

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

Contrôleur de robot ménager

1 Objectif

L'objectif de ce TD est de vous laisser structurer, déstructurer et restructurer un statechart décrivant le code de contrôle d'un robot ménager permettant d'aspirer le sol mais aussi de faire d'autres choses comme attraper des objets ou dessiner sur le sol. Plus précisément le but est de contrôler les différents actionneurs du robots simulé dans un environnement virtuel en fonction des informations données par ses capteurs. Le robot à contrôler est basé sur l'environnement cyberbotique webots (<https://cyberbotics.com/>). Cet environnement autorise plusieurs robots et scènes mais le statechart attendu sera restreint au contrôle du robot PolyCreate défini dans le dépôt suivant:<https://github.com/jdeantoni/polytechWebotsController>. Dans ce dépôt vous trouverez en plus de l'environnement PolyCreate une couche d'abstraction Java au-dessus de l'API distante cyberbotique. bien qu'elle ne soit pas suffisante pour tous les objectifs du projet, elle peut être utilisée comme point de départ.

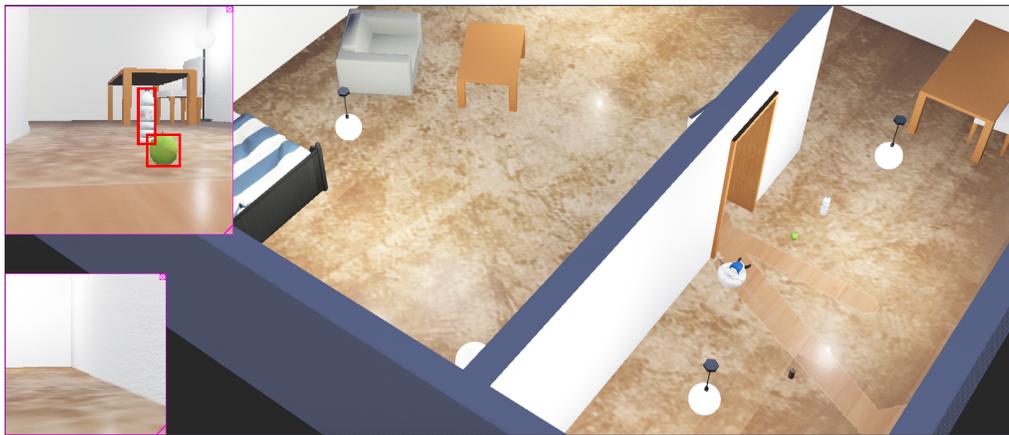


Figure 1: A view of the proposed Hover simulator

L'exercice se fera en utilisant l'environnement Yakindu et le code généré, à intégrer à votre contrôleur de robot sera en Java. De plus vous travaillerez en binôme¹.

2 MVP

Pour la version minimale de votre projet, le robot doit se déplacer dans l'environnement en évitant un maximum les obstacles et sans rester bloquer, sans tomber dans les escaliers ou les trous (s'il y en a) et en évitant de taper dans les objets. Également, il ne devra pas franchir les murs virtuels qui délimitent des espaces où le robot ne doit pas aller. Enfin il devra couvrir un maximum la pièce afin qu'elle soit propre.

¹ Attention, En fonction du contrôle continu, il est fortement possible que vous n'ayez pas les mêmes notes

En tant que MVP, la réalisation parfaite des contraintes ci dessus avec des choix de conception éclairés permet d'avoir la moyenne (10/20). La réalisation du comportement du MVP sans une architecture correcte et réfléchi entre le contrôle et le code métier ne vaut rien. Bien sûr vous ferez valider régulièrement les comportements par le client.

3 Extensions

Vous devez faire autant d'extensions que possible. Vous pouvez également proposer vos propres extensions (n'hésitez pas à m'en parler d'abord). Bien sûr vos extensions seront documentées et leur conception justifiée dans le rapport. Vous n'êtes pas obligé de les prendre dans l'ordre.

Contournement propre des objets

Cette extension contrôle le robot afin qu'il contourne proprement les pieds de table ou autre éléments présent dans la pièce afin de nettoyer au mieux ce parties.

Rangement des objets présents dans la scène

Cette extension contrôle le robot afin qu'il déplace les objets déplaçables de la scène et les amène dans un endroit prédéfini (de votre choix). Pour cela il utilisera la pince qui se trouve sur l'arrière. Tout cela devra se faire de manière opportunistes lors du nettoyage. Cette action pourra être activée/désactivée par une entrée utilisateur.

Gestion du crayon

Cette extension permet de définir des mots qui seront écrits par le robot pour permettre... de rien du tout... juste d'écrire des mots sur le sol :). À noter que cela nécessite d'interagir avec l'utilisateur.

À suivre ?

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

4 Rendu du projet

Pour le rendu de ce projet, vous m'invitez sur le système de suivi de version de votre choix (git de préférence). En plus du code de votre projet, vous ajouterez un dossier *report* contenant un pdf qui met en avant vos choix. Le rapport commencera succinctement par dire ce que vous avez implémenté (e.g., MVP + option X et Y). Ensuite, ce rapport ne devra pas expliquer les banalités directement lisible sur le state chart mais au contraire mettre en exergue pourquoi vous avez choisi une structuration particulière de votre state machine plutôt qu'une autre. Dis autrement, nous vivons dans un monde de compromis et il n'existe que rarement une solution miracle avec uniquement des avantages. Dans ce cadre, votre solution n'est la bonne que si vous êtes conscient de ses avantages et inconvénients par rapport aux autres choix envisageables et que vous avez fait vos choix en connaissance de cause. Ces différents éléments seront donc présentés dans le rapport sous forme d'un ensemble de sous sections.

Ensuite, vous présenterez l'activité de V&V que vous avez réalisé dans l'outil que nous prendrons en main dans un prochain TP), à partir de votre modélisation du contrôle du robot (le code associé se trouvera dans un sous dossier *VnV* sur le git). Quelles sont les questions que vous avez posé au système, à quelles catégories correspondent elles (safety/liveness, verification/validation) ? Pourquoi est-ce que cela a été difficile ou impossible à mettre en place ?

Enfin une dernière section présentera une petite prise de recul sur votre travail. Qu'auriez vous fait différemment ? pourquoi ? etc.

Le deadline pour la finalisation du projet et du rapport sur le git est le 14 Décembre 2021 à 23:59.