ROBOTPACKING

DOCUMENTAZIONE PDDL APP

* Migliorati Giorgio matricola 742281
* Pagarini Mattia matricola 741964

Indice

1. Obiettivo del progetto pag. 2
2. Dominio pag. 4
   1. Tipi
   2. Predicati
   3. Funzioni
   4. Azioni
3. Problema pag. 7
   1. Inizializzazione
   2. Oggetti
   3. Obiettivi
4. Planner pag. 9

1. OBIETTIVO DEL PROGETTO

L’applicazione PDDL “RPS” ha come obiettivo quello di formulare un piano di azioni per un robot collaborativo all’interno di un supermercato. Il suo compito principale è quello di imbustare gli articoli della spesa: terminati gli acquisti del cliente e scansionati dal cassiere, il braccio robotico si reca in tutte le posizioni necessarie per ottenere gli articoli da imbustare e consegna i sacchetti pieni in una posizione nota. Tiene inoltre conto della capacità dei sacchetti e dispone di una propria batteria, da ricaricare necessariamente dopo una serie di operazioni, secondo parametri impostati.

Dall’apposita applicazione, è possibile scegliere la tipologia dei sacchetti, controllare lo stato di avanzamento del braccio meccanico, pagare la spesa e accedere ad una propria area riservata che permette di ottenere vantaggi e coupon.

Gli incarichi verranno caricati nel problema PDDL, che attraverso un planner compatibile pianificherà l’ordine delle azioni, così da risolvere il problema.

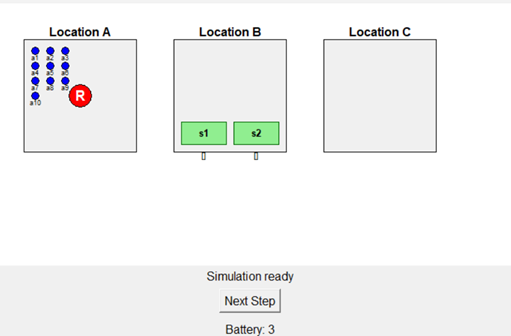
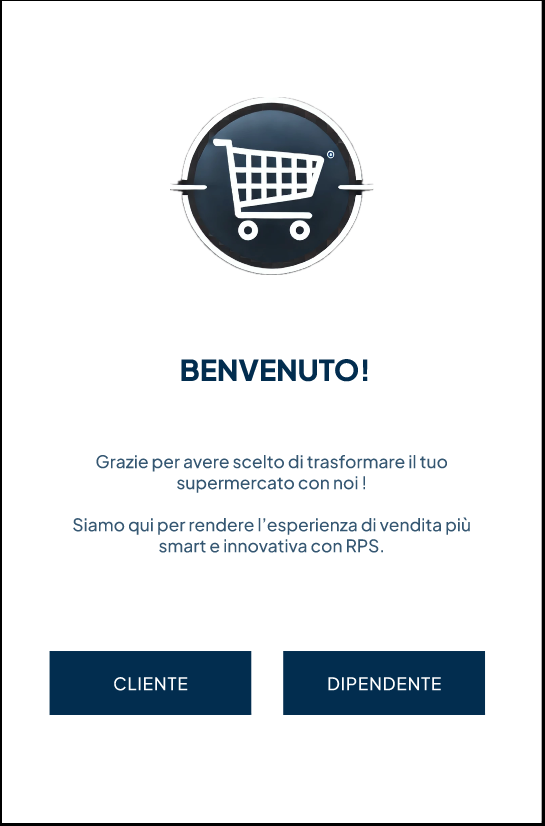
Il robot, essendo collaborativo, andrà ad affiancare il lavoro dell’operatore in cassa e fornirà al cliente i sacchetti con i suoi articoli acquistati. Il punto di partenza in cui il robot afferra gli articoli, il punto in cui li imbusta e la posizione di arrivo in cui deposita i sacchetti sono fissi e si ipotizza, data la presenza del barman, la continua disponibilità di sacchetti e la possibilità di essere sempre ricaricabile.

Tutti i sacchetti sono caratterizzati da una capacità di 5 articoli, implicazione che porta dover cambiare di volta in volta i sacchetti e a rendere quindi fondamentale la presenza di un operatore per fare in modo di intervenire al momento opportuno.



Bozza grafica dell’implementazione del robot collaborativo

Alcune schermate dell’applicazione (mock-up) e dell’interfaccia grafica realizzata in python.



Applicazione completa:



2. DOMINIO

L’applicazione necessita di un planner numerico, che supporti i fluents.

