

ÉTUDIER UNE MÉTHODE DE PROGRAMMATION POUR LE DÉPLACEMENT DU ROBOT PLASTIVORE

Objectifs

Connaissances (à savoir)

- ✦ Savoir retrouver, dans le document ressource, ou ailleurs, les commandes Python utiles à la programmation.

Capacités (à savoir faire)

- ✦ Être capable de développer des compétences numériques à travers la recherche d'informations et l'exploitation de données et documents numériques ;
- ✦ Être capable de programmer la carte micro:bit.
- ✦ Être capable de recopier un programme et de le flasher sur la carte micro :bit avec le logiciel Mu Editor.

ACTIVITÉ 1 : ROBOT MAQUEEN –ÉVITER L'OBSTACLE

SA S'approprier	TBM		AN Analyser - Raisonner	TBM		RE Réaliser	TBM		VA Valider	TBM		COM Communiquer	TBM	
	MS			MS			MS			MS			MS	
	MF			MF			MF			MF			MF	
	MI			MI			MI			MI			MI	

TBM : Très Bonne Maîtrise (en autonomie et dans un contexte nouveau) MS : Maîtrise Satisfaisante (en autonomie dans un contexte connu) MF : Maîtrise Fragile (avec aide) MI : Maîtrise Insuffisante

Le robot Maqueen micro:bit est un robot contrôlé par la carte micro:bit que vous avez programmé pour afficher du texte et des images. Le robot est petit, maniable et facile d'utilisation avec beaucoup de fonctionnalités. Ce robot se programme sous Python avec le module maqueen à rajouter sur la carte.

Le but de l'activité est de programmer le robot afin qu'il évite les obstacles.

Principe de fonctionnement : robot Maqueen éviteur d'obstacle – niveau débutant

Le principe est très simple : On avance tant que le capteur de distance ne détecte pas d'obstacles.

- Si un obstacle très proche est détecté, on recule pendant 2 secondes et on tourne pendant 1 seconde
- Si un obstacle est détecté à moins de 2 cm, on introduit de l'aléatoire avec trois comportements possibles : on recule et on tourne, on tourne ou on tourne sur place.
- pour illustrer le fonctionnement du capteur de ligne, on émet un bip à chaque fois qu'une ligne est franchie (joint de carrelage par exemple).

Voici le programme de démonstration. Vous pouvez le modifier à loisir pour changer le comportement du robot.

```

1 from microbit import *
2 from maqueen import Maqueen
3 from random import randint
4
5 mq=Maqueen()
6 mq.setVitesse(50)
7 while True:
8     mq.avance()
9     pin12.write_digital(0) # LED rouge de devant droite
10    pin8.write_digital(0) # LED rouge de devant gauche
11    d=mq.distance()
12    if d<5 or d>2000:      # obstacle très proche
13        mq.recule()
14        pin12.write_digital(1)
15        pin8.write_digital(1)
16        sleep(2000)
17        mq.moteurDroit(0)
18        sleep(1000)

```

```
19 elif d<20: # obstacle proche
20     mq.son_r2d2()
21     r=randint(1,3)
22     if r==1:
23         mq.recule()
24         pin12.write_digital(1)
25         pin8.write_digital(1)
26         sleep(2000)
27         mq.moteurGauche(0)
28         sleep(1000)
29     elif r==2:
30         mq.moteurDroit(50)
31         mq.moteurGauche(0)
32         sleep(1000)
33     elif r==3:
34         mq.moteurDroit(-50)
35         mq.moteurGauche(50)
36         sleep(1000)
37
38 if pin14.read_digital() == 0 and pin13.read_digital() == 0 :
39     # detection de franchissement de ligne
40     mq.son_bip()
```



Les documents ressources sont situés après l'appel 2.

RE



Appel n°1 :

COM

Faire vérifier la programmation de la carte micro :bit avec le logiciel Mu Editor puis brancher la carte micro :bit au robot maqueen et observer les mouvement du robot.

