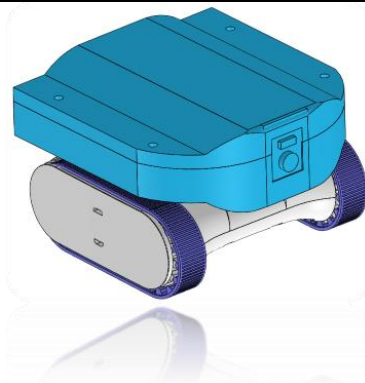


SNT	Spécialité	SNT	Niveau	Seconde
	Dimension	Scientifique et technique.	TP	Photo1
	Objectif	Découvrir la représentation d'une image par un assemblage de photos.	Durée :	1+ 1.5h
Activité		Programmer une prise de vue pour obtenir une image stable.		
Chapitre de connaissances abordées		Photographie numérique.		
Compétences développées		Identifier les étapes de la construction de l 'image finale. Expliciter des algorithmes associés à la prise de vue.		
Connaissances associées		Programmation python		
SUPPORT		Robot de surveillance Appbot Riley Virtuel		
Problématique technique		Comment fonctionnent les objets du quotidien qui nous facilitent la vie ?		
<b>1. Conditions générales</b> Ressources logicielles : <ul style="list-style-type: none"><li>- Logiciel de pilotage Mywiz avec interface Python.</li></ul> Ressources numériques : <ul style="list-style-type: none"><li>- Dossier technique du robot.</li></ul>				
<b>2. Pré requis</b>		- Avoir les notions de base du langage Python.		Activité avec ordinateur : Logiciel Mywiz installé sur le poste élève
<b>3- Conditions particulières de réalisation :</b>  <b>Professeur :</b> Présentation de la problématique, du système et du travail demandé aux élèves.  <b>Elèves :</b> Les élèves doivent : Séance 1 – 1h <ul style="list-style-type: none"><li>▪ Identifier les étapes de la construction d'une photo panoramique à partir d'un guide sur la photographie panoramique. (première partie A)</li></ul> Séance 2 – 1h30 <ul style="list-style-type: none"><li>▪ Tester avec le logiciel Myviz en mode simulé, les programmes python de test fournis.(deuxième partie B)</li><li>▪ Expliciter des algorithmes associés à la prise de vue.(deuxième partie B)</li><li>▪ Proposer une modification du programme python.(deuxième partie B)</li></ul>				
<b>4- Résultats attendus</b> <ul style="list-style-type: none"><li>- Document de synthèse cohérent.</li><li>- La fiche de formalisation complétée.</li></ul>		<b>5- Critères et Indicateurs de réussite :</b> <ul style="list-style-type: none"><li>- La rigueur dans la démarche expérimentale,</li><li>- La qualité des documents rendus,</li></ul>		

## Activités

## Programmer une prise de vue pour obtenir une image stable.

### Problématique technique

Comment fonctionnent les objets du quotidien qui nous facilitent la vie ?

### Organisation de la séance

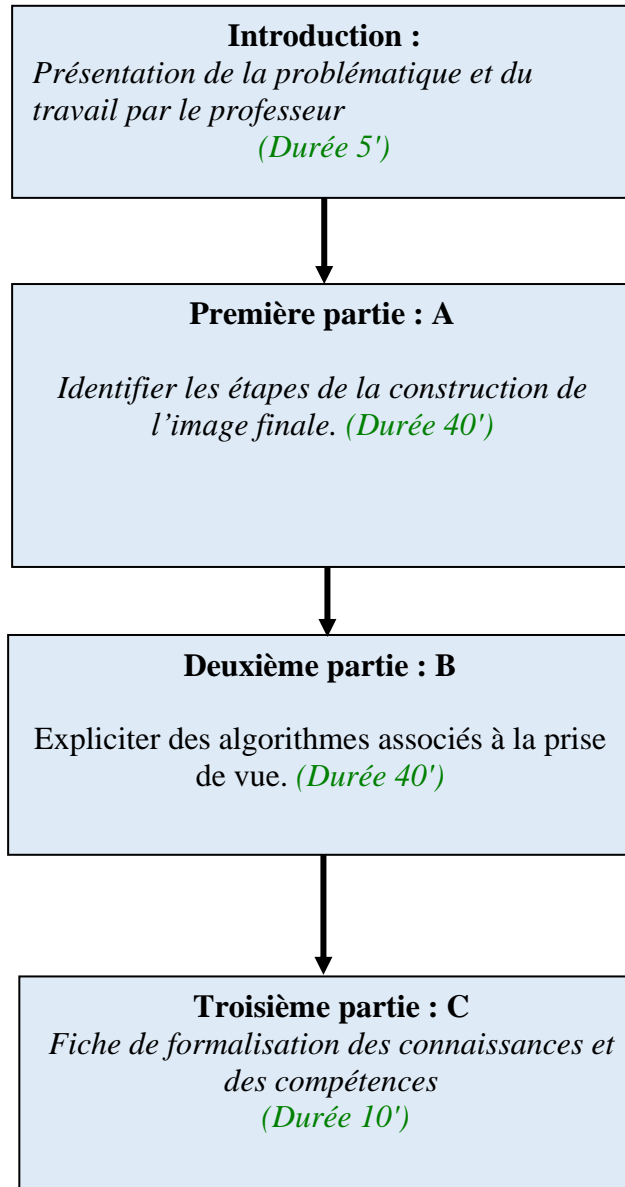
En présence sur l'ordinateur de l'élève, du logiciel de programmation et de simulation Mywiz complété du modèle du robot :

