarde pour les importations et exportations. À noter que l'interpréteur devra passer par le package contrôleur comme n'importe quelle autre commande.

Diagramme de séquence de conception – Livrable 2

Création pays de forme irrégulière

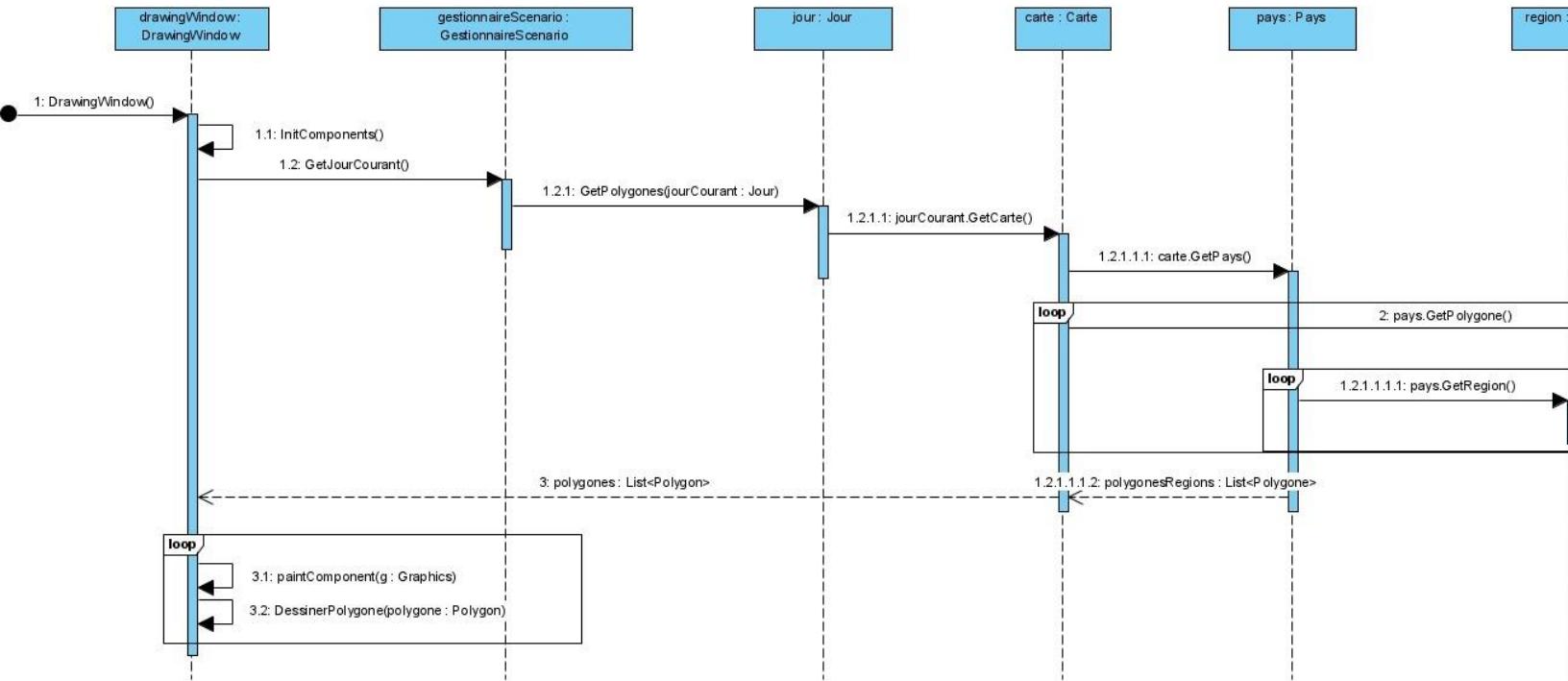
La création d'un pays de forme irrégulière se représente par la communication de deux classes, le *DrawingPanel* et le *GestionnaireCreation*. La création commence par l'appel de la fonction *EditCarte()* qui change le mode de vue pour le mode de création de cartes.

