SIMULATION D'UNE ÉQUIPE DE ROBOT POMPIERS

équipe Teide 79

14 Novembre 2016

Introduction

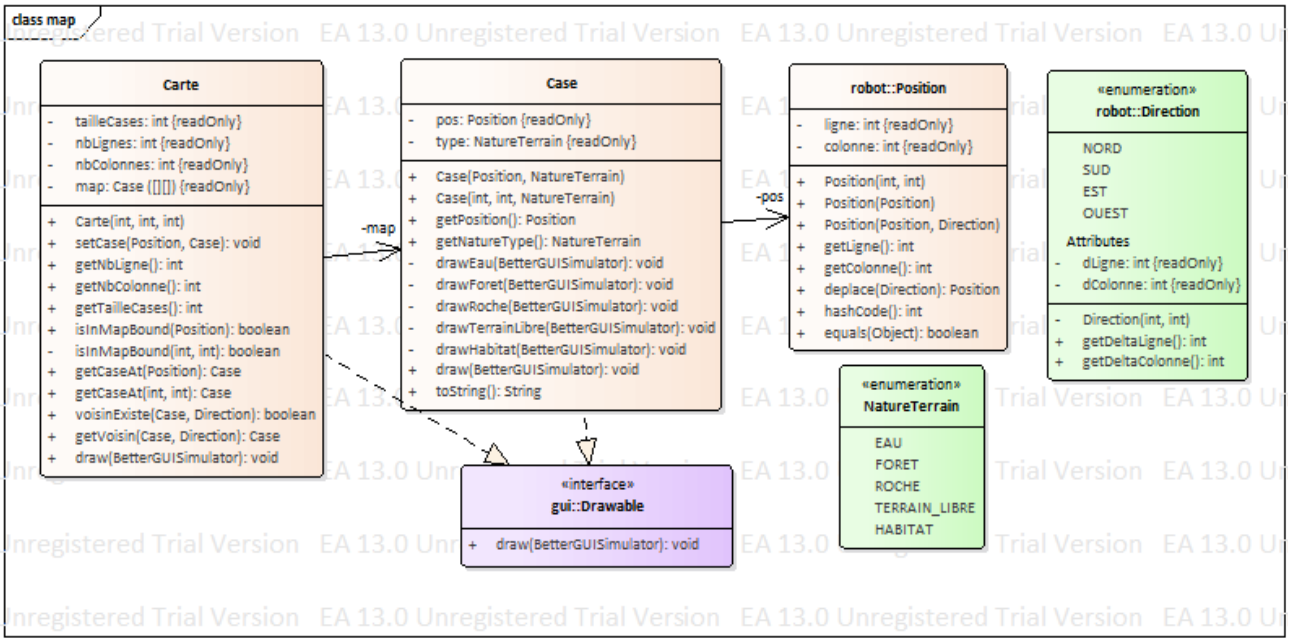
Ce projet est une application permettant de simuler une équipe de robots pompiers évoluant de manière autonome dans un environnement naturel. La simulation a pour objectif la gestion de ces robots en fonction de leurs capacités de déplacement, de leur puissance et de leur nombre. Un algorithme de calcul de trajectoire est utilisé et permet la réalisation des simulations.

Dans ce projet de conception logiciel en Java, les aspects d'héritage, d'encapsulation, l'abstraction et les collections sont notamment utilisées.