#### Les élèves doivent :

- Identifier les étapes de la construction d'une photo panoramique
- Tester avec le logiciel Myviz en mode simulé, les programmes python de test fournis.
- Expliciter des algorithmes associés à la prise de vue.
- Proposer une modification du programme python.

## Description des activités pendant la séance.

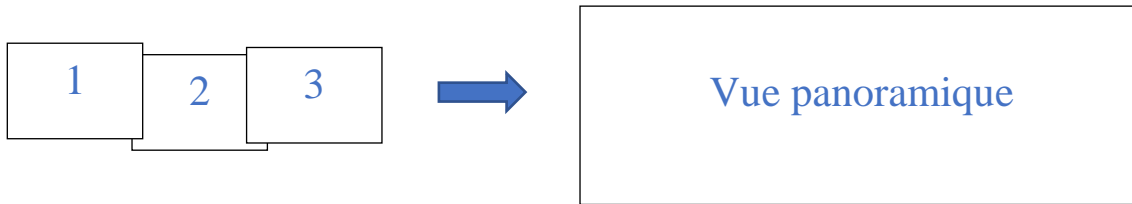
En présence sur l'ordinateur de l'élève, du logiciel de programmation et de simulation Mywiz complété du modèle du robot et des programmes nécessaires, les élèves doivent :



---

**Première partie A - Identifier les étapes de la construction de l'image finale.**

**Principe :**



A partir du document « La photographie panoramique et le robot » :

**QA-1** Enoncer le principe de réalisation d'une vue à 180° par assemblage de photos.

M. Durand souhaite installer un système de surveillance de zone de son entrepôt à l'aide du robot. Le robot envoie une photo régulièrement avec un plan large (prise de vue à 180°) des différentes zones à surveiller.

