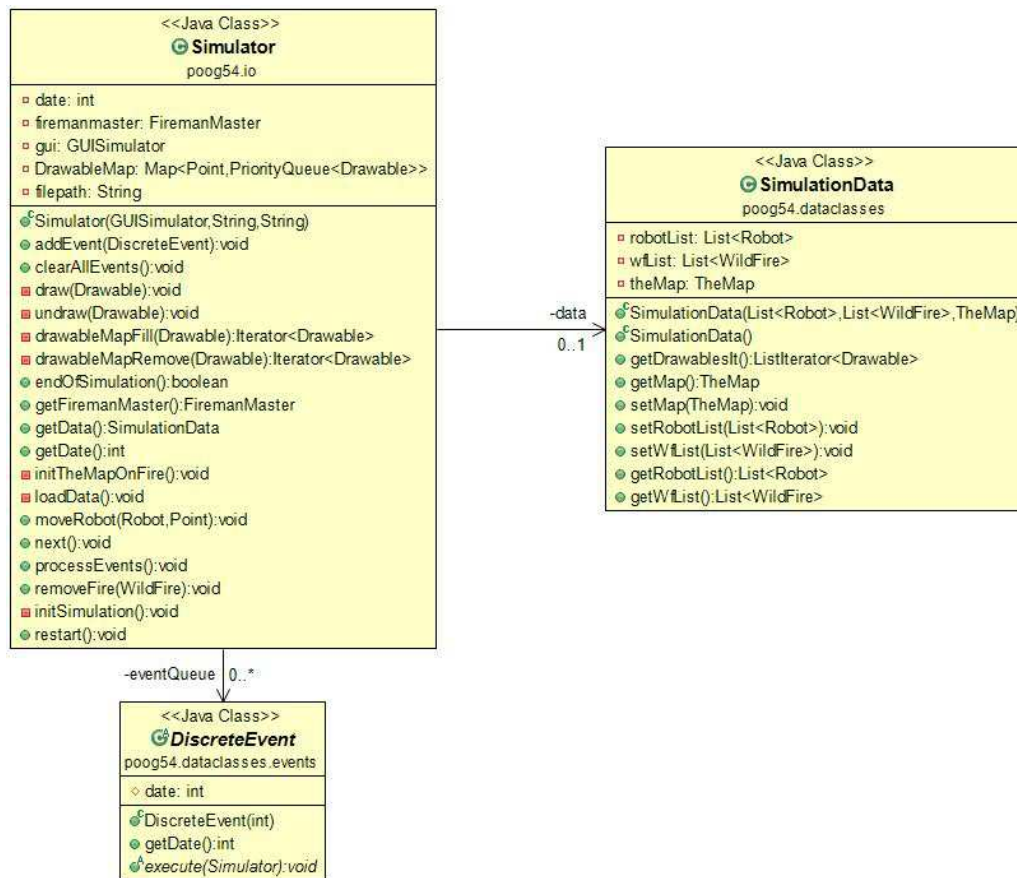
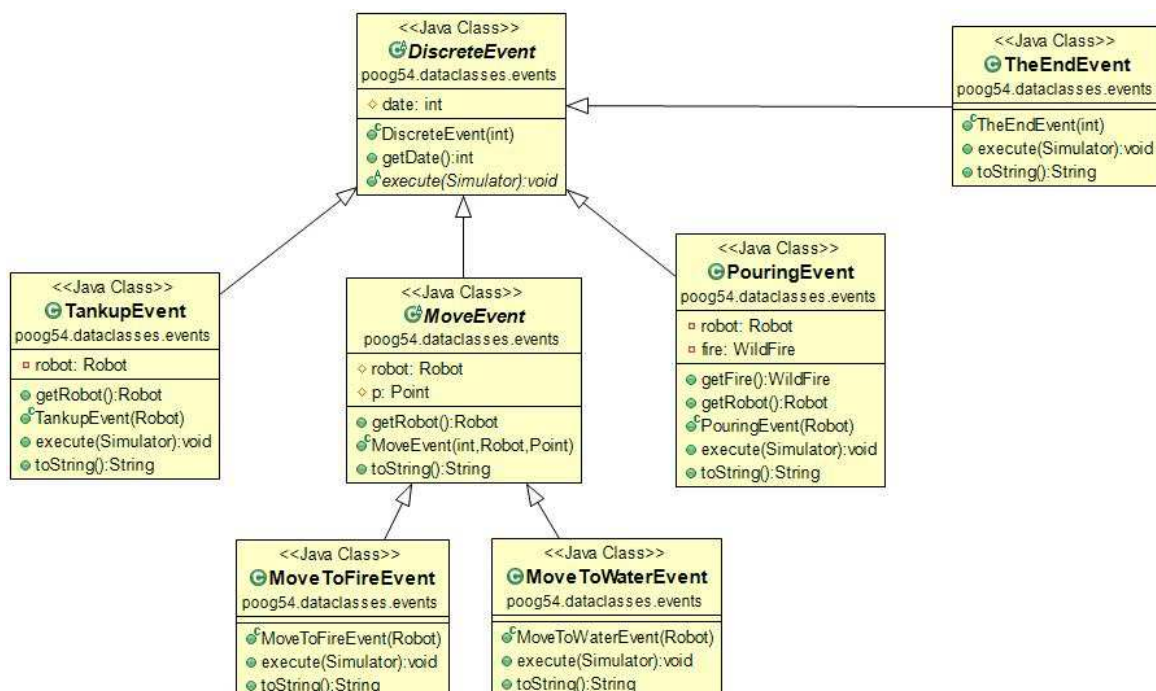


