Épidém.io

Sprint 2 : du 16/10/2025 au 20/11/2025 (35 jours)

Travail prévu

Athène

1. Ajout des états : sain, infecté, immunisé, mort
2. Ajout des paramètres simples : nombre initial d'infectés, probabilité de transmission…
3. Mise en place de la propagation (première étape sans déplacement)
4. Gestion de la mortalité, immunité après guérison (oui/non), immunodéprimés…
5. Premier déplacement aléatoire temporaire simple pour chaque personne (mouvement limité par les bords) pour tester pour la contagion
6. Organisation des données collectées par itération pour le module statistiques

Adrien

1. Ajout de la visualisation dans l'interface de la simulation
2. Ajout de la partie extraction et visualisation des données dans l'interface
3. Liaison entre le modèle et l'interface : toile affichant les points colorés selon l'état
4. Rafraîchissement automatique de la toile toutes les x seconde(s) (approche à déterminer) avec mise à jour des paramètres
5. Boutons fonctionnels pour lancer, mettre en pause et réinitialiser
6. Info bulle pour que les utilisateurs puissent comprendre exactement à quoi correspondent les paramètres

Travail réalisé

Athène

Nous avions mal compris les attentes du projet, notamment par rapport au chiffrement du temps à passer sur le projet, qui nous semblait être d'environ 30h/personne sur le semestre. Cependant, lors de la dernière séance, il nous a été indiqué que cette estimation était très au-dessus de ce qui avait été estimé pour l'évaluation. En plus de cela, le temps de chaque étape avait été surestimé lors du rendu du cahier des charges. En effet, comme c'est le premier projet de ce type que nous réalisons, plusieurs tâches étaient plus simples que ce que nous avions en tête. Ces nouvelles informations sur les attentes sur le projet nous ont amené à réorganiser les tâches et les objectifs. Les missions de ce sprint ont donc été réévaluées.

Pour rééquilibrer la charge de travail, nous avons ajouté dans les missions liées à chaque sprint le fait d'inclure la documentation de format pdoc et de faire les tests liés aux nouveaux ajouts avec ptest, comme indiqué à la fin de la dernière séance. Il n'y a pas eu besoin de changer plus la répartition des tâches sur les différents sprints, car le nombre d'heures estimé pour les tâches correspond à celui voulu dans le cadre de la licence.

Cette approche nous permet de continuer dans la direction que nous voulions dans notre projet pour obtenir le logiciel prévu, mais de réorganiser les tâches pour équilibrer la charge de travail.

Voici les tâches qui avaient déjà été réalisées lors du sprint précédent parce qu'elles étaient involontairement comprises dans d'autres tâches :

1. Ajout des états : sain, infecté, immunisé, mort, ce qui est traité dans la classe Personne grâce à un attribut lié à chaque personne. On lie aussi une couleur à la personne en fonction de son état (par exemple : rouge pour les morts, vert pour les sains…);
2. Ajout des paramètres simples : création d'une classe Maladie avec les attributs liés à une maladie, comme le taux de létalité, la distance d'infection, le taux de risque de transmission, s'il y a une immunité après la guérison ou non et le temps de guérison ;
3. Mise en place de la propagation : première étape sans déplacement, on a vérifié que la propagation se passe bien ;
4. Gestion de la mortalité, immunité après guérison (oui/non), immunodéprimés : ces attributs ont été ajouté à la classe Personne ;
6. Organisation des données collectées par itération pour le module statistiques : il y a une structure de dataframe lié à la classe Simulation qui stocke à chaque itération les statistiques liés (nombre de personnes saines, malades…). La structure dataframe a été choisie car c'est celle qui est utilisée pour réaliser des schémas sur matplotlib, qui est la bibliothèque la plus utilisée pour faire des représentations visuelles sur Python.

Voici donc les tâches réalisées lors de ce sprint qui étaient initialement prévues :

5. Premier déplacement aléatoire temporaire simple pour chaque personne (mouvement limité par les bords) pour tester pour la contagion : pour réaliser cette étape, on récupère pour chaque personne sa position, et on va ajouter une valeur aléatoire entre -5 et 5 en x, et faire la même chose avec une autre valeur aléatoire entre -5 et 5. Il y a des vérifications pour s'assurer que la personne ne sorte pas de la fenêtre (x ou y inférieur à 0, ou x supérieur à la largeur ou y supérieur à la hauteur). Une fois que la nouvelle position est calculée, on met à jour l'attribut de position de l'utilisateur ;

Voici les tâches qui ont été ajoutées après rééquilibrage :

6. Prise en compte de la distance de contagion et de la probabilité de transmission selon la proximité
7. Première boucle main qui initialise la dispersion d'une maladie avec une initialisation locale des paramètres pour vérifier que la dispersion se passe comme prévu. Cette étape était nécessaire car cela permet de tester les algorithmes, et qu'Adrien puisse avoir des données sur lesquelles tester ses features ;
8. Mise à jour du diagramme de classes ; -> question : comment déterminer ce qui est un attribut public ou privé ?
9. Tests sur l'implémentation initiale des classes grâce à pytest ;
10. Clarification de la documentation grâce à pdoc ;
11. Rédaction du rapport de sprint