**QA-2** Quelles sont les contraintes pour réaliser un assemblage de qualité ?

**QA-3** Quelles sont les difficultés que M. Durand risque de rencontrer ?

**QA-4** Quel est l'intérêt du robot pour cette tâche ?

**QA-5** Quel est l'intérêt technique du logiciel hugin ?

## Deuxième partie B - Expliciter des algorithmes associés à la prise de vue.

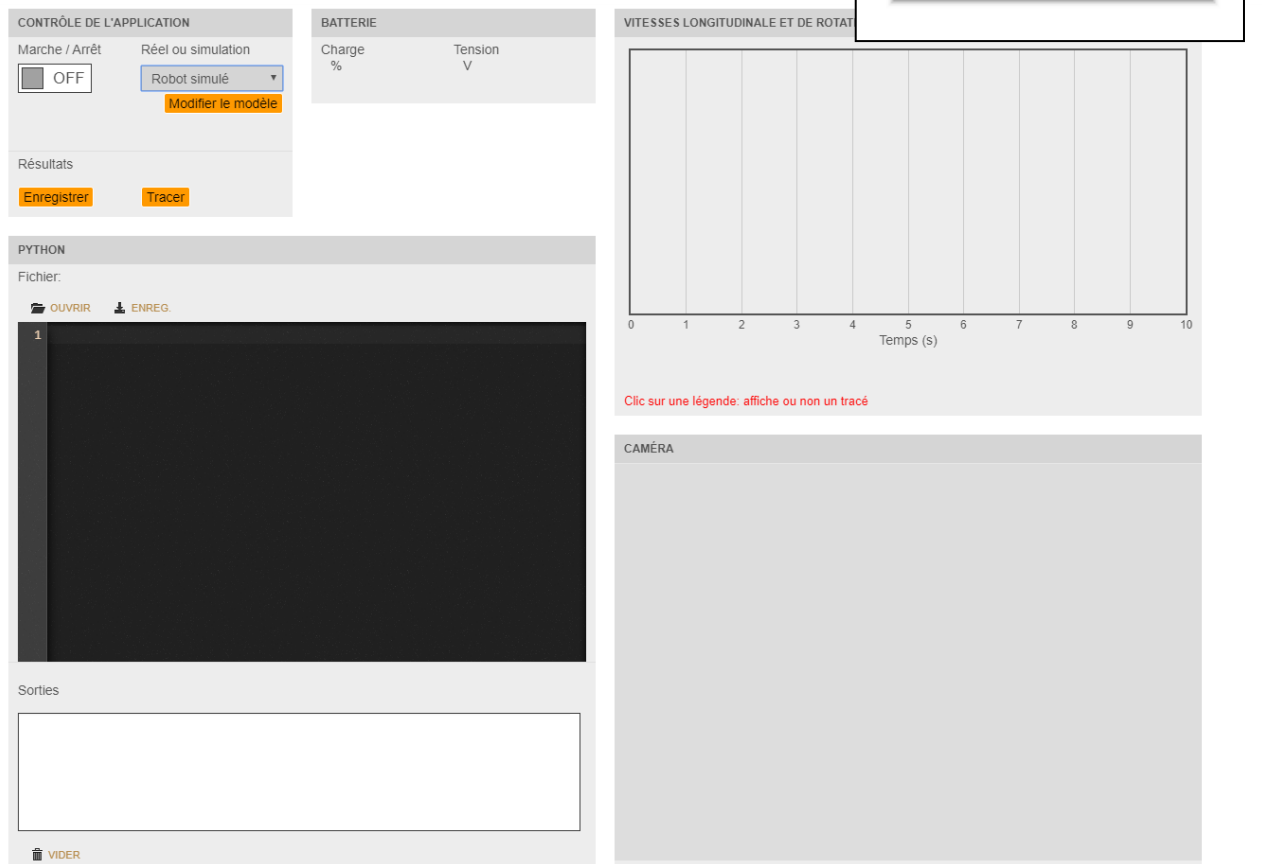
Ressources logicielles :

Environnement = RobotSNT\_API.json

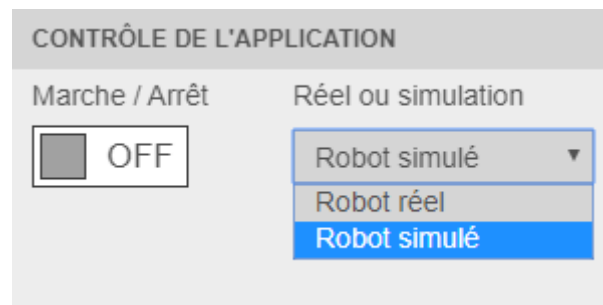
Programme = Robot\_SNT\_1.py

### Découvrir le simulateur Myviz

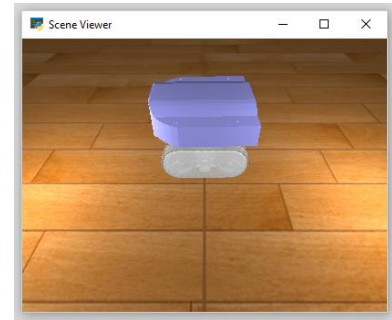
- ✚ Lancer le logiciel Myviz.
- ✚ Ouvrir dans MyViz le fichier " RobotSNT\_API.json ".