Simulateur à événements discrets

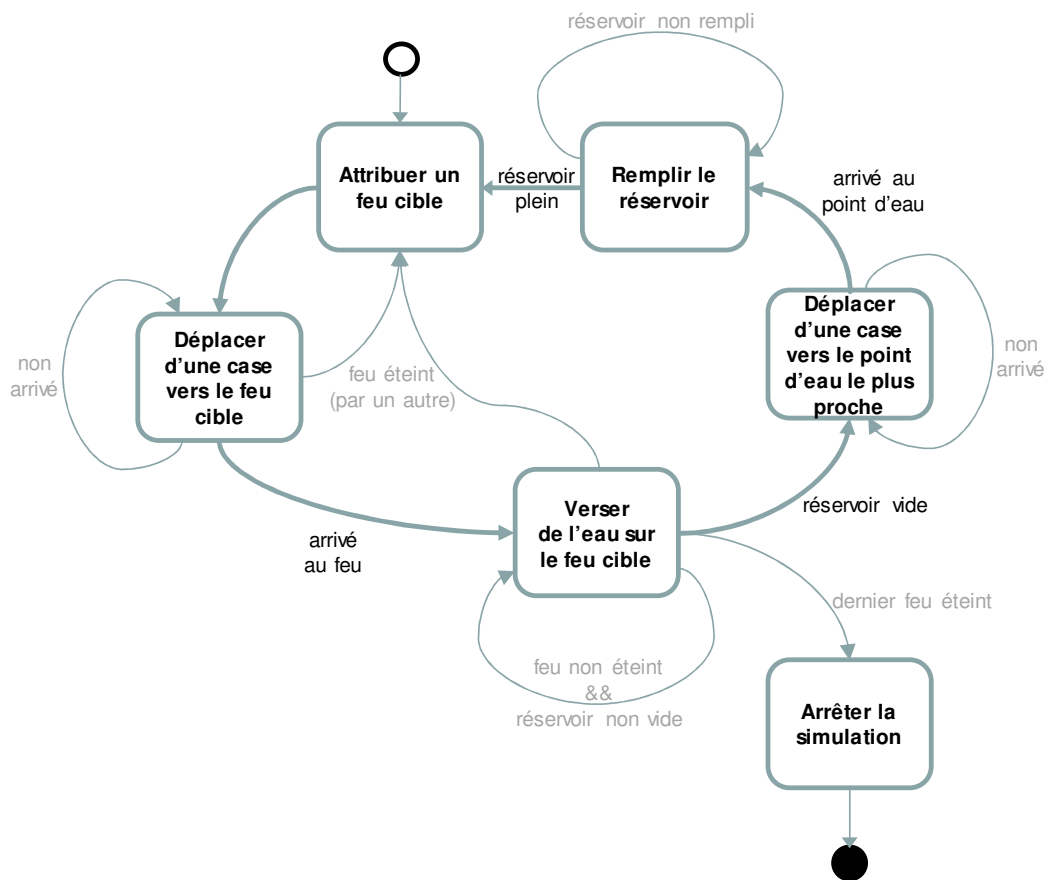
Le simulateur possède une liste *PriorityQueue* d'événements. Cette liste trie automatiquement les événements par ordre chronologique : le critère de priorité est la date des événements discrets.