Par la suite, la



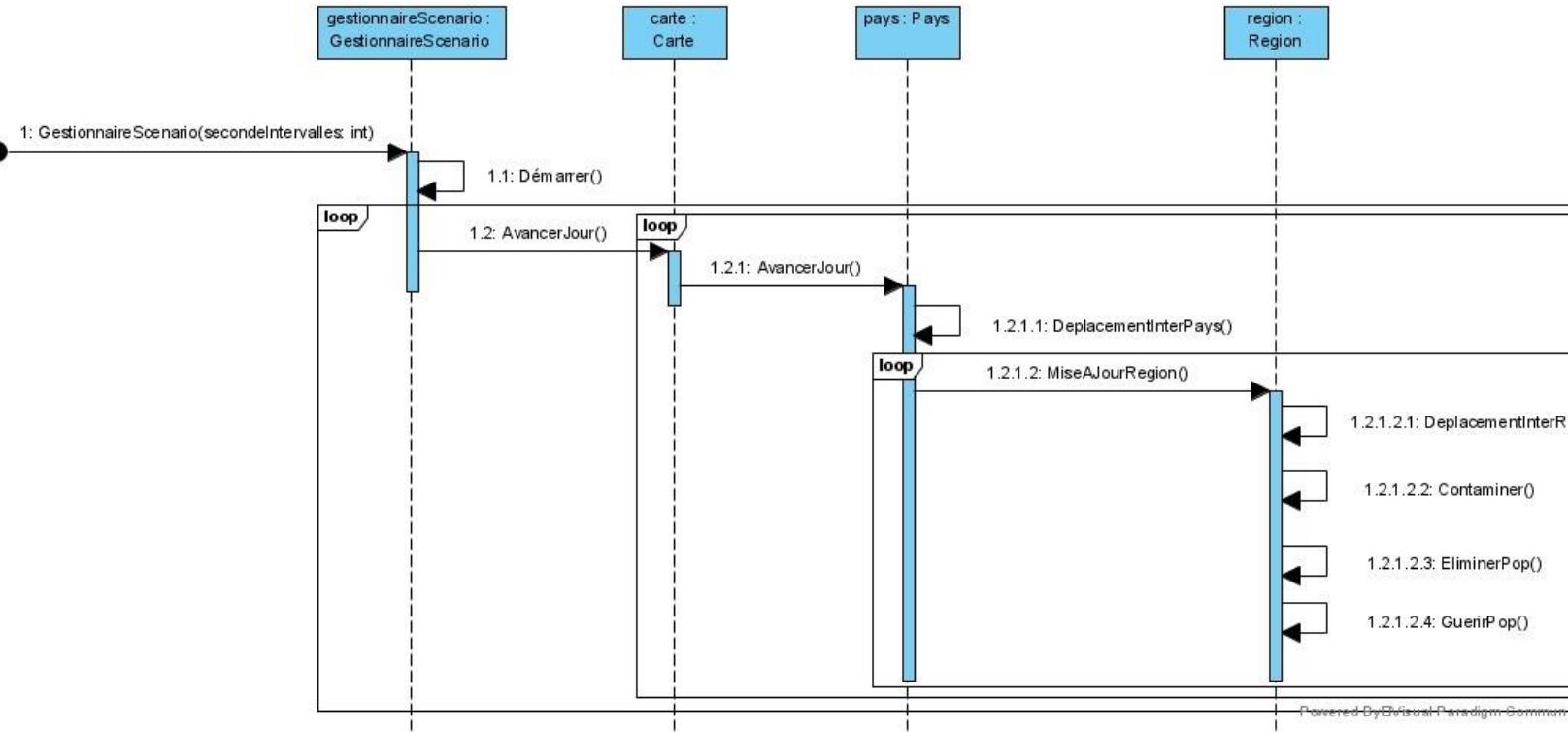
fonction *ChoisirFormeIrreguliere()* dit au programme que les prochaines interactions avec la fenêtre seront pour la création d'une forme irrégulière. Ensuite, l'utilisateur peut, à répétition, faire un clic dans la fenêtre qui ajoute un point dans le polygone en cours de création. Lorsque l'utilisateur appuie sur une touche de clavier précise comme la touche « espace », la fonction *DessinerPolygone()* est ensuite appelée pour faire afficher le polygone à l'écran.

Ensuite, la deuxième boucle contenant les fonctions *MouseDrag()*, *UpdatePoint()* et *Repaint()* se chargent d'ajuster le point que l'utilisateur est en train de déplacer. Pour finir, la fonction *CreerPays(Pays pays)* crée le pays selon le polygone en paramètre.

