

I. Objectifs

Ce premier TP comporte trois objectifs :

- Prendre en main le simulateur Webots et ses outils de développement
- Découvrir le robot Thymio et ses capteurs embarqués
- Développer un premier algorithme de déplacement du robot.

Nous utiliserons le simulateur Webots pour l'ensemble des Tps de ce module. Il permet de simuler un robot Thymio dans un environnement entièrement contrôlé. Il est possible de piloter le robot tout en récupérant les données des capteurs embarqués. Nous pourrions donc faire réagir le robot à sa perception de l'environnement.

Attention, il est indispensable de conserver tout au long des TPs les programmes réalisés. En effet, vous utiliserez lors des prochains TPs des programmes que nous réaliserons aujourd'hui.

II. Découverte de Webots

a. Configuration du Simulateur

Le simulation Webots est disponible dans le répertoire TP EEA – Webots.

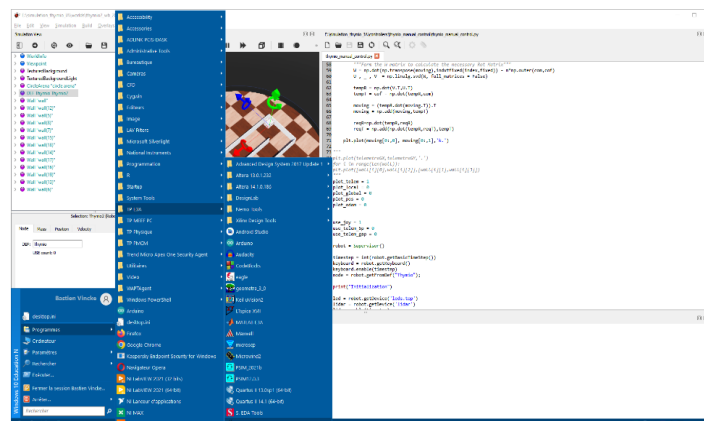


Figure 1 - Raccourci Webots

L'environnement de simulation est disponible sur Ecampus. L'archive fait un peu plus de 200mo car elle contient une version de python compatible avec votre environnement de travail.

Lors de la première utilisation du simulateur, il est nécessaire de configurer la version de Python utilisée dans les préférences du simulateur.

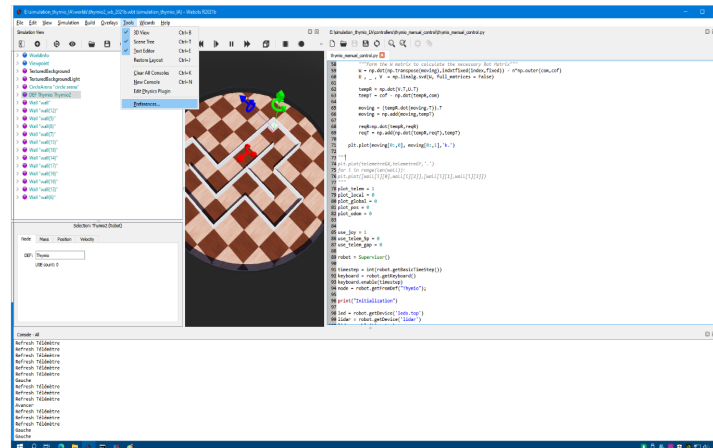


Figure 2 - Configuration de Python sous Webots

Il faut renseigner le bon chemin. La version de python est incluse dans le package que vous venez de télécharger. Vous devez bien renseigner ce chemin.

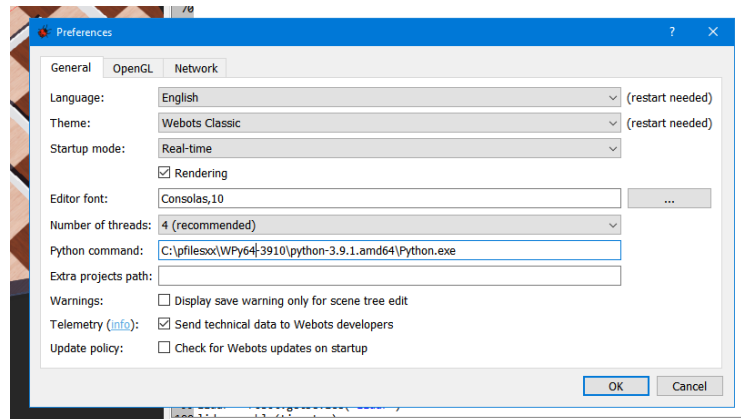


Figure 3 - Chemin vers l'exécutable Python

Ouvrez le monde contenant notre robot en ouvrant le fichier « world/thymio2_wb_2021b.wbt »

Liens utiles (voir indispensable) :

Le guide global : <https://cyberbotics.com/doc/guide>

La prise en main de l'interface utilisateur : <https://cyberbotics.com/doc/guide/the-user-interface>

La doc du robot thymio sous webots : <https://www.cyberbotics.com/doc/guide/thymio2?version=R2021b>

a. Prise en main du robot

A l'aide de la documentation du robot :

1. Donner les principales caractéristiques mécaniques du Thymio
2. Identifier les actionneurs disponibles sur votre robot
3. Identifier les capteurs embarqués

b. Déplacement du point de vue, Déplacement du robot

Le comportement du robot est géré par un « contrôleur ». Ce contrôleur est programmé en langage Python. Pour éditer le contrôleur, réalisez un clic droit sur le robot puis « edit controller ».