## 2.1 tipi

Il programma definisce una serie di tipi, o oggetti:

* *Location*à Rappresenta un tipo generico di luogo;
* *Locatable*à Rappresenta un tipo di oggetto che può essere localizzato in uno spazio;
* *Bot, articolo e sacchetto* (extends Locatable)à Rappresentano rispettivamente il robot che si muove nello spazio, e gli oggetti con cui lui interagisce;
* *Robot* (extends Bot) à Rappresenta il robot;
* *Battery-level* à Rappresenta il livello di carica della batteria;
* *Capacità-level*à Rappresenta il livello di capacità di un sacchetto, cioè la sua capacità di contenere articoli.

## 2.2 predicati



Indica che un oggetto, che è di tipo locatable, si trova in una specifica posizione, che è di tipo location.





Indica che un braccio sta tenendo un articolo (della spesa) o un sacchetto.



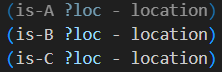
Indica che il braccio del bot è vuoto, ovvero non tiene alcun oggetto.



Indica che un articolo si trova su un sacchetto.



Indica che esiste un percorso tra due location (?from e ?to), cioè è possibile muoversi da una location all'altra.



Definisce 3 location: location A, location B e location C





Indica che la batteria e il sacchetto hanno un determinato valore e questi valori hanno delle transazioni da uno stato iniziale a uno stato successivo.



Indica che il sacchetto è pieno.

## 2.3 funzioni

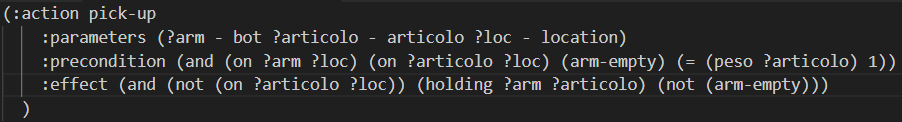


Restituisce il peso dell'articolo, che è di tipo articolo. Il valore che questa funzione restituirà dipende dall'articolo specifico.



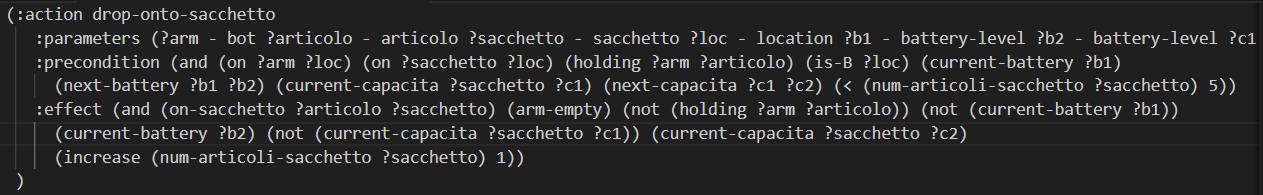
Restituisce il numero di articoli contenuti in un sacchetto, che è di tipo sacchetto.

## 2.4 azioni



Questa azione rappresenta il momento in cui un bot raccoglie un articolo da una location. Precondizioni: Il bot è nella posizione ?loc e l'articolo si trova anch'esso in quella posizione; il braccio del bot è vuoto (non sta già tenendo nulla); l'articolo ha un peso di 1 unità.

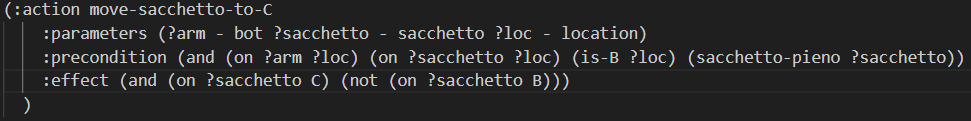
Effetti: L'articolo non è più sulla location (?loc), ma è ora tenuto dal bot e il braccio del bot non è più vuoto (sta ora tenendo un articolo).



Il bot lascia un articolo nel sacchetto. Questo accade solo se il bot si trova nella posizione giusta e il sacchetto ha abbastanza capacità per contenere l'articolo.

Precondizioni: Il bot è nella posizione giusta e sta tenendo un articolo; il sacchetto si trova anche nella stessa posizione, la posizione del bot è B, la batteria del bot e la capacità del sacchetto cambiano da uno stato all'altro e il sacchetto non è ancora pieno (contiene meno di 5 articoli).

