

Introduction

Dans le cadre du cursus technicien, nous sommes amenés à réaliser un travail de diplôme qui dure du 19 avril aux 11 juin. Durant cette période, plusieurs évaluations intermédiaires sont prévues, la première se situant 10 jours après le début du travail de diplôme. La seconde évaluation est prévue le 17 mai, la troisième le 31 mai et la dernière est le rendu final le 11 juin et dure 9 jours contrairement aux autres sprints qui durent 10 jours.

Il est nécessaire de réaliser un poster pour ce travail ainsi que de remplir un journal de bord comprenant nos activités et nos réflexions.

Le but de mon projet est de simuler une propagation du covid dans une simulation dite individu-centrée. Ce qui signifie que des individus sont simulés et agissent selon leur planning. S'ils sont infectés c'est en cas de contact avec une autre personne infectée. Les données utilisées sont des données officielles et sont maintenues à jour aussi souvent que possible.

WPF

Le programme WPF est le cœur de l'application, il réunit toutes les sections du projet et les gère.

Simulation

La simulation génère tous les objets nécessaires au fonctionnement de celle-ci. Ses paramètres peuvent être modifiés depuis l'UI. Ses paramètres concernant le virus sont écrits dans un fichier XML. La simulation gère aussi la temporalité permettant la propagation et les déplacements.

UI

L'interface utilisateur est gérée par les grilles WPF permettant un affichage responsive. Elle permet à l'utilisateur de modifier les paramètres de la simulation ainsi que les paramètres d'affichage des graphiques.

XML

Les paramètres jugés fixes du virus sont stockés dans un fichier XML.

Graphiques

Les graphiques sont créés par la librairie liveCharts qui permet l'affichage de nombreux types de graphiques ainsi qu'un grand contrôle sur ceux-ci. Les données sont mises à jour en temps réel et des animations sont intégrées à la librairie.

Unity

Le programme Unity s'occupe de gérer l'interface graphique qui comprend les bâtiments, véhicules et individus. L'interface est animée en fonction de la temporalité de l'application WPF. La simulation et l'interface graphique avancent donc ensemble. La communication s'effectue à travers un pipeline nommé. Les données de la simulation sont envoyées par celui-ci. Le programme Unity est intégré directement dans le projet WPF.

Résumé

Covid propagation est une application permettant de visualiser l'évolution du covid au sein d'un environnement peuplé d'individus uniques. La visualisation se fait à l'aide de graphiques (Colonne, courbe, circulaire) et de données telles que le nombre d'infecté actuellement ou le nombre de rétablissement ainsi qu'une interface graphique permettant la visualisation des individus, des lieux, des véhicules et de leurs déplacements.

L'utilisateur a une certaine liberté dans les paramètres de la simulation comme pour la création de la population en choisissant leur nombre et l'âge moyen.

Des mesures contre le virus peuvent être prises, limitant ou stoppant sa propagation.

Abstract

9. Cahier des charges

9.1. Titre

Covid propagation

9.2. Fonctionnalités

- Simulation
 - Population
 - Mesures
 - Hôpitaux
 - individus
 - Patient à risque
 - Âge
 - Décès dû au virus
 - Famille
 - Cercle d'amis
 - "Vie" *Calendrier*
 - Virus
 - propagation
 - effets sur les individus
 - De "Aucun"
 - À "Grave"
 - Hôpitaux
 - Places limitées
 - Mesure de sécurités
 - Port du masque
 - Quarantaine
 - Confinement global
 - Distanciation
- Graphiques

- Informations sur la population
 - Décès
 - Rétablissements
 - Infecté
 - Sains
- Informations sur le virus
 - Dangersité

9.3. Matériel et logiciels

- Pc techniciens
- Visual studio 2019
- Une connexion internet
- Github

9.4. Prérequis

- C#
- Visual studio 2019

9.5. Descriptif complet du projet

9.5.1. Méthodologie

Scrum

9.5.2. Description de l'application

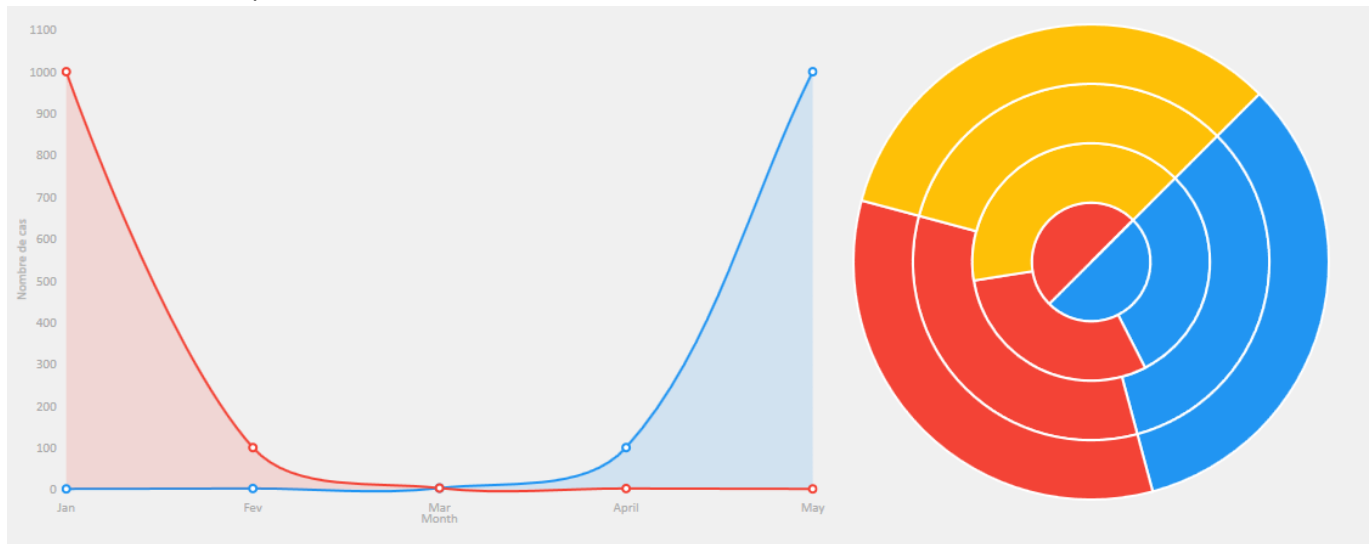
Simuler un grand nombre de personnes possédant toutes des variables différentes (âge, résistance immunitaire, etc...), y introduire le virus et observer sa propagation. Il est possible d'affecter des mesures de sécurité, tel que le port du masque ou la distanciation pour observer la possible réduction de la propagation. L'affichage permet de voir en temps réel la propagation du virus et permet de visualiser chaque individu distinctement au besoin. Des graphiques sont aussi présents pour avoir une idée en chiffres de ce que signifie l'affichage.

9.5.2.1. Graphique

Les données des graphiques sont choisies par l'utilisateur et donc personnalisable. Plusieurs graphiques peuvent être affichés en même temps. Leur position est définie par l'utilisateur au sein de la page de l'application.

