Genetics 2.1

Introduction de la GeneticMemory

F. de Coligny, S. Muratorio - 23.5.2017

Portée

Cette documentation concerne les modules (modèles de crosissance) de Capsis s'appuyant sur la bibliothèque Genetics.

Contexte

Dans une simulaton dans Capsis, il est possible pour économiser de la mémoire en cas de simulations gourmandes de restreindre la mémorisation des étapes calculées par un module en jouant sur un *Memorizer*. Le Memorizer par défaut (*DefaultMemorizer*) garde tout en mémoire, le *FrequancyMemorizer* garde les étapes suivant une fréquence (e.g. 1 sut 5), le *CompactMemorizer* garde la première et la dernière calculée à tout moment.

Genetics propose une procédure pour retrouver un genotypable (arbre, poisson...) dans la simulation s'il n'est pas présent dans l'étape courante de simulation, en s'appuyant sur l'historique Capsis.

L'utilisaton des Memorizers autres que le DefaultMemorizer contrarie le bon fonctionnement de cette procédure.

Solution

Une adaptation est proposée pour doter la bibliothèque Genetics de sa propre mémoire, de facon à ne pas s'appuyer sur l'historique configurable de Capsis. La **GeneticMemory** est un objet unique pour toute une simulation, elle est portée par les instances de GeneticScene du module (e.g. PDGStand, LubStand, BidStand...).

La GeneticMemory est créee dans le constructeur de la classe scene, sa référence elle est ensuite partagée par toutes les scènes de la simulation.

La méthode GeneticTools.searchGee () a été déplacée dans GeneticMemory et adaptée pour chercher dans la mémoire de la bibliothèque.

Chque fois q'une scène est créée dans un module Capsis :

- à la fin d'initializeModel ()
- à chaque itération de processEvolution ()

Il faut appeler la méthode memorize (prevScene, scene) de GeneticMemory et lui passer la scène précédente ainsi que la scène créée.

Dans le cas ou la scène créée est la scène initiale, on passe null pour la scène précédente.

La méthode memorize () ajoutera dans GeneticMemory une image MemoGeneticScene de la scène, et maintiendra une table permettant de trouver la MemoGeneticScene précédente pour permettre les recherches ultérieures.

Lors d'un appel à la méthode getGenotype () sur un genotypable pour lequel le génotype n'a pas encore été spécifié, les identifiants des parents seront utilisés pour chercher les genotypables parents, d'abord dans la scène courante, puis si non trouvés, dans les scènes précédentes. A partir des génotypes des parents, on pourra construire le génotype de l'individu concerné.

Note : les parents peuvent avoir disparu s'il sont morts ou ont été coupés.

Adaptation du module PDG

Le module PhysioDemoGenetics (PDG) présente la particularité de n'avoir à tout moment q'eune instance de scène (PDGStand) dans le projet. En effet, la méthode getEvolutionBase () ne s'appuie pas sur clone (), mais renvoit toujours la même scène.

La méthode memorize (prevScene, scene) n'est donc pas suffisante pour renseigner la GeneticMemory : prevScene et scene sont le même objet.

PDG utilise une méthode dérivée : memorize (prevScene, prevDate, scene) pour renseigner correctement les informations qui permettront de naviguer dans l'historique.

Adaptation de Luberon

Lubéron a une structure plus classique que PDG : les scènes sont historisées normalement dans le projet Capsis, dans des objets distincts.

L'emploi de la méthode GeneticMemory.memorize (prevScene, scene) est dons adaptée.

Adaptation d'un modèle à Genetics 2.1, exemple de Luberon

Dans LubStand

- ajouter la variable d'instance
```
private GeneticMemory geneticMemory;
```
- Dans le constructeur
```
geneticMemory = new GeneticMemory();
```
- ajouter l'accesseur suivant
```
public GeneticMemory getGeneticMemory() {
    return geneticMemory;
    }
```

Dans LubModel

- à la fin d'intializeModel ()
```
// Store initScene in GeneticMemory // fc+som-22.5.2017
initStand.getGeneticMemory().memorize(null, initStand);
```
- à chaque itération dans processEvolution (), au moment de la création de la nouvelle étape
```
// fc+som-22.5.2017 params: prevStand, prevDate, newStand
// In PDG, prevStand = newStand (REVOLUTION, stand is not cloned)
refStand.getGeneticMemory().memorize(refStand, newStand);
```

On n'appelle pas GeneticMemory.memorize () sur les étapes d'intervention. Cela reviendrait à garder en mémoire des étapes incomplètes qui n'aideraient pas à la recherche des parents ultérieurement.

Reste à voir

Dans GeneticTools, la méthode **actualizeMultiGenotype** () utilise une méthode GeneticTools.**searchLastMultiGenotype**(gee) qui cherche un multiGenotype dans l'historique Capsis. Il faudrait étudier de l'appuyer sur la GeneticMemory.

Le champ GeneticTools.**lastKnownStep** qui servait principalement dans GeneticTools.searchGee () devrait être supprimé. Attention à son utilsation par GeneticTools.phi ().