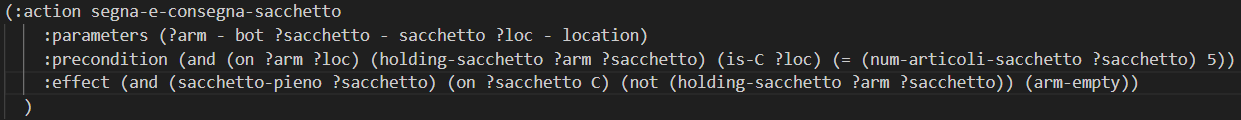
Effetto: L'articolo è ora nel sacchetto, e il bot non sta più tenendo l'articolo, la batteria del bot e la capacità del sacchetto vengono aggiornate e il numero di articoli nel sacchetto aumenta di 1.



Il bot sposta un sacchetto dalla posizione B alla posizione C.

Precondizioni: Il bot è nella posizione giusta e sta tenendo il sacchetto; il sacchetto è pieno e il sacchetto si trova nella posizione B.

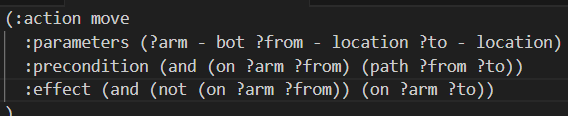
Effetti: Il sacchetto si sposta dalla posizione B alla posizione C.



In bot consegna un sacchetto che è pieno (contenente 5 articoli) nella posizione C.

Precondizioni: Il bot sta tenendo un sacchetto e il sacchetto è pieno e si trova nella posizione C.

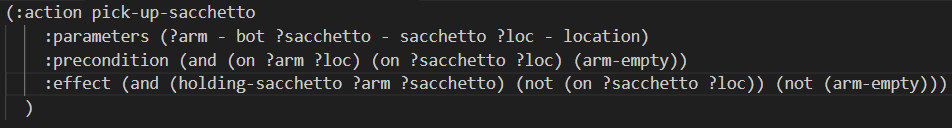
Effetti: Il sacchetto è ora nella posizione C; il bot non sta più tenendo il sacchetto e il braccio è ora vuoto (non sta più tenendo nulla).



Il bot si sposta da una posizione (?from) a un'altra (?to), a condizione che esista un percorso tra queste due posizioni.

Precondizioni: Il bot si trova nella posizione ?from ed esiste un percorso tra ?from e ?to.

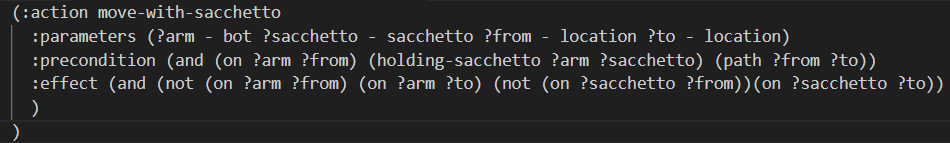
Effetti: Il bot si sposta dalla posizione ?from alla posizione ?to.



Il bot raccoglie un sacchetto dalla location in cui si trova.

Precondizioni: Il bot è nella posizione giusta, il sacchetto è in quella posizione e non sta già tenendo nulla.

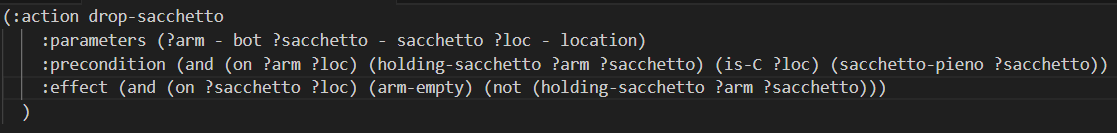
Effetti: Il bot ora tiene il sacchetto e il sacchetto non è più nella posizione in cui si trovava.



Il bot si sposta da una posizione (?from) a un'altra (?to) mentre tiene un sacchetto.

Precondizioni Il bot è nella posizione ?from e tiene un sacchetto ed esiste un percorso tra la posizione ?from e la posizione ?to.

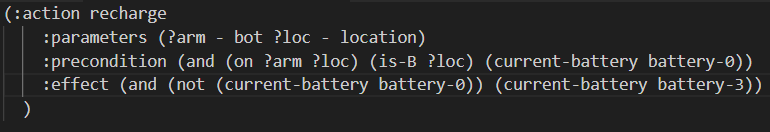
Effetti: Il bot si sposta dalla posizione ?from alla posizione ?to e il sacchetto si sposta anch'esso dalla posizione ?from alla posizione ?to.



Il bot lascia un sacchetto in una posizione C, se il sacchetto è pieno.

