

SUJET DE TP NOTÉ DE JAVAFX (INTRODUCTION AUX IHM, M2105)

Quentin Couland, IUT de Laval

04/05/2020

Le code source de votre application et un rapport seront à rendre sur la plateforme UMTice. Le contenu du rapport est précisé au début du sujet.

Description générale

Lors de ce TP noté, vous allez concevoir et développer une application JavaFX. L'objectif est de vous faire manipuler les composants, layouts, etc. de JavaFX, afin de produire une application fonctionnelle, et ainsi vous montrer que les connaissances acquises sont applicables.

Pour ce faire, vous utiliserez les ressources suivantes :

- Les **cours magistraux** disponibles sur **UMTICE**
- La **documentation officielle de JavaFX** disponible ici : <https://docs.oracle.com/javase/8/javafx/api/toc.htm>
- **Scene Builder** : <https://gluonhq.com/products/scene-builder/>

Modalités d'évaluation

Seront évalués :

- Nombre de fonctionnalités correctement implémentées
- Qualité globale du code (organisation des fichiers, noms des variables et méthodes, commentaires au sein du code, indentation)
- Respect des consignes

Instructions pour le rendu

Archive

Votre archive devra être de la forme suivante :

- **Nom_Prenom_tpjavafx.zip** (Exemple : Couland_Quentin_tpjavafx.zip). Vérifiez bien le format de l'archive. Si ce format n'est pas respecté, une pénalité sera appliquée à la note finale (-1).
- A la racine de l'archive : le **rapport**, le dossier contenant **votre code** et le **.jar** de votre application.

Rapport

Le rapport devra impérativement contenir (dans l'ordre suivant) :

- Une explication succincte de la hiérarchie de votre archive (un petit schéma avec les dossiers, une explication de la logique de découpage de votre projet si elle est spéciale, etc.)
- Écrivez (et soyez honnête) ce que vous avez fait ou non. Si vous avez commencé à faire quelque-chose et que ça ne fonctionne pas, précisez-le aussi
- Un diagramme de classes

Vous êtes bien sûr libre de rajouter des choses (du contenu, des images, des dessins, des compliments, etc.), mais il faut au minimum cette base. **Le rapport comptera dans la note finale.**

Description du projet

Connaissez-vous les simulateurs d'épidémie ?

Leur objectif peut être d'anticiper, d'analyser ou de prévenir la propagation d'une épidémie au sein d'une population. Les supports sont multiples : texte, équations mathématiques, visualisation, etc. Dernièrement, et de manière assez surprenante, leur nombre a beaucoup augmenté sur le net. Étant moi-même assez curieux, je me suis lancé, en partant de zéro, dans une simulation visuelle. Vous pouvez trouver le code ici : github.com/Coul33t/ContagionSim.

On essaye !

Essayez de lancer le code, puis observez cette beauté en action. Chaque balle à l'écran représente une personne. Lorsque deux personnes se touchent, elles partent dans des directions opposées par rapport à leur trajectoire initiale.

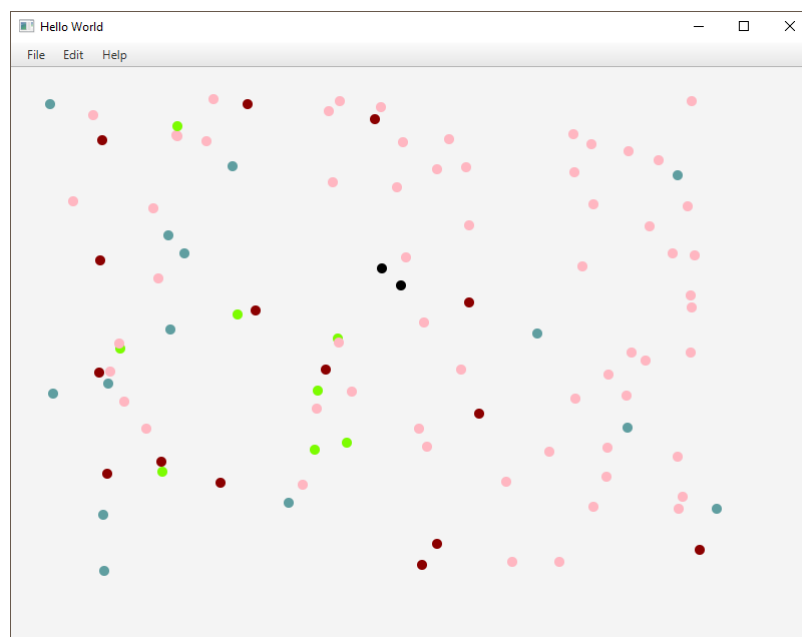


Figure 1: Un exemple du jeu de simulation de contagion

Les individus sont colorés de manière différente en fonction de leur état :

