

La fouloscopie au service de l'épidémiologie



Sommaire

I – Introduction	2
II – Partie Théorique	4
1) Les modèles SEIR et SEIRS	4
2) Le jeu de la vie	6
3) Les différentes options ajoutées	7
III – Aspect Pratique	9
1) Simuler une épidémie grâce au Jeu de la vie	9
A) Comment caractériser une épidémie ?	9
2) Notre programme	11
A) Description	11
IV – Les simulations	14
1) Le programme réalisé en première année	14
2) Première version du nouveau programme	14
3) Les 2 ^{ème} et 3 ^{ème} version du nouveau programme	16
4) La 4 ^{ème} version du programme	16
5) 5 ^{ème} et dernière version du programme	17
V – Conclusion	20
VI – Bibliographie	20

I – Introduction

A l'image de la crise sanitaire que nous traversons depuis maintenant plus d'un an, le thème de cette année, Santé et Prévention, nous a orienté sur le thème de l'épidémiologie qui est aujourd'hui au cœur de l'actualité.

Nous nous sommes d'abord demandé comment cette science fonctionnait, et comment elle évoluait. C'est alors que nous avons pensé à la fouloscopie, un terme nouveau, mais qui décrit en réalité une science qui existe depuis bien plus longtemps : l'étude des foules. La question qui se pose tout naturellement est donc : Comment la fouloscopie sert-elle l'épidémiologie ?

Les objectifs de notre TIPE ont été de répondre dans un premier temps à cette question en première année, puis dans un second temps de créer notre propre simulation d'épidémie qui soit la plus réaliste possible, grâce aux données apportées par la fouloscopie.

Cette année a donc servi à créer ce logiciel permettant de générer des simulations mais aussi à le rendre accessible et facile d'usage.

(Toutes les démarches scientifiques effectuées dans ce dossier sont tirées d'article, de documentation ou de codes sources libres de droit – ceux-ci sont trouvables dans le VI en annexe/bibliographie)



Planning :

Septembre	Octobre	Novembre	Décembre
-Brainstorming de la marche à suivre pour l'année -Création du schéma algorithmique du programme SEITM	- Programmation de L'algorithme SEITM -Programmation de l'interface graphique (Ce qui s'affiche à l'écran) -Programmation de l'interface Homme-Machine (Ce qui va interagir avec l'utilisateur)		

Janvier	Février	Mars	Avril	Mai
- Terminer le programme SEITM -Lister les Variables à ajouter	-Création Nouvelles Variables : - Densité - Mutation - Vaccination - Tension Hospitalière		- Comparaison Programme P1 et SEITM - Conclusion Apports et limites du programme	- Préparation du dossier de fin d'année

Cahier des charges :

-Objectif :

Réaliser un programme informatique permettant la simulation d'un modèle épidémiologique, ici le modèle SEITM, et s'assurer de la facilité d'utilisation et d'exploitation des mesures

-Algorithme :

- Création d'un algorithme Fonctionnel pouvant simuler une épidémie sur une grille 10*10 minimum sur une durée de 31 jours, soit 1 mois, minimum. Sans limites pour le maximum de jours simulés
- Implémentation de 3 facteurs extérieurs minimum

-Interface Graphique :

- Création d'une interface graphique aérée (Chartre Graphique)
- Création de pages explicative pour faciliter l'utilisation et la compréhension du programme pour l'utilisateur
- Affichage des courbes et enregistrement de celles-ci dans un fichier tier
- Affichage des paramètres entrées par l'utilisateur lors de la simulation et enregistrement de ceux-ci dans un fichier tier

-Interface Homme-Machine :

- Création de 2 liens permettant de rejoindre le site de l'ESEO et d'ouvrir le compte-rendu de P1 (qui explique le modèle SEITM)
- Permettre de rentrer des paramètres pour les simulations depuis l'interface graphique
- Pouvoir arrêter et relancer la simulation si besoin sans relancer le programme



II – Partie Théorique

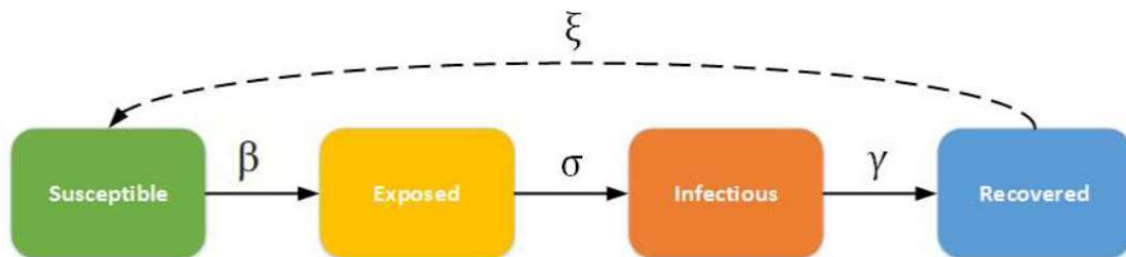
1) Les modèles SEIR et SEIRS

En Première année nous nous sommes concentrés sur la définition d'une épidémie et de l'épidémiologie Et comment l'appliquer sur une foule En la considérant de façon statistique c'est à dire en appliquant des taux sur des nombres sans prendre en compte les personnes comme des individus.

Nous avons donc étudié l'épidémiologie et avons pris connaissance des modèle SEIR et SEIRS

On rappelle le fonctionnement des modèles SEIR et SEIRS qui sont la base de notre modèle SEITM que nous avons créé à partir des deux autres modèles

Les modèles SEIR et SEIRS se présentent sous la forme suivante :



Ici, chaque case représente un état chez l'individu lors d'une épidémie.

- S : Susceptible : La personne est saine, et non immunisée, donc susceptible d'être infectée.
- E : Exposed : La personne a été exposée et contaminée par l'agent infectieux, mais n'est néanmoins pas encore infectieuse, (elle ne transmet pas la maladie).
- I : Infectious : La personne est infectée depuis un certain moment, et est désormais infectieuse, c'est-à-dire qu'elle peut contaminer d'autres personnes.
- R : Recovered : La personne n'est plus contaminée et a gagné une immunité contre l'agent contagieux.
- S (Seulement pour le modèle SEIRS) : Susceptible : La personne est revenue à l'état initial, c'est-à-dire saine, et non immunisée.

Seul le modèle SEIRS prend en compte la boucle retour c'est-à-dire la possibilité de redevenir malade. L'intérêt est donc de gagner en réalisme car chaque personne infectée peut le redevenir dans le cas d'une épidémie virale

Dans le premier programme crée l'année dernière on considérait ces cases comme des variables

Et donc les lettres de passage représentaient la proportion à passer au stade suivant

Sur le schéma ces proportions correspondent aux flèches positionnées entre chaque case :

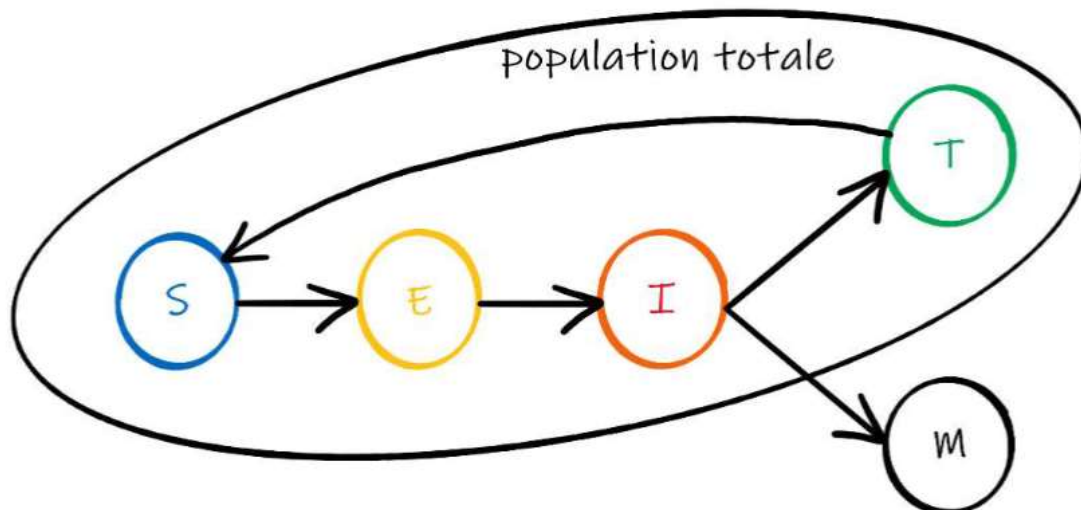
$S \rightarrow E$: taux d'infection (noté β) C'est le taux qui indique la probabilité qu'une personne "Infectious" contamine une personne "Susceptible"

$E \rightarrow I$: taux d'incubation (noté σ) C'est le taux qui indique combien, en moyenne, de personnes deviennent infectieuses après avoir été infectée, basé donc de la durée moyenne d'incubation (qui vaut $1/\sigma$).

