



alpha*ai*

L'EDUCATION À L'INTELLIGENCE ARTIFICIELLE

TELECOMMANDER ALPHAI

PRÉREQUIS

CONNAISSANCES

- Programmation Python
 - Boucles *while* et conditions *if/elif/else*.
 - Manipulation de *list*.
 - Créer et appeler une fonction.

MATÉRIEL

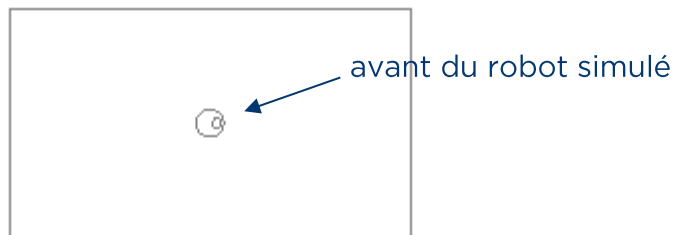
- Le logiciel AlphaI installé sur un ordinateur.
- Un ordinateur avec une carte Wi-Fi ou Bluetooth pour communiquer avec le robot.
- Un robot AlphaI.

PRISE EN MAIN DU ROBOT

Test du module alphai pour contrôler le robot simulé

Le module `alphai.py` permet d'envoyer au robot des instructions écrites en python.

Pour que le module fonctionne, le logiciel « **AlphaI** » doit être lancé (en cliquant sur l'icône correspondante ou à partir du menu démarrer). En l'absence d'une connexion avec le vrai robot, un petit robot simulé apparaît en bas à droite.



Dans votre éditeur Python favori, créez un nouveau projet à l'emplacement de votre choix et configurez l'environnement Python pour qu'il contienne le module `alphai` en tapant dans la console IPython ou un terminal :

```
pip install alphai-api
```

Cette commande va télécharger et installer le module et ses dépendances.

Ouvrez un nouveau document python ou redémarrez la console, recopiez-y puis exécutez le code suivant :

```
from alphai import *  
  
motor(30, -30, 5)
```

Chargement de toutes les fonctions du module `alphai`

Si la première ligne du programme cause une erreur, c'est que la librairie `alphai` n'a pas été installée correctement : appelez le professeur pour demander de l'aide. Si la ligne `motor(30, -30, 5)` cause une erreur, c'est probablement que le logiciel AlphaI n'a pas été démarré.

S'il n'y a pas d'erreur, vous observez dans le logiciel le robot simulé tourner sur lui-même pendant 5 secondes.

La commande `motor(left, right, duration)` envoie les valeurs `left` et `right` aux moteurs gauche et droit du robot pendant la durée `duration`, après quoi il s'arrête. Les valeurs `left` et `right` doivent être comprises entre -50 et +50. Si la durée n'est pas précisée, alors le robot continue d'avancer jusqu'au prochain appel `motor(0,0)`.

Vous trouverez à la fin de cette feuille de TP l'ensemble des fonctions incluses dans la librairie `alphai`. Consultez-les dès maintenant pour en prendre connaissance.

Connexion au robot AlphaI

- Allumez le robot (l'interrupteur se trouve en dessous). Il effectue un petit mouvement puis clignote en blanc une fois qu'il est prêt à recevoir une connexion.
- Notez le numéro du robot inscrit sur sa plaque du dessous.
- Choisissez si vous voulez vous connecter en Wi-Fi ou en Bluetooth (*nous recommandons la connexion **Bluetooth**, qui est préférable dès qu'il y a plus de 4 robots ou de nombreux réseaux Wi-Fi déjà présents dans la salle*).

Connexion Wi-Fi

- Connectez l'ordinateur au Wi-Fi du robot : cherchez le réseau Wi-Fi qui commence par ALPHAI et se termine par le numéro de votre robot : le mot de passe est identique au nom du wifi

Connexion Bluetooth

- Dans le logiciel, menu Connexions > Bluetooth, sélectionner dans la liste le nom (numéro) de votre robot.
- Si votre robot n'est pas dans la liste, cliquez sur « Mon robot n'est pas dans la liste » et suivez les instructions. Votre robot est alors ajouté à la liste où vous pouvez le sélectionner.

- Dans le logiciel, cliquez sur le bouton « connexion »  pour vous connecter au robot.

Vous voyez apparaître son niveau de batterie  en bas à droite.

Exécutez à nouveau le même code que tout à l'heure : à présent, c'est le vrai robot qui tourne sur lui-même !

Il est possible et parfois utile d'effectuer la connexion dans votre script python plutôt qu'en cliquant « connexion » dans le logiciel, en utilisant les fonctions `connect_bluetooth(n)` (en remplaçant `n` par le numéro de votre robot) ou `connect_wifi()`, ainsi que `disconnect()` pour se déconnecter. Le code dans ce cas-là se présente ainsi :

```
from alphai import *  
  
connect_bluetooth(n)  
motor(30, -30, 5)  
disconnect()
```

Cette fois c'est le vrai robot qui tourne sur lui-même !

Premier script

Vous allez à présent créer un script Python dans votre environnement de programmation favori.