Budget points : 50

Adrien

Dans le but de représenter au mieux la simulation et son évolution, nous avons opté pour une représentation graphique par point où chaque point représente une personne. Ces derniers ont une couleur indiquant leur état : sain, infecté, immunisé ou mort. Il a donc fallu trouver une façon de faire représenter ces points de manière à ce qu'il s'intègre dans le reste de l'interface du projet. Après quelque recherche sur la réalisation de graphiques ou de nuages de points avec la bibliothèque PyQT, j'ai pris la décision d'utiliser la bibliothèque PyQtGraph. Cette bibliothèque est vastement utilisée et recommandée lors de l'affichage de graphiques (plots) comme des nuages de point.

Concernant les tâches réalisées, toutes celles planifiées ont été effectuées :

1. Ajout de la visualisation dans l'interface de la simulation
2. Liaison entre le modèle et l'interface : toile affichant les points colorés selon l'état
3. Rafraîchissement automatique de la toile toutes les x seconde(s) (approche à déterminer) avec mise à jour des paramètres
4. Boutons fonctionnels pour lancer, mettre en pause et réinitialiser
5. Textes descriptifs qui s'affichent lors du survol d'un paramètre pour que les utilisateurs puissent comprendre exactement à quoi ils correspondent.
6. Ajout de la partie extraction et visualisation des données de la simulation en temps réel dans l'interface

Travail prévu pour le prochain sprint

Objectif : algorithme de déplacement des personnes (foules) et amélioration de l'interface

Athène

1. Tests de plusieurs modèles de déplacement (par exemple : aléatoire simple, inertiel, avec évitement, Boids ajusté aux foules humaines…)
2. Comparaison leurs impacts sur la propagation, leur fidélité par rapport au phénomène à représenter et la stabilité visuelle
3. Implémentation de l'approche choisie
4. Mise à jour de la documentation
5. Mise en place des tests

