

Praktikum Anwendungen wissensbasierter Systeme - Intelligente Systeme

Tischdecken im Robotersimulator

Abgabetermin: Mittwoch, 13.07.2005

Die finale Aufgabe dieses Praktikums wird sein, einen Tisch mit 3 Tellern und 3 Tassen zu decken.

Aufgabe 1:

Die beiden Teller (IDs 3 und 4) und die Tasse (ID 5) auf der Küchenzeile, sowie eine der beiden Tassen (IDs 9 und 10) im rechten Schrank sollen auf den Tisch gestellt werden. Dazu muss natürlich die Schranktür geöffnet und gegebenenfalls der Stuhl (ID 16) verschoben werden.

Natürlich sollten dann die Teller und Tassen an einer vernünftig benutzbaren Position stehen.

Dazu stehen Methodenrumpfe in der Datei `entities.gr.lisp` bereit, die noch ergänzt werden müssen.

Hinweise:

- Auf der Praktikumsseite stehen wieder verschiedene Dateien zur Verfügung (gepackt in `agent.tar.gz` und `worlds.tar.gz`): 2 Weltbeschreibungen, eine mit Stuhl (`with-chair.world`) für die finale Vorführung und eine ohne Stuhl (`kitchen-and-livingroom.world`) zum Testen, außerdem `entities.g.lisp` mit den Definitionen der Ziele zu den in `entities.gr.lisp` enthaltenen zu implementierenden Zielroutinen.
 - Die Ziele sind im Einzelnen:
 - b21-set-vel** setzt die Geschwindigkeit des Roboters,
 - b21-at-angle** dreht,
 - b21-at-point** fährt den Roboter und
 - b21-at-pose** fährt den Roboter an eine Position mit bestimmter Orientierung.
 - b21-right-at-pose** bewegt den rechten Arm,
 - b21-right-gripper-at-pos** verändert die Öffnung des rechten Greifers,
 - b21-right-gripper-set-vel** setzt die Geschwindigkeit des rechten Greifers und
 - b21-right-gripper-closed** schließt den rechten Greifer.
 - pick-up-entity** hebt eine Entität (Tasse oder Teller) auf,
 - put-down-entity** stellt eine Entität ab,
 - cupboard-door-opened** öffnet die Schranktür,
 - cupboard-door-closed** schließt die Schranktür und
 - chair-at-pose** verschiebt den Stuhl an eine Position.
- Die ersten acht Zielroutinen sind bereits definiert. Ebenso gibt es eine einfache Implementierung zum Aufheben einer Tasse, die als Referenz für die eigene Implementierung verwendet werden kann.
- Für die seitenabhängigen Ziele gibt es natürlich auch das Pendant für die linke Seite, z.B. `b21-left-at-pose`.
- Die Zustandsvariablen sind:
 - b21-pose:** Die Position und Rotation des Roboters.
 - b21-point:** Die Position des Roboters.
 - b21-angle:** Die Orientierung des Roboters.

b21-right-arm-pose: Position des rechten Arms.

b21-right-gripper-pos: Position des rechten Greifers (linker und rechter Finger).

b21-right-gripper-vel: Geschwindigkeit des rechten Greifers.

Hier gibt es auch wieder die entsprechend seitenverkehrten Variablen.

- Die Koordinatenursprünge der einzelnen Gegenstände können via Taste `a` angezeigt werden. Um die genauen Positionen der Einzelteile bestimmen zu können hier noch ein paar Daten:

Für den Stuhl gilt:

Beinlänge: 0.45m

Sitz: 0.35×0.35×0.02m

Rückenlehnenhöhe: 0.45m

Höhe des oberen Querbalkens: 0.08m

Für den Schrankgriff im Koordinatensystem der Schranktür (gegeben durch Öffnungswinkel der Schranktür) gilt:

x: 0.1m

y: -0.015

z: 1.05

- Um einzelne Teile eurer Steuerung ausprobieren zu können ohne den Agenten starten zu müssen soll folgendes Code-Snippet dienen:

```
(in-package "LISPCL")
(pc-start)

;Linker Arm
(set-pose-rel-euler (ias-pp-get 21)
 :x 0.420 :y 0.225 :z 0.820 :ax 90deg)
(set-pose-rel (ias-pp-get 22) :x 0.05 :y 0.03)

;Rechter Arm
(set-pose-rel-euler (ias-pp-get 11)
 :x 0.460 :y 0.065 :z 0.820 :ax -90deg)
(set-pose-rel (ias-pp-get 12) :x 0.05 :y 0.03)

(pc-stop)
```