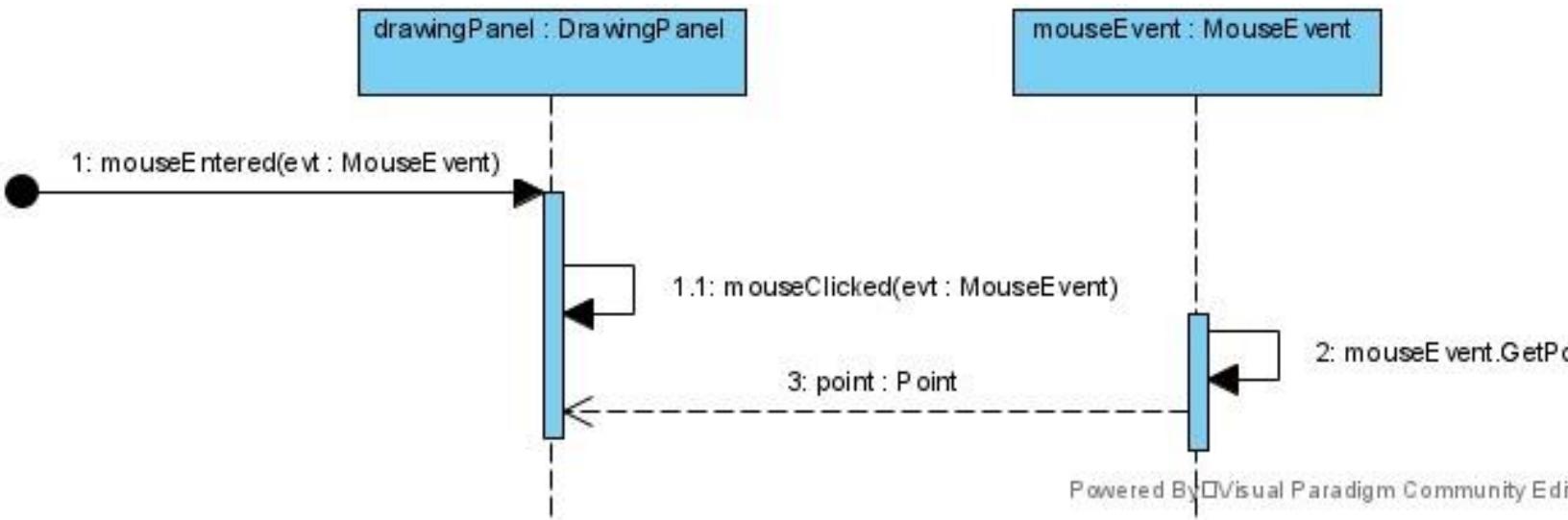
L'affichage de la vue du monde

Le diagramme commence lorsque le système fait appel au `DrawingWindow()`. Il initialise avec `InitComponents()` qui appelle par la suite `GetJourCourant()`. Par la suite on appelle `GetPolygones()` sur ce jour courant, ce qui ira chercher toutes les informations relatives à la représentation de la carte mondiale du jour courant. Lorsque que l'algorithme est allé chercher la carte du jour courant avec `GetCarte()`, il pourra bouclé sur `GetPolygone()` des pays puis le retourner au `DrawingWindow()` sous forme de liste `List<Polygon>`. Par la suite, le `DrawingWindow()` dessine chacun des polygones de la liste à l'aide de ses méthodes `paintComponent()` & `DessinerPolygone()`.