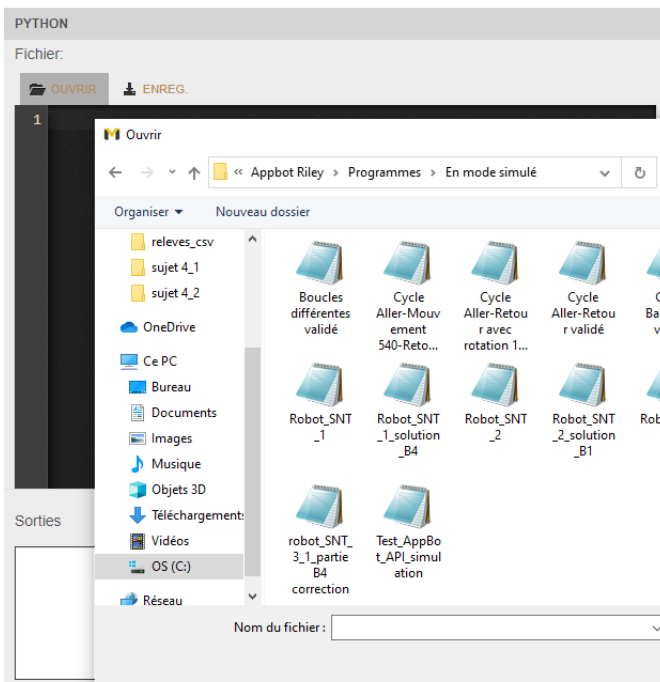
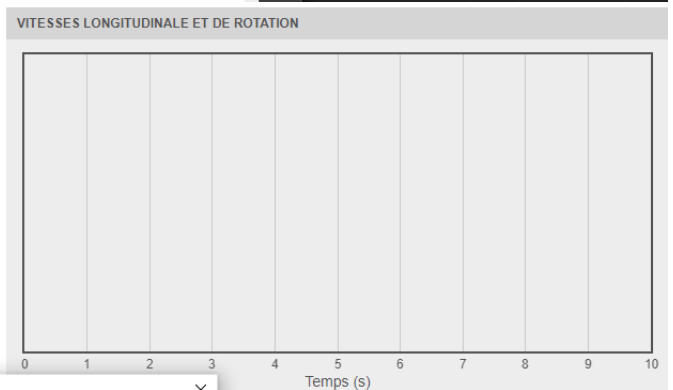
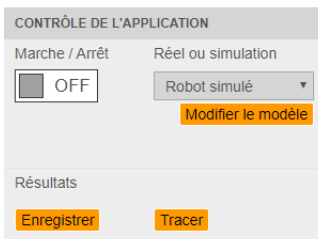
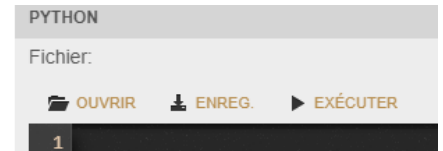
- ✚ Choisir Robot simulé
- ✚ Cliquer sur OFF → ON



Le modèle 3D simulé apparaît dans une fenêtre séparée du logiciel.



 Dans la console Python, cliquer sur ouvrir :



 Choisir le programme Python = Robot\_SNT\_1.py

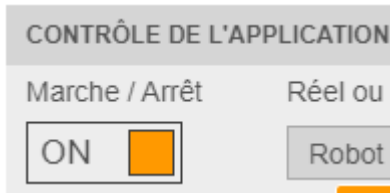
Ce programme est adapté au robot afin que le téléphone fixé prenne une série de trois photos espacées d'un certain angle pour réaliser un plan à 180°. Par programmation on laisse un temps suffisant (attente) afin que l'opérateur puisse prendre une photo ou bien selon le téléphone, que le mode rafale soit paramétré avec ce même temps en chaque prise pour une prise de vues en automatique.

- ✚ Cliquer sur EXECUTER et observer le déroulement dans la fenêtre affichage des courbes et visualiser le modèle3D simulé qui reproduit toutes les commandes fonctions Python incluses dans le programme

**QB-1** Vérifier par simulation le fonctionnement.

**Remarque :**

*Entre chaque essai si l'on veut en faire plusieurs, il sera nécessaire de réinitialiser le robot. Pour ce on mettra le robot sur « off » dans la console et on la remettra sur « on » ensuite.*



**QB-2** En vous aidant de la documentation technique donnée en annexe donner la signification des instructions suivantes.

```
from API import AppBot_API
```

```
AppBot.Tourner(30, 1.5)
```

```
time.sleep(5)
```

**QB-3** En vous aidant des textes ci-dessous, remettre dans l'ordre les étapes de fonctionnement (algorithme) du programme présenté.

Vous remplacerez les ..... par les valeurs numériques du programme.

*Actions (sorties):*

- *Tourner vers la droite à une vitesse de .... pendant .... secondes.*
- *Tourner vers la gauche à une vitesse de .... pendant .... secondes.*

*Commentaires :*

- *Afficher « Tourner vers la ..... à une vitesse de .... pendant .... secondes »*
- *Afficher « prise de la .... photo »*
- *Afficher « Fin de la prise de vue »*

*Controles :*

- *Attendre ... secondes.*
- *Attendre .... secondes le temps que la caméra prenne la ... photo.*

**QB-4** On souhaite maintenant que le robot reprenne sa position initiale en fin de cycle après avoir pris la dernière photo.