La classe *DiscreteEvent* est une classe abstraite. Elle est dérivée en événements spécifiques :



Chaque robot est géré par une machine d'état dédiée. Au démarrage de la simulation, on considère que tous les robots ont leurs réservoirs pleins. La simulation commence alors par l'affectation d'un feu cible à chaque robot (cf. chapitre stratégie).



Chaque état décrit ci-dessus correspond à l'exécution de l'action associée à un événements discrets, via la méthode *execute()*.

On remarque que le déplacement d'un robot vers un feu peut être annulé si celui-ci a été éteint par un autre robot. Pour obtenir cette réactivité, les événements discrets sont créés et ajoutés au fur et à mesure de la progression du robot.

Stratégie

La stratégie permettant d'éteindre au plus vite tous les feux est portée par le chef pompier. Le chef pompier assigne un feu cible à chaque robot présent sur la carte, via la méthode *orderRobotToFire(Robot, Simulator)*. Le robot emprunte le plus court chemin pour se rendre sur les lieux, verse son eau et fait le plein (si nécessaire) au point d'eau le plus proche. Les robots connaissent la carte et se déplacent de manière autonome.

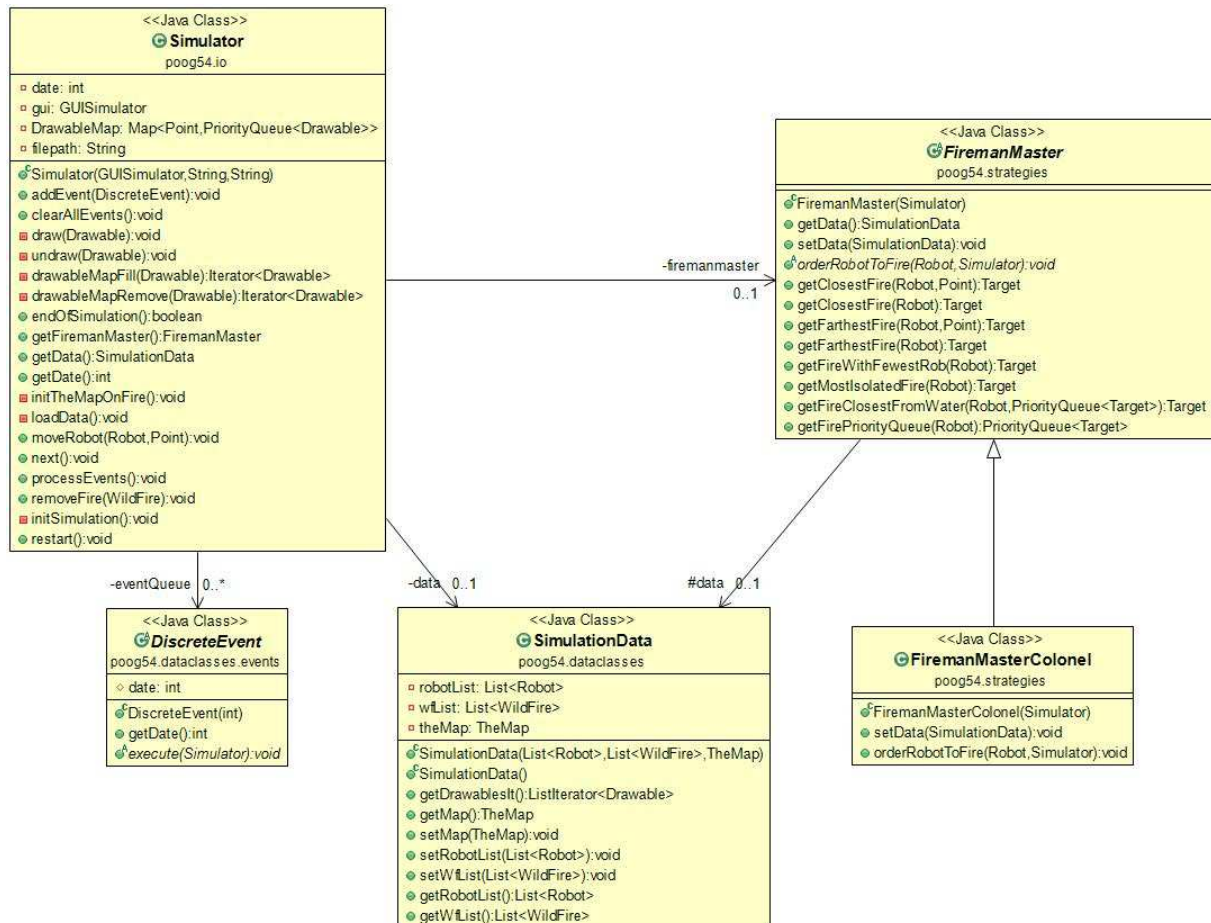
Lorsqu'un feu est éteint, tous les robots qui lui étaient affectés demandent un nouveau feu cible au chef pompier. Le cycle se répète alors jusqu'à l'extinction de tous les feux.

