



Note di implementazione - Task Manager

Prossima volta @Jun 6, 2021

Avviare il sistema Baxter

Implementazione - Task_Manager

Environment - cubetti, mano e tutto il resto

Task manager body

Assegnazione compiti - braccio destro

Condizioni varie - braccio destro

Assegnazione compiti - braccio sinistro

Condizioni - braccio sinistro

Esecuzione dei task

Controlli - esecuzione del task

Lettura delle informazioni da Unity

Debug @Jun 6, 2021

Errori a runtime

Come avviare il controllore

Prima esecuzione del task manager

https://github.com/programmatoreSeduto/SOFAR-ASSIGNMENT/tree/task_manager

Prossima volta @Jun 6, 2021

- nuova esecuzione del programma, correggi gli ultimi errori.

Avviare il sistema Baxter

semplicemente usa il launch file *human_baxter_collaboration* che trovi nell'omonimo package.

Implementazione - Task_Manager

Il compito è semplice: baxter deve spostare i cubetti blu dalla loro posizione attuale al goal. Ciclo di funzionamento del nodo:

- leggi la posizione degli oggetti (in parallelo rispetto all'elaborazione del task manager)
- cerca un cubetto da spostare dal lato sinistro al lato destro del tavolo

- cerca un cubetto da spostare dal lato destro del tavolo al goal
- esegui i due comandi in sequenza
- (aspetta un attimo e) ricomincia

Environment - cubetti, mano e tutto il resto

Anzitutto, che genere di informazioni escono da Unity?

```
frames:
-
  header:
    seq: 1
    stamp:
      secs: 29
      nsecs: 0
    frame_id: "pelvis"
  pose:
    position:
      x: 1.3234540224075317
      y: 0.09743370860815048
      z: 1.0112618207931519
    orientation:
      x: -0.21526095271110535
      y: 0.002820029854774475
      z: -0.9763075113296509
      w: 0.021871313452720642
```

```
>> rostopic info unity_tf
Type: human_baxter_collaboration/UnityTf

Publishers:
* /server_endpoint (http://3c27404a5b91:35787/)

Subscribers:
* /unity_tf_publisher (http://3c27404a5b91:36725/)
```

```
# human_baxter_collaboration/UnityTf
Header header
geometry_msgs/PoseStamped[] frames

# geometry_msgs/PoseStamped
std_msgs/Header header
geometry_msgs/Pose pose

# geometry_msgs/Pose
geometry_msgs/Point position
geometry_msgs/Quaternion orientation
```

Ora, di che informazioni ho bisogno?

- ~~posizione del gripper~~ (@Jun 6, 2021 non servono per via del meccanismo con *baxter_at_home*)
- posizione della mano dell'operatore
- posizione di tutti i blocchi rossi (quelli riservati all'operatore umano)
- posizione di tutti i blocchi blu (quelli che BAXTER deve sistemare)
- posizione del centro del tavolo

Task manager body

Il task manager assegna ad ogni braccio un compito rappresentato come messaggio command:

```
# controller_baxter/command.srv
# arm: "left", "right"
string arm
# pos: "box", "center"
string pos
# cube: E, M, C, G, I
string cube
---
bool ok

# controller_baxter/at_home.srv
# arm: "left", "right"
string arm
---
bool at_home
```

Assegnati i due compiti, il task manager manda i comandi al controller uno ad uno: prima quello sul braccio sinistro, poi quello sul braccio destro.

Assegnazione compiti - braccio destro

Seleziona tra i cubetti presenti, uno che soddisfi queste caratteristiche:

- Si trova sul lato destro del tavolo
- non è vicino alla mano dello human operator oltre la "soglia di pericolo" (definisci una distanza di allarme)
- possibilmente è più lontano possibile dallo human operator
- alcuni blocchi possono essere messi uno sopra l'altro. Se un blocco rosso si trova sopra un blocco blu, il blocco blu non è accessibile.

Nota: non ce ne frega nulla dell'efficienza in questo contesto, perciò possiamo anche trascurare la posizione del gripper per ora.

Supponiamo che i blocchi non caschino dal tavolo, o che l'umano non li sposti proprio mentre Baxter sta calcolando. Supponiamo anche che l'operatore usi solo la mano destra, e che esso non tocchi i blocchi di Baxter, mai.

Come implementare la scelta?