Precondizioni: Il bot sta tenendo un sacchetto; il sacchetto si trova nella posizione C ed è pieno.

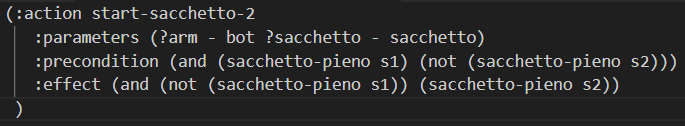
Effetti: Il sacchetto è lasciato nella posizione C; Il bot non tiene più il sacchetto ed è ora vuoto.



Azione di Ricarica. Il bot si ricarica se si trova nella posizione B e ha una batteria scarica.

Precondizioni: Il bot è nella posizione B; la batteria del bot è al livello 0 (scarica).

Effetti: Il livello della batteria del bot viene aggiornato a 3.



Il bot inizia a usare un altro sacchetto, in quanto il primo sacchetto (pieno) è già stato consegnato

Precondizioni: il sacchetto s1 è pieno e il sacchetto s2 non è pieno.

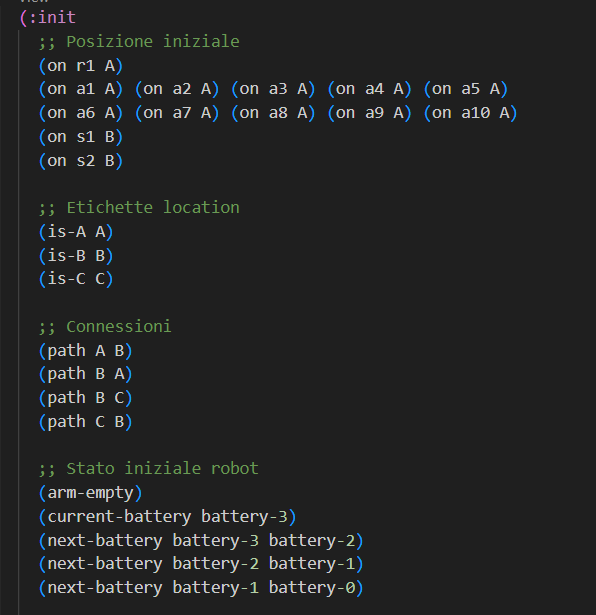
Effetti: Il sacchetto s2 diventa pieno.

3 problema

All’interno del problema, ci sono tre parti principali da definire.

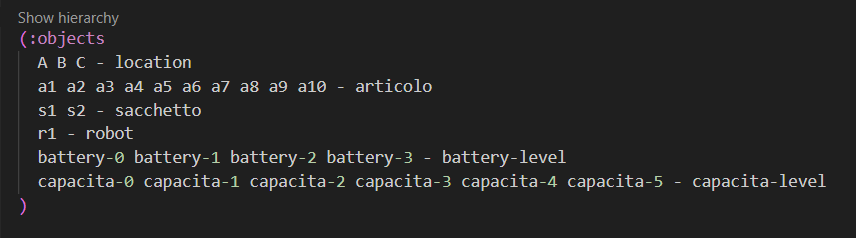
## 3.a inizializzazione

Predicati di stato iniziale: definisce le posizioni iniziali degli oggetti,  i vari passaggi che possono compiere gli oggetti e lo stato iniziale del braccio robotico.

(

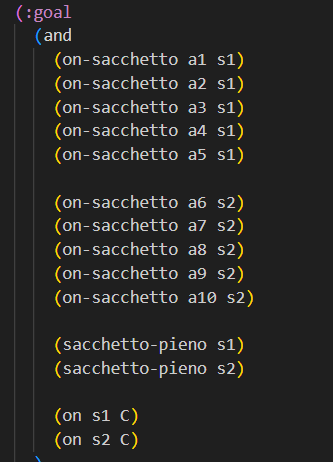
## 3.b OGGETTI

Definire i vari oggetti che partecipano e le caratteristiche di alcuni di questi.



## 3.c GLI OBIETTIVI

Definire i diversi obiettivi che devono essere completati, cioè il riempimento dei 2 sacchetti e il loro spostamento nel punto di consegna.



4. planner

Come già specificato lungo la documentazione, l’applicazione definisce funzioni e si basa in generale sui fluents, definiti a partire dalla versione 2.1 di PDDL. Perciò, è necessario utilizzare un planner numerico, con specifico supporto ai fluents.

Per avviare l’interfaccia grafica, basta scaricare il codice in python presente nel repository su GitHub.