$I \rightarrow R$: taux de récupération (noté γ) C'est le taux qui indique combien en moyenne de personnes vont guérir chaque jour, basé donc sur la durée moyenne d'infection D . (On a $\gamma = 1/D$).

$R \rightarrow S$: taux de perte d'immunité (noté ξ) C'est le taux qui indique combien de personnes en moyenne reviennent au stade de "Susceptible", basé donc sur la durée moyenne d'immunité.

Notre modèle, Le modèle SEITM prend en compte l'état M correspondant aux personnes décédées lors de l'épidémie



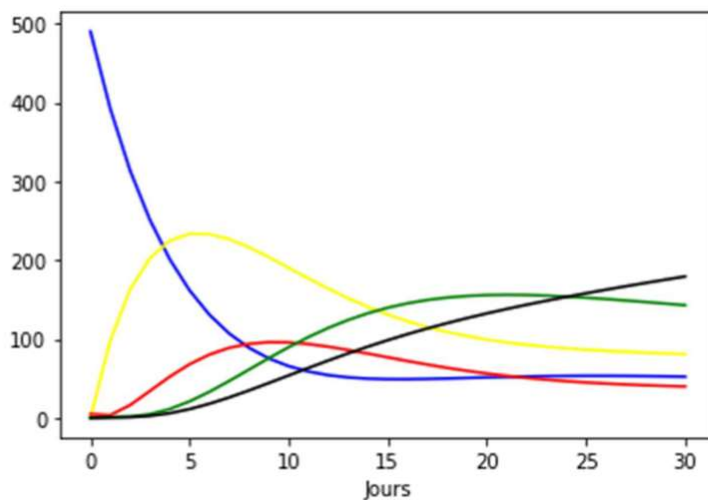
Le but de ce modèle est encore une fois de gagner en réalisme avec le comportement d'une épidémie : Le taux de mortalité influe sur sa capacité à se répandre.

Pour le calcul du nombre d'individus dans chaque état pour un jour donné, nous avons réalisé une suite matricielle

$$\begin{pmatrix} S \\ E \\ I \\ T \\ M \end{pmatrix}_n = \begin{pmatrix} 1-e & 0 & 0 & d & 0 \\ e & 1-f & 0 & 0 & 0 \\ 0 & f & 1-c-g & 0 & 0 \\ 0 & 0 & g & 1-d & 0 \\ 0 & 0 & c & 0 & 1 \end{pmatrix}^n * \begin{pmatrix} S_0 \\ E_0 \\ I_0 \\ T_0 \\ M_0 \end{pmatrix}$$

Ici Chaque Variables S E I T et M sont des suites récurrentes et il suffit de faire un produit matriciel n fois pour obtenir les données du jour n

Cet algorithme nous permet donc d'obtenir des courbes de ce type :



- Bleu : population saine
- Jaune : population infectée mais non infectieuse
- Vert : population temporairement immunisée
- Rouge : population infectieuse
- Noire : population décédée

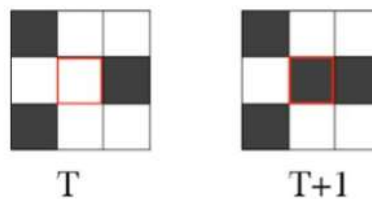
(Le code couleur restera le même dans la suite du compte-rendu)

Nous observons que ce modèle est très prédictible et donc peu représentatif car bien que chaque épidémie ait un comportement semblable (Ex : effet de vague), chaque épidémie va être différente de la précédente et ce à cause de différents facteurs.

2) Le jeu de la vie

Le Facteur principal est le comportement Humain, afin de pouvoir l'étudier nous avons créé un second algorithme Basé sur le fonctionnement du jeu de la vie :

Le jeu de la vie est donc un algorithme « cellulaire », c'est-à-dire un algorithme qui va appliquer des « règles » aux cellules d'une grille, créer par John Horton Conway en 1970. Ainsi, le jeu de la vie applique certaines règles aux cellules les faisant passer de l'état cellule vivante à l'état cellule morte.

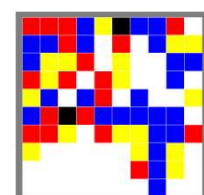


Ce second programme va donc se baser sur ce type d'algorithme pour comprendre la diffusion d'une épidémie dans une foule statique. En effet nos cellules pourront appartenir à 5 états différents correspondant à notre système SEITM

Ainsi les cellules infectées auront un impact sur les cellules vivantes et auront un pourcentage de chances de devenir des cellules mortes. Nous pourrons donc observer la propagation de l'agent infectieux et nous pourrons appliqués des contraintes sur certaines cellules pour voir l'impact sur la propagation.

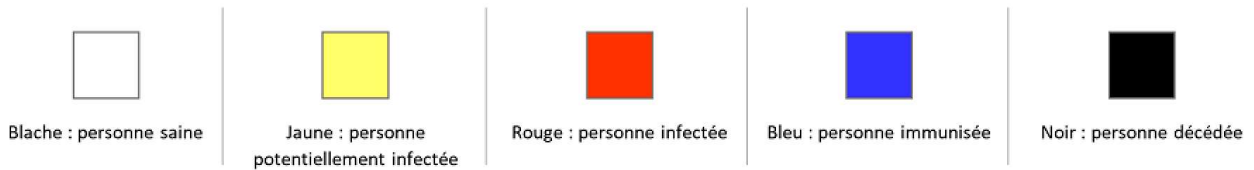
La Grille va donc se présenter de la manière suivante :

L'utilisateur pourra sélectionner la taille de la grille les différentes variables influant le comportement de l'épidémie et la durée de la simulation.



De plus il pourra mettre en place d'autres options influant l'épidémie

Une grille où chaque cellule représente une personne ; chaque cellule pouvant avoir 5 états :



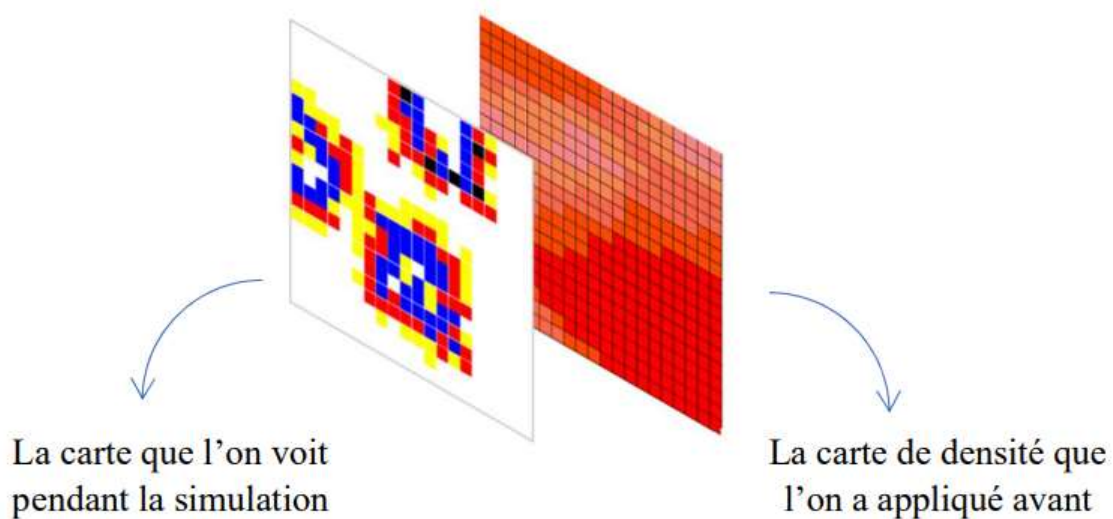
Ces nouvelles options seront facultatives au lancement du programme mais auront un effet sur le déroulement de la simulation.

3) Les différentes options ajoutées

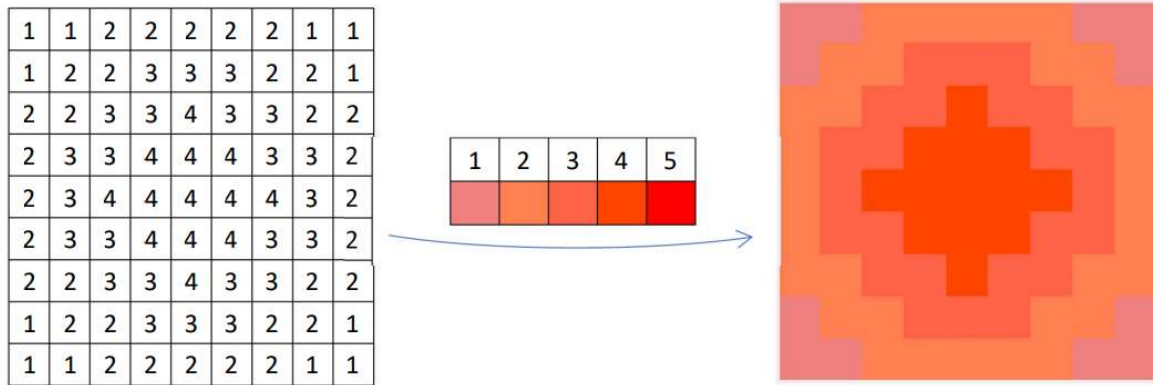
La première option est la mise en place d'une carte de densité :

Pour simuler une épidémie, un des paramètres sur lequel on peut jouer est la densité.

