J.-Y. Tigli
Polytech Nice Sophia
SI5
2023

Travail à rendre sur les architectures comportementales des Systèmes Intelligents Autonomes

1 Partie 1 du rendu

Lors du dernier TD vous avez a minima implémenté un algorithme d'évitement d'obstacle pour le Khepera sur le simulateur Webots.
Dans ce travail à rendre, vous allez rajouter un comportement de recherche de source lumineuse. Le robot se dirige alors vers cette source.

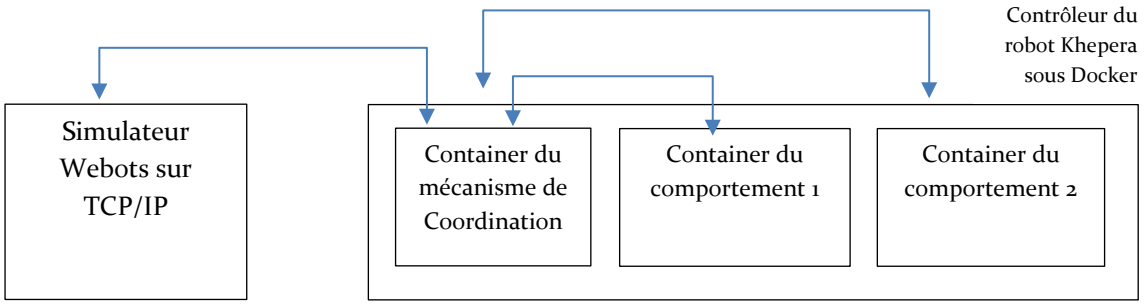
Question : Après avoir implémenté et testé ces deux comportements sur le simulateur Webots choisissez le mécanisme de coordination le plus adapté pour permettre au robot d'exhiber ces deux comportements simultanément et pertinemment. Il s'agit alors pour le robot de chercher une source lumineuse toute en évitant les obstacles.

- Vous produirez alors une V1 de votre solution.
- Dans un fichier README vous justifierai votre choix et la procédure d'installation de votre démo.

2 Partie 2 du rendu

Question : La seconde partie de ce travail consiste à mettre en œuvre une architecture logicielle à base de micro-services sous docker. Il s'agira alors d'utiliser un serveur Webots et de containeriser chacun des comportements ainsi que le mécanisme de coordination.

L'architecture sera alors la suivante :



Les protocoles de communication utilisés entre Simulateur – Containers et entre Containers sont libres et de votre choix.

- Vous produirez alors une V2 de votre solution.
- Dans le fichier README vous détaillerez la procédure d'installation de votre démo.

3 Modalité de rendu et deadline

*Pour votre rendu, merci de **créer un canal SIA-GR-X où X** est le numéro de votre groupe et de m'y inviter. Vous déposerez alors les solutions **v1 et v2** de votre travail sous forme de packages à installer ou un lien pour les trouver. Un **README** avec les noms des membres de votre groupe, les réponses aux questions ci-dessus, la procédure d'installation et de test sera fourni avec les solutions.*

***La date limite de rendu est fixée au 7 février minuit,** dépôt sur votre canal SIA-GR-X faisant fois.*

Travail à rendre sur les architectures comportementales des Systèmes Intelligents Autonomes

4 Quelques pointeurs utiles

Comment utiliser les lightsensors du Khepera ?

Seuls les khepera 1 et 2 sont dotés de capteurs de proximité Infrarouge. Les khepera 3 et 4 utilisent des capteurs Ultrasons. Vous utiliserez donc le khepera 2 pour ses travaux.

Plus d'indications sont fournies ici : https://cyberbotics.com/doc/reference/lightsensor

Comment rajouter une light dans l'environnement (world) de mon robot ?

https://cyberbotics.com/doc/guide/tutorial-3-appearance

Le but de ce tutoriel est de vous familiariser avec certains nœuds liés au rendu graphique. Des simulations de bonne qualité peuvent être créées très rapidement lorsque ces nœuds sont utilisés de manière adéquate. Une bonne qualité graphique n'améliore pas seulement l'expérience de l'utilisateur, elle est également essentielle pour les simulations où les robots perçoivent leur environnement (traitement d'image de la caméra, suivi de ligne, etc.).

Comment dialoguer avec Webots sur le réseau TCP/IP ?

Webots documentation: Interfacing Webots to Third Party Software with TCP/IP (cyberbotics.com)

Webots offre des API de programmation pour les langages suivants : C/C++, Java, Python et MATLABTM. Une telle interface peut être aussi mise en œuvre par le biais d'un protocole TCP/IP que vous pouvez définir vous-même.
Webots est livré avec un exemple permettant d'interfacer un robot Khepera simulé via TCP/IP avec n'importe quel programme tiers capable de lire et d'écrire sur une connexion TCP/IP.
Cet exemple s'appelle "khepera_tcpip.wbt", et se trouve dans le répertoire "WEBOTS_HOME/projects/robots/k-team/khepera1/worlds" de Webots. Le robot Khepera simulé est alors contrôlé par le contrôleur "tcpip" qui se trouve dans le répertoire "controllers" du même projet. Ce petit contrôleur en C est fourni avec le code source complet dans "tcpip.c", de sorte que vous pouvez le modifier pour répondre à vos besoins.
Un exemple de client est fourni dans "client.c". Ce client peut être utilisé comme modèle pour écrire un client similaire en utilisant le langage de programmation de votre logiciel tiers.