

Simulation de population

Qualité de Développement - R4.02

Le thème de ce TP est d'améliorer une preuve de concept de simulation de population.

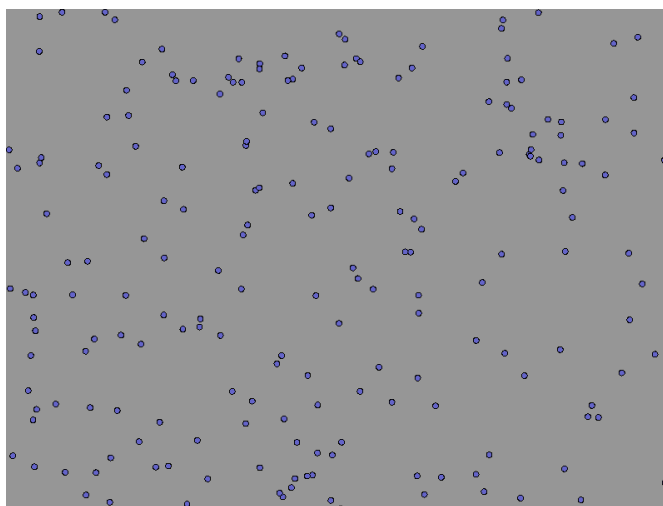
L'objectif de cette simulation de population est à terme d'évaluer l'influence de mesures sanitaires sur l'évolution d'une épidémie.

Vous trouverez un prototype dans [cette archive](#).

Vous devriez être en mesure de compiler puis lancer l'exécutable en lançant la commande `cargo run` dans un terminal à l'intérieur du projet.

La première compilation peut prendre un peu de temps puisque qu'elle requiert l'installation des bindings Rust de la bibliothèque SFML.

Vous devriez obtenir une fenêtre semblable à ceci :



Première impression

En examinant le code, remplissez le document "EvaluationQualite.md" qui se trouve dans le projet. Vous devrez l'ajouter à votre Git, **il sera évalué**.

Améliorations attendues

Fonctionnalités

À terme, on souhaite simuler la propagation d'une épidémie au sein d'une population et mesurer l'influence de mesures sanitaires sur cette propagation. Pour cela, vous êtes en charge de réaliser une application à partir du prototype qui aura les fonctionnalités supplémentaires suivantes :

1. Simulation d'une épidémie :
 - a. Ajout d'un état vis-à-vis d'une maladie : saint, malade, guéri, mort.

- b. Paramètres d'évolution de la maladie :
 - Probabilité de guérir ou de mourir quand on est malade.
 - c. Ajout de la propagation de la maladie avec des paramètres :
 - Rayon de contagion
 - Probabilité de contamination.
 - d. Déplacements aléatoires. Ajouter des changements aléatoires de direction pour les individus au cours de la simulation.
2. Récupération de données sur l'épidémie
 - Nombre de personnes dans chaque état à chaque pas de temps,
 - Bilan global, c'est-à-dire nombre final de personnes dans chaque état,
 - Durée de l'épidémie, c'est-à-dire temps qu'il faut pour que les états ne changent plus.
 3. Lecture de la configuration dans un fichier.
 - Taille de l'environnement.
 - Nombre d'entités.
 - Différents paramètres.
 4. Mesures sanitaires :
 - Port du masque : influence sur le taux de transmission.
 - Vaccination : influence sur le taux de survie et sur le taux de transmission.
 5. Amélioration de la finesse du comportement :
 - Comportement de déplacement individuel, pour pouvoir simuler des mesures de confinement.
 - Notion de lieu spéciaux où toutes les personnes doivent se rendre (supermarchés, travail, écoles).

Qualité

Cet outils de simulation vise à fournir des données fiables aux agences de gestion médicales, il est donc important que les résultats donnés soient fiables. Les requis sont amenés à évoluer.

Il faudra donc prêter un soin particulier aux critères de qualité suivants :

- Testabilité
- Maintenabilité
- Extensibilité
- Performance

Structuration

Organisez les données et leurs méthodes, sans hiérarchie de classe puisqu'il n'y a pas d'héritage en Rust, mais à l'aide de **modules**.