Pour ce faire, on applique une carte de densité à la matrice de départ : dans les zones de plus forte densité, l'épidémie se propagera plus rapidement. Il faut imaginer une seconde matrice derrière la première :

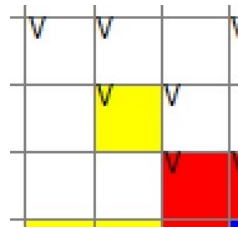


Cette deuxième carte ou matrice (celle de droite ci-dessus) est en réalité une grille contenant une valeur dans chaque cellule, on associe ensuite un gradient de couleur pour caractériser le gradient de densité :

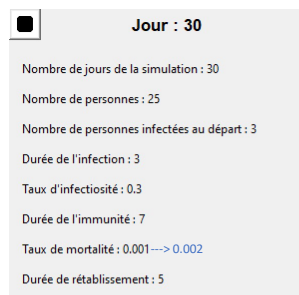


Enfin, lorsque on exécute le programme, à chaque itération (chaque jour) on multiplie la probabilité d'être infecté par un coefficient qui dépend de la valeur de la grille de densité correspondante.

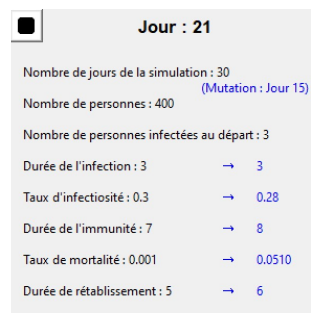
Le Second Facteur est la mise en place de la vaccination Les personnes vaccinées vont voir leur durée de rétablissement se réduire réduisant ainsi la mortalité et la contagiosité :



Le Troisième Facteur est la tension hospitalière qui prends en compte le nombre de personnes infectées par rapport au nombre total de personnes dans la population de base et va augmenter le taux de mortalité en conséquent à partir d'un certain seuil.



Le dernier Facteur est la mutation. En effet une mutation dans une maladie va changer drastiquement son comportement en changeant la durée dans laquelle l'individu va rester dans les différents états et les taux de passages d'état à état.



III – Aspect Pratique

1) Simuler une épidémie grâce au Jeu de la vie

A) Comment caractériser une épidémie ?

Tout d'abord, une épidémie est régie par plusieurs paramètres, c'est d'ailleurs cette complexité qui met en évidence le fait que, malgré la précision et la multiplication des paramètres, la simulation reste une approximation voire une « prédiction probable » de la réalité. En effet, le comportement humain est lui-même un paramètre des plus aléatoire à prendre en compte. De ce fait, pour mieux comprendre cela, expliquons ce qui diffère du jeu de Conway.

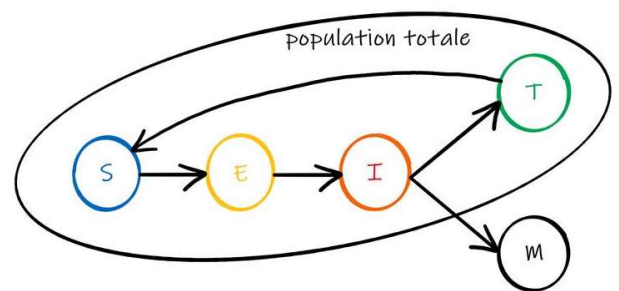
a) Les règles de base d'une épidémie

Premièrement, ici la cellule a plus que 2 états, à savoir :

- Sain
- Exposé
- Infecté
- Temporairement Immunisé
- Mort



Schématisation du cycle des états



Ensuite, il semble logique de rappeler que dans la réalité lorsque la cellule meurt, elle ne peut plus redevenir vivante.

Enfin, il faut également souligner que le passage d'un état à un autre est effectué selon une certaine probabilité.

b) Quels paramètres utiliser ?

Si on retrace l'histoire de l'épidémiologie, au cours des dernières années, certains paramètres, en plus de ceux de bases semblent revenir.

Pour ce qui est des paramètres de bases, à savoir les probabilités de passages d'un état à un autre, ils sont alors au nombre de 5 si on se réfère au schéma précédant (ceux-ci ont également été présenté en amont du dossier, dans la partie théorique cf ...) ; on peut également mentionner le nombre de jours, le nombre de personnes, et le nombre d'infecté au jour 0.

Pour ce qui est des paramètres « additionnels », ils vont permettre d'affiner la simulation, et de la rendre plus précise, mais aussi plus complexe à réaliser ; nous en avons choisi :

- La densité de population
- La vaccination
- La tension hospitalière
- La mutation

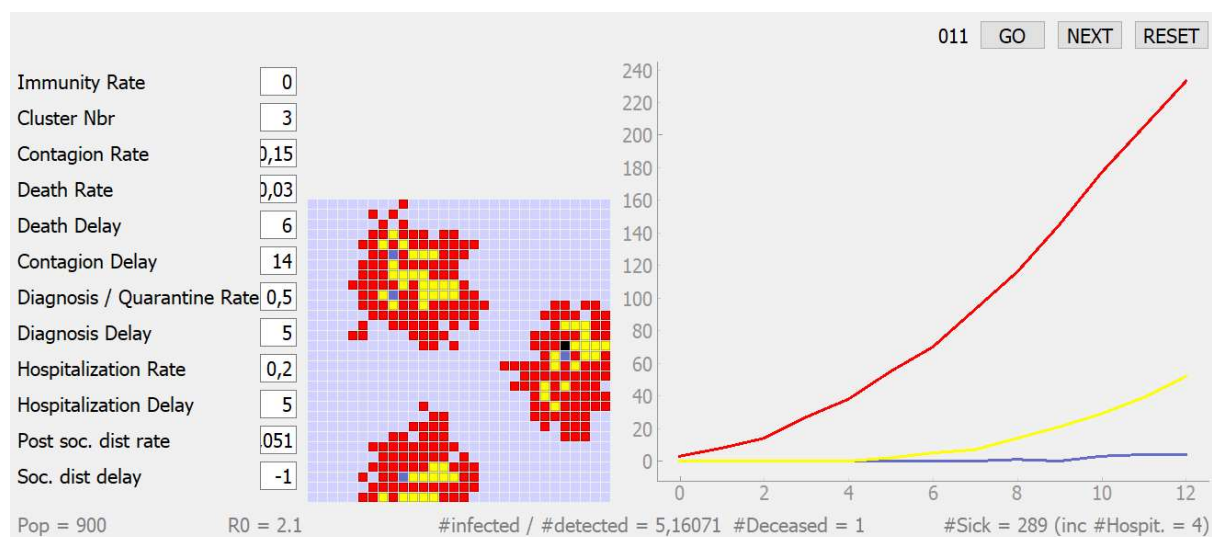
Enfin, à l'origine comme indiqué dans notre sujet, nous devons traiter l'aspect foulescopie de l'épidémiologie : tout le travail qui lie ces deux aspects, tant de manière théorique que pratique ont été travaillé l'année dernière. Cependant, c'est en ce début d'année que nous avons décidé de nous concentrer principalement sur la simulation d'épidémie sans y incorporer toute la recherche liée à la foulescopie. Ce domaine, qui correspond à l'étude des foules, peut être vu comme un autre paramètre de premier plan à prendre en compte lors de la simulation d'épidémie. Néanmoins, au vu des capacités et du temps qui était à notre disposition, nous avons préféré garder ces recherches sans les ajouter à la programmation pour pouvoir nous concentrer sur la partie épidémiologie, cœur de notre TIPE étant donné le thème de celui : santé et prévention.

C'est donc grâce à ces précisions, et différentes explications que l'on peut lister les paramètres sur lesquels l'utilisateur pourra jouer lors des simulations, à savoir :

- | | | |
|----------------------------|-------------------------|---------------------------------|
| - Taux de mortalité | - Densité de population | - Nombre de jour |
| - Taux de perte d'immunité | - Vaccination | - Nombre de personnes |
| - Taux d'incubation | - Tension Hospitalière | - Personnes infectées au jour 0 |
| - Taux d'infection | - Mutation | |
| - Taux de récupération | | |

c) Programme existant

Ainsi, avec une liste des paramètres et des règles bien définies, nous avons également regarder si d'autres avaient créé un jeu de la vie épidémiologique. Au cours de nos recherches, nous sommes tombés sur une simulation disponible en libre accès et libre de droit sur GitHub (lien disponible en annexe) qui ressemble à cela :



On retrouve bien différents paramètres (ici la distanciation sociale a été ajouté), ainsi qu'une grille avec les différentes cellules représentant les individus (à petite échelle) ou les clusters (à grande échelle). De plus, on note également la présence d'un graphe permettant de visualiser l'évolution dans le temps des états d'une cellule possible.

