



Mini-projet

Éléments de programmation par objets avec Java

Présentation

Ce mini-projet a pour but la réalisation d'un programme mettant en œuvre la plupart des points vus jusqu'au cours 7 (TD/TME 6). Il représente un travail d'environ 5h30. Il est demandé de proposer et d'implémenter un programme qui réalise une simulation de récolte se déroulant sur un terrain représenté sous la forme d'un tableau à 2 dimensions contenant des ressources que des agents doivent récolter.

Par exemple, cela peut être la simulation de la récolte de pollen par une colonie d'abeilles et sa transformation en miel. Les agents sont les abeilles qui se déplacent sur le terrain qui contient du pollen et produisent du miel. Le pollen et le miel sont des ressources. D'autres exemples (non exhaustifs) de simulation : récoltes de nourriture par les fourmis d'une fourmilière ou par d'autres animaux, plantations et récoltes de fruits, légumes ou céréales, construction et expansion d'une ville (époque moderne, futuriste, médiévale, romaine...) et récolte des impôts, monde imaginaire où des créatures déposent et récoltent des ressources dans un monde...

Le programme à réaliser doit respecter le cahier des charges suivant :

- la simulation se déroule sur un terrain de ressources représenté à l'aide de la classe **Terrain**.
Remarque : contrairement aux ressources, les agents ne sont pas mémorisés sur le terrain.
- la simulation gère des **Ressource** placées sur le terrain, elles ne peuvent pas être déplacées mais peuvent être retirées du terrain, et des agents qui ont la capacité de se déplacer sur le terrain en changeant de case et qui peuvent récolter des ressources ou en produire.

Les classes **Terrain** et **Ressource** sont fournies, accompagnées de leur documentation. Leur représentation UML client est donnée en Annexe.

Feuille de route

1. Choisir le thème de la simulation, identifier les agents et les ressources, et rédiger une description des interactions entre les agents, les ressources et le terrain (ce qui en fait une simulation). Noter qu'il existe d'autres classes que les agents et les ressources. Par exemple, une ruche n'est pas un agent n'est pas une ressource, mais elle a des coordonnées sur le terrain comme les agents.
Par exemple, dans la simulation d'un essaim d'abeilles, une abeille se déplace case par case sur le terrain, récolte le pollen qu'elle découvre et produit du miel qui est ensuite déposé dans la ruche.
2. Comprendre comment utiliser les classes **Terrain** et **Ressource** en étudiant la documentation fournie et en expérimentant avec la classe **TestTerrain** aussi fournie. Le terrain est représenté dans cette classe par un tableau à 2 dimensions de **Ressource**. Une ressource est caractérisée par un type et une quantité. Le type de la ressource est mémorisé sous la forme d'une chaîne de caractères (par exemple : "Pollen" ou "Miel") donnée lors de la création de la ressource. La quantité est donnée sous la forme d'un entier naturel.
3. Programmer la classe pour représenter des agents. Les agents ne sont pas mémorisés sur le terrain. Un agent possède deux attributs qui correspondent à sa position sur le terrain (numéros de ligne et de colonne). Cette classe doit contenir les deux méthodes suivantes (ainsi que tous les attributs et méthodes que vous jugerez utile de rajouter pour votre simulation) :
 - **distance(x,y)** qui rend la distance (euclidienne) séparant l'objet courant de la case de coordonnées (x,y) ;
 - **seDeplacer(xnew,ynew)** qui change la position de l'objet courant sur le terrain et le place en (xnew, ynew).
4. Écrire la classe **Simulation** qui contient un terrain, une liste ou un tableau d'agents, et une liste ou un tableau de ressources. Cette classe possède un constructeur qui place aléatoirement *m* ressources

sur le terrain et génère n agents. Vous pouvez ajouter dans cette classe la gestion des règles du jeu de votre simulation. Cette classe doit réaliser le travail suivant :

- (a) initialiser le terrain avec des ressources ;
 - (b) réaliser plusieurs récoltes et productions de ressources par les agents ;
 - (c) afficher des statistiques sur les quantités récoltées et produites.
5. Écrire la classe `TestSimulation` contenant un `main()`, dans laquelle des essais de simulation sont effectués et qui produit un log à l'écran avec les différentes actions réalisées et leurs résultats.