1 L'Implémentation

1.1 Le package map

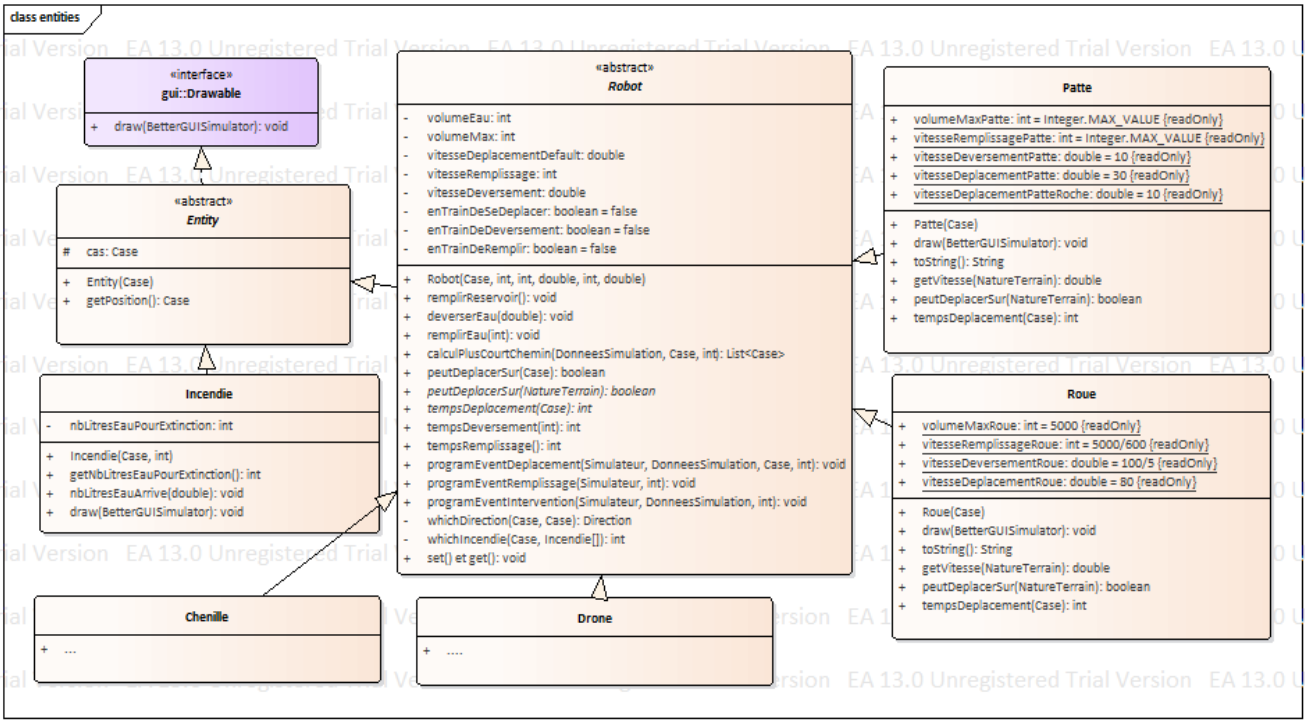
Dans ce projet la carte où se déroule la simulation est composée de Case. Un Case est définie par sa position et par son type de terrain. Une class Position a été ajouté pour encapsuler les coordonnées lignes,colonnes sur un carte. Pour l'affichage, dans la class Case et Carte on override la méthode Draw() de interface Drawable.



1.2 Le package entité

Les robots et les incendies sont des entitées qui se situent sur une case et sont affichées sur leur case par la méthode draw().

La class Robot contient des attributs représentant la vitesse, le volume d'eau, l'etat et des méthode permettant de trouver la plus court chemin, de calculer du temps de déplacement, d'ajouter des évènements, de remplir le réservoir ou de déverser de l'eau.

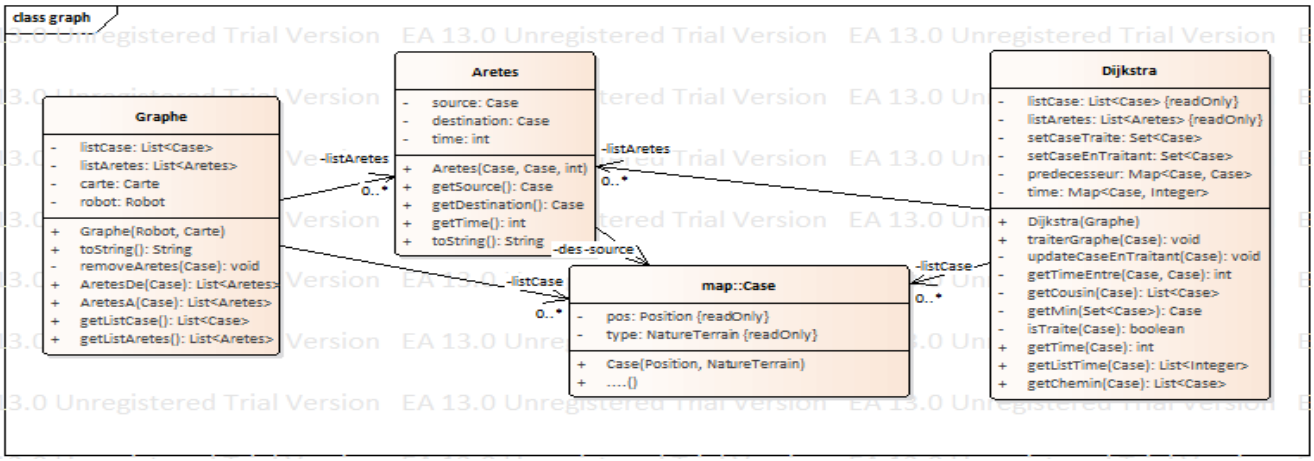


1.3 Le package graphe

Dans le projet, on utilise l'Algorithme de Dijkstra , la Graphe est créé avec :