L'interface graphique est fournie par [LiveChart](#). Les données sont directement fournies par l'application ainsi que les échelles de grandeurs qui sont ajustées automatiquement. Les graphiques à courbes et en forme

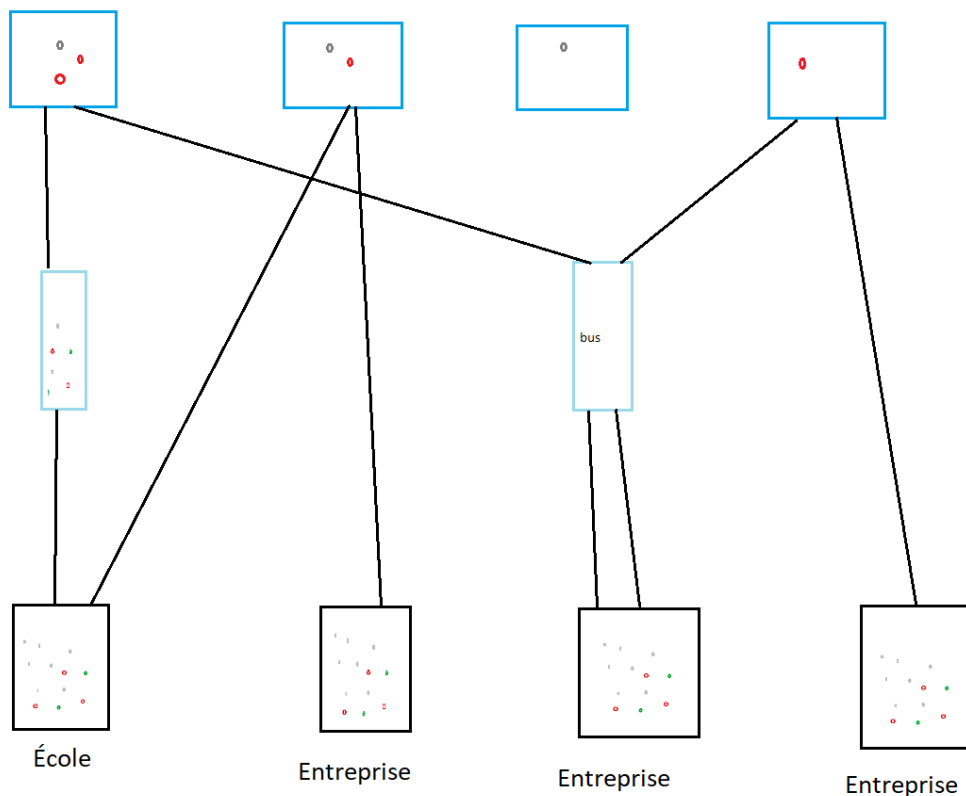
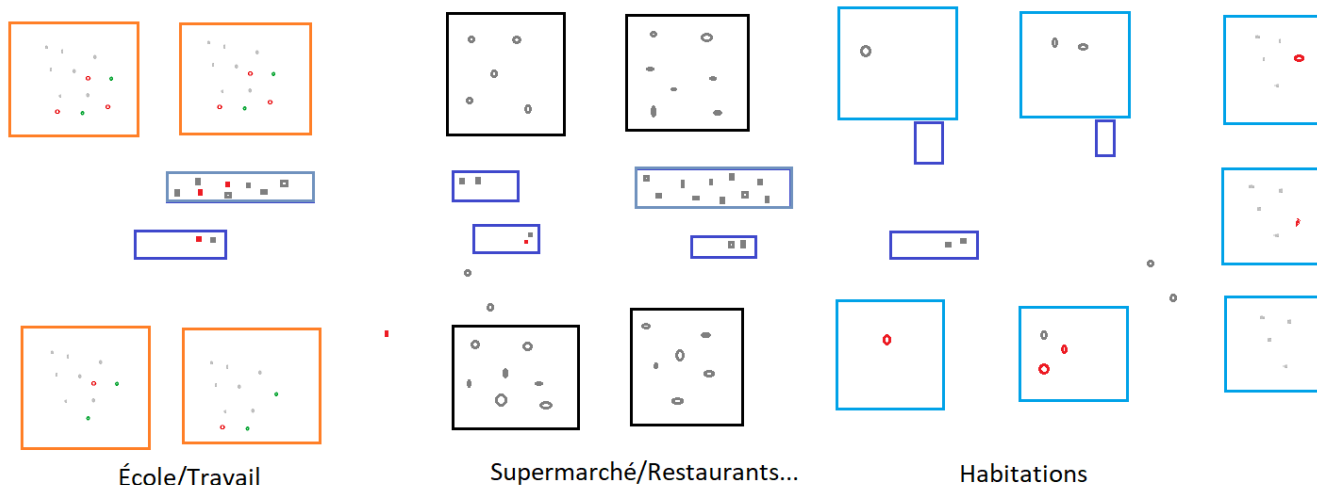
camembert sont disponibles.



9.5.2.2. Interface graphique

En plus des graphiques, une interface graphique affichant les individus ainsi que leur lieu de travail, habitation et déplacement est disponible. Elle permet d'avoir une visualisation plus naturelle de la situation. Elle est très simple car simuler une ville est une tâche trop complexe et longue pour être ajoutée au projet. Il s'agit donc d'une aide visuel simple de la simulation. Il n'y a donc pas de routes ou autres éléments complexe

similaires. Voici deux exemples d'interface graphique :



9.5.2.3. propagation

La propagation se fait à l'aide de calcul et de différentes variables. 1000 m² contenant 10 individus à l'intérieur aura de faibles chances de transmettre le virus. Le même nombre de personnes dans un espace clos de 10 m² aura des résultats totalement différents.

La température est prise en compte ainsi que les mesures telles que le masque. Le masque réduit les chances de transmettre le virus. La température, elle, fait varier la durée de vie du virus à l'extérieur d'un hôte. La complexité de ce type de calcul étant d'une difficulté largement supérieure aux compétences acquises en tant que technicien, je me base sur cette fiche Excel réalisée par des professionnels. Elle est très bien documentée et sourcée.

[Fiche excel](#)**9.5.2.4. Population**

La population est constituée d'objets C# généré partiellement, aléatoirement en fonction des paramètres de la simulation. Ils informent la simulation en cas de changement d'état (sain, infecté, etc...). Des itérations sont faites dans la simulation pour calculer si un individu est infecté ou non durant le temps écoulé. Il a un planning simple à suivre dans sa journée qui peut être constituée de par exemple :

- Être dans son habitation
- Prendre le bus
- Travailler
- Prendre le bus
- Faire les courses dans un supermarché
- Prendre le bus
- Et finalement rentrer chez soi

Ce planning est différent en fonction des individus même si vaguement le même. Durant sa journée, il croisera d'autres individus et à chaque itération, il aura des chances d'être infecté si des personnes aux alentours le sont. En fonction du lieu, il rencontrera des personnes différentes, parfois les mêmes comme dans son travail où ses collègues sont fixes. Dans le bus, des variations seront possibles. Son cercle d'amis ainsi que sa famille, lorsqu'il se trouve dans son habitation, seront les individus risquant de le contaminer.