Ce faisant, nous avons décidé de nous inspirer de ce programme pour créer le nôtre.

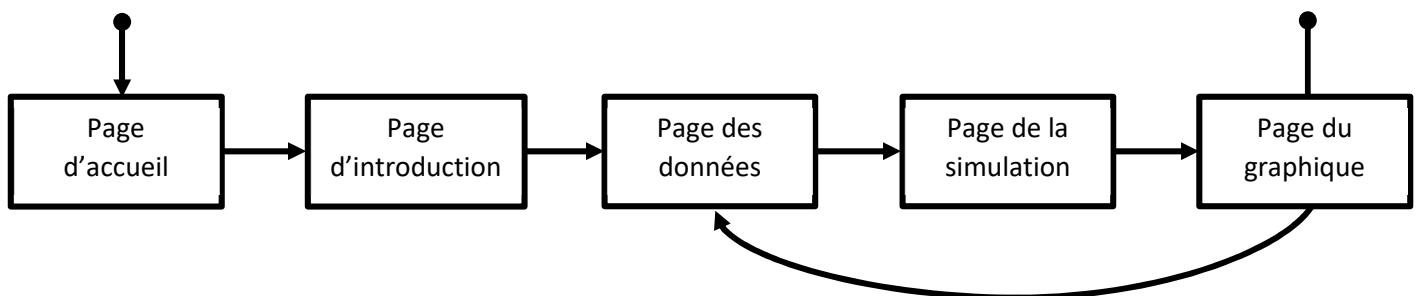
2) Notre programme

A) Description

Pour ce genre de simulation, nous voulions développer un programme simple d'utilisation, c'est-à-dire tant compréhensible visuellement que par sa prise en main. Ainsi, comme il sera précisé plus tard, celui-ci une fois téléchargé nécessite seulement une application permettant de traduire le langage de programmation utilisé, à savoir python, pour lancer le programme.

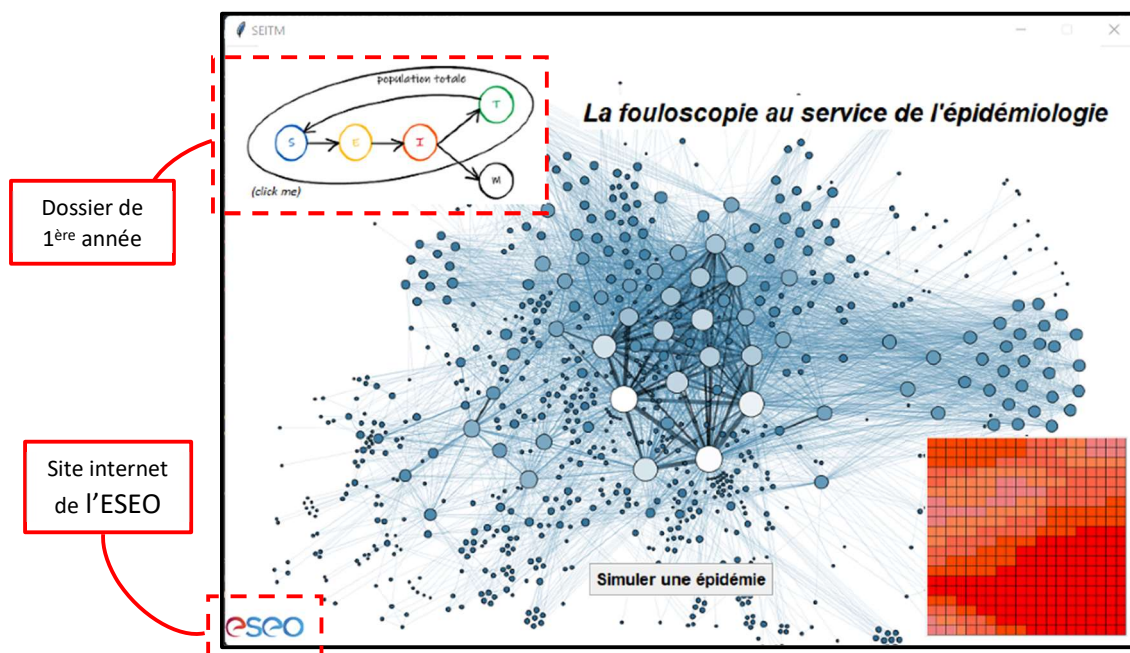
a) Structure du programme

Pour ce genre de projet, nous avons dû découper la programmation en trois temps : premièrement le jeu de la vie épidémiologique de base, deuxièmement l'ajout de paramètres pour préciser les simulations, et troisièmement le développement de l'interface homme-machine. Après avoir divisé notre travail, nous avons établi un cahier des charges correspondant aux différentes fenêtres du programme, à avoir le schéma ci-dessous :



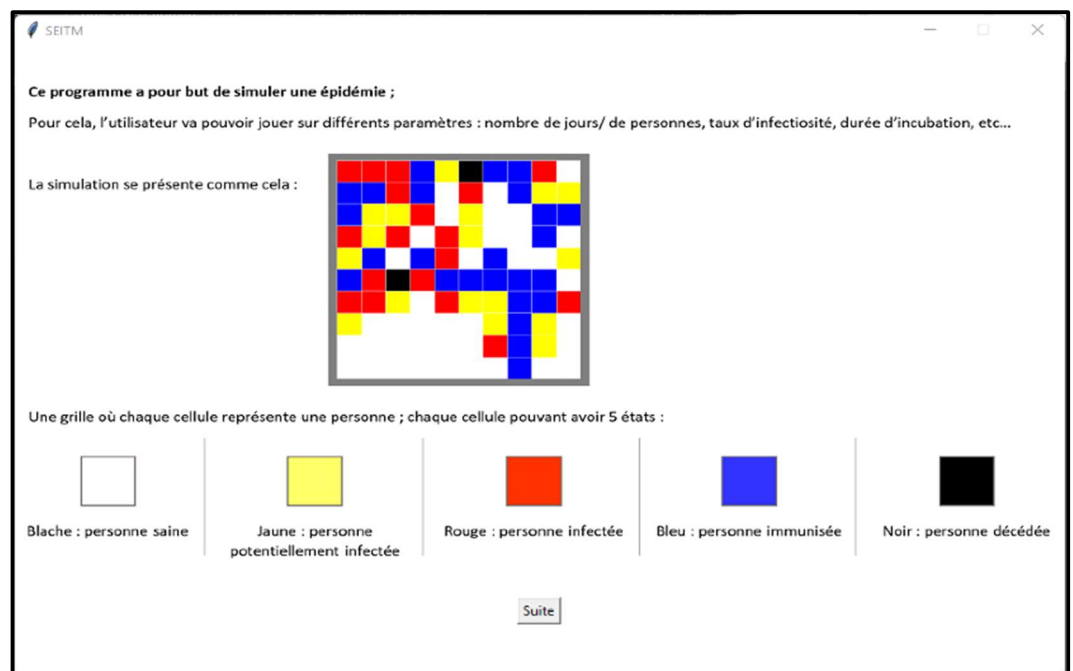
Ce faisant, nous avons ensuite réalisé chacune des fenêtres :

- La page d'accueil :



- La page d'introduction :

