

Modélisation et robotique

Exercices 3

L'objectif de ce TP est de vous permettre de prendre rapidement en main le langage Python, en écrivant de petits programmes permettant de réaliser différentes tâches dans un *micro-monde*.

Dans ce micro-monde, il y a un jardin entouré de murs de pierre, qui occupe la moitié du terrain. Le personnage principal est un jardinier, et sa tâche sera se créer un beau jardin potager. Au tout début, le micro-monde ressemble à celui de la figure 1.

On voit que le jardinier a déjà disposé dans son jardin une rangée de 11 légumes d'une façon particulière. Pour l'instant, la plupart des légumes sont éparpillés hors du jardin. Le jardinier devra les ramasser et les placer à l'intérieur du jardin.

Plus précisément, le jardinier souhaite disposer encore 4 rangées de légumes exactement comme la première rangée.

Lorsqu'il aura fini, le micro-monde devra ressembler à celui de la figure 2

Tout programme python utilisant ce micro-monde devra commencer ainsi :

```
from gardenworld import *
init('info2_1')

avance()      # avance d'une case
tournegauche() # pivote d'un quart de tour
touredroite() # pivote d'un quart de tour (autre sens)
cherche()     # cherche s'il y a un légume sous les pieds du jardinier
ramasse()     # ramasse le légume sous les pieds du jardinier
depose()      # dépose l'un des légumes ramassés
```

Ces instructions ont des abréviations : `av()`; `tg()`; `td()`; `ra()`; `ch()`; `dp()`

Ensuite, on a plusieurs instructions à disposition (en plus de l'ensemble des instructions python). Le jardinier peut pivoter sur lui-même, avancer tout droit, ramasser des légumes, et les déposer. Le jardinier peut ramasser autant de légumes qu'il veut avant de les déposer (il a un grand panier).

Question A. Promenade. Ouvrez un interpréteur python et essayez ces commandes.

- essayez `ramasse()`, `cherche()` lorsque le personnage se trouve sur des légumes. Que font et que renvoient ces fonctions ?
- Essayez `avance()` lorsque le jardinier est bloqué contre un mur ou de l'eau. Que font et que renvoient ces fonctions ?

Question B. Collecte. Si le jardinier a ramassé plusieurs légumes différents, il peut appeler la fonction `depose(nomlegume)` avec en paramètre le nom du légume à déposer. Par exemple `depose('citrouille')`. Ramassez un chou et une carotte, puis déposez la carotte à la place du chou et vice-versa.



FIGURE 1 – Le jardin



FIGURE 2 – Le jardin à la fin

Question C. Exercices sur les Fonctions, Boucles, If Créez un fichier `exo1.py` dans l'environnement python Spyder pour cet exercice.

- Écrivez une fonction `demitour()`, et testez la
- Écrivez une boucle qui appelle 10 fois la fonction `avance()`
- Écrivez une fonction `cours(x)` qui appelle x fois la fonction `avance()`
- Écrivez une fonction `toutdroit()` qui avance jusqu'à que le jardinier soit bloqué contre un obstacle
- Écrivez une fonction `ramassetout()` qui avance et ramasse les légumes jusqu'à que le jardinier soit bloqué. Cette fonction doit afficher à l'écran le nom de chaque légume ramassé.
- Écrivez une fonction `cherchelegume()` qui renvoie `True` si le jardinier est sur un légume.
- Testez vos fonctions avec un petit programme.

Question D. Exercices sur les listes Dans un terminal, lancez python et faites :

- Créez une liste vide que vous nommerez `Panier`,
- positionnez le jardinier sur un légume, et écrivez l'instruction qui rajoute à la liste `Panier` le nom de ce légume.

Question E. Programme complet. Dans un fichier `exo2.py`, avec l'environnement Spyder :

- Créez une liste vide que vous nommerez `Panier`.
- Créez une fonction `ramasseEnregistre()` qui ramasse un légume et rajoute son nom à la liste `Panier`.
- Créez la fonction `typeDeLegumes(listeLegumes)` qui prend en parametre une liste de légumes en renvoie une liste dans laquelle chaque type de légume n'apparaît qu'une fois. Par exemple : `typeDeLegumes(['chou', 'salade', 'chou', 'citrouille', 'citrouille'])` devra renvoyer `['chou', 'salade', 'citrouille']` Créer une fonction `Compte(nomLegume)` qui comptera le nombre de légumes de type `nomLegume` dans le panier.
- Créez une fonction `resume()` qui affiche le nombre de légumes de chaque espèce ramassé jusqu'à présent.
- Créez une fonction `ramasseTout()` qui ramasse l'ensemble des légumes hors du jardin, et affiche le nombre de légume de chaque type.
- Créez une fonction `ObserveRangee()` qui va aller tout droit jusqu'au prochain obstacle, en observant les légumes sous les pieds du jardinier. Cette fonction devra renvoyer une liste des légumes observés dans l'ordre. Si le jardinier est positionné en haut à gauche de son jardin, l'appel à cette fonction renverra la liste `['citrouille', 'salade', 'chou', ...]`
- Créez une fonction `RecopieRangee(listeRange)` qui prend en paramètre une liste de noms de légumes, et qui dépose chaque légume de la liste dans l'ordre.
- Assemblez le tout pour que le jardiner finalise son jardin.

1 Exercices Supplémentaires

Question F. Du chou! Créez une fonction `duChou()` qui renvoie vrai si parmi les 10 derniers légumes ramassés, il y a du chou.

Question G. Directions. Créez des fonctions `haut()`, `bas()`, `gauche()`, `droite()` qui permettent de diriger le jardinier plus facilement.