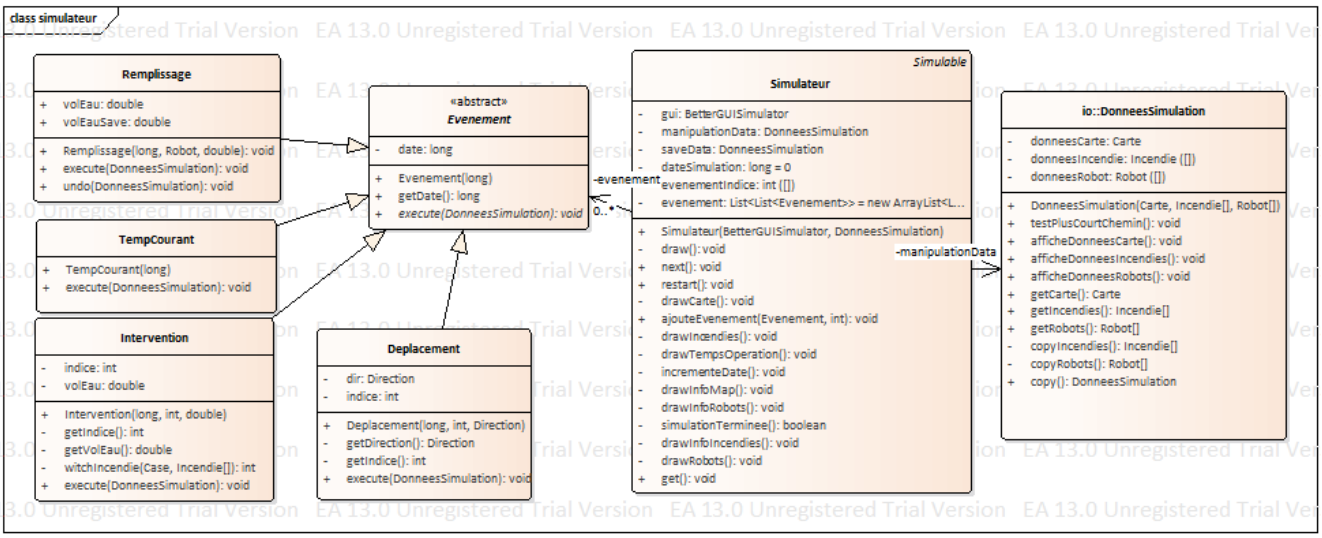
- Les sommets sont les cases sur lesquelles le robot peut se déplacer.
- Les arêtes sont les pairs des cases entre lesquelles le robot peut se déplacer
- Le poids d'une arête est égal à la durée de déplacement entre deux ses cases.

Dans la classe Dijkstra, il existe deux méthodes publiques qui renvoient des cases du plus court chemin et l'instant correspondant aux lesquelles



1.4 Le package simulateurs

L'évènement est héritée par des class comme Remplissage, TempCourant, Intervention, Déplacement qui représentent des évènements réels.



1.5 Les exceptions

Un IllegalArgumentException est lancé quand un robot sort de la carte.

```
1. public void setCase(Case c) throws IllegalArgumentException
2. public Case getCaseAt(int lig, int col) throws IllegalArgumentException
3. public Case getVoisin(Case source, Direction dir) throws IllegalArgumentException
```

Des Exceptions en lecture de donnée .

```
1. public static DonneesSimulation creeDonnees(String fichierDonnees) throws FileNotFoundException, Da
   taFormatException
2. private void lireCarte() throws DataFormatException
3. private void lireCase(int lig, int col) throws DataFormatException
4. ...
```

2. Les tests et les résultats

Les scénarios simple fonctionnent. Les fonctions de calcul du plus court chemins sont en place cependant les stratégies ne fonctionne pas correctement. Une ébauche de stratégies simple et plus avancé ont été mis en place, mais ces stratégies ne sont pas opérationnelles car il reste le réaprovisionement en eau qui pose problème.