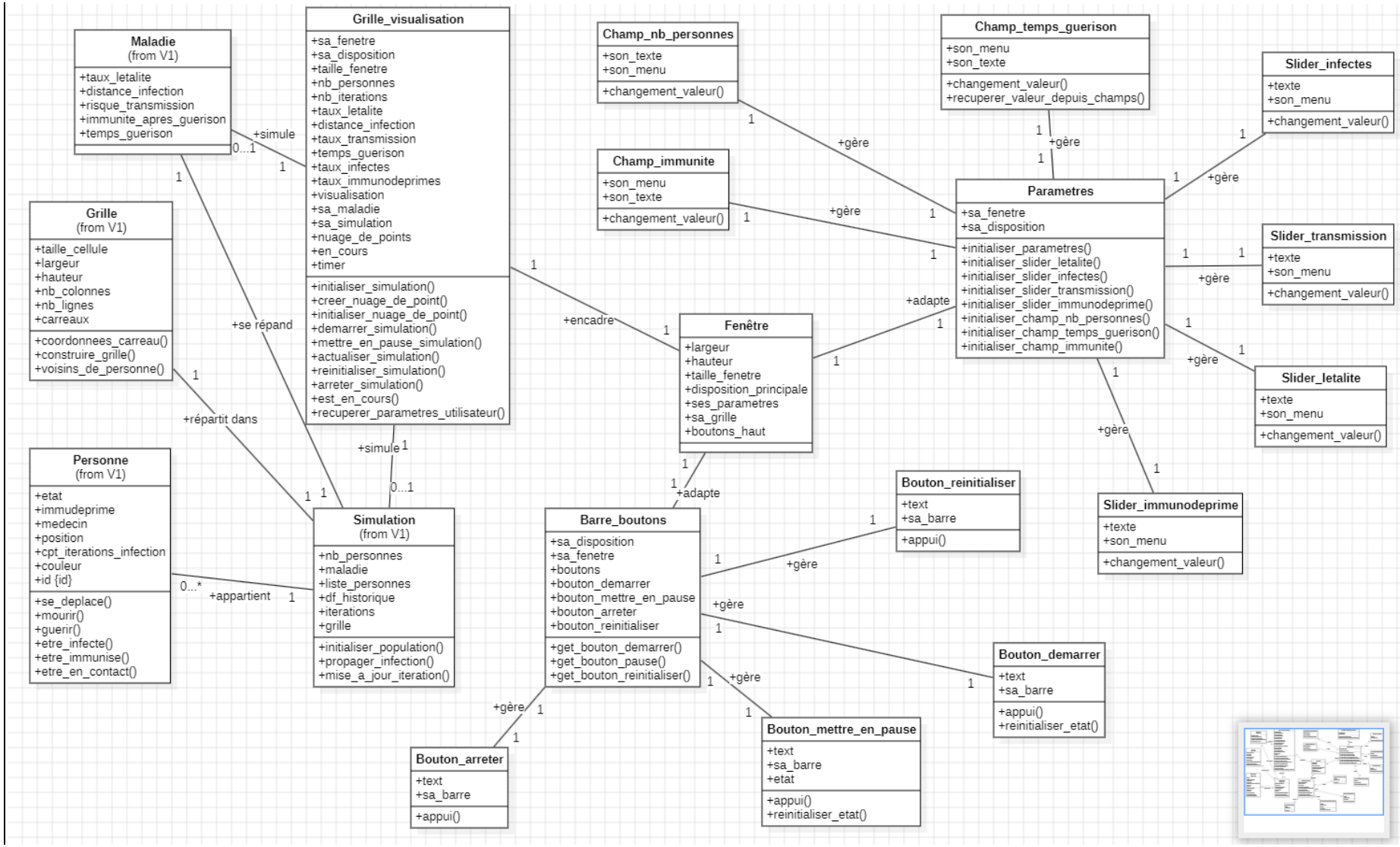
Adrien

1. Documenter les différentes classes du package view
2. Ajouter des tests de non régression
3. Améliorer l'interface et corriger les bugs éventuels
4. Adapter la toile représentant la simulation en fonction de la taille des personnes
5. ajout de cercles rouges autour des points pour montrer la distance d'infection entre deux points lorsque l'utilisateur règle la distance d'infection pour qu'il puisse bien se représenter la distance

Mise à jour des diagrammes

Diagramme de Gantt

Diagramme de classes


Maladie
(from V1)
+taux_letalite
+distance_infection
+risque_transmission
+immunite_apres_guerison
+temps_guerison
Grille_visualisation
+sa_fenetre
+sa_disposition
+taille_fenetre
+nb_personnes
+nb_iterations
+taux_letalite
+distance_infection
+taux_transmission
+temps_guerison
+taux_infectes
+taux_immunodeprimes
+visualisation
+sa_maladie
+sa_simulation
+nuage_de_points
+en_cours
+timer
+initialiser_simulation()
+creer_nuage_de_point()
+initialiser_nuage_de_point()
+demarrer_simulation()
+mettre_en_pause_simulation()
+actualiser_simulation()
+reinitialiser_simulation()
+arreter_simulation()
+est_en_cours()
+recuperer_parametres_utilisateur()
Champ_nb_personnes
+son_texte
+son_menu
+changement_valeur()
Champ_temps_guerison
+son_menu
+son_texte
+changement_valeur()
+recuperer_valeur_depuis_champs()
Slider_infectes
+texte
+son_menu
+changement_valeur()
Champ_immunite
+son_menu
+son_texte
+changement_valeur()
Parametres
+sa_fenetre
+sa_disposition
+initialiser_parametres()
+initialiser_slider_letalite()
+initialiser_slider_infectes()
+initialiser_slider_transmission()
+initialiser_slider_immunodeprime()
+initialiser_champ_nb_personnes()
+initialiser_champ_temps_guerison()
+initialiser_champ_immunite()
Slider_transmission
+texte
+son_menu
+changement_valeur()
Grille
(from V1)
+taille_cellule
+largeur
+hauteur
+nb_colonnes
+nb_lignes
+carreaux
+coordonnees_carreau()
+construire_grille()
+voisins_de_personne()
Fenêtre
+largeur
+hauteur
+taille_fenetre
+disposition_principale
+ses_parametres
+sa_grille
+boutons_haut
Slider_letalite
+texte
+son_menu
+changement_valeur()
Personne
(from V1)
+etat
+immudeprime
+medecin
+position
+cpt_iterations_infection
+couleur
+id {id}
+se_deplace()
+mourir()
+guerir()
+etre_infecte()
+etre_immunise()
+etre_en_contact()
Simulation
(from V1)
+nb_personnes
+maladie
+liste_personnes
+df_historique
+iterations
+grille
+initialiser_population()
+propager_infection()
+mise_a_jour_iteration()
Barre_boutons
+sa_disposition
+sa_fenetre
+boutons
+bouton_demarrer
+bouton_mettre_en_pause
+bouton_arreter
+bouton_reinitialiser
+get_bouton_demarrer()
+get_bouton_pause()
+get_bouton_reinitialiser()
Bouton_reinitialiser
+text
+sa_barre
+appui()
Slider_immunodeprime
+texte
+son_menu
+changement_valeur()
Bouton_demarrer
+text
+sa_barre
+appui()
+reinitialiser_etat()
Bouton_arreter
+text
+sa_barre
+appui()
Bouton_mettre_en_pause
+text
+sa_barre
+etat
+appui()
+reinitialiser_etat()
+simule
0...1
1
+se répand
+répartit dans
+encadre
+gère
+adapte
+simule
0...1
0...*
+appartient
+gère
+adapte
+gère
+gère
+gère
+gère
+gère
+gère
+gère
+gère
+gère
+gère