9.5.2.4.1. Temporalité

Le quotidien des individus est défini par la simulation lors de leur création. Elles peuvent évoluer avec l'âge des individus.

Une itération est équivalente à ~30min dans la simulation. À chaque itération, chaque individu calcul ses chances d'attraper le virus en fonction de son environnement et des mesures prises. Elle permet aussi à un individu d'évoluer dans son quotidien en passant d'une tâche à une autre par exemple. Leur permettant aussi de changer de lieu et tous les événements liés à l'agenda des individus ainsi que la propagation du virus. La "durée" de la simulation est définie par l'utilisateur et peut donc durer plusieurs jours.

9.5.2.4.2. Individus

Les individus possèdent différents paramètres qui vont modifier leur quotidien ainsi que leur résistance au virus. La valeur la plus essentielle est l'âge de ces personnes. L'âge permet de contribuer à la modification de la résistance au virus. Il modifie aussi le quotidien en définissant si la personne va travailler, va à l'école, est libre de faire ce qu'il souhaite ou rien si trop jeune. L'âge évolue avec le temps de la simulation.

Chaque individu a un entourage qui peut le contaminer. Il possède un cercle d'amis avec lequel il peut y avoir des contacts à domicile, et avec lequel il y aura des contacts en extérieur. Il a aussi une famille avec qui les contacts se font majoritairement à domicile même s'il peut y avoir des déplacements groupés. Par exemple déposer des enfants à l'école, aller au restaurant en famille. Finalement, il a des collègues/camarades qui sont des contacts qui se trouve dans les écoles ou lieu de travail et qui sont ne définit pas ceux-ci.

Les moyens de transport des individus sont choisis par la simulation en fonction des paramètres de celle-ci. Un individu possédant une voiture aura beaucoup moins de risque de propager le virus qu'en prenant les

transports publics. Il est cependant possible que d'autres personnes du cercle familial ou du cercle d'amis utilisent le même véhicule. De ce fait, il n'est pas forcément 100% sécurisé. Les transports publics eux ont des risques élevés, car beaucoup de monde se situe dans le même véhicule de taille moyenne. En plus de cela, les individus sont en contact avec des étrangers qui peuvent varier en fonction des jours augmentant encore plus les chances de contagion.

La résistance au virus des individus définit si la personne a des symptômes en cas d'infection, si elle est asymptomatique, ou si elle a besoin de soins. Ce paramètre est défini par pourcentage. De 100% à 90% de résistance, l'individu est asymptomatique. De 90% à 50% de résistance, l'individu a des symptômes tels que la toux. De 50% à 10% de résistance, la personne est hospitalisée. Et finalement, à moins de 10%, l'individu est hospitalisé et risque la mort.

- Plus ce paramètre est haut moins les effets du virus sont présents
- 90-100 => asymptomatiques
- 90-50 => symptômes normaux
- 50< => hospitalisations
- 10< => décès

Chaque individu créé commence avec une valeur entre 80 et 100. Sachant qu'environ 5% de ces individus ont plus de 90 de résistance. Des maladies peuvent entrer en compte et baisser la résistance naturelle. Plus l'âge est élevé, plus l'individu sera impacté par un grand nombre de maladies et celles-ci seront plus dangereuses.

Les maladies sont inspirées de maladie réelle impactant l'effet du covid. Cependant, dans la simulation, elle n'affecte que la résistance au virus. Ces maladies apparaissent de façons aléatoires et plus fréquemment sur les individus dont l'âge est élevé. Elles ne se propagent pas. Elles sont en partie assignées au départ par la simulation puis apparaissent avec le temps. Elles réduisent la résistance au covid de 1% à 20% en fonction de la maladie et de l'âge de la personne.

9.5.2.4.3. Hopitaux / écoles / entreprise

Ces différents lieux fonctionnent de façon similaire. Ils ont tous des individus en leur sein qui peuvent se transmettre le virus. Ils ont des tailles différentes en fonction du nombre de personnes pouvant être à l'intérieur.

Les hôpitaux fonctionnent légèrement différemment. Ils ont des patients ainsi que des membres du staff de l'hôpital. Il y a donc des différences de mesures et quantités. Les patients sont là de manière temporaire en fonction du nombre de personnes attrapant le covid.

Les écoles ont une situation similaire en ayant des élèves ainsi que des profs qui ont des mesures et quantités différentes.

Les entreprises fonctionnent en groupe d'individus, similaire aux classes des écoles mais sans personnel ayant des mesures différentes des autres.

9.6. Protocols de tests

Ce projet étant en c#, je vais utiliser les tests unitaires intégrés dans visual studio.

Les tests unitaires ne garantissant pas qu'il n'y ait aucun bug dans l'application, je vais créer des scénarios que je testerai avant et après chaque implémentation de fonctionnalités. Ces scénarios auront pour but de couvrir

un maximum de possibilités pour éviter l'apparition de bug dû à une modification du code ou l'ajout d'une fonctionnalité. Ils permettent aussi de trouver d'éventuels des problèmes d'ergonomie en me plongeant à la place d'un utilisateur.