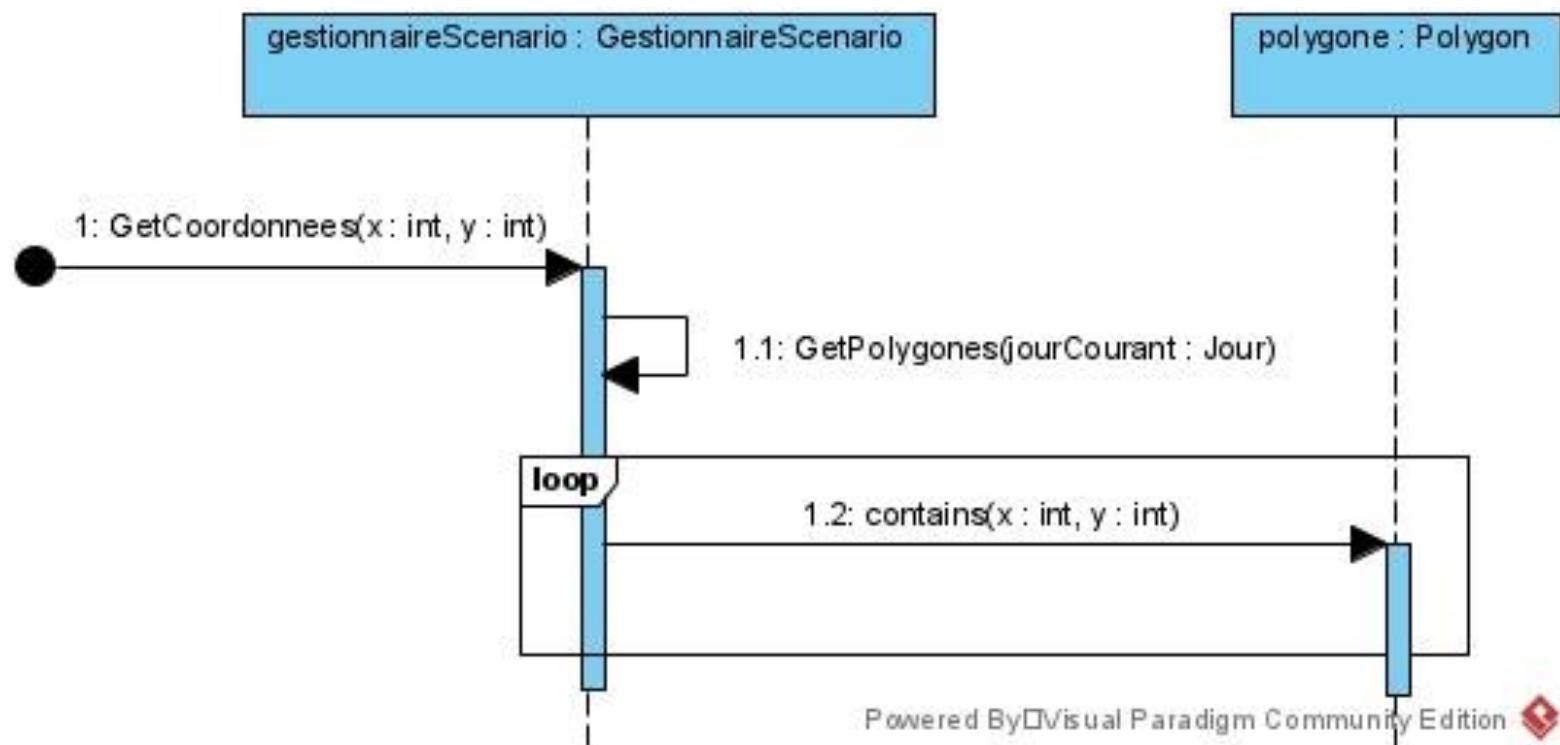
Fonctionnement d'un pas dans le temps dans la simulation mettant à jour l'information



Le diagramme commence avec l'appel au constructeur du **GestionnaireScenario**. Celui-ci appelle ensuite *Démarrer()* qui commence l'appel en boucle de la méthode *AvancerJour()* sur la carte courante. Cette carte va ensuite également appeler *AvancerJour()* sur tous les pays de la carte. Ces pays vont par la suite calculer leurs déplacements avec l'appel *DeplacementInterPays()*. Chaque région de tous les pays exécute la fonction *MiseAJourRegion()*. À leur tour, les régions se chargent de déplacer, contaminer, éliminer et guérir la population avec l'appel des fonctions respectives.