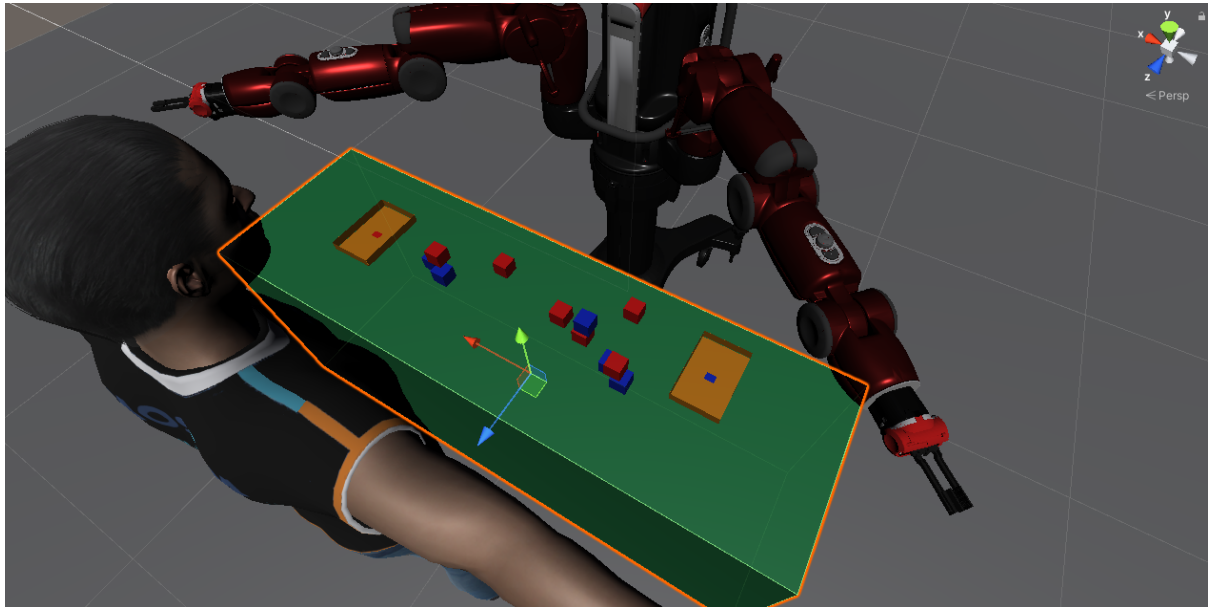
- ~~Fai una copia dell'array di blocchi rossi e di blocchi blu (magari falla nel task_manager_body così non devi rifarla tutte le volte) e copia anche altre eventuali informazioni per eliminare il problema della concorrenza~~
- ~~escludi tutti i blocchi sul lato destro del tavolo (filter su una condizione)~~
- ~~se rimangono blocchi, vai avanti~~
- ~~escludi tutti i blocchi entro la soglia critica~~
- ~~se rimangono blocchi, vai avanti~~
- ~~escludi tutti i blocchi non accessibili~~
- ~~se rimangono blocchi, vai avanti~~
- ~~ordina i blocchi in base a quello più lontano dalla mano dell'operatore~~
- ~~prendi il primo della lista~~

Il messaggio da inviare per ogni comando è questo:

```
arm: "right"
pos: "box"
cube: "E"    # cube: E, M, C, G, I
```

Condizioni varie - braccio destro

Una nota sul posizionamento degli assi di Unity riguardo al tavolo. La x è negativa nel lato destro, e z cresce allontanandosi dal robot.



nota: il frame del tavolo è orientato esattamente come il frame assoluto.

Blocco nel lato destro del tavolo:

```
# Il blocco si trova sul lato destro del tavolo?
# elem : ( 'name', Pose )
def block_on_right_table( elem ):
    global frame_table

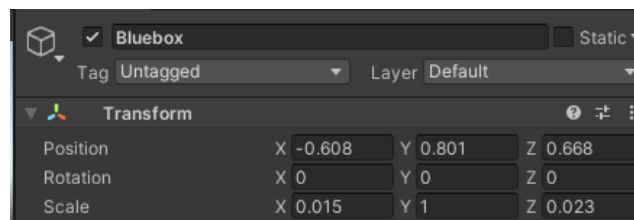
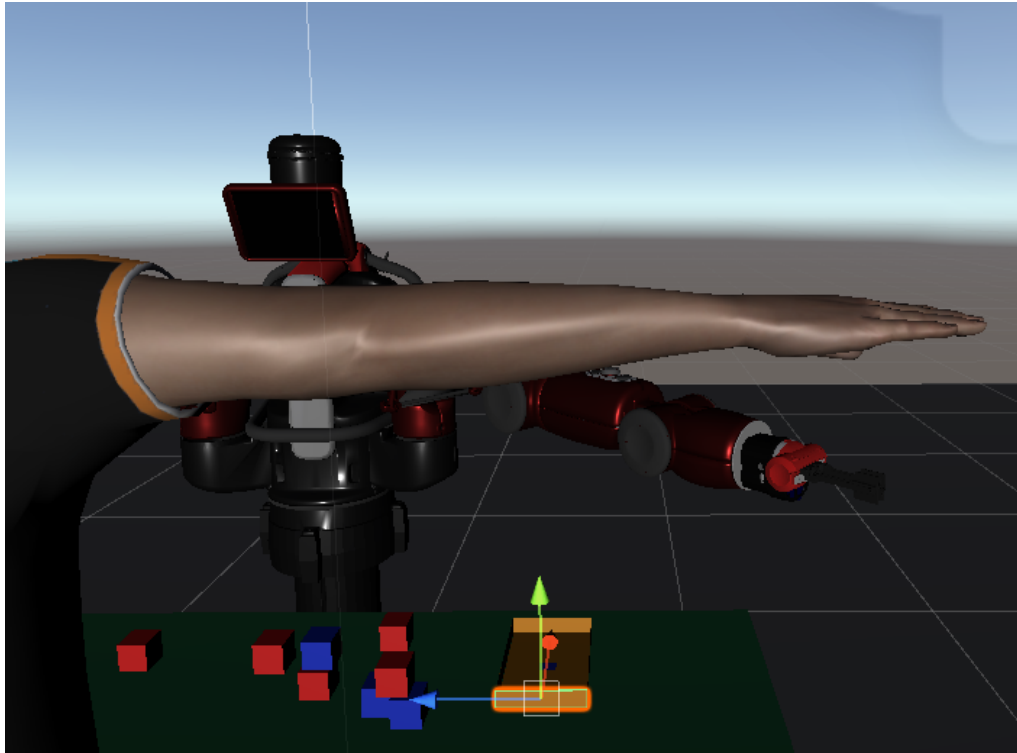
    P = [ elem[1].position.x, elem[1].position.z ]
    O = [ frame_table.position.x, frame_table.position.z ]

    # ritorna false se (supponiamo l'origine del tavolo perfettamente centrata)
    # x non Ã" in [ O[0], O[0] - sz_table[0]/2 ]
    # y non Ã" in [ O[1] - sz_table[2]/2, O[1] + sz_table[2]/2 ]

    if( P[0] > O[0] || P[0] < (O[0] - sz_table[0]/2) ):
        return False
    if( P[1] < (O[1] - sz_table[2]/2) || P[1] > (O[1] + sz_table[2]/2) ):
        return False

    return True
```

Blocco non troppo vicino alla mano. Quanto è grande più o meno la mano per Unity? e com'è posizionato il punto centrale più o meno?