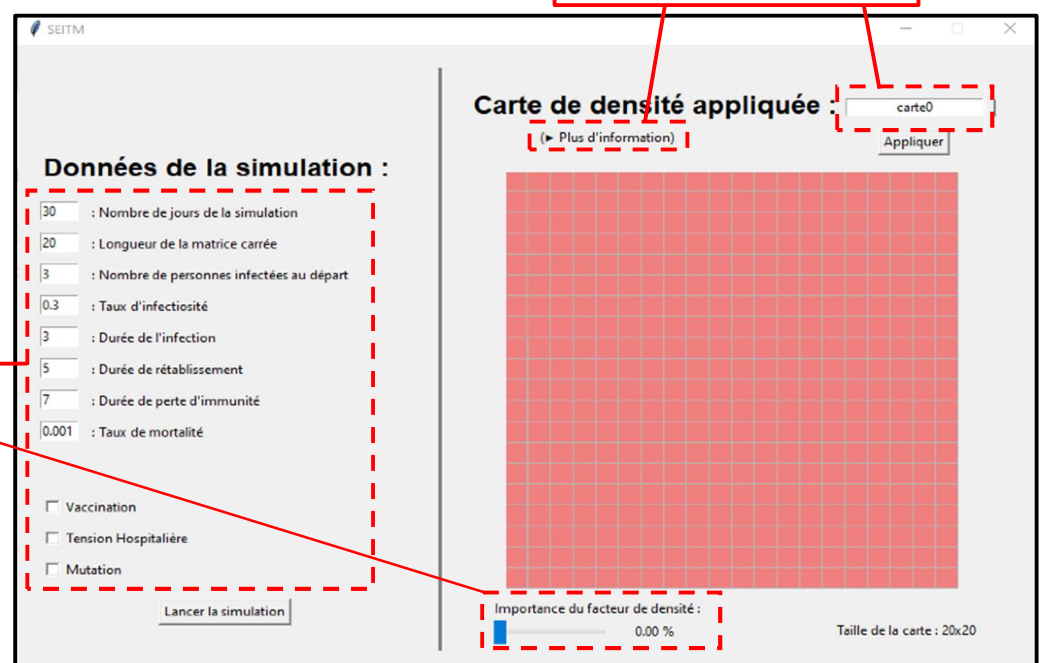
Brève explication du concept de jeu de la vie épidémiologique



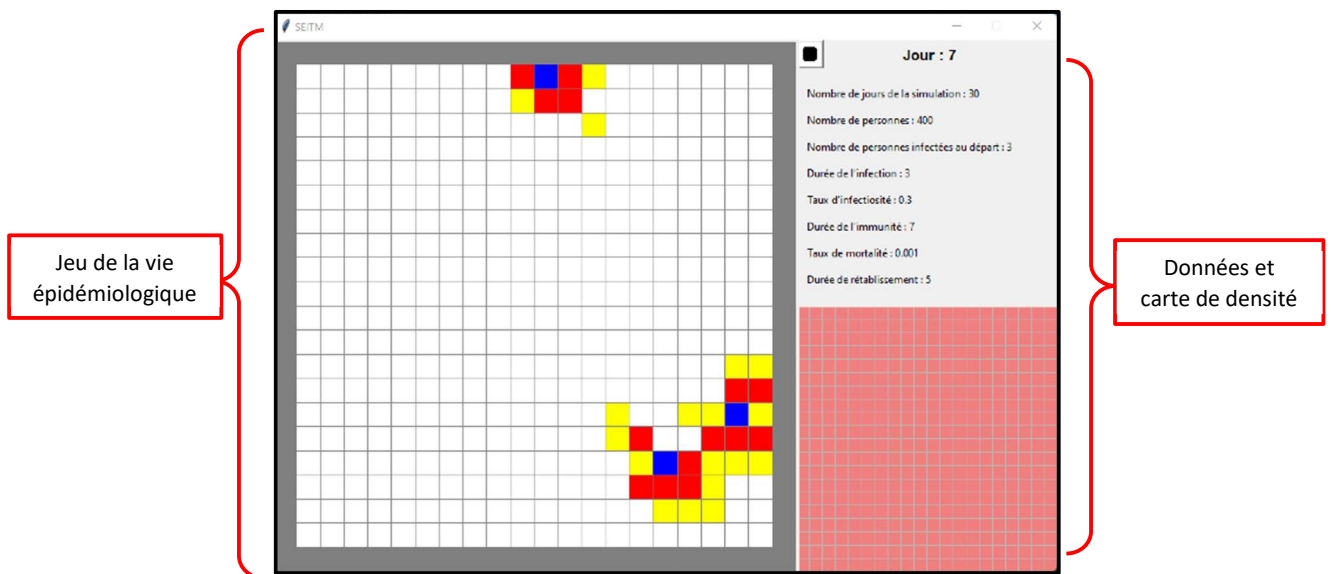
- La page des données :

Carte de densité appliquée avec prévisualisation, et un tutoriel pour créer sa carte

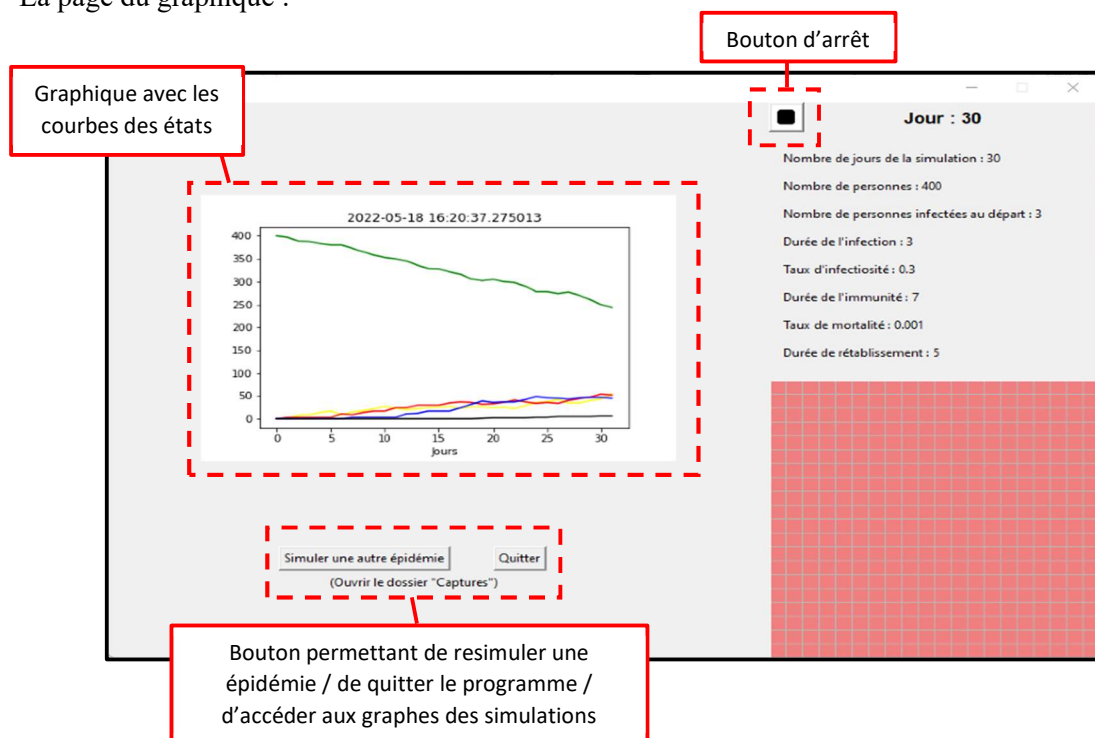
Différents paramètres sur lesquels l'utilisateur peut jouer pour faire la simulation de son choix



- La page de la simulation :



- La page du graphique :



Une fois qu'une simulation est terminée, un fichier texte contenant toutes les données de celle-ci ainsi que les graphiques sont enregistrés dans un sous dossier ; le tout ayant pour nom la date et heure de la simulation.

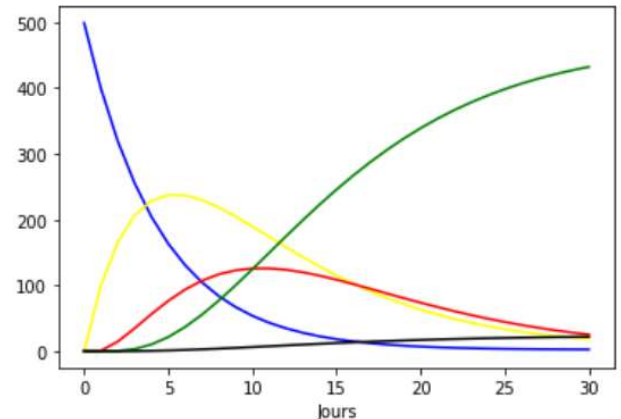
IV – Les simulations

1) Le programme réalisé en première année

En parallèle de la recherche théorique sur notre sujet, nous avons aussi réalisé en première année, un premier programme sur Python basé sur le modèle SEITM.