- Bleu : individu sain, non contaminé
- Vert clair : individu contaminé, mais non diagnostiqué
- Marron : individu contaminé et diagnostiqué
- Rose : individu immunisé (a été malade et a survécu)
- Noir : individu décédé

Voici les quelques règles qui régissent la simulation :

1. Au début, une personne est infectée (mais non diagnostiquée) et tout le monde se balade librement
2. Chaque collision entre un individu sain et un individu infecté entraîne la contagion de l'individu sain
3. Dès qu'un individu est diagnostiqué, il ralentit sa marche

4. Au bout d'un certain temps, l'individu guéri ou meurt
5. Si il guéri, il continue de marcher au ralenti, mais ne peut pas être réinfecté
6. Si il meurt, il ne bouge plus et ne rentre plus en collision avec les autres individus
7. Dès qu'un certain pourcentage de la population est diagnostiquée, le confinement se met en place : les gens se déplacent plus lentement

Les trucs à faire

Les points marqués **OBL** sont les choses à faire obligatoirement. Si tous ces points sont fait, la note finale tournera autour de 15/20. Les points marqués **OPT** sont des améliorations optionnelles ; optionnelles dans le sens " il ne faut pas toutes les faire ", mais il faut quand même en faire quelques-unes pour arriver à 20/20. Vous êtes libres de proposer d'autres améliorations optionnelles. Si vous le faites, n'oubliez surtout pas de préciser dans le rapport ce que vous avez fait en plus. Si vous n'êtes pas sûr que ça puisse rapporter des points en plus, n'hésitez pas à me demander (avant de commencer à le coder, de préférence).

Certaines de ces modifications vous demanderons de toucher au code que j'ai créé, d'autres seulement de toucher à l'interface. Une partie de votre travail sera bien entendu de s'approprier le code existant ; aussi, n'hésitez pas à me poser des questions. Il en va de même pour le TP noté : ce n'est pas parce-que c'est noté que je en vous aiderai pas.

OBL Changez l'interface de manière à intégrer :

- OBL** Des éléments pour modifier les paramètres suivants : nombre d'individus, vitesse des individus, taille des individus, pourcentage de personnes infectées pour déclencher le confinement, taux de contagion du virus, taux de mortalité du virus, temps avant un possible diagnostic
- OBL** Un bouton pour faire commencer la simulation
- OBL** Un bouton pour recommencer la simulation
- OBL** Ajoutez un écran d'accueil sur lequel les paramètres seront réglés (il y aura donc deux interfaces différentes : une avec les paramètres, une avec la simulation)
- OBL** Ajoutez un compteur, en temps réel, à côté de la simulation qui montre les statistiques (nombre de sain / infectés / diagnostiqués / sains / décédés)
- OBL** Ajoutez une troisième interface, qui affichera un graphique montrant l'évolution du nombres de personnes étant dans chaque état au cours du temps
- OPT** Ajoutez d'autres statistiques (vitesse moyenne des individus, nombre de contagion dans les X dernières itérations, etc.)
- OPT** Ajoutez une chance de contagion (au lieu d'une contagion systématique) lors du contact de deux individus
- OPT** Ajoutez une vitesse aléatoire aux individus, puis également une vitesse aléatoire diminuée lors du confinement (afin d'être plus représentatif du cas réel)
- OPT** Ajoutez un taux de contamination (chance d'être contaminé lors d'un contact), plutôt qu'une contamination systématique
- OPT** Créer plusieurs types de populations différentes : jeunes, âge moyen, âge avancé, chaque type pondérant différemment les risques de contagion et la vitesse de déplacement (par exemple)
- OPT** Ajouter des problèmes possibles (difficulté de déplacement, accès aux soins limités, système immunitaire affaibli, etc.)