Maquettes

UI

Page Simulation

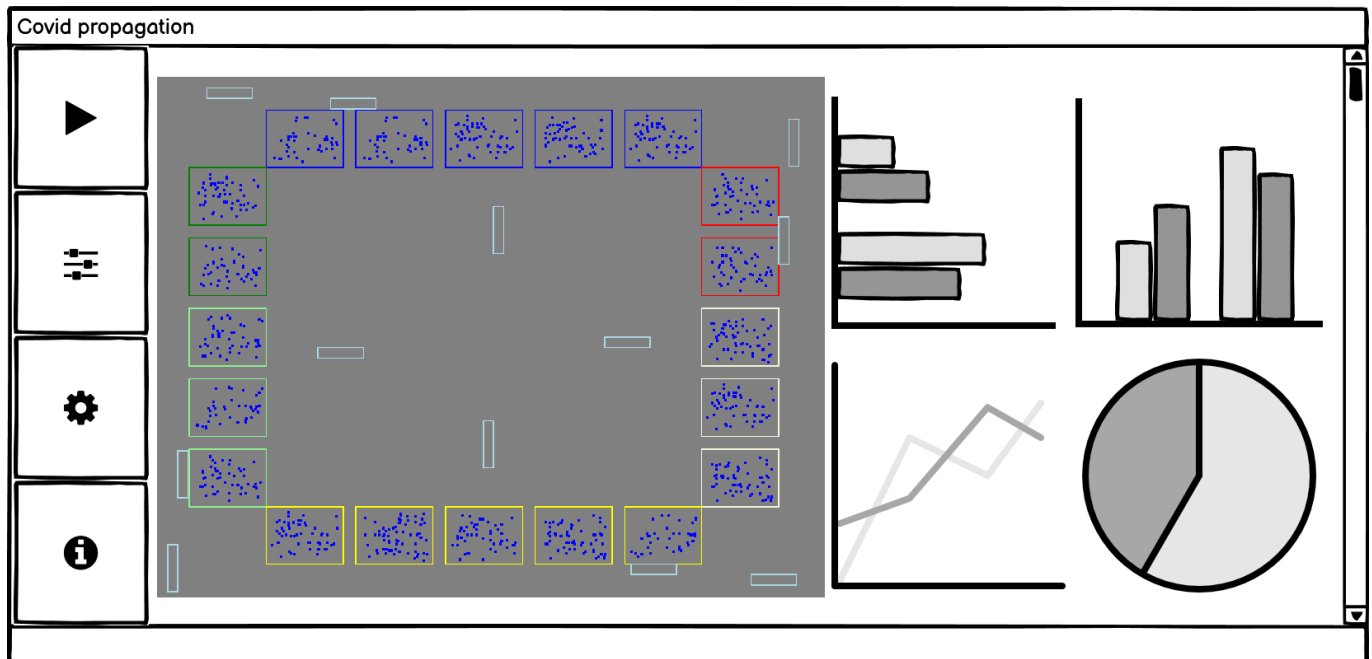


Figure 2: Maquette page de simulation

La page simulation permet la visualisation de la simulation via les graphiques, l'interface graphique ou des données brutes.

Page Paramètres graphiques

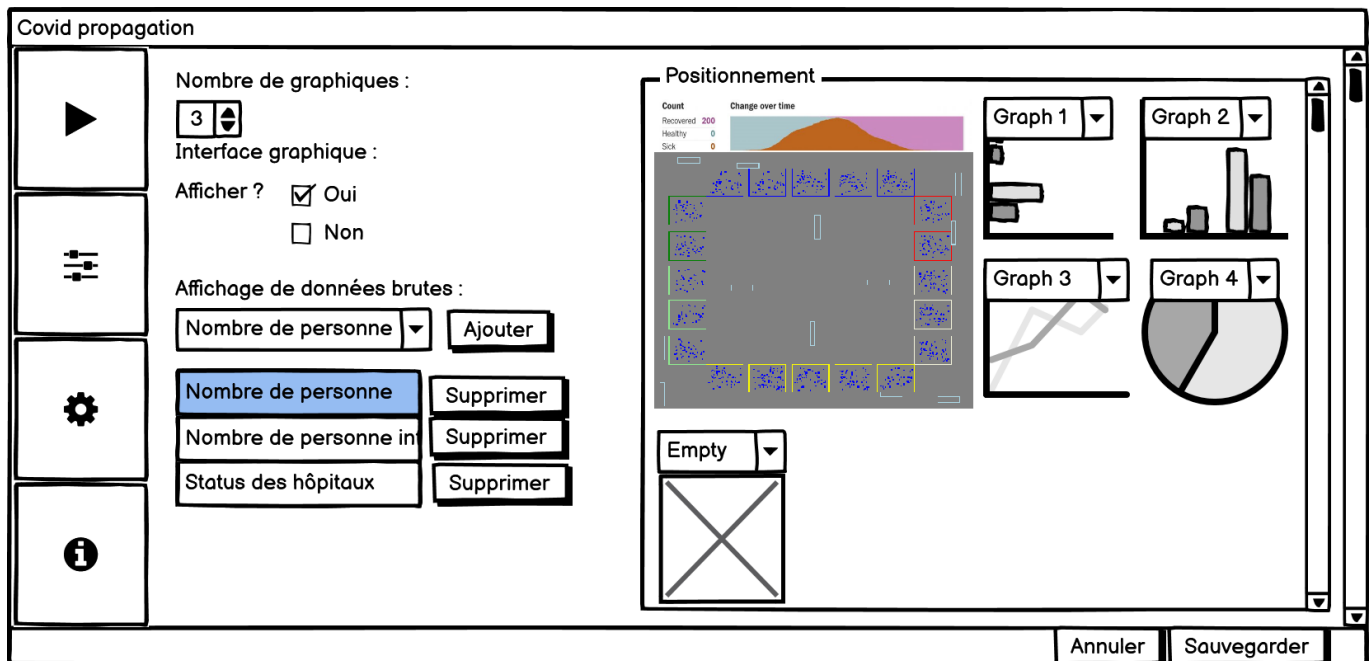


Figure 3: Maquette page de paramètres graphiques 1

Cette page permet à l'utilisateur de modifier l'interface graphique qui s'affichera dans la page simulation.

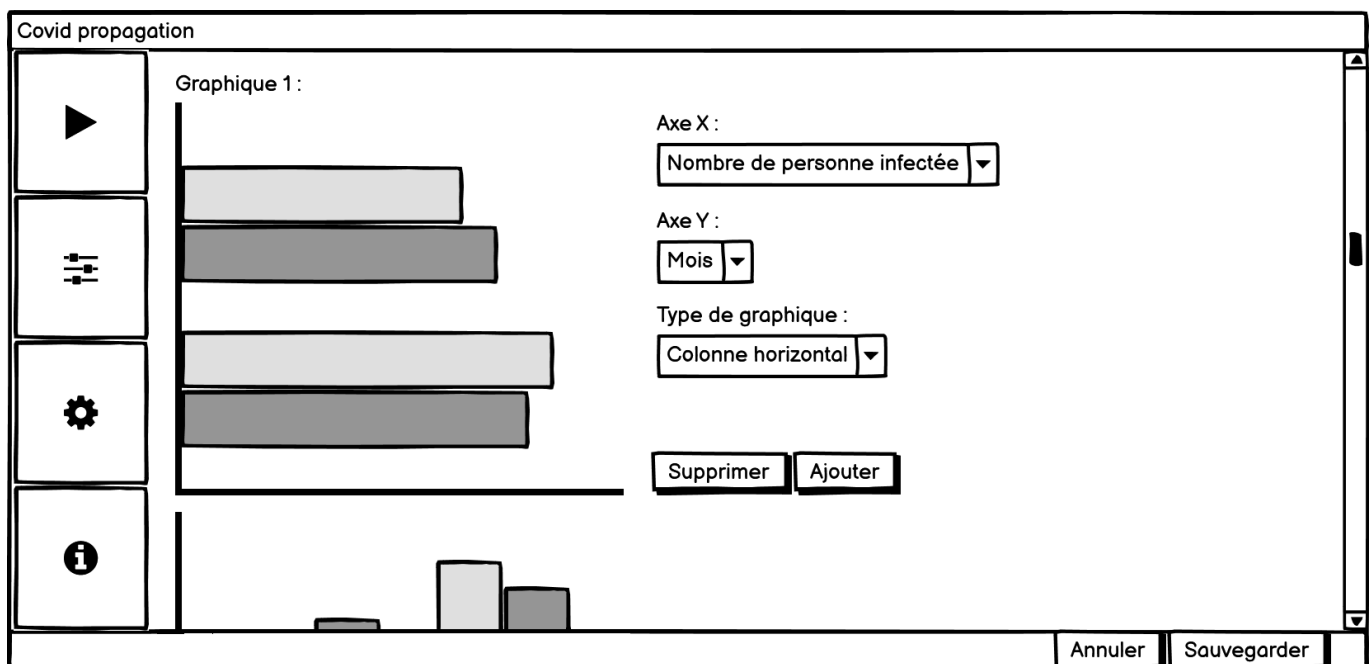


Figure 4: Maquette page de paramètres graphiques 2

Cette section permet de modifier en détail les paramètres d'un graphique.