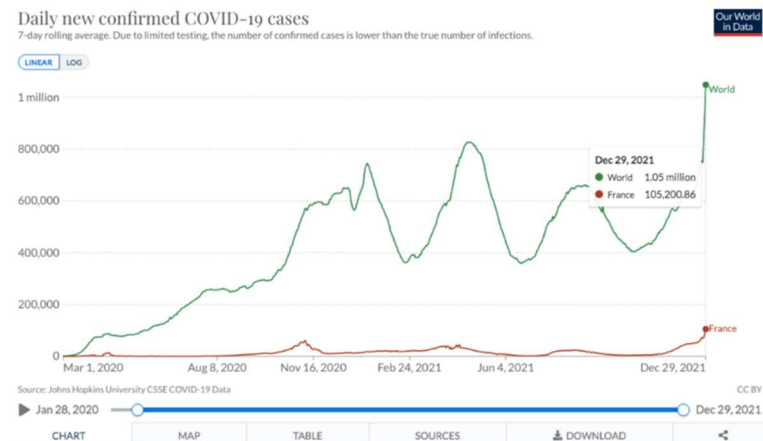
La version finale de ce programme nous renvoyait, pour les paramètres suivants, ces courbes :

```
a=1      #nombre de personnes infectées
b=500    #nombre total de personnes
c=0.01   #taux de mortalité
d=0.001  #taux de perte d'immunité
e=0.2    #taux d'infection
f=0.14   #taux d'incubation
g=0.2    #taux de récupération
nbrj=31  #nombre de jours
```



On peut observer avec ce programme que les courbes qu'il renvoie sont lisses, et que, tant que l'on rentre les mêmes paramètres, le programme renvoie toujours les mêmes courbes, ce qui ne correspond pas vraiment à la réalité.

Lorsque l'on compare avec des courbes tirées d'épidémies réelles, on se rend compte que les courbes ressemblent plus à des 'pics', comme avec l'épidémie de Covid-19 par exemple :



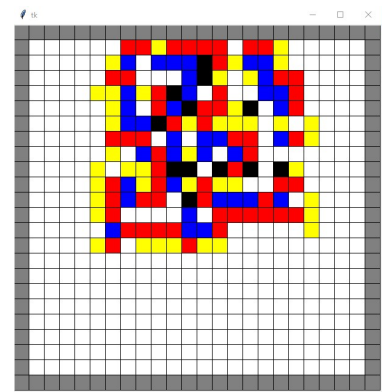
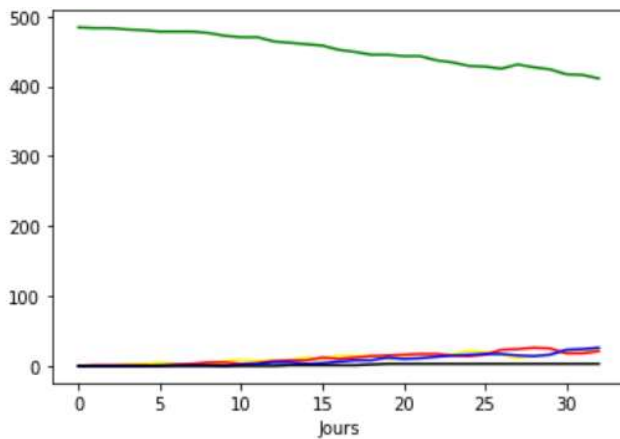
2) Première version du nouveau programme

Notre premier programme n'étant pas satisfaisant, nous avons décidé de changer de méthode de construction des courbes, et plutôt que de calculer des valeurs pour chaque jour avec des fonctions prédéfinies, nous avons eu l'idée d'ajouter une part d'aléatoire au programme, afin d'avoir des courbes non plus lisses, mais avec des pics, comme vu précédemment.

De plus, nous avons aussi décidé de ne plus partir du principe qu'une fois que quelqu'un guérissait de la maladie, il y était immunisé à vie. Nous avons donc ajouté un paramètre qui détermine la durée de l'immunité après la guérison, faisant en sorte qu'une personne puisse être réinfectée.

Afin d'illustrer la simulation, nous avons également créé une partie graphique grâce à la bibliothèque python Tkinter, où l'on voit l'évolution de l'épidémie simulée en "temps réel" (Au jour par jour), avec une matrice carrée de cellules dont les dimensions sont rentrées au début de la simulation avec les autres paramètres.

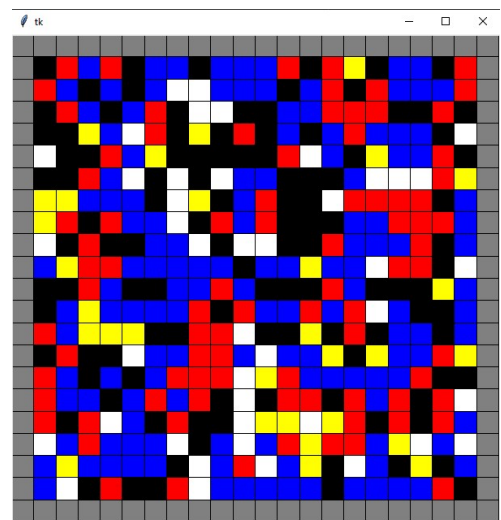
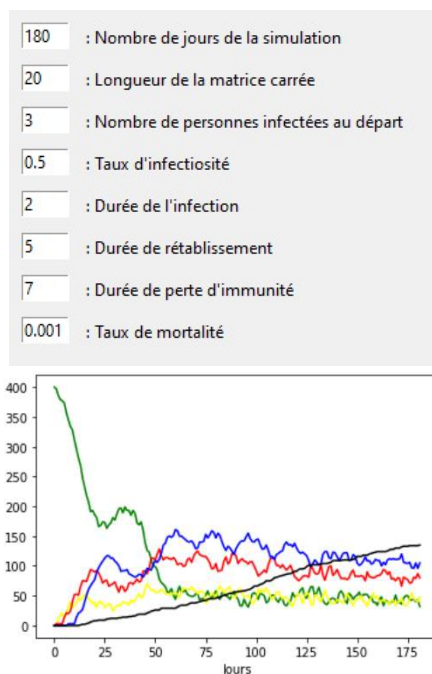
En reprenant les mêmes paramètres qu'à la première simulation, on obtient les courbes suivantes :



Matrice carré 22 par 22 montrant l'évolution de l'épidémie en temps réel

On a des courbes bien plus réalistes pour les paramètres donnés, qui changent même si l'on rentre les mêmes paramètres en entrée, et on commence même à observer les pics attendus.

Cependant, les courbes ne sont pas vraiment exploitables, on a donc changé les paramètres entrés pour le nouveau programme, et opté pour ces paramètres :

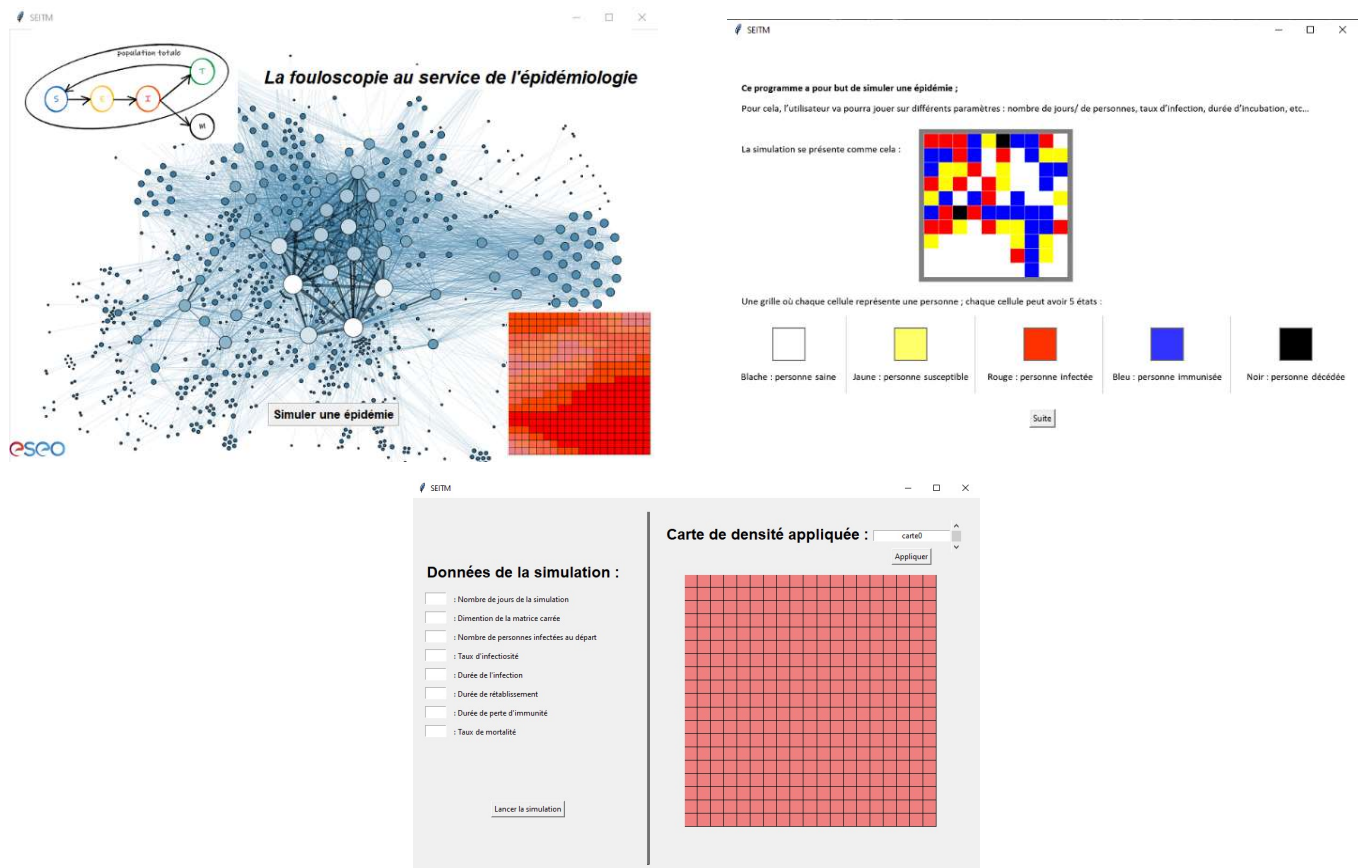


Matrice carré 20 par 20 montrant l'évolution de l'épidémie en temps réel

Les courbes obtenues sont plus satisfaisantes avec ces paramètres, et on peut même observer des "vagues", avec différents pics distincts, comme lors d'une vraie épidémie.

