

ÉCOLE POLYTECH'NICE SOPHIA
SI4
2018-2019

Julien Deantoni

Finite State Machine: TD numéro 6, 7, 8, ?

Contrôleur de robot artificier

Objectif

L'objectif de ce TD est de vous laisser structurer, déstructurer et restructurer un statechart décrivant le code de contrôle d'une application de robotique. Plus précisément le but est de contrôler un petit robot explorateur dans un environnement virtuel. Dans cet environnement se trouve des bombes de peintures qui explosent après un temps donné. Le robot doit aller chercher les bombes et les mettre dans un endroit sûr avant qu'elles n'explosent.

L'environnement de simulation à utiliser est V-REP: <http://www.coppeliarobotics.com/>. Bien que V-REP puisse utiliser de nombreux robots et environnements différents, vous vous limitez à l'environnement et au robot fournis dans la scène se trouvant ici: <https://github.com/jdeantoni/PolyBot/tree/master/scenes>

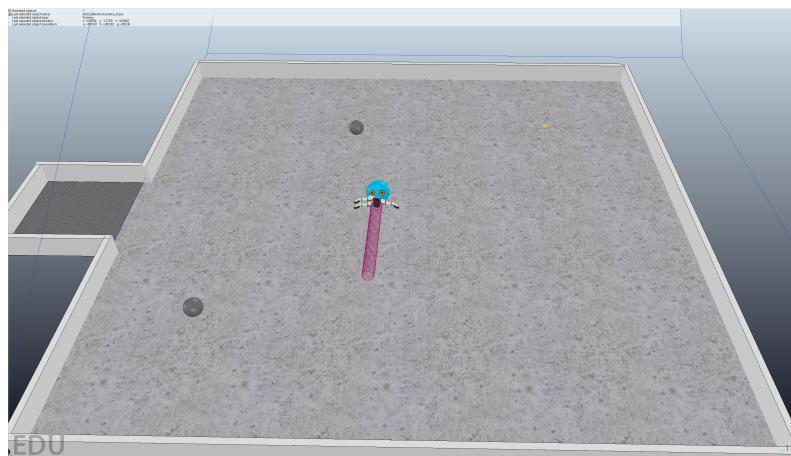


Figure 1:
fig:screenshot

L'exercice se fera en utilisant l'environnement Yakindu et sera en Java.

MVP

Soit la scène référencée ci dessus. Pour la version minimal de votre programme, le robot devra aller mettre chacune des bombes de peintures explosives dans la zone dont le sol est plus foncé (coordonnées -25; 500). Le robot évolue dans un repère dont le point (0 ; 0) est en bas à gauche de la vue initiale de la scène. La séparation entre le code applicatif en Java et le statechart écrit avec Yakindu sera à justifier.

Si un robot avec un comportement smooth et précis est toujours agréable, ce n'est clairement pas le but de ce TD. Le but est de manipuler des statecharts et de bien comprendre ce qui fait parti du code métier et du code de contrôle.

un projet (pur) Java avec une interface de communication avec le simulateur (`fr.unice.polytech.deantonio.vrep.polybot.PolyRob`) ainsi que deux algorithmes de tests (vraiment non structurés) sont fournis ici: <https://github.com/jdeantonio/PolyBot>.

Vous pourrez modifier l'emplacement des bombes de peintures et du robot mais pas le reste de la scène. La scène fournie est simple mais des obstacles (immobiles dans un premier temps et pourquoi pas mobiles) pourraient y être placés.

Vous devez faire autant d'extensions que possible :) Vous pouvez également proposer vos propres extensions (n'hésitez pas à m'en parler d'abord).

Détection et gestion des obstacles immobiles

Cette extension permet à votre robot de naviguer dans la scène en présence d'obstacles immobiles, en les évitant.

Détection et gestion des obstacles mobiles

Cette extension permet à votre robot de naviguer dans la scène en présence d'obstacles mobiles, en les évitant.

Gestion de l'ordre de gestion des bombes de peinture

Il peut être intéressant de gérer un ordre particulier pour amener les bombes de peinture dans un lieu sûr. Cette extension permet de le faire.

À suivre ?

D'autres propositions d'extensions arriveront peut-être au fil de l'eau.