Une proposition de hiérarchie de modules est représentée ci-dessous :

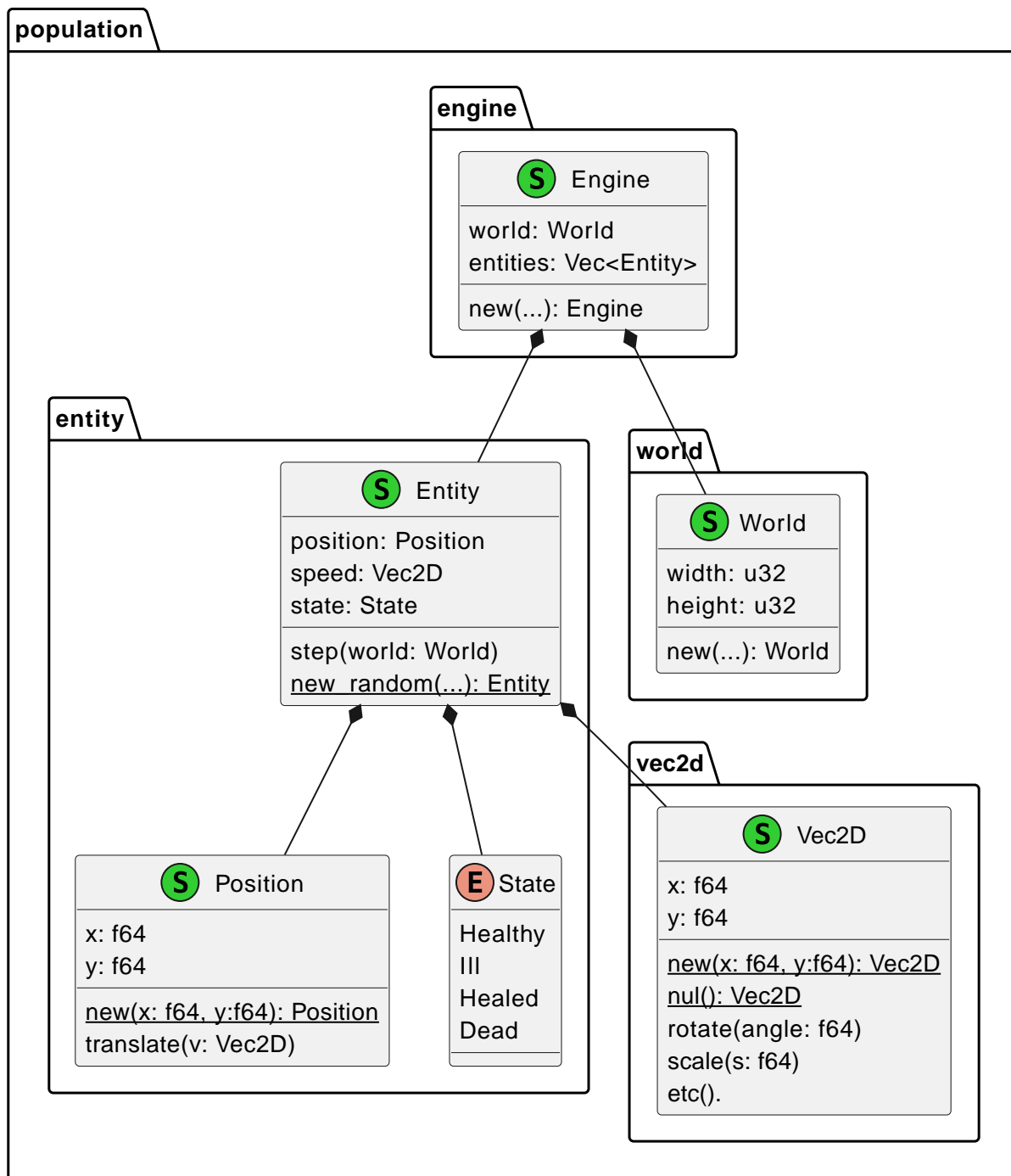


Figure 1. Population modules hierarchy. (This is a **suggestion**, you can adapt it to a better design.)