3) Les 2^{ème} et 3^{ème} version du nouveau programme

Dans les 2^{ème} et 3^{ème} version du programme, on a surtout amélioré l'interface graphique en Tkinter, pour la rendre plus pratique et plus agréable à utiliser :



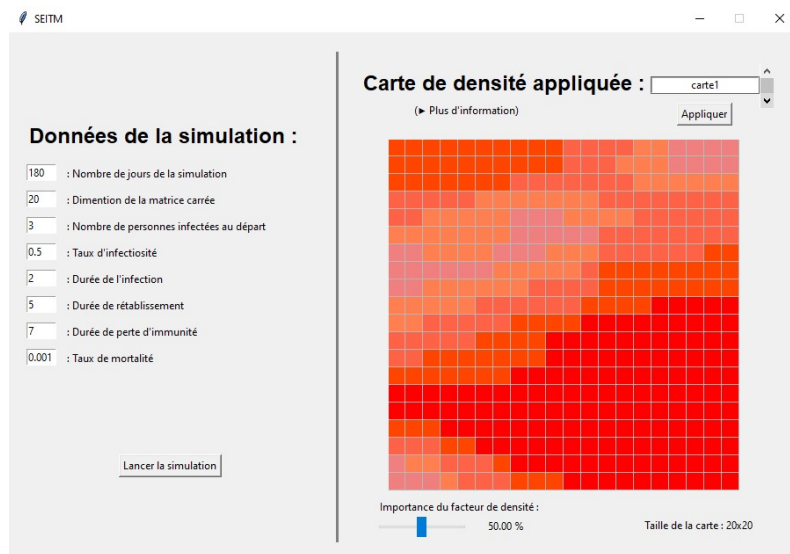
Les différents éléments présents sur ces pages ayant déjà été présentés précédemment.

4) La 4^{ème} version du programme

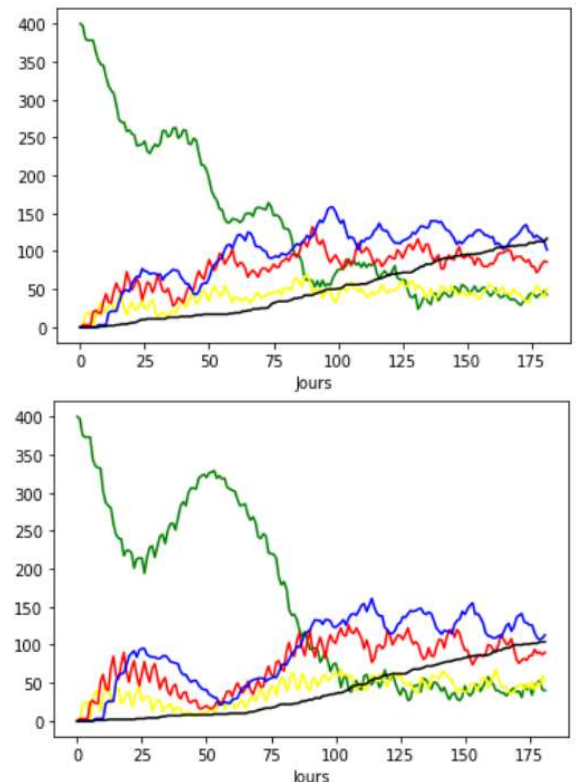
A partir du moment où le programme principal était terminé, nous avons commencé à nous concentrer sur la création d'options activables ou désactivables, dans le but d'améliorer le nombre de cas de figures simulables par le programme, et ainsi nous rapprocher le plus possible de la réalité.

Dans la 4^{ème} version du programme, nous avons donc commencé par l'option de densité de population déjà présentée précédemment

Un exemple avec les mêmes paramètres, la carte 1 appliquée, et une importance du facteur de densité fixée à 50% :



(Plus la case est rouge sur la carte, plus la densité est élevée)



Pour deux simulations différentes avec les mêmes paramètres, on remarque certains changements par rapport à la simulation sans l'option de densité de population, sans pour autant changer la tendance de chacune des courbes.

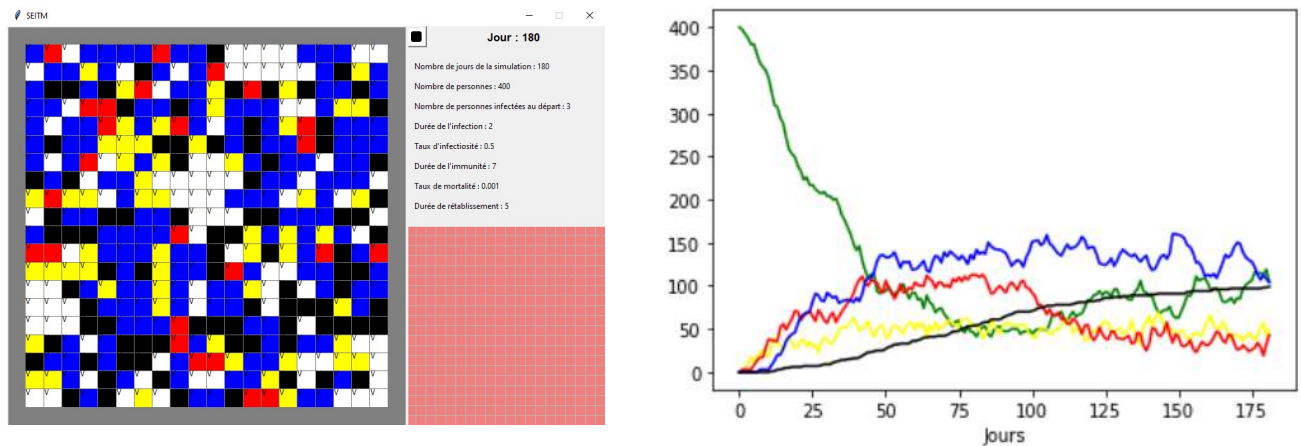
5) 5^{ème} et dernière version du programme

Après la densité de population, qui était la plus grosse option à ajouter, nous avons ajouté tous les autres facteurs codés séparément dans un seul et même programme final, en plus de quelques améliorations d'ergonomie tel qu'un bouton pour relancer une simulation, ou des valeurs pré-rentrees.

Les différents facteurs ajoutés étant :

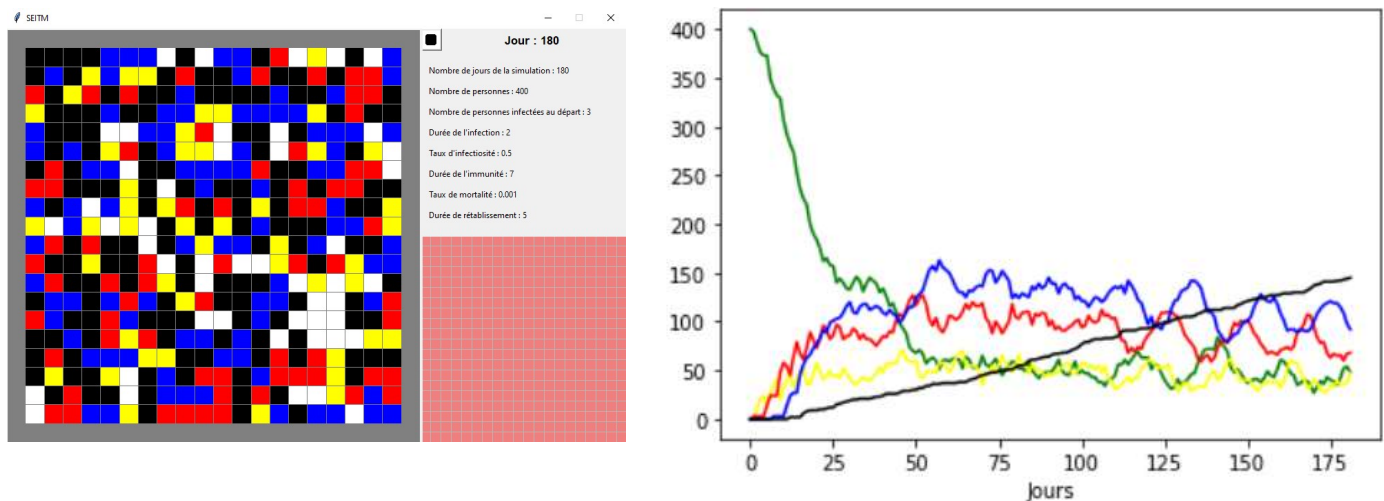
- La présence / découverte ou non d'un vaccin contre la maladie
- La prise en compte ou non de la capacité hospitalière
- La possibilité ou non pour la maladie de muter