Page Paramètres

Covid propagation






   	Nombre d'individus : <input type="text" value="1000"/>	Moyenne d'âge : <input type="text" value="40"/>
	Nombre d'infectés : <input type="text" value="5"/>	Asymptomatique ? <input checked="" type="checkbox"/> Oui <input type="checkbox"/> Non
	Quantité d'hôpitaux : <input type="text" value="1"/>	Vitesse :  50 itérations par <input type="text" value="Minutes"/>
	Pourcentage d'habitation : <input type="text" value="60%"/>	
	Pourcentage d'entreprise : <input type="text" value="60%"/>	
	Pourcentage de restaurant/Supermarché : <input type="text" value="60%"/>	
Paramètres par défaut Annuler Sauvegarder		

Figure 5: Maquette page de paramètres

Cette page permet à l'utilisateur de modifier les paramètres de la simulation.

Page Informations

Covid propagation

   	Technologies <p>Les technologies utilisées dans cette application sont : Unity, LiveCharts, Sources.</p>	Méthodologie <p>La méthodologie utilisée pour développer cette application est basée sur les principes de la programmation orientée objet (POO) et du développement modulaire.</p>
	Source de données <p>Les données utilisées pour la simulation sont issues de sources fiables et actualisées.</p>	Technologies <p>Les technologies utilisées dans cette application sont : Unity, LiveCharts, Sources.</p>
	Sources LiveCharts Unity	

Figure 6: Maquette page d'informations

Cette page sert d'aide à l'utilisateur sur le fonctionnement de l'application et contient aussi des informations sur sa logique et ses sources.

Interface Graphique

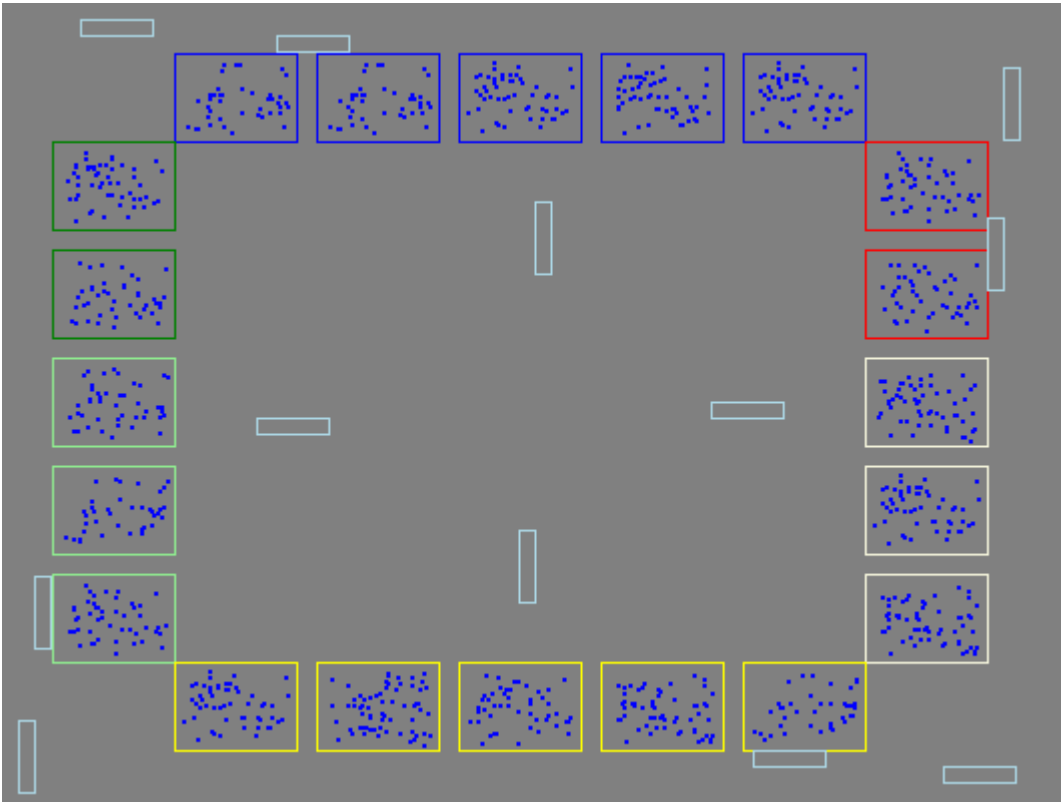



Figure 7: Maquette interface graphique

L'interface graphique permet de visualiser ce qui se passe dans la simulation. Les batiments, les véhicules et les individus sont affich  ainsi que leur d placements et leur status.

9.7. Persona

9.7.1. Utilisateur exp riment 

Persona Utilisateur exp riment 



Age: 30 ans
Travail: M decin
Nom: Ashley

Personnalit 

Ashley ayant beaucoup de patients concern  par le virus, elle souhaite avoir un moyen fiable de v rifier l'efficacit  de certaines mesures dans diff rentes conditions.

Objectifs

- Ashley souhaite comparer diff rentes situations avec les m mes mesures.
- Elle souhaite ensuite comparer les r sultats pour v rifier les mesures les plus efficaces dans diff rentes situations.

Aisance informatique

Utilisation web

Utilisation d'application

Bureautique

Smartphones

Attentes et connaissances

Elle est tr s   l'aise sur le web ainsi que dans l'utilisation d'applications qu'elle utilise pour son travail. Elle a donc une bonne exp rience avec les applications en g n rales. Elle souhaite que l'application ait une interface claire et que les fonctionnalit s soient en nombres. Cependant, elle recherche avant tout la pr cision des donn es ainsi qu'une simulation proposant des r sultats similaires   la r alit .

Figure 1: Persona expérimenté

9.7.2. Utilisateur inexpérimenté



Figure 2: Persona inexpérimenté

9.8. User stories

9.8.1. Ashley

En tant que Ashley

Je veux comparer différentes situations avec différentes personnes en prenant des mesures identiques

Afin de pouvoir observer les différences et déterminer quelles mesures est efficaces dans quelle situation.

scénarios Je crée sans soucis une situation à l'aide de l'application. Pour ce faire, j'entre différents paramètres, tel que le nombre de personnes, les mesures prises pour limiter la transmission ainsi que d'autres paramètres.

J'observe la simulation et prends note des résultats.

Une fois terminée, j'en lance une autre avec certains paramètres différents et prends note des résultats.

Je compare les résultats avec la simulation précédente et effectue ma conclusion.