AN



Appel n°2 :

COM

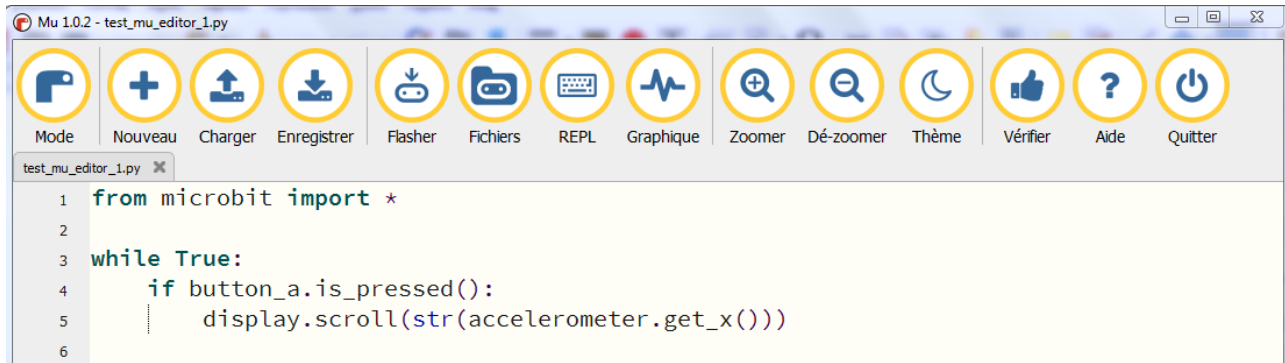
Vous devez être capable d'expliquer au professeur les différentes ligne de commande du programme.

Ressource pour paramétrer les ordinateurs et la carte micro :bit

Pour commencer, vous devrez avoir lu la ressource sur [l'installation de la librairie maqueen](#) sur le logiciel [Mu Editor](#). Remarque il faudra installer le dossier *maqueen* dans le dossier *mu_code* qui se trouve dans le dossier *utilisateurs*. Il faut installer le fichier *maqueen.py* sur votre carte micro:bit.

Ressource sur le logiciel Mu Editor

Le logiciel Mu Editor [1aIDEaide] permet une programmation en microPython, en lignes de texte. Il peut être installé sur l'ordinateur, une version portable est également disponible. Une documentation en ligne du logiciel [12] permet d'explorer les différentes fonctionnalités.



Dans le cadre d'une utilisation basique le logiciel Mu Editor présente déjà un grand intérêt : la possibilité de « Flasher » directement la carte micro:bit. Il n'est donc plus nécessaire de télécharger puis de transférer le programme au format hexadécimal sur la micro:bit, ce qui est un gain de temps et d'énergie !

Ressources pour les commandes fournies par le module maquen

- **avance(vitesse)** : avance en ligne droite. *vitesse* est un nombre entre 0 et 100. Ce paramètre est optionnel. Si non spécifié, c'est la dernière vitesse spécifiée lors de *avance()* ou *setVitesse()* qui sera utilisée.
- **recule()** : fait marche arrière.
- **stop()** : stoppe les moteurs
- **moteurDroit(vitesse)** : fait tourner la roue droite.
- **moteurGauche(vitesse)** : fait tourner la roue gauche.
- **getVitesse()** : renvoie la vitesse paramétrée par *setVitesse()* ou *avance()*
- **setVitesse()** : change la valeur de la vitesse utilisée par *avance*, *recule*, *moteur**
- **distance()** : renvoie la distance (en cm) lue par le capteur ultrason
- **son_r2d2()** et **son_bip()** : effets sonores

Ressources pour le langage Python (le micro Python en est proche) dans votre classeur ou bien visible sur votre tablette :

<http://thenard-prof.fr/wp-content/uploads/2019/12/1-MS-Document-ressources-PYTHON.pdf>