Proposer une modification du programme – sauvegarder dans votre dossier personnel– le tester en simulation



**FICHE DE FORMALISATION DES CONNAISSANCES ET DES COMPETENCES****1- CONNAISSANCES ABORDEES DU PROGRAMME**

Connaissances abordées du programme	Savoir appris maîtrisé	Je saurai en parler	Non maîtrisé
Programmer une prise de vue pour obtenir une image stable			
Découvrir la représentation d'une image par un assemblage de photos.			

**2- COMPETENCES ABORDEES DU PROGRAMME**

Compétences abordées du programme	Acquis	Je saurai refaire avec de l'aide	Non acquis
Identifier les étapes de la construction de l'image finale.			
Expliciter des algorithmes associés à la prise de vue.			

## Annexes :

## ANNEXE - Fonctions de l'API

L'API de pilotage contient les fonctions suivantes :

### AppBot\_API()

- Description : fonction d'initialisation à exécuter impérativement au début de chaque programme. Elle lance en particulier en tâche de fond le client Websocket destiné à envoyer les ordres de pilotage au robot.

### Avancer(vitesseLongitudinale, duree=-1)

- Paramètres :
  - vitesseLongitudinale : réel ou chaîne de caractères (expression Python valide du type ' $3 * \text{math.sin}(t)$ ',  $t$  étant reconnu comme le temps courant). Consigne de vitesse longitudinale (m/s), saturée en interne entre  $-0.5$  et  $0.5$  m/s
  - duree : réel (optionnel). Durée de la manœuvre en secondes. Valeur par défaut :  $-1$  (la valeur reste constante jusqu'à ce que la commande soit interrompue par un CTRL-C)
- Description : cette fonction donne une consigne de vitesse longitudinale au robot pendant une certaine durée

### Tourner(vitesseRotation, duree=-1)

- Paramètres :
  - vitesseRotation : réel ou chaîne de caractères (expression Python valide du type ' $3 * \text{math.sin}(t)$ ',  $t$  étant reconnu comme le temps courant). Consigne de vitesse de rotation (deg/s), saturée en interne entre  $-180$  et  $180$  deg/s
  - duree : réel (optionnel). Durée de la manœuvre en secondes. Valeur par défaut :  $-1$  (la valeur reste constante jusqu'à ce que la commande soit interrompue par un CTRL-C)
- Description : cette fonction donne au robot une consigne de vitesse de rotation autour de la verticale pendant une certaine durée

**Mouvement(vitesseLongitudinale, vitesseRotation, duree=-1)**

- Paramètres :
  - vitesseLongitudinale : réel ou chaîne de caractères (expression Python valide du type '3 \* math.sin(t)', t étant reconnu comme le temps courant). Consigne de vitesse longitudinale (m/s), saturée en interne entre  $-0.5$  et  $0.5$  m/s
  - vitesseRotation : réel ou chaîne de caractères (expression Python valide du type '3 \* math.sin(t)', t étant reconnu comme le temps courant). Consigne de vitesse de rotation (deg/s), saturée en interne entre  $-180$  et  $180$  deg/s
  - duree : réel (optionnel). Durée de la manœuvre en secondes. Valeur par défaut : -1 (la valeur reste constante jusqu'à ce que la commande soit interrompue par un CTRL-C)
- Description : cette fonction donne une combinaison de consignes de mouvement au robot pendant une certaine durée

**Terminer()**

- Description : fonction à exécuter impérativement à la fin de chaque programme.

**Programme « robot\_SNT\_1 .py »**

```
import time

from API import AppBot_API
AppBot = AppBot_API('simulation')

print("Rotation a la vitesse de 30 deg/s pendant 1.5 s")
AppBot.Tourner(30, 1.5)
print("prise de la premiere photo")
time.sleep(5)

print("Rotation a la vitesse de 30 deg/s pendant 1.5 s")
AppBot.Tourner(-30, 1.5)
print("prise de la deuxieme photo")
time.sleep(5)

print("Rotation a la vitesse de 30 deg/s pendant 1.5 s")
AppBot.Tourner(-30, 1.5)
print("prise de la troisieme photo")
time.sleep(5)

print("fin de la prise de vue")
AppBot.Terminer()
```