La condizione: la distanza è maggiore della soglia di sicurezza, altrimenti non puoi prendere il blocco e devi aspettare.

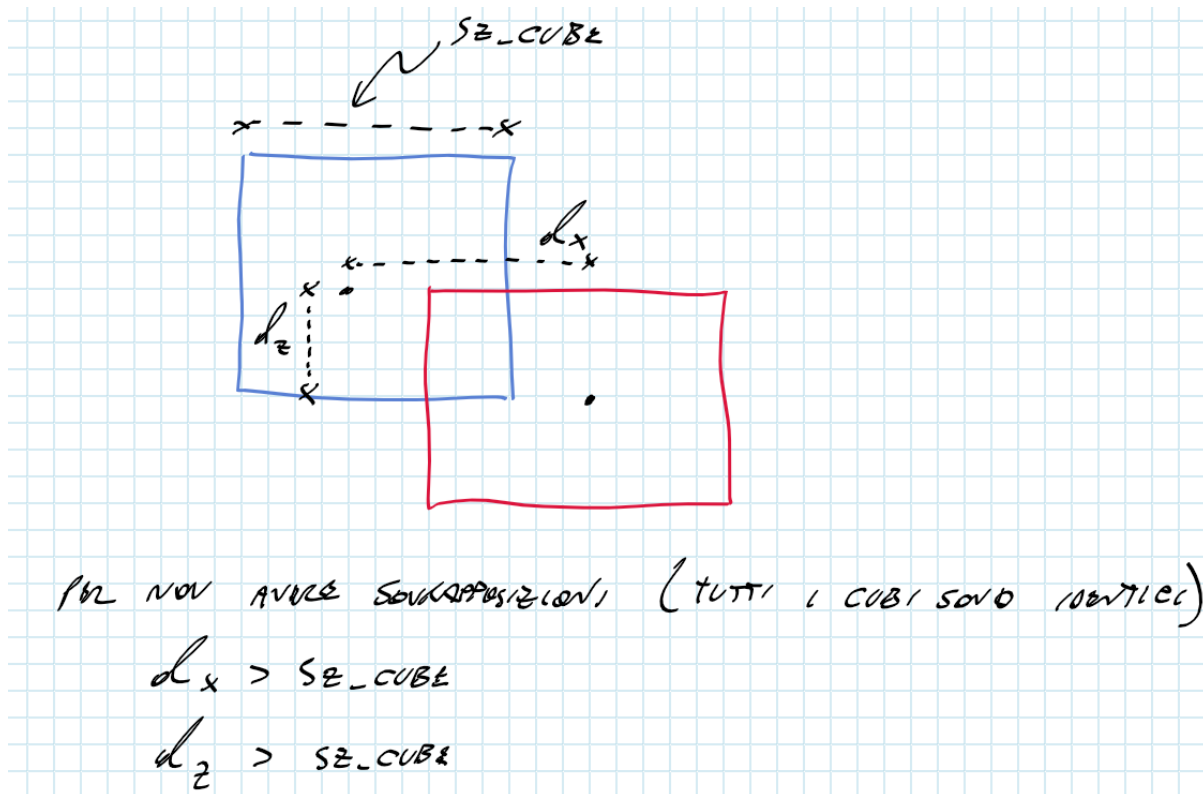
```
# segnala se il blocco Ã" troppo vicino all'operatore
# elem : ( 'name', Pose )
def block_not_near_to_hand( elem ):
    global frame_human_hand

    # calcola la distanza tra l'origine della mano e quella del blocco
    P = [ elem[1].position.x, elem[1].position.z ]
    O = [ frame_human_hand.position.x, frame_human_hand.position.z ]
    dist = math.sqrt( (P[0]-O[0])**2 + (P[1]-O[1])**2 )

    True if ( dist >= sz_hand ) else False
```

Ora, come verificare l'accessibilità di un blocco? Si vede guardando come sono posizionate le origini. Oppure guardando l'asse y?

L'asse y in questo caso non mi serve a nulla, perchè comunque dovrei individuare la sovrapposizione e quindi se le coordinate piane sono quelle che sono. Inoltre assumo che i blocchi non possano "compenetrarsi" in qualche modo, quindi se due blocchi sono sovrapposti allora le loro coordinate piane sono "abbastanza vicine". Vedi disegno:



Se non valgono queste condizioni sulle coordinate piane, c'è sicuramente sovrapposizione.

```
# il blocco è accessibile?
# elem : ( 'name', Pose )
def block_is_accessible( elem_b, elem_r ):
    dx = math.abs( elem_b.pose.x - elem_r.pose.x )
    dz = math.abs( elem_b.pose.z - elem_r.pose.z )

    return True if (dx < sz_cube and dz < sz_cube) else False
```

Assegnazione compiti - braccio sinistro

La procedura è grossomodo la stessa:

- copia di tutte le liste necessarie
- trova tutti i blocchi sul lato sinistro del tavolo
- se rimangono blocchi considerabili, vai avanti
- escludi i blocchi non accessibili
- se non rimangono blocchi, vai avanti
- ordina i blocchi in base a quello più lontano
- prendi il primo blocco e costruisci il task

```
arm: "left"
pos: "center"
cube: "E" # cube: E, M, C, G, I
```

Condizioni - braccio sinistro

L'unica condizione che cambia rispetto al braccio destro è quella per identificare i blocchi sul lato sinistro. La valutazione della coordinata z non cambia. La valutazione della x invece deve essere invertita. Nota solo la condizione sulla prima coordinata, l'altra è la stessa:

```
# Il blocco si trova sul lato destro del tavolo?
# elem : ( 'name', Pose )
def block_on_left_table( elem ):
    global frame_table

    P = [ elem[1].position.x, elem[1].position.z ]
    O = [ frame_table.position.x, frame_table.position.z ]

    if( P[0] < O[0] or P[0] > (O[0] + info.sz_table[0]/2) ):
        return False
    if( P[1] < (O[1] - info.sz_table[2]/2) or P[1] > (O[1] + info.sz_table[2]/2) ):
        return False

    return True
```