Vous n'aurez pas besoin des "Traits" pour obtenir une structure déjà cohérente. Vous pouvez vous référer au [support de cours sur la structuration](#) pour construire votre hiérarchie de modules.

À ce stade, laissez la partie "graphique" (SFML) dans le main.

Structures de données

Commencez par déclarer les structures de données dont vous aurez besoin.

Fonctions

En utilisant la méthodologie TDD, implémentez les fonctions dont vous aurez besoin.

À ce stade vous devez avoir obtenu le même fonctionnement que le prototype mais avec une qualité de code bien meilleure.

Pensez à commiter et pousser votre travail.

C'est une bonne idée de faire un "tag" sur cette version (ce commit)

Épidémie - Propagation

Ne commencez pas cette partie tant que vous n'avez pas terminé la partie précédente.

Comme première amélioration et toujours en mode TDD, faites en sorte que l'état des individus évolue en fonction de :

- Leur état courant
- L'état des autres individus dans un rayon de contamination
- Une probabilité de contamination
- Une probabilité de survie

Commitez, poussez et taguez cette version.

Production de résultats

Maintenant que vous avez une première version de simulation il est temps de produire des résultats. Faites en sorte de produire des fichiers de log contenant les différentes données intéressantes.

Vous choisirez le format du fichier judicieusement pour pouvoir générer des graphs.

Configuration depuis un fichier

Ajoutez la possibilité de configurer la simulation à partir d'un fichier. Choisissez un format de fichier pour lequel des parseurs existent, json est un bon exemple a priori.

Mesures sanitaires globales

Ajoutez la possibilité d'activer des mesures sanitaires globales :

- Port du masque : influence la probabilité de contamination lorsque des individus sont à proximité.

- Vaccination : influence sur la probabilité de contamination et sur la probabilité éde décès.

Comportements individuels

Afin de pouvoir évaluer l'impact de mesures sanitaires s'appliquant aux individus comme le télétravail ou la restriction des contacts sociaux à un groupe restreint, vous allez ajouter un comportement individuel de déplacement aux individus.

Ajout de lieux particuliers dans l'environnement

Vous ajouterez dans l'environnement différents types de lieux :

- Domicile
- Travail
- Magasins
- Lieu publique
- École (pour certains individus)

Vous ferez ensuite en sorte que les individus possèdent une liste de lieux et naviguent entre ces lieux pour simuler les croisements plus fréquents que lors de déplacements aléatoires. Procédez de manière incrémentale.

Mesures sanitaires liées aux déplacements

Ajoutez la possibilité d'activer des mesures sanitaires liées aux déplacements :

- Télétravail, les individus ne vont plus au travail.
- Confinement : seuls quelques individus vont au travail, les écoles sont fermées, les fréquences de déplacement sont réduites.

Évaluation

Le code produit sera évalué selon les critères suivants :

- Qualité du code en terme de :
 - Structuration, lisibilité et clarté du code,
 - Tests réalisés,
 - Documentation, incluant des exemples-tests lorsque cela est pertinent.
- Avancement des fonctionnalités demandées.

L'ordre des critères est pertinent : un code de meilleure qualité mais avec moins de fonctionnalités sera plus valorisé lors de l'évaluation.

Extensions "bonus" suggérées

Si vous avez terminé les fonctionnalités attendues avec un niveau de qualité satisfaisant, vous pouvez proposer des améliorations de la simulation.

Vous pouvez par exemple travailler sur :

- La génération de graphs à partir des données produites (cf. plotters).
- L'amélioration de l'aspect graphique de la simulation.
- La possibilité de lancer la simulation en mode "headless", sans l'affichage graphique.
- La parallélisation de la simulation, notamment en ce qui concerne la simulation des comportements individuels.