Attention, pour prendre en compte la modification du contrôleur et mettre à jour le comportement du robot, il faut enregistrer le fichier puis remettre à zéro la simulation.

Attention, ce fichier contrôleur doit déclencher de manière de manière périodique une étape du simulateur le simulateur webots. Il n'est pas possible de réaliser une boucle infinie dans le contrôleur sans appeler cette étape de simulation. Dans ce cas, le simulateur serait bloqué en attente de récupérer l'exécution.

Déplacement automatique du robot

4. En vous servant de l'exemple fourni, réaliser un programme pilotant les deux moteurs pour permettre au robot de se déplacer.
5. Programmer une rotation à 90 degrés (avancer d'environ un carreau, tourner de 90 degrés, avancer d'un carreau etc ...)
6. Généraliser la rotation un angle de 'x' degrés

Déplacement piloté par l'utilisateur

Webots permet de récupérer les touches du clavier qui ont été appuyées. On souhaite permettre à l'utilisateur de piloter les déplacements du robot à l'aide du clavier.

On utilisera la documentation associée à cette fonction :

<https://cyberbotics.com/doc/reference/keyboard?tab-language=python>

7. Programmer un pilotage par le clavier du robot (avancer, reculer, tourner à droite, tourner à gauche, s'arrêter).
Il est indispensable de pouvoir piloter son robot à l'aide du clavier pour pouvoir lui faire réaliser des trajectoires adaptées à nos traitements futurs.

III. Première mesures capteurs

Le Thymio dispose de 7 capteurs de distance disposés tout autour de lui.

Utilisez la fonction « `robot.getDevice('prox.horizontal.*')` » puis prendre l'accès à un capteur puis la fonction « `.enable(timestep)` » pour régler la fréquence d'échantillonnage du capteur. Il suffira d'utiliser la fonction « `.getValue()` » pour récupérer la distance.

8. Programmer la récupération des valeurs des 7 capteurs de distance
9. Tester le bon fonctionnement des capteurs, leur portée etc ...
10. Réaliser un algorithme simple permettant au robot de se déplacer dans le labyrinthe sans toucher de murs.

IV. Prise approfondie en main du Thymio

On souhaite maintenant créer un nouveau monde qui permettra de tester les capteurs sol du Thymio.

11. Créer un « wizard – new project directory ». Renseigner sur les informations nécessaires et sélectionner l'option « Add a rectangle arena »
12. Ajouter le robot Thymio dans l'environnement :

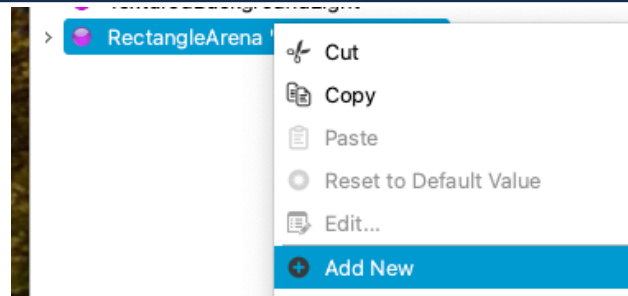


Figure 4 - Ajout du robot Thymio

Sélectionner le Thymio dans PROTO nodes – robots – mobysa.

13. Pour ajouter la ligne noire, nous allons utiliser une ligne toute faite. Télécharger le fichier TrackOne.proto dans le répertoire proto de votre simulateur.
14. Ajouter un node avec le « + » en haut à gauche et sélectionner « Trackone »
15. Modifier la taille de votre environnement pour faire rentrer la piste dedans.
16. Modifier la propriété « Floorappearance » en « base node »

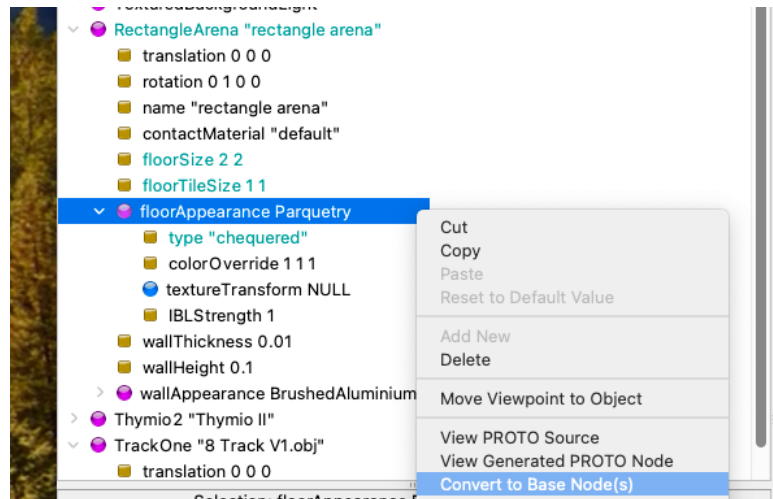


Figure 5 – Conversion en Base node

17. Réinitialiser la couleur du sol



Figure 6 - Réinitilisation de la couleur du sol

18. Créer un nouveau controleur : Wizard – New robot controller, choisir le langage Python
Ne pas oublier de changer le controleur dans les propriétés du robot Thymio
19. Récupérer les valeurs des capteurs de ligne
20. Proposer et coder un algorithme simple de suivi de ligne