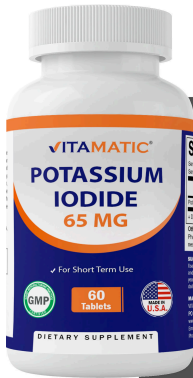


Projet IODeliv



Contexte

En cas d'accident nucléaire, les services de l'Etat (préfecture, SDIS) doivent distribuer des pilules d'iode à la population à proximité afin de saturer les amygdales et éviter la fixation de poussière radioactive. Les pilules ayant une durée de vie courte, elles doivent être distribuées peu après un accident.

Vous êtes à la tête d'IODeliv, une compagnie qui assure la livraison par drones de pilules d'iode dans une zone escarpée. IODeliv est un "middle-man" entre les services de l'Etat et les opérateurs de drone. IODeliv ne possède pas et n'opère pas de drone, mais s'attache ces services auprès

d'opérateurs de drone.

Les services de l'Etat s'adressent à vous pour livrer un ou plusieurs sacs de pilules dans des zones escarpées, avec des priorités sur ces zones (p. ex. en fonction du nombre de personnes dans une zone donnée). Les services veulent aussi un état de la mission toutes les heures, et être prévenus si un échec de délivrance a lieu. Pour ce faire, ils utilisent une page web fournie par IODeliv, dans laquelle ils peuvent sélectionner des zones sur une carte, et cliquer sur un bouton "Lancer la livraison" (vous n'avez pas à réaliser cette page web).

Votre mission

Vous devez concevoir un module logiciel qui permet d'envoyer des instructions à des opérateurs de drone pour qu'ils opèrent ces drones, à partir de bases réparties sur le territoire. Chaque drone a des capacités (charge utile, autonomie). Chaque drone doit effectuer un chemin de livraison, pour livrer plusieurs zones les unes à la suite des autres. Quand ils se posent, une personne des autorités locales (mairie) prend le nombre de pilules pour sa zone. Les drones doivent rendre compte de l'avancement de leur mission (zone atteinte à telle heure, échec comprenant les zones non livrées, succès).

Chaque opérateur de drone a ses drones tous situés dans leur local respectif. Un opérateur doit s'enregistrer auprès d'IODeliv pour pouvoir livrer les pilules.

Instructions

L'idée est de simuler une situation de client/fournisseur .

1. Donc il faut vous mettre en binôme, pour que vous vous entendiez sur ce qu'il y a à réaliser.
2. Définissez ensemble les services offerts par IODeliv à l'Etat et aux opérateurs, avec des diagrammes de cas d'utilisation, et un diagramme de classe (3/4h)

3. Définissez les services offerts par les opérateurs de drone à IODeliv en termes de (squelette de) classes (avec des `pass`), de méthodes typées avec unités, et de contrats logiciels (1/2h). **!!!! SANS DEFINIR L'IMPLEMENTATION !!!**

Ex:

```
class Drone:
    @icontract.ensure(lambda self, result: result > 0 *
ureg.second)
    def get_autonomy(self): -> pint.Quantity: pass
```

4. Puis de façon séparée :
Vous reprenez les classes que vous avez définies ensemble.
Vous implémentez les services chacun de votre côté, en les testant de façon unitaire.
5. Vous assemblez les composants ensuite et vous testez/faites marcher le système entier.
6. Déposez le code source dans l'"assignement" idoine sur e-campus

Détails d'implémentation

L'objectif de ce BE n'est **pas** d'implémenter des algorithmes ~~sophistiqués~~. Pour le calcul des chemins et l'allocation à plusieurs drones, vous pouvez utiliser un algorithme très simple sous-optimal: par exemple, s'il y a 20 zones à livrer, on assignera à un premier drone les n premières zones possibles en fonction de la capacité d'emport du drone et de la population à visiter.

Pour simuler la mission et le vol de drones (notamment la durée de vol), nous vous fournirons du code qui permettra de le faire.