Esecuzione dei task

Alcune annotazioni preliminari su cosa dovrebbe fare questa parte di codice:

- eliminare un blocco dalla lista quando quello raggiunge il goal (lo vedi dai topic)...
- ...o quando esce fuori dal tavolo (in quel caso infatti Baxter non sarebbe in grado di recuperare quel blocco caduto a terra)

funzionamento:

- ricevuti i due task... estrai i nomi dei blocchi
- Invia entrambi i comandi insieme (i gripper potrebbero scontrarsi se i blocchi sono troppo vicini? per il momento ipotizziamo di no)
- attendi che entrambi i gripper abbiano raggiunto la posizione di base
- verifica la posizione dei due blocchi: il blocco che ha raggiunto il goal viene eliminato dalla lista di blocchi da considerare.

Codice di *baxter_at_home*:

- *baxter_at_home_server* è il nome del servizio
- di default la risposta contiene il campo *at_home* a false; se il braccio si avvicina oltre una certa tolleranza alla posizione di riferimento (casa) allora quel parametro scatta a true.
- la richiesta è di tipo *controller_baxter/at_home*, che ha un solo campo *arm*: left o right.

```
#include <ros/ros.h>
#include <geometry_msgs/PoseStamped.h>
#include <tf2_msgs/TFMessage.h>
#include <moveit/move_group_interface/move_group_interface.h>
#include <human_baxter_collaboration/BaxterTrajectory.h>
#include <human_baxter_collaboration/UnityTf.h>
#include <controller_baxter/at_home.h>
#include <sensor_msgs/JointState.h>
#include <moveit_msgs/RobotState.h>
#include "iostream"
#include <tf/tf.h>

//NAMESPACES
using namespace std;
using namespace moveit;
using namespace planning_interface;
using namespace geometry_msgs;

//GLOBAL VARIABLES
sensor_msgs::JointState messaggio_joint, left_arm_msg_joint, right_arm_msg_joint;
```

```

//SUBSCRIBER CALLBACKS
void joints_callback(const sensor_msgs::JointState& msg );

//SERVICES
bool server_callback(controller_baxter::at_home::Request &req,controller_baxter::at_home::Response &res);

//FUNCTIONS
bool very_close(double A ,double B);

int main(int argc, char** argv)
{
    // Initialize the ROS Node
    ros::init(argc, argv, "baxter_at_home");
    ros::NodeHandle n;
    ros::Subscriber joint_state_sub = n.subscribe("baxter_joint_states", 10,joints_callback);
    ros::ServiceServer baxter_at_home_server = n.advertiseService("baxter_at_home_server",server_callback);

    while(ros::ok()){
        ros::spinOnce();
    }
}

void joints_callback(const sensor_msgs::JointState& msg)
{
    right_arm_msg_joint=msg;
    right_arm_msg_joint.name.erase(right_arm_msg_joint.name.begin());
    right_arm_msg_joint.name.erase(right_arm_msg_joint.name.begin()+7,right_arm_msg_joint.name.begin()+18);
    right_arm_msg_joint.position.erase(right_arm_msg_joint.position.begin());
    right_arm_msg_joint.position.erase(right_arm_msg_joint.position.begin()+7,right_arm_msg_joint.position.begin()+18);

    left_arm_msg_joint=msg;
    left_arm_msg_joint.name.erase(left_arm_msg_joint.name.begin());
    left_arm_msg_joint.name.erase(left_arm_msg_joint.name.begin(),left_arm_msg_joint.name.begin()+7);
    left_arm_msg_joint.name.erase(left_arm_msg_joint.name.begin()+7,left_arm_msg_joint.name.begin()+11);
    left_arm_msg_joint.position.erase(left_arm_msg_joint.position.begin());
    left_arm_msg_joint.position.erase(left_arm_msg_joint.position.begin(),left_arm_msg_joint.position.begin()+7);
    left_arm_msg_joint.position.erase(left_arm_msg_joint.position.begin()+7,left_arm_msg_joint.position.begin()+11);
}

bool server_callback(controller_baxter::at_home::Request &req,controller_baxter::at_home::Response &res){

    res.at_home=false;
    sensor_msgs::JointState messaggio_joint;
    if(req.arm=="right") messaggio_joint=right_arm_msg_joint;
    if(req.arm=="left")  messaggio_joint=left_arm_msg_joint;
    if(req.arm!="left"&& req.arm!="right") return false;

    cout<<"joint0: "<<messaggio_joint.position[0]<<endl;
    cout<<"joint1: "<<messaggio_joint.position[1]<<endl;
    cout<<"joint2: "<<messaggio_joint.position[2]<<endl;
    cout<<"joint3: "<<messaggio_joint.position[3]<<endl;
    cout<<"joint4: "<<messaggio_joint.position[4]<<endl;
    cout<<"joint5: "<<messaggio_joint.position[5]<<endl;
    cout<<"joint6: "<<messaggio_joint.position[6]<<endl;

    if((very_close(messaggio_joint.position[0],0.52)||very_close(messaggio_joint.position[0],-0.52))&&
    very_close(messaggio_joint.position[1],-1.22)&&
    very_close(messaggio_joint.position[2],0) &&
    very_close(messaggio_joint.position[3],1.72) &&
    very_close(messaggio_joint.position[4],0) &&
    very_close(messaggio_joint.position[5],0.75) &&
    very_close(messaggio_joint.position[6],0))
        {res.at_home=true;}

    return true;
}

bool very_close(double A ,double B){
    if(A<B+0.1 && A>B-0.01)return true;
    return false;
}

```

Controlli - esecuzione del task

Come fare il check del goal?

