Simulateur d'Évolution
Rapport Itération 3 - Groupe EF-4

Romain Peyremorte, Pierre Rossi, Maël Berrier, Killian Brisset, Simon Demarty

21 mai 2022

Table des matières

1 Introduction

Succinctement, notre projet est un simulateur générique d'évolution dans lequel l'utilisateur a le choix de tous les paramètres. Notre équipe est décomposée à peu près comme suit :

- Pierre : Espece
- Romain : Ressource
- Maël : Simulation
- Simon : Individu
- Killian : Mise en relation des codes précédents & affichage

Dans la suite, la partie représente des thématiques correspondant à peu près à la répartition des équipes.

2 Environnement & Ressources

Au moment de la création d'un environnement nous avons ajouté la création de montagnes et de points d'eau de telle sorte a rendre notre simulation plus réaliste.

Les montagnes et les points d'eau suivent des lois normales de telle sorte a créer un aspect aléatoire. Pour stocker la création de ces éléments, nous les avons utilisé un tableau de la même taille que l'environnement, et de valeur par défaut 0. L'eau représente la valeur -3, et les montagnes ont une valeur comprise entre 1 et 5. La création d'une methode "getType" permet de renvoyer l'altitude de chaque position.

Au départ environnement faisait apparaitre une seule ressource lorsqu'on utilisait la méthode apparaitre, mais maintenant cela a été modifié de telle sorte que cette méthode renvoie une liste de ressources.

3 Individus & Simulation

Lors de cette itération, nous avons remarqué que certaines des fonctions de déplacement d'individu ne fonctionnaient pas comme nous le souhaitions. Nous avons donc dû chercher des alternatives plus convaincantes et fonctionnelles. Cela était dû à l'architecture même des fonctions.

Toujours dans le code des individus, nous avons cherché à faciliter la mise à jour de leurs états à la fin de chaque itération. Pour servir ce but, certains calculs ont été déplacés de "Simulation" à "Individu". Ensuite, nous avons donc pu utiliser à bon escient une fonction d'itération dans la classe "Individu" afin de mettre à jour les valeurs de ces derniers. Cela facilite donc le travail de "Simulation".

Finalement, nous avons créer une nouvelle fonction : "deplacementVision". Cette fonction permet d'approcher de manière plus réaliste le comportement de chaque animal. Celui-ci cherche désormais ce dont il a besoin dans son champ de vision et se déplacera donc vers cela, au lieu de se deplacer de manière aléatoire constamment. S'il n'a besoin de rien, il se déplacera au hasard.

4 Affichage

Pour l'affichage de fond on créé un JPanel dont chaque pixel prend une couleur différente en cohérence avec la valeur que nous retourne "getType" en fonction de la position du pixel.

On réalise l'affichage des individus et des ressources sur un Jpanel qui se superpose au JPanel de l'affichage de fond. On affiche actuellement un carré de 5×5 pour chaque individu/ressource de couleur différente selon l'especes/ressource.

L'ajout du choix de l'environnement a été aussi ajouté dans le menu de choix.

5 Architecture du code

Pour choisir les espèces que l'on va ajouter et leur nombre, nous sommes passés par les systèmes Observer-Observable. Nous avons créé une map composée du nom de chaque espèce et leur nombre, comme ça nous pouvons actualiser l'affichage de l'évolution.

En outre, puisque nous avons mis en commun nos codes, nous nous sommes rendu compte de problèmes mineurs de compatibilité. Aussi, nous avons dû expliquer ce que telle ou telle fonction faisait exactement (type de retour, à quoi ce retour correspond). Nous avons donc du faire une relecture approfondie de nos codes a tous afin d'apporter des solutions à ces petits problèmes de compatibilité.

6 Améliorations futures

Il nous reste cependant plusieurs aspects du code que nous aimerions améliorer.

Pour commencer, nous avons donné des valeurs pour chacunes des espèces de manières complétement aléatoire. Cela donne des comportement qui bien que cocasse, ne reflète pas trop ceux que nous aimerions représenter. Cela ne demande pas beaucoup de temps, et relève de l'amélioration mineure, mais cela ajouterai un plus non négligeable.

Notre application ayant pour ambition d'être aussi générique que possible, il manque la possibilité pour l'utilisateur de créer ses propres espèces et environnements. Cela est faisable car il suffit de mettre a profit les constructeurs déjà existants. Cependant, ce ne sera pas facilement implémenté car il y a la gestion de l'affichage, et cela change (un peu) notre code en initialisation de simulation. En outre, lors du choix de l'environnement, nous aimerions pouvoir choisir un pourcentage d'humidité (création d'un nombre points d'eau cohérant avec ce chiffre) et faire un système analogue pour les montagnes.

Un autre problème que nous rencontrons actuellement, est l'impossibilité d'utiliser nos boutons de navigation dans la simulation de la manière que nous avions prévue. Nous ne portons pas nos effort dessus pour l'instant car cela n'est pas handicapant pour apprécier la simulation.

Avec la limite précédente, vient la gestion de l'historique de la simulation pour pouvoir "retourner dans le passée". Il s'agirait de stocker les informations de la simulation à différents moments dans des fichiers textes annexes.

Finalement, nous voudrions améliorer la présentation de notre application afin de la rendre plus attrayante graphiquement parlant.