9.8.2. Kanan

En tant que Kanan

Je veux vérifier l'efficacité de différentes mesures prises pour éviter la propagation du covid

Afin de afin de me donner une idée concrète et visuelle de l'efficacité de ses mesures.

scénarios Je lance l'application et cherche à créer une simulation. Une fois trouvé, je peux voir les mesures qui apparaissent clairement, d'autres paramètres sont disponibles mais je n'y touche pas. Une fois la simulation lancée, je vois un message m'indiquant que celle-ci commence. Des aides sont disponibles me permettant de comprendre les données qui sont affichées. Après avoir terminé cette simulation, j'en lance une autre en désactivant les mesures. Je relance la simulation et observe la différence entre les deux simulations.

9.9. Diagramme d'activité

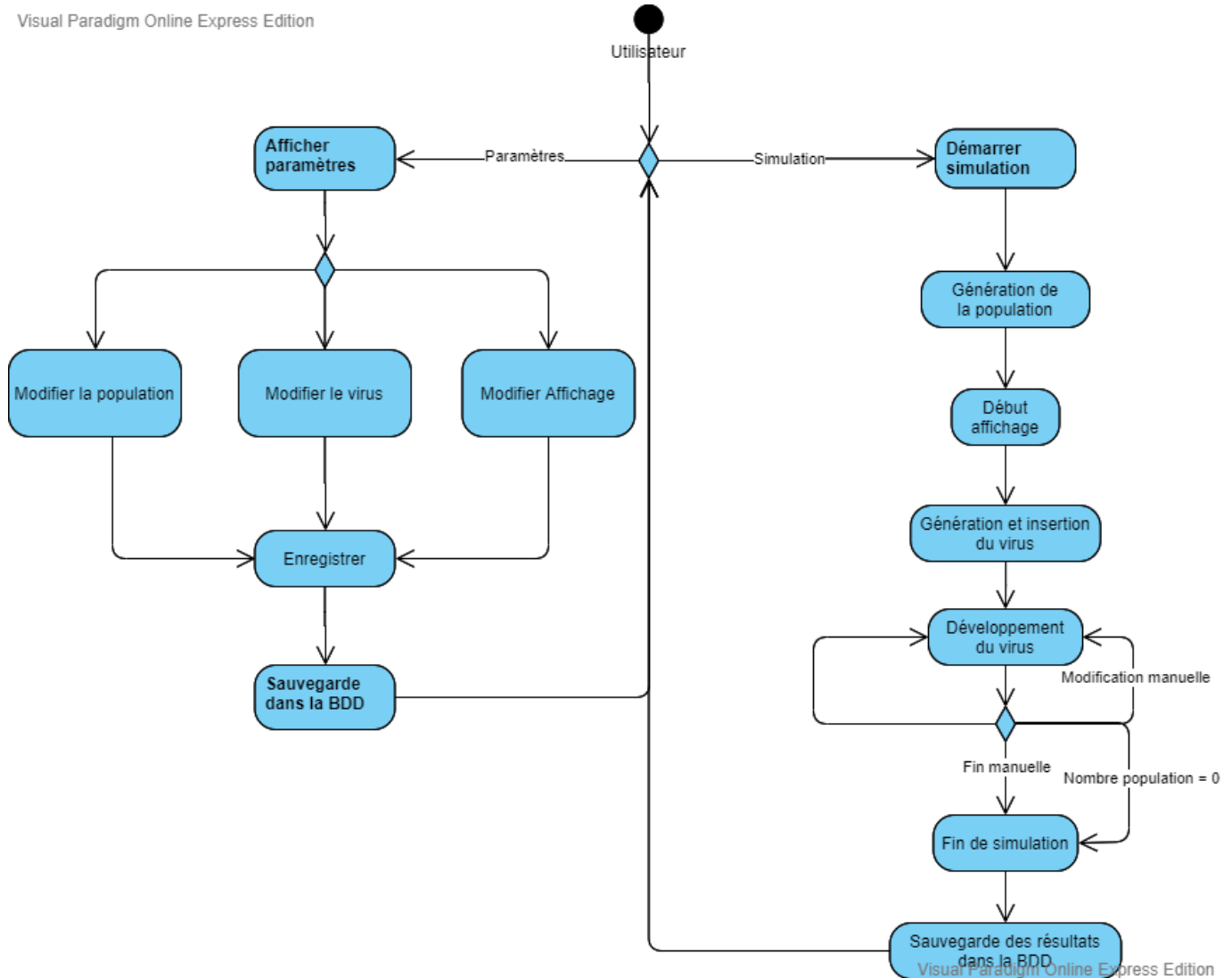


Figure 3: Diagramme d'activité

9.10. Planning

<https://docs.google.com/spreadsheets/d/1tSplbcDVvGnzMhEN71UDwPOxEy0oapQSSbxzjqXt3RA/edit?usp=sharing>

9.11. Diagramme de classe initial

8. Technologies utilisées

8.1. C#

C# est un langage de programmation orienté objet développé dans les années 2000 par Microsoft. Sa première version a été adoptée comme standard international en 2002 par Ecma. Il est régulièrement mis à jour, des versions majeures sont publiées tous les 2 à 3 ans environ. La dernière version de C# est la version 8.0 et c'est avec celle-ci que j'ai développé l'application. Son environnement de développement Visual Studio permet de créer des applications windows facilement.

8.2. Microsoft Visual studio

Microsoft Visual Studio est une suite de logiciels disponible sur windows et mac. La dernière version qui est la version utilisée dans la réalisation de ce projet est la version 2019.

Il permet de générer des services web XML, des applications web ASP .NET, des applications Visual basic, Visual C++, Visual C#. C#, C++ et basic utilisent tous les même IDE, ce qui permet de partager certaines ressources.

8.3. WPF

Windows Presentation Foundation (WPF) ou nom de code Avalon est une spécification graphique de .NET 3.0. Il utilise le XAML qui le rapproche d'une page HTML avec un système de balise. Il est apparu en 2006.

WPF comparé à WinForms permet par exemple l'affichage d'une interface responsive et l'utilisation du GPU pour certaines fonctionnalités.

8.4. Unity

Unity est un moteur de jeu développé par Unity Technologies. Il est majoritairement utilisé par des petits studio et des indépendant pour la création de jeux. Il est compatible avec le C# et le javascript qui permet de réaliser les scripts. Il permet de développer des jeux compatibles avec Windows, Mac OS X, iOS, Android, TV OS, PlayStation 3, PlayStation Vita, PlayStation 4, Xbox 360, Xbox One, Xbox One X, Windows Phone 8, Windows 10 Mobile, PlayStation Mobile, Tizen, Oculus Rift, Wii U, Nintendo 3DS, Nintendo Switch, WebGL.

8.5. XML

XML qui est un acronyme pour Extensible Markup Language. C'est un langage de balises et fait parti du sous-ensemble du standard Generalized Markup Language (SGML). Il a été créé en 1999.

Le but premier du XML étant de permettre au SGML d'être utilisé sur le web de la même manière que l'html. Dans mon cas, il permettra de stocker certaines données du programme.