- siano d0 e d1 i lati della scatola
- Allora P deve avere [0] tra goal-dx/2 e goal+dx intorno a O[0]
- Lo stesso dicasi per la coordinata [1]

Il seguente codice ha un problema, dovuto al fatto che il punto che viene controllato si trova al centro dell'area piana del blocco. Quindi, questo codice controlla che ci sia almeno una parte di area piana comune tra blocco e goal, considerandolo dentro anche quando si trova parzialmente fuori. *Per i fini del progetto, questa cosa non dovrebbe rappresentare un problema.*

```
# il blocco blu si trova nel goal?
# elem : String
def block_in_blue_goal( elem ):
    global frame_block_blue, frame_goal_blue

    # supponiamo che il blocco esista nella lista
    # SOLO SE questa funzione viene utilizzata da task_execute()
    # e non richiamata da altre parti del codice

    # posizione del blocco e del goal
    P = [ frame_block_blue[elem].position.x, frame_block_blue[elem].position.z ]
    O = [ frame_goal_blue.position.x, frame_goal_blue.position.z ]

    # ritorna true se il blocco si trova all'interno della scatola
    if( P[0] < (O[0] - info.sz_goal[0]/2 ) or P[0] > (O[0] + info.sz_goal[0]/2 ) ):
        return False
    if( P[1] < (O[1] - info.sz_goal[2]/2 ) or P[1] > (O[1] + info.sz_goal[2]/2 ) ):
        return False

    return True
```

Lo stesso identico controllo può essere implementato per i blocchi rossi. Tutti i blocchi rossi che hanno raggiunto il goal semplicemente vengono eliminati dalla lista.

```
# il blocco rosso si trova nel goal?
# elem : String
def block_in_red_goal( elem ):
    global frame_block_blue, frame_goal_blue

    # il blocco esiste
    # SOLO SE questa funzione viene richiamata da
    # task_manager_body
    # altrimenti potrebbe sollevare eccezione

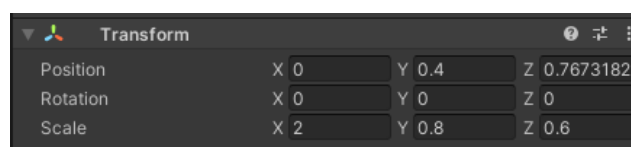
    # posizione del blocco e del goal
    P = [ frame_block_blue[elem].position.x, frame_block_blue[elem].position.z ]
    O = [ frame_goal_red.position.x, frame_red_blue.position.z ]

    # ritorna true se il blocco si trova all'interno della scatola
    if( P[0] < (O[0] - info.sz_goal[0]/2 ) or P[0] > (O[0] + info.sz_goal[0]/2 ) ):
        return False
    if( P[1] < (O[1] - info.sz_goal[2]/2 ) or P[1] > (O[1] + info.sz_goal[2]/2 ) ):
        return False

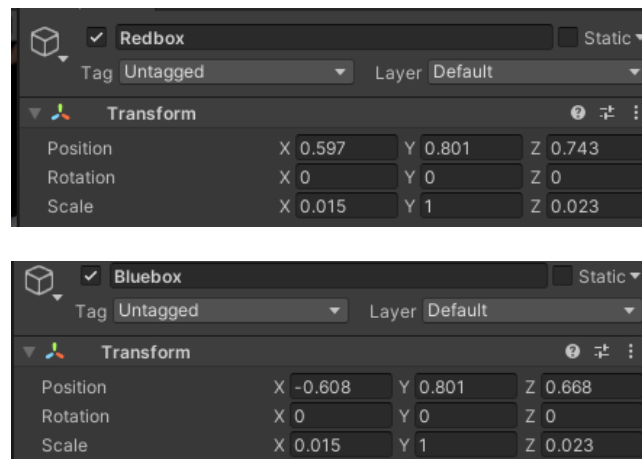
    return True
```

Lettura delle informazioni da Unity

MANCA LA PUBBLICAZIONE DELLA POSIZIONE DEL TAVOLO MANNAGGIA A LORO! Hard coding...



A questo punto posso fare hard coding anche per le posizioni dei goal, tanto esattamente come il tavolo non possono cambiare posizione.



Debug @Jun 6, 2021

- Anzitutto... l'importazione del codice Python direttamente da src in file dipendenti. Cosa non funziona? Mi spiego: se volessi creare un file contenente solo costanti e metterlo in src, dovrei essere libero di farlo no? Vediamo di provare subito, creando un nuovo package semplicissimo che contenga due file: uno è il file py principale, e l'altro quello da cui il file py prende le informazioni. Semplice stampa a video, e vediamo che succede.

nuovo package: ros_ws py_test

due file: main.py, e data.py.

Entrambi eseguibili: chmod ...

Provo a mettere l'interprete per entrambi i file, anche per quello di dati. Nel codice che ho scritto non c'era.

main.py:

```
#!/usr/bin/env python

import data
import rospy

print( "main!" )

rospy.init_node( "test_py_main" )
print(data.mydata)
```

data.py:

```
#!/usr/bin/env python

print( "data!" )
mydata = 21
```

Aggiustatina al makefile: stavolta rovo a registrare anche il file data.

```
catkin_install_python(PROGRAMS
  scripts/main.py
  scripts/data.py
  DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
)
```

Ora vediamo che succede.

```
root@3c27404a5b91:~/ros_ws# rosrun py_test main.py
data!
main!
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/py_test/main.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/py_test/scripts/main.py", line 9, in <module>
    print(data.mydata)
AttributeError: module 'data' has no attribute 'mydata'
```

vede entrambi i file e li esegue ... però non gli piace quel modulo. Dev'esserci qualcosa che non va nell'importazione. Ricerchina stupida di Python.