On fait une simulation avec l'option Vaccin activée et toujours les mêmes paramètres entrés :



On observe que si le vaccin est découvert à la moitié de la simulation (ici, au 90^{ème} jour), à la fin de la simulation, quasiment toute la population est vaccinée à la fin de la simulation. De plus, on observe au niveau des courbes qu'à partir du 90^{ème} jour, le nombre de personnes infectées et infectieuses diminue progressivement pour se stabiliser vers le 125^{ème} jour. De même, le nombre de personnes mortes n'augmente quasiment plus au niveau du 125^{ème} jour.

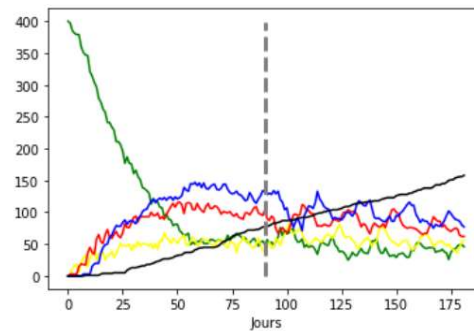
On fait une simulation avec l'option Tension Hospitalière activée, et toujours les mêmes paramètres entrés :



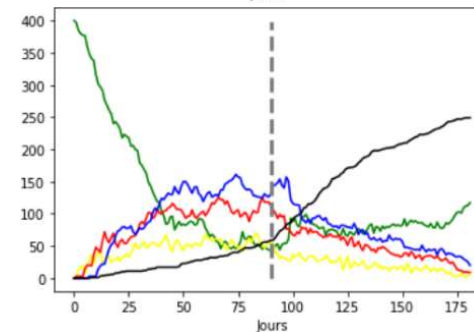
On observe que lorsque l'on prend en compte la capacité hospitalière, le nombre de morts augmente lorsque l'on observe des pics pour la courbe rouge. De ce fait, le nombre de morts à la fin de la simulation est plus élevé que dans les autres simulations

On fait une simulation avec l'option de activée, et toujours les mêmes paramètres entrés :

Nombre de jours de la simulation : 180	(Mutation : Jour 90)
Nombre de personnes : 400	
Nombre de personnes infectées au départ : 3	
Durée de l'infection : 2	→ 3
Taux d'infectiosité : 0.5	→ 0.51
Durée de l'immunité : 7	→ 6
Taux de mortalité : 0.001	→ 0.0100
Durée de rétablissement : 5	→ 5



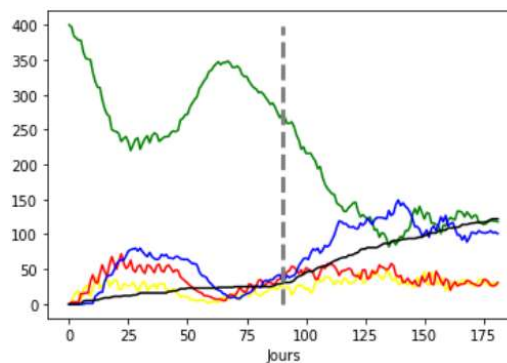
Nombre de jours de la simulation : 180	(Mutation : Jour 90)
Nombre de personnes : 400	
Nombre de personnes infectées au départ : 3	
Durée de l'infection : 2	→ 1
Taux d'infectiosité : 0.5	→ 0.31
Durée de l'immunité : 7	→ 6
Taux de mortalité : 0.001	→ 0.0410
Durée de rétablissement : 5	→ 4



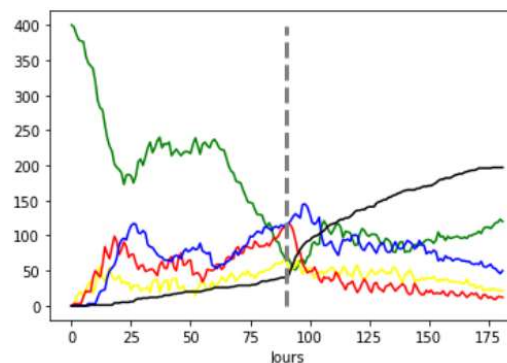
On observe qu'une mutation peut complètement changer l'évolution de la maladie, et donc des courbes.

On a enfin, fait deux simulations finales, avec toutes les options activées en même temps, et toujours les mêmes paramètres en entrée :

Nombre de jours de la simulation : 180	(Mutation : Jour 90)
Nombre de personnes : 400	
Nombre de personnes infectées au départ : 3	
Durée de l'infection : 2	→ 1
Taux d'infectiosité : 0.5	→ 0.28
Durée de l'immunité : 7	→ 6
Taux de mortalité : 0.031	→ 0.0310
Durée de rétablissement : 5	→ 4



Nombre de jours de la simulation : 180	(Mutation : Jour 90)
Nombre de personnes : 400	
Nombre de personnes infectées au départ : 3	
Durée de l'infection : 2	→ 3
Taux d'infectiosité : 0.5	→ 0.31
Durée de l'immunité : 7	→ 7
Taux de mortalité : 0.081	→ 0.0810
Durée de rétablissement : 5	→ 4



On remarque que, dans la première simulation, malgré la nette augmentation du taux de mortalité du à la mutation de la maladie, la vaccination permet tout de même de ralentir la progression de la maladie, et de limiter le nombre de morts.

Dans la seconde simulation, l'augmentation significative du taux de mortalité cause un bond du nombre de morts, qui se stabilise par la suite grâce à la vaccination qui se développe plus lentement.

V – Conclusion

Pour résumer le travail accompli au cours de cette année : L'objectif était de créer un logiciel permettant à quiconque d'apprécier l'évolution d'une épidémie et de comprendre l'impact de la foule sur le comportement de l'épidémie.

Afin de répondre à cette objectif une longue phase de réflexion était nécessaire pour comprendre comment allier un algorithme fonctionnel et une interface graphique simplifié rendant l'utilisation possible pour tout le monde.

Ce dossier, qui est un condensé de tout ce que nous avons appris cette année est donc l'accomplissement de notre objectif et une réponse à la problématique posée en première année : La foulescopie sert l'épidémiologie en lui apportant sans cesse de nouveaux paramètres pour affiner les simulations, et de nouveaux outils permettant de modéliser de plus en plus précisément des foules d'individus, qui sont à la base de la propagation d'une épidémie.

Pour conclure ce dossier, il est important de rappeler que malgré tous les paramètres que l'on peut prendre en compte dans les simulations, ce ne sont, comme leur nom l'indique, que des simulations.

Elles ne permettent que de faire une hypothèse plus ou moins précise de ce qui va se passer selon les conditions que l'on pose, et non une prédiction exacte, car il y aura toujours une part d'aléatoire, d'imprévu peu importe la situation.

Malgré tout, ces simulations nous permettent tout de même de voir l'importance de certains paramètres et leur influence sur l'évolution d'une épidémie, nous permettant ainsi d'anticiper et de limiter les dégâts d'une épidémie en adaptant notre réponse, comme pendant la crise du Covid-19.

VI – Bibliographie

Epidémiologie : <https://docs.idmod.org/projects/emod-hiv/en/latest/model-seir.html>

Foulescopie : [continuum-crowds.pdf \(washington.edu\)](https://continuum-crowds.pdf.washington.edu)

<https://www.thelancet.com/journals/lancet/article/PIIS0140673608609782/fulltext>

[Qu'est ce que la Foulescopie ? - Carnets de foulescopie \(mehdimoussaid.com\)](https://mehdimoussaid.com)

http://uel.unisciel.fr/mathematiques/sys_diff/sys_diff_ch02/co/apprendre_ch2_03.html

https://www.wku.edu/da/covid-19-research/excel_sir_model.php

[Outbreak — Melting Asphalt](#)

[GitHub - julienbordet/spread: Python disease spreading and visualisation](#)

<https://covid.idmod.org/#/ToolsInstituteForDiseaseModeling/covasim>: COVID-19 Agent-based Simulator (Covasim): a model for exploring coronavirus dynamics and interventions