8.6. LiveCharts

LiveCharts est une librairie C# permettant de créer des graphiques. Il permet d'inclure une grande quantité de graphiques à des projets, de lier les données au code. Lorsque les données changes, les graphiques s'adaptent automatiquement et sont animés. Les graphiques sont personnalisables et interactable. Il est même possible d'importer des cartes composées de régions en tant que graphique.

En plus d'ajouter énormément d'éléments graphiques et animations, LiveCharts est très performants et peut par exemple afficher des graphiques contenant plus de 100'000 points tout en restant fluide.

8.7. JSON

JavaScript Object Notation (JSON) est un format de données dérivé de la notation des objets javascript. Il permet d'afficher la structure d'une information comme par exemple un objet C# ainsi que ses données. C'est le format de données qui me permet de communiquer avec le programme Unity depuis le programme WPF.

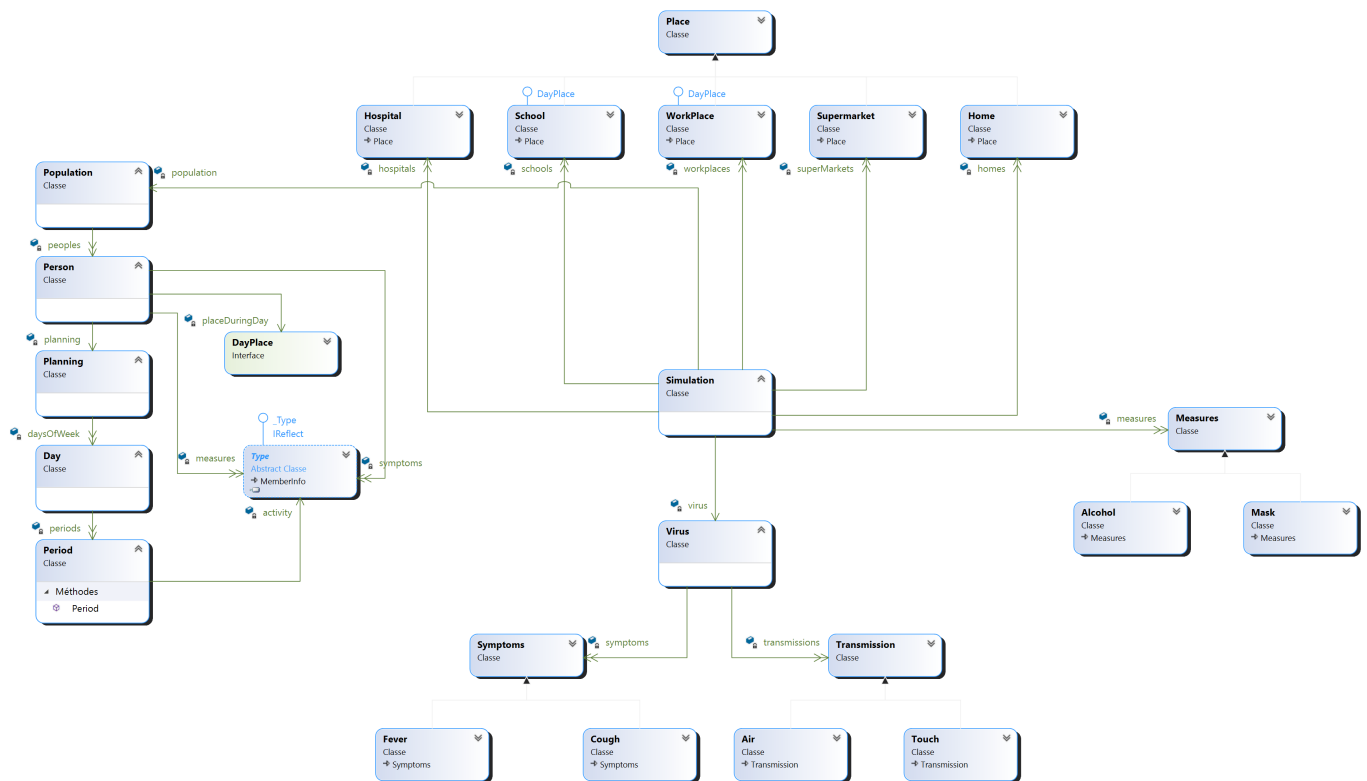


Figure 4: Diagramme de classe initial

9.12. Interactions

9.12.1. Menu principal

- Affiche un preview de l'affichage de la simulation
- Btn Paramètres
 - Population
 - Remplace l'affichage actuel se situant à droite pour afficher les paramètres de la population
 - Virus
 - Remplace l'affichage actuel se situant à droite pour afficher les paramètres du virus
 - Affichage
 - Remplace l'affichage actuel se situant à droite pour afficher les paramètres de l'affichage
- Btn lancer la simulation
 - Change l'affichage de la totalité de l'application, affiche une barre de chargement indiquant l'état de création de la simulation.

9.12.2. Population

Affiche une page avec les paramètres suivant :

- Écoles / lieux de travail
 - Différentes selon l'âge
 - Zone de transmission

- Familles / Cercles d'amis
 - Transmission
- Moyenne d'âge de la population
 - Permet de modifier la moyenne d'âge de la population de 1 à ~100
 - Permet de délimiter une limite d'âge maximal et minimal
 - Il est possible de le laisser par défaut
- Nombre d'individus
 - Le nombre d'individus simulé dans une population
 - La limite n'est pas définie par le programme
 - L'utilisateur connaît les limites de sa machine
- Mesures
 - Permet de sélectionner plusieurs mesures
 - Les mesures ont un pourcentage d'efficacité
 - Permet de réduire les chances de propagation du virus
 - Affecte différemment le virus en fonction de la mesure
 - Pourrait totalement contrer un virus
 - Peut être modifier par l'utilisateur jusqu'à un niveau de 100% de protection
 - Valeur par défaut défini par des études sur le sujet
 - Appliquer uniquement sur certaines parties de la population
 - Infectés
 - Sains
 - À risques
- Cercle social
 - Ami
 - Famille
 - Collègues
 - ...
 - Transmissions accrues
 - Rencontres inclusent dans le planning journalier des individus
- Hôpitaux
 - Il y a plusieurs hôpitaux avec les options :
 - Copier
 - Coller
 - Appliquer sur tout
 - Permet de modifier le nombre d'hôpitaux
 - Permet de modifier le nombre de places
 - Stabilise les individus y étant admis
 - Réduit leurs chances de décès
 - Nécessite du personnel qui peut être infecté pour fonctionner
 - Mesures du personnel :
 - Permet de sélectionner plusieurs mesures
 - Les mesures ont un pourcentage d'efficacité
 - Permet de réduire les chances de propagation du virus
 - Affecte différemment le virus en fonction de la mesure
 - Pourrait totalement contrer un virus
 - Peut être modifier par l'utilisateur jusqu'à un niveau de 100% de protection

- Btn annuler
 - Annule les modifications faites à l'hôpital
 - Réaffiche les données précédemment affichées
- Btn sauvegarder
 - Sauvegarde les paramètres choisis par l'utilisateur

9.12.3. Virus

Affiche une page avec les paramètres suivant :