Diffusion de l'information concernant un pays/région sous la souris

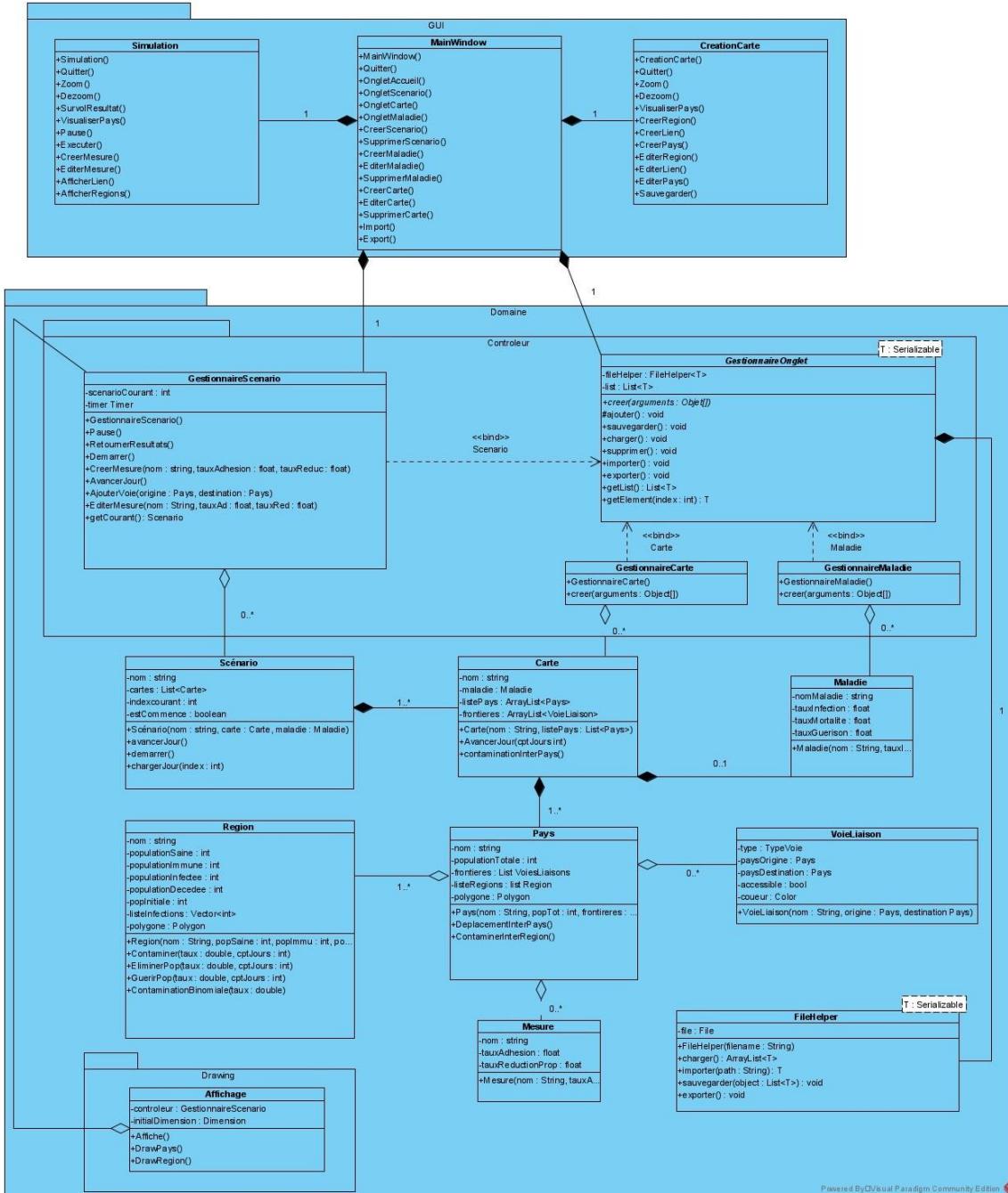
Le diagramme commence lorsque la souris entre sur le panneau de dessin **DrawingPanel**. À tout instant, ce panneau est à l'écoute d'évènement dont celui qui nous intéresse le plus, *mouseClicked(MouseEvent evt)*. La méthode *GetPoint()* est liée à cet évènement et retourne le point avec les coordonnées x et y de la souris au moment du clic.

Diffusion de l'information concernant un pays/région sous la souris (suite)

L'obtention d'information concernant un pays ou une région qui se trouve sous la souris commence par l'envoi des coordonnées x et y de la souris au contrôleur **GestionnaireScenario**. Ce contrôleur récupère ensuite tous les pays et régions. Une fois le tout récupéré, on boucle sur toutes les formes récupérées afin de savoir si le point se trouve dans la forme à l'aide de `Polygon.contains(int x, int y)`.

Diagramme de classe de conception – Livrable 3

Notre précédent diagramme de classe de conception était erroné sur plusieurs points. Premièrement, nous avions des erreurs au niveau de l'architecture même du programme. Nos contrôleurs ne faisaient pas partie du domaine et nous avions mis un package « interpréteur » pour représenter les imports/exports de fichiers, ce que nous avons changé.



Au niveau du domaine, nous avons effectué quelques modifications, principalement pour gérer les statistiques reliées à l'infection. Par exemple, nous avons ajouté un compteur de jour pour que les infections/guérisons apparaissent après 2 semaines (inspirés du Coronavirus). Nous avons aussi ajouté un vecteur de personnes infectées pour contaminer les personnes en «différé» (au 14e jour, nous guérissons ou éliminons les infectés du premier jour). Nous nous sommes aussi, à la demande du client, débarrassés de la classe jour et utilisons uniquement une liste de carte dans Scenario pour sauvegarder nos jours écoulés.

Comme nous avons décidé de ne pas faire qu'une fenêtre pour l'application complète, nous avons opté pour un design de contrôleur plus adéquat à nos besoins. En effet, chaque onglet aura son contrôleur et son type relié ce qui facilitera le processus de manipulations des données et de la sauvegarde de celles-ci. En effet, comme chaque onglet découle du GestionnaireOnglet, ils partagent des fonctions communes tels que l'exportation et l'importation, l'ajout, la sauvegarde et la suppression. La nouvelle classe FileHelper fait partie du GestionnaireOnglet et est de type générique pour gérer uniformément la sérialisation de chaque modèle (Maladie, Carte, Scénario).