Créez un nouveau fichier Python et recopiez le code suivant :

```
from alphai import *
```

Liste des imports. L'astérisque sélectionne tous les éléments du module alphai.

```
if __name__ == '__main__':  
    connect_bluetooth(n)  
    motor(30,30,1)  
    disconnect()
```

Point d'entrée du code : fonction **main**, tous les appels à des fonctions devront se trouver à l'intérieur de ce bloc main.

Puis exécutez-le : le robot avance pendant une seconde puis s'arrête.

La commande `motor(left, right, duration)` envoie les valeurs `left` et `right` aux moteurs gauche et droit. Ces valeurs doivent être comprises entre -40 et +40. Naturellement, on arrête le robot avec `motor(0, 0)`.

Vous trouverez à la fin de cette feuille de TP l'ensemble des fonctions incluses dans la librairie `alphai`. Consultez-les dès maintenant pour en prendre connaissance.

Programmation d'une séquence

Pour faciliter la suite de votre codage, définissez les cinq fonctions suivantes. Faites des fonctions très simples : chacune doit seulement réaliser un appel à la fonction `motor`.

```
def forward()
```

Fait avancer le robot tout droit.

```
def backward()
```

Fait reculer le robot.

```
def stop()
```

Arrête le robot.

```
def left()
```

Fait pivoter à gauche le robot.

```
def right()
```

Fait pivoter à droite le robot.

Enfin, créez une fonction `walk_around()` qui fait faire une séquence d'actions au robot en utilisant ces cinq fonctions. Testez-la en l'appelant dans le bloc `main`.

Programmation du téléguidage

Lecture des touches pressées

À l'aide de la fonction de l'API `wait_for_key` (voir les explications en dernière page) et d'une boucle `while` dans la fonction `main`, imprimez dans la console avec la fonction `print` les touches pressées sur le clavier. Pour éviter d'écrire une boucle infinie, prévoyez une touche spéciale qui permet de quitter la boucle, par exemple la touche « échap » (« `esc` » en anglais) :

```
keys = []
while "esc" not in keys:
    # n'oubliez pas l'indentation
```

Programmation du téléguidage

Toujours à l'aide de la fonction `wait_for_key` et d'une boucle `while`, créez une fonction `remote_control()` qui réagit aux touches directionnelles de telle sorte que le robot se déplace selon les instructions données au clavier. Pour récupérer la dernière touche enfoncée, sélectionnez le dernier élément dans la liste grâce à l'indice -1. Attention : la liste peut être parfois vide !

Pensez à faire stopper le robot lorsqu'aucune touche n'est pressée. Testez votre code en téléguidant le robot !

Pour aller plus loin

Nous allons pouvoir organiser des courses entre vos robots téléguidés, mais vous aurez aussi intérêt à améliorer votre programme en ajoutant de nouvelles commandes comme proposé ci-dessous... Ces améliorations ne sont que des propositions, n'hésitez pas à créer vos propres améliorations !

Différents virages

Créer deux nouvelles fonctions que vous pourrez utiliser lorsque la fonction `wait_for_key` renvoie une liste de plusieurs touches :

```
def forward_left()
```

Fait avancer le robot vers la gauche lorsqu'on appuie simultanément sur les flèches du haut et de gauche. Ainsi on pourra faire faire au robot à la fois des virages serrés avec `left()` (flèche de gauche seule) et des virages larges avec `forward_left()` (flèches du haut et de gauche).

```
def forward_right()
```

De même fait avancer le robot vers la droite lorsqu'on appuie simultanément sur les flèches du haut et de droite.

Freinage et boost

Ajouter des variables d'entrée à toutes vos fonctions pour pouvoir moduler la vitesse du robot, ou la force des virages, en fonction d'autres touches qui seront pressées.

Par exemple ajoutez une variable d'entrée `speed` à toutes vos fonctions (par exemple `forward(speed)`), et appelez les fonctions avec une vitesse plus faible lorsqu'on appuie en même temps sur la barre d'espace (freinage), et plus forte lorsqu'on appuie en même temps sur la touche « x » (« boost »).

Pilotez votre robot et ajustez les paramètres de vitesse pour réussir à rendre le robot le plus maniable et le plus rapide possible. N'oubliez pas de vous exercer également à doubler des concurrents !

Programmation d'une séquence

Sur la touche de votre choix, faire réaliser au robot une petite chorégraphie, par exemple pour célébrer la victoire !

Annexe : API de la librairie alphai

En programmation, la liste des fonctions accessibles d'un programme s'appelle API, pour *Application Programming Interface*, Interface de Programmation d'Application. Elle est utilisée par les programmeurs pour savoir comment interagir avec le programme. Vous trouverez l'API du module python **alphai** en suivant ce lien : [documentation de la bibliothèque alphai-api](#).

Vous l'aurez remarqué, le nom des fonctions est en anglais. Il est coutume dans le monde de la programmation d'écrire le code en anglais, pour qu'il puisse être compris par n'importe quel autre programmeur autour du monde, quelle que soit sa nationalité.