- Effet sur le corps
 - Permet de modifier le pourcentage de propagation en fonction du symptôme (toux)
 - Les effets mortels nécessitant une hospitalisation
- Moyens de transmissions
 - Sont impacté par les symptômes (en incrémentant l'efficacité)
 - Sont impacté par les mesures (en décrémentant l'efficacité)
- Durée
 - Permet de définir la durée durant laquelle le virus prend effet
- Asymptomatique
 - Permet de définir si oui ou non il y a des asymptomatiques
 - Permet de définir le pourcentage d'asymptomatiques

9.12.4. Affichage

Affiche une page avec les paramètres suivant :

- Graphiques
 - Permet de sélectionner différents styles de graphiques à afficher
 - Permet de sélectionner une donnée au choix en X et en Y
 - Un exemple du graphique avec les données est affiché à côté de la barre de sélection
 - Plusieurs graphiques possibles à sélectionner
- Affichage d'une "carte" permettant une visualisation plus simple

9.12.5. Simulation

Affiche une page :

- Affichage d'une barre de chargement lors de la génération de la simulation
 - Évolue en fonction du nombre d'individus créé
- Affiche les graphiques sélectionnés
 - Onglets permettant de sélectionner quel graphique affiché
 - Possibilité d'afficher jusqu'à 4 graphiques sur le même onglet
- S'actualise toutes les secondes (environ)

9.13. Livrables

- Mind Map
- Planning
- Rapport de projet

- Manuel utilisateur
- Journal de travail ou LogBook
- Résumé / Abstract

Communications

Nous nous sommes mis d'accord sur le fait de se contacter au moins une fois par semaine pour vérifier l'avancement du projet ou poser différentes questions. Sachant que nous pouvons nous contacter à tout moment par mails. Nous avons établi des intervalles réguliers chaque semaines:

- Mardi --> Meet
- Vendredi --> En personne

Conclusion

Difficultés envisagée

Planning

Les plannings sont le centre des individus. Ils contrôlent les évènements des individus permettant les déplacements, les rencontres et donc les contaminations. Il faut donc qu'ils soient fonctionnelles rapidement pour permettre le teste de toutes les autres fonctionnalités de la simulation. C'est avant tout la création des plannings qui risque de me compliquer la tâche. J'ai déjà pu créer des plannings de tests. Dû à leur structure qui est composée de jours qui sont eux-même composés de nombreuses périodes. La création de ceux-ci est donc très compliquée manuellement. Il va donc me falloir trouver un moyen d'automatiser le système ainsi que de le rendre partiellement aléatoire pour garantir une certaine variété.

GUI

Après avoir réalisé le travail de stage, qui consistait en la création d'une interface graphique pour la simulation en WindowsForms, je me suis rendu compte de la limitation de celui-ci. Il me faut donc changer d'environnement. Il me faudra commencer par en trouver un nouveau et si celui-ci est externe à Visual Studio, il me faudra l'y intégrer. Cette intégration risque d'être compliquée à mettre en place et d'ajouter des limitations en terme de communication entre ceux-ci.

Liste des tâches du sprint 1

Avant de commencer à coder, je vais prendre un certain temps pour réfléchir à la structure du code que j'ai prévu. Essayer de trouver d'éventuelles défaut que je peux combler dès le départ.

Je vais ensuite trouver le meilleur moyen d'optimiser les performances de la simulation majoritairement en terme de threads et d'utilisation du GPU pour calculer l'affichage de l'interface graphique.

L'étape précédente me permettra aussi de rechercher des possibilités pour réaliser l'interface graphique. Mon attention se portant pour le moment sur WPF et Unity.

Après que les bases soient posées, je vais pouvoir commencer à créer l'UI. En comptant toutes les pages nécessaires pour l'utilisateur. La responsivité de la fenêtre ainsi qu'un thème personnalisé donnant une

interface plus sophistiquée.

Une fois les bases de l'interface posées, je vais commencer à coder la population. Tous les paramètres leur étant liés seront créés durant ce sprint. En commençant par leur résistance au virus ainsi que la maladie réduisant cette résistance. Puis les moyens de transports comprenant les lieux où ils vont ainsi que leur cercles sociaux. Puis l'âge qui a pour principe de modifier certains paramètres déjà établis. Et pour finir le planning des individus ainsi que les symptômes du virus sur eux.

Organisation

Planification

Pour la planification du travail de diplôme, j'ai décidé d'utiliser excel qui permet de réaliser un planning simple et très compréhensible. Le planning me servant de fil conducteur et de moyen d'organiser l'ordre d'exécution des tâches que j'ai créées.

Tâches

Le traçage des tâches s'effectue sur github en suivant le modèle de scrum. Les tâches à effectuer sont dans une section "To Do", les tâches qui sont en cours, sont dans la section "In progress" et finalement les tâches terminées sont dans la section "Done".

Les sprints sont tous séparés ayant des tâches différentes.

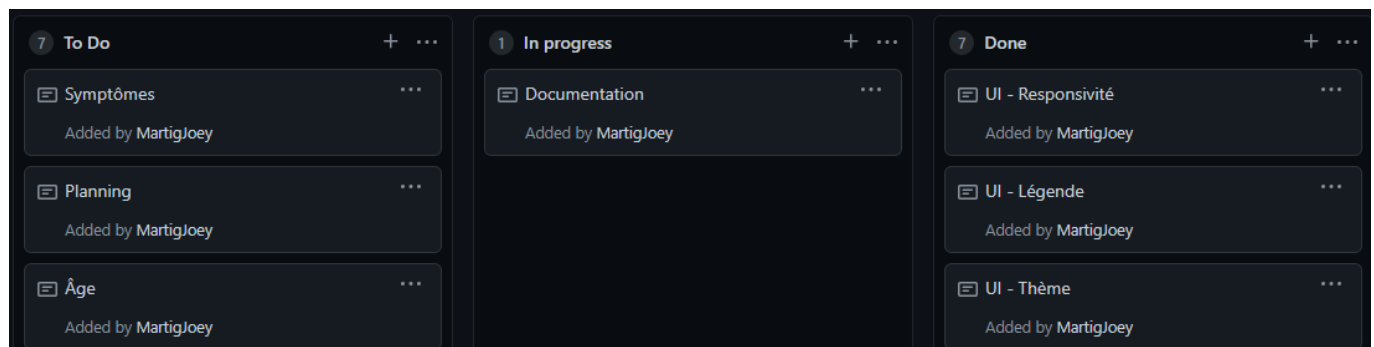


Figure 8: Gestion des tâches

Versionning - Backup

Le versionning est fait à l'aide de github. Au moins deux sauvegardes sont faites chaque jour. Une à midi et une en fin de journée. En cas de perte de données, je ne perds qu'une demi-journée dans le pire des cas.

Ayant eu des problèmes avec git par le passé. (Corruptions de fichiers - conflits) J'ai décidé de faire une sauvegarde supplémentaire sur un disque dur externe. La fréquence de sauvegarde étant plus faible mais suffisante étant donné qu'il s'agit d'une sauvegarde de secours.