Remarque : vous pouvez ajouter d'autres attributs et méthodes aux classes demandées, vous pouvez aussi proposer d'autres classes.

Modalités et compte-rendu

Ce mini-projet est à réaliser seul ou en binôme.

Le compte-rendu final de ce projet sera composé d'un fichier archive (zip ou tar ou tgz) contenant les fichiers suivants :

- le fichier libreoffice `description.odt` (ou sa version word `description.doc`) complété avec les informations demandées. En particulier, toutes les classes écrites dans ce projet seront mises (copier/coller) dans ce fichier.
- un répertoire avec tous les fichiers java écrits pour ce projet. Un répertoire doc avec une documentation javadoc des classes écrites sera apprécié.

Barème indicatif

La note finale (sur 20) sera déterminée de la façon suivante :

sur 2 points	: originalité de la solution proposée
sur 5 points	: qualité de la conception et de la solution, respect des bonnes pratiques de la POO
sur 6 points	: mise en œuvre des contraintes détaillées dans le fichier <code>description.odt</code>
sur 2 points	: lisibilité du code java (indentation, noms de méthodes et de variables,...)
sur 2 points	: exemples de simulations réalisées (présentées grâce aux logs produits)
sur 3 points	: qualité du rapport (fichier <code>description.odt</code>)

Annexe : classes fournies

Trois classes sont fournies pour ce projet :

- la classe `Terrain` pour gérer un terrain sous la forme d'un tableau d'éléments. Vous ne pouvez/-devez pas modifier cette classe, pour cela elle n'est fournie que compilée (format `.class`).
- la classe `Ressource` qui sert à définir le contenu d'une case du terrain. Cette classe n'est pas modifiable et est fournie compilée.
- la classe `TestTerrain` qui fournit un exemple d'utilisation des 2 classes précédentes. Le source de cette classe est fourni.

La documentation des classes `Terrain` et `Ressource` est fournie dans le répertoire `doc/` de l'archive `MiniProjet2021.tgz`. Dans ce répertoire, il faut ouvrir le fichier `index.html` avec un navigateur internet.

Pour utiliser les 2 classes `Terrain` et `Ressource`, vous devez mettre leurs fichiers class dans le même répertoire que vos fichiers java.

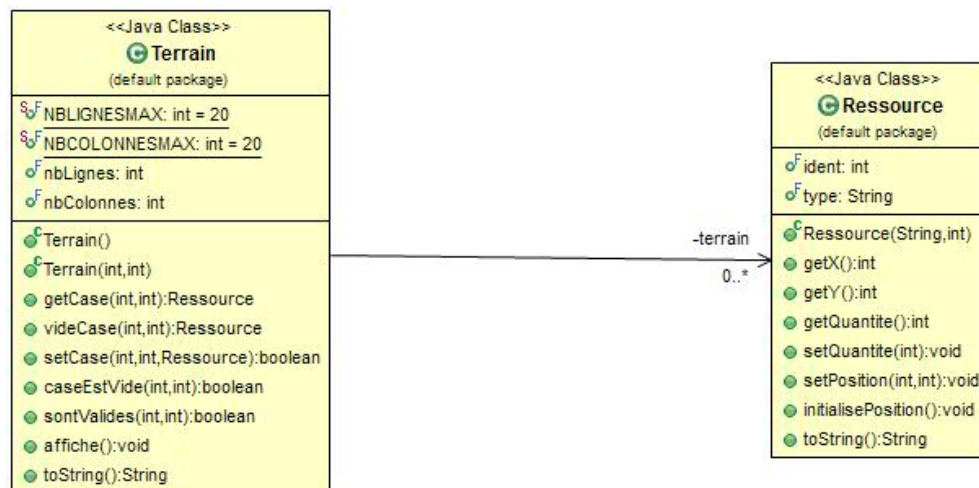


FIGURE 1 – Diagramme des classes fournies (vision client)