

Pour mettre en place l'environnement logiciel de travail utilisé dans ce TME, suivez la procédure décrite ci-après:

1. télécharger le simulateur (roborobo3_jar_2018.tgz)
`http://pages.isir.upmc.fr/~bredeche/Teaching/IA/Ressources`
2. dans un répertoire où vous aurez placé l'archive du simulateur, désarchivez:
`tar xvfz roborobo3_jar_2018.tgz`
3. dans le répertoire du simulateur, installez et compilez le simulateur:
`make -j8`
4. pour tester:
`./roborobo -l config/tutorial.properties`
Remarques: tapez « h » dans la fenêtre de simulation pour avoir de l'aide. Tapez « d » pour accélérer la simulation (3 modes). Tapez "f" pour activer le suivi d'un agent en particulier. Tapez "<tab>" pour passer d'un agent à l'autre.

Au cours de ce TME, vous allez utiliser plusieurs projets. Chaque projet correspond à un point de départ pour un exercice. Les projets se trouvent dans le répertoire **prj/** et sont nommés *TME_...*.

Vous ne modifierez que les classes (...)Controller(.h/.cpp) et (...)WorldObserver(.h/.cpp). **La classe (...)Controller** permet de programmer le comportement d'un robot, **la classe (...)WorldObserver** permet de manipuler l'environnement. Il est possible de faire tous les sujets en modifiant uniquement les méthodes **step()** de ces deux classes, et (éventuellement) le constructeur de la classe (...)Controller.

Les robots dont vous disposez ont 12 senseurs, positionnés comme indiqué sur l'image ci-dessous. L'avant du robot dispose de plus de senseurs.



Il s'agit d'un programme en C++, cependant il n'est pas nécessaire de maîtriser le C++ pour faire le TME. **Tous les éléments nécessaires** pour vous aider à répondre aux questions se trouvent dans l'exemple fourni avec le projet qui se trouve dans le répertoire **/prj/Tutorial**. Pour commencer, étudiez le code de la fonction **step()** de la classe *TutorialController*. Cette méthode *step()* vous montre comment écrire un comportement dépendant des entrées sensorielles (les valeurs de effecteurs dépendent d'une combinaison linéaire des entrées), et illustre comment accéder aux données capteurs qui vous seront utiles au cours du TME.

En plus des classes mentionnées, vous pouvez modifier le fichier de paramètres **TME(...).properties** du répertoire **/config/** — ce fichier vous permet de changer facilement les paramètres de votre simulation, par exemple le nombre de robots (par défaut: *gInitialNumberOfRobots* vaut 200), voire les obstacles présents ou la carte de l'environnement. Il n'est pas nécessaire de modifier les fichiers *properties* pour faire le TME.

EXERCICE 1: Braitenberg (prise en main et rappel)

Projet: <roborobo3>/prj/TMEbraitenberg
Commande: ./roborobo -l config/TMEbraitenberg.properties

Ecrivez un comportement d'évitement d'obstacles. Pour mémoire, les connexions dans un robot de Braitenberg sont excitatrices ou inhibitrices (ie. pas de seuils).

Evaluation: implémentation, identification de cas difficiles.

EXERCICE 2 : Agrégation

Projet: <roborobo3>/prj/TMEaggregation
Commande: ./roborobo -l config/TMEaggregation.properties

Ecrivez un comportement d'agrégation. Chaque robot doit se diriger vers le robot le plus proche. Vous pouvez augmenter le nombre de robots (*gInitialNumberOfRobots*) en éditant votre fichier *properties* dans le répertoire *config/*.

Evaluation: avec la touche « entrée » vous pouvez prendre le contrôle d'un robot. Déplacer vous afin de perturber le comportement d'agrégation. Si votre robot est bloqué, vous pouvez changer de robot avec la touche tabulation (shift+tab pour revenir au précédent).

EXERCICE 3 : Dispersion

Projet: <roborobo3>/prj/TMEdispersion
Commande: ./roborobo -l config/TMEdispersion.properties

Ecrivez un comportement de dispersion. Chaque robot doit s'éloigner du robot le plus proche. Lorsqu'il ne voit rien, il tourne sur lui même.

Evaluation: avec la touche « entrée » vous pouvez prendre le contrôle d'un robot. Déplacer vous afin de perturber le comportement de dispersion.