A quanto pare il metodo di importazione è corretto, però per qualche strana ragione, ROS non è d'accordo. Dev'esserci qualcosa che non quadra nei percorsi.

Interessante cosa succede apportando queste modifiche:

main.py:

```
#!/usr/bin/env python

# import data
from data import myfunct
import rospy

print( "main!" )

rospy.init_node( "test_py_main" )
# print( data )
# print( myfunct() )
```

data.py:

```
#!/usr/bin/env python

print( "data!" )

mydata = 21
def myfunct():
    return "->21<-"
```

output:


```
root@3c27404a5b91:~/ros_ws# rosrun py_test main.py
data!
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/py_test/main.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/py_test/scripts/main.py", line 4, in <module>
    from data import myfunct
ImportError: cannot import name 'myfunct' from 'data' (/root/ros_ws/devel/lib/py_test/data.py)
```

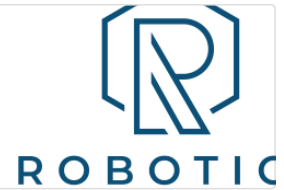
... perchè diavolo sta cercando la libreria in devel piuttosto che in src???

<https://duckietown.mit.edu/media/pdfs/1rpRisFoCYUm0XT78j-nAYidlh-cDtLCdEblaBCnx9ew.pdf>

[ROS] How To Import a Python Module From Another Package - The Robotics Back-End

In this tutorial I'll show you how to import a Python module from another package. ROS allows you to do that, and as you'll see it's quite simple.

 <https://roboticsbackend.com/ros-import-python-module-from-another-package/>



ok non era così facile in effetti ... ecco cosa ho fatto per risolvere:

1. crea nel package una cartella include/nome_del_package/
2. crea dentro questa cartella tutti i file py che ti servono da importante
3. crea un file setup.py nella home del package, vedi sotto per il contenuto
4. CMakeLists: decommenta `catkin_python_setup()` proprio all'inizio
5. `catkin_make`, e tutto dovrebbe andare.

file setup.py:

```
from distutils.core import setup
from catkin_pkg.python_setup import generate_distutils_setup
d = generate_distutils_setup(
    packages=['py_test'],
    package_dir={'': 'include'}
)
setup(**d)
```

main.py:

```
#!/usr/bin/env python

# import data
from py_test.data import myfunct, mynumber
import rospy

print( "main!" )

rospy.init_node( "test_py_main" )
print( mynumber )
print( myfunct() )

print( "importazione come modulo (con un altro nome)" )
import py_test.data as info
print( "mynumber = " + str( info.mynumber ) )
```

data.py:

```
#!/usr/bin/env python

print( "data!" )

mynumber = 21

def myfunct():
    return "->21<-"
```

Ora l'importazione funziona! Andiamo avanti.

- Callback Unity errata. Vedi la precedente versione:

```
# callback topic unity
def callback_unity( data ):
    global frame_human_hand, frame_block_red, frame_block_blue, frame_table, frame_goal_blue, frame_goal_red

    # etichette dei blocchi
```

```

labels_red = frame_block_red.keys()
labels_blue = frame_block_blue.keys()

# etichetta del frame da Unity
name = data.header.frame_id

if ( name in frame_block_red ):
    frame_block_red[ name ] = data.pose
elif ( name in frame_block_blue ):
    frame_block_blue[ name ] = data.pose
elif ( name == info.name_human_hand ):
    frame_human_hand = data.pose
else:
    # ignora il messaggio
    pass

```

Se i messaggi arrivassero in sequenza, sarebbe corretta. Peccato che no nsia così: i messaggi arrivano in un unico pacchetto che contiene tutti i frame di Unity in un array di *frames*. Il codice dà questo errore:

```

[ERROR] [1622989133.894343]: bad callback: <function callback_unity at 0x7f01e4e04e50>
Traceback (most recent call last):
  File "/opt/ros/noetic/lib/python3/dist-packages/rospy/topics.py", line 750, in _invoke_callback
    cb(msg)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 74, in callback_unity
    name = data.header.frame_id
AttributeError: 'UnityTf' object has no attribute 'header'

```

- Filter() è una funzione, e non un membro!
- Questo è strano:

```

Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 406, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 401, in main
    task_manager_body()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 214, in task_manager_body
    task_arm_right = get_task_arm_right()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 253, in get_task_arm_right
    if( len(bluebox) > 0 ):
TypeError: object of type 'filter' has no len()

```

Ora, quella len che gli da tanto fastidio è stata ottenuta così (e simili):

```

bluebox = filter( block_on_left_table, bluebox )

```

Mi possibile che filter() ritorni un oggetto che non è una lista? E che il tipo che sputa fuori non sia compatibile con len()?

How To Use the Python Filter Function | DigitalOcean

The Python built-in filter() function can be used to create a new iterator from an existing iterable (like a list or dictionary) that will efficiently filter out elements using a function that we provide. In this tutorial, we'll review four different


<https://www.digitalocean.com/community/tutorials/how-to-use-the-python-filter-function>



A quanto pare, quello che esce dalla filter va poi convertito in lista:

How to find the length of a "filter" object in python

Thanks for contributing an answer to Stack Overflow! Please be sure to answer the question. Provide details and share your research! Asking for help, clarification, or responding to other answers. Making statements based on opinion; back them up with

 <https://stackoverflow.com/questions/19182188/how-to-find-the-length-of-a-filter-object-in-python>



così dovrebbe andare:

```
bluebox = list(filter( block_on_right_table, bluebox ))
```