Lorsqu'un robot a terminé de remplir son réservoir, il demande un feu cible au chef pompier.

La classe *FiremanMaster* (chef pompier) est abstraite. Elle met à disposition quelques algorithmes de recherche de feux cibles pour les classe filles spécifiques. Chaque classe fille porte une stratégie

différente et redéfinie la méthode *orderRobotToFire(Robot, Simulator)*. Elles sont nommées par un grade : 1^{ère} classe, sergent, major... Plus le grade est élevé, plus la stratégie est évoluée. Seule la stratégie *Colonel* sera détaillée dans ce document, les stratégies de grade inférieur ayant surtout servi à la mise au point de la machine d'état.

Un chef pompier a accès à toutes les données de la simulation pour pouvoir élaborer sa stratégie et affecter les feux en fonction de leurs positions et intensités, la position des robots et leurs caractéristiques, la position des points d'eau.



Le chef pompier de classe *Colonel* choisit les feux cibles de manière différente selon le type de robot :

- Robot à pattes : affecté au feu le plus proche de sa position courante. L'objectif est de minimiser ses déplacements. Le robot à pattes est très efficace une fois sur place, mais les temps longs de marche d'approche nuisent à son rendement.
- Robot à roues ou chenilles : affecté au feu le plus proche d'un point d'eau. Ces types de robots effectuent beaucoup d'aller-retours pour remplir leurs réservoirs. Cette distance est alors bien plus grande que celle parcourue lors de l'approche initiale. L'objectif est donc de minimiser la longueur des aller-retours, quelle que soit la distance d'approche à effectuer.
- Drone : affecté à un 'gros' feu, le plus éloigné des autres robots. L'objectif est de mettre à profit leur vitesse de déplacement élevée (sur tout terrain) pour atteindre les feux qui sont les plus inaccessibles pour les robots terrestres. Cependant, leur temps de remplissage étant excessivement longs, le chef pompier veille à affecter un feu dont l'intensité dépasse la capacité du réservoir du drone. Le second objectif est donc de ne pas gaspiller l'eau des drones. S'il ne reste que des 'petits' feux, le drone est affecté au feu le plus isolé.