- Altro problema con la filter: questa si aspetta due argomenti, e non tre a quanto dice questa schermata.

```
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 406, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 401, in main
    task_manager_body()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 217, in task_manager_body
    task_arm_left = get_task_arm_left()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 307, in get_task_arm_left
    bluebox = list(filter( block_is_accessible, bluebox, redbox ))
TypeError: filter expected 2 arguments, got 3
```

niente, non si può fare. La modifica è presto fatta comunque. Guardiamo in faccia quella funzione per il filter:

```
# il blocco e' accessibile?
# elem : ( 'name', Pose )
def block_is_accessible( elem_b, elem_r ):
    dx = math.abs( elem_b.pose.x - elem_r.pose.x )
    dz = math.abs( elem_b.pose.z - elem_r.pose.z )

    return True if (dx < info.sz_cube and dz < info.sz_cube) else False
```

Dovrei reimplementarla in modo che sia la funzione stessa a fare lo scorrimento elemento per elemento. Inoltre questa implementazione è errata, perchè il fatto che ci siano due aree piane sovrapposte non significa che il blocco rosso sia necessariamente sopra quello blu. Quindi, bisogna considerare anche la y oltre a z e x.

Modifica: la funzione prende un solo argomento.

```
# il blocco e' accessibile?
# elem : ( 'name', Pose )
def block_is_accessible( elem_b ):
    global frame_block_red

    for elem_r in frame_block_red.values():
        dx = math.abs( elem_b[1].pose.x - elem_r.pose.x )
        dz = math.abs( elem_b[1].pose.z - elem_r.pose.z )
        # se dy < 0 allora il blocco rosso si trova sopra quello blu
        dy = elem_b[1].pose.y - elem_r.pose.y

        if (dx < info.sz_cube and dz < info.sz_cube and dy < 0):
            return False

    return True
```

Mi è venuto un dubbio terribile: gli elem sono a posto dalle altre parti? ricorda che sono delle ennuple, e io devo considerare solo il secondo elemento di ogni ennupla in queste valutazioni...

E ora posso mettere a posto quei filter.

- Per qualche strana ragione non mi trova il valore assoluto.

```
[INFO] [1622990922.696552]: task_manager online
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 414, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 409, in main
    task_manager_body()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 225, in task_manager_body
    task_arm_left = get_task_arm_left()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 315, in get_task_arm_left
    bluebox = list(filter( block_is_accessible, bluebox ))
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 171, in block_is_accessible
    dx = math.abs( elem_b[1].pose.x - elem_r.pose.x )
AttributeError: module 'math' has no attribute 'abs'
```

Potrebbe essere perchè si tratta di una built-in.

abs() in Python - GeeksforGeeks

The abs() function is used to return the absolute value of a number. Syntax: abs(number)
number : Can be integer, a floating point number or a complex number The abs() takes only one argument, a number whose absolute value is to be returned. The argument can

➦ <https://www.geeksforgeeks.org/abs-in-python/>



- Un problema con pose:

```
[INFO] [1622991033.837483]: task_manager online
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 414, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 409, in main
    task_manager_body()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 225, in task_manager_body
    task_arm_left = get_task_arm_left()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 315, in get_task_arm_left
    bluebox = list(filter( block_is_accessible, bluebox ))
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 171, in block_is_accessible
    dx = abs( elem_b[1].pose.x - elem_r.pose.x )
AttributeError: 'Pose' object has no attribute 'pose'
[INFO] [1622991034.340173]: task_manager offline
```

Infatti si chiama *position* e non *pose*...

```
# il blocco e' accessibile?
# elem : ( 'name', Pose )
def block_is_accessible( elem_b ):
    global frame_block_red

    for elem_r in frame_block_red.values():
        dx = abs( elem_b[1].position.x - elem_r.position.x )
        dz = abs( elem_b[1].position.z - elem_r.position.z )
        # se dy < 0 allora il blocco rosso si trova sopra quello blu
        dy = elem_b[1].position.y - elem_r.position.y

        if (dx < info.sz_cube and dz < info.sz_cube and dy < 0):
            return False

    return True
```

Errori corretti finora (ne rimangono un po'):

```
root@3c27404a5b91:~/ros_ws/src/controller_baxter/scripts# rosrun controller_baxter task_manager.py
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 328
    task_execute( task_right, task_left ):
                ^
SyntaxError: invalid syntax

root@3c27404a5b91:~/ros_ws/src/controller_baxter/scripts# rosrun controller_baxter task_manager.py
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 11, in <module>
    import controller_baxter.sim_infos as info
  File "/root/ros_ws/src/controller_baxter/include/controller_baxter/sim_infos.py", line 37, in <module>
    info.server_baxter_at_home = "baxter_at_home_server"
NameError: name 'info' is not defined

root@3c27404a5b91:~/ros_ws/src/controller_baxter/scripts# rosrun controller_baxter task_manager.py
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 405, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 384, in main
    rospy.init( "task_manager" )
AttributeError: module 'rospy' has no attribute 'init'

root@3c27404a5b91:~/ros_ws/src/controller_baxter/scripts# rosrun controller_baxter task_manager.py
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 405, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 396, in main
    srv_baxter_home = rospy.ServiceProxy( info.server_baxter_at_home )
TypeError: __init__() missing 1 required positional argument: 'service_class'
[INFO] [1622989133.891659]: task_manager offline
[ERROR] [1622989133.894343]: bad callback: <function callback_unity at 0x7f01e4e04e50>
Traceback (most recent call last):
  File "/opt/ros/noetic/lib/python3/dist-packages/rospy/topics.py", line 750, in _invoke_callback
    cb(msg)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 74, in callback_unity
    name = data.header.frame_id
AttributeError: 'UnityTf' object has no attribute 'header'

root@3c27404a5b91:~/ros_ws/src/controller_baxter/scripts# rosrun controller_baxter task_manager.py
[INFO] [1622989363.495959]: task_manager online
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 406, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 401, in main
    task_manager_body()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 214, in task_manager_body
    task_arm_right = get_task_arm_right()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 250, in get_task_arm_right
    bluebox = bluebox.filter( block_on_right_table, bluebox )
AttributeError: 'dict_items' object has no attribute 'filter'
[INFO] [1622989363.497772]: task_manager offline
root@3c27404a5b91:~/ros_ws/src/controller_baxter/scripts# rosrun controller_baxter task_manager.py
[INFO] [1622989472.537838]: task_manager online
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 406, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 401, in main
    task_manager_body()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 214, in task_manager_body
    task_arm_right = get_task_arm_right()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 253, in get_task_arm_right
    if( len(bluebox) > 0 ):
TypeError: object of type 'filter' has no len()
[INFO] [1622989472.540102]: task_manager offline
root@3c27404a5b91:~/ros_ws/src/controller_baxter/scripts# rosrun controller_baxter task_manager.py
[INFO] [1622989888.739695]: task_manager online
```



```

Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 406, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 401, in main
    task_manager_body()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 217, in task_manager_body
    task_arm_left = get_task_arm_left()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 307, in get_task_arm_left
    bluebox = list(filter( block_is_accessible, bluebox, redbox ))
TypeError: filter expected 2 arguments, got 3
[INFO] [1622989889.242312]: task_manager offline
root@3c27404a5b91:~/ros_ws/src/controller_baxter/scripts# roslaunch controller_baxter task_manager.py
[INFO] [1622990922.696552]: task_manager online
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 414, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 409, in main
    task_manager_body()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 225, in task_manager_body
    task_arm_left = get_task_arm_left()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 315, in get_task_arm_left
    bluebox = list(filter( block_is_accessible, bluebox ))
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 171, in block_is_accessible
    dx = math.abs( elem_b[1].pose.x - elem_r.pose.x )
AttributeError: module 'math' has no attribute 'abs'
[INFO] [1622990923.199566]: task_manager offline
root@3c27404a5b91:~/ros_ws/src/controller_baxter/scripts# roslaunch controller_baxter task_manager.py
[INFO] [1622991033.837483]: task_manager online
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 414, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 409, in main
    task_manager_body()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 225, in task_manager_body
    task_arm_left = get_task_arm_left()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 315, in get_task_arm_left
    bluebox = list(filter( block_is_accessible, bluebox ))
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 171, in block_is_accessible
    dx = abs( elem_b[1].pose.x - elem_r.pose.x )
AttributeError: 'Pose' object has no attribute 'pose'
[INFO] [1622991034.340173]: task_manager offline
root@3c27404a5b91:~/ros_ws/src/controller_baxter/scripts# roslaunch controller_baxter task_manager.py
[INFO] [1622991294.318720]: task_manager online
^C[INFO] [1622991312.158433]: task_manager offline

```

Errori a runtime

- Vedi questo errore,

```

File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 436, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 431, in main
    task_manager_body()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 241, in task_manager_body
    task_execute(task_arm_right, task_arm_left)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 387, in task_execute
    left_at_home = ( srv_baxter_home( req ) ).at_home
TypeError: 'NoneType' object is not callable

```

dovuto a questo genere di istruzione: chiamo il servizio, e di quello che mi ritorna tento di leggere un valore interno. Purtroppo, non funziona, non so perchè.

```
left_at_home = ( srv_baxter_home( req ) ).at_home
```

```
# modifica:
left_at_home = srv_baxter_home( req )
left_at_home = left_at_home.at_home
```

Non è proprio elegantissimo fare questi magheggi coi tipi, ma funziona, quindi va bene.

... scherzavo, non va:

```
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 438, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 433, in main
    task_manager_body()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 241, in task_manager_body
    task_execute(task_arm_right, task_arm_left)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 388, in task_execute
    left_at_home = srv_baxter_home( req )
TypeError: 'NoneType' object is not callable
```

SOLUZIONE: OCCHIO AI GLOBAL! Occhio quando scrivi sui dati globali: se il nme globale non compare dopo `global`, il valore non verrà scritto sulla variabile globale ma su una locale che poi andrà a perdersi alla chiusura del programma. Attenzione!

Come avviare il controllore

avviare ROS e il launch file

```
roscore &
roslaunch human_baxter_collaboration human_baxter_collaboration.launch &
```

(su windows) avviare il pacchetto `MMILauncher.exe`

sempre su windows, fai partire Unity e lancia la simulazione. Dovresti vedere sulla consolle di ROS un messaggio del tipo

```
ROS-Unity Handshake received, will connect to 192.168.65.2:5005
```

lancia la simulazione

A QUESTO PUNTO puoi far partire il controller:

```
roslaunch controller_baxter baxter_at_home &

# ignora i warning da qui
roslaunch controller_baxter controller_baxter &
```

Dovresti vedere questo su consolle se il nodo `controller_baxter` è stato lanciato correttamente:

```
[1622991828.338282000]: Loading robot model 'baxter'...
[ WARN] [1622991828.366259400]: Link display has visual geometry but no collision geometry. Collision geometry will be left empty. Fix your URDF file by explicitly specifying collision geometry.
[ WARN] [1622991828.366382800]: Link left_hand_accelerometer has visual geometry but no collision geometry. Collision geometry will be left empty. Fix your URDF file by explicitly specifying collision geometry.
[ WARN] [1622991828.366439100]: Link left_hand_camera has visual geometry but no collision geometry. Collision geometry will be left empty. Fix your URDF file by explicitly specifying collision geometry.
[ WARN] [1622991828.366476700]: Link left_hand_range has visual geometry but no collision geometry. Collision geometry will be left empty. Fix your URDF file by explicitly specifying collision geometry.
[ WARN] [1622991828.382643900]: Link right_hand_accelerometer has visual geometry but no collision geometry. Collision geometry will be left empty. Fix your URDF file by explicitly specifying collision geometry.
[ WARN] [1622991828.382720900]: Link right_hand_camera has visual geometry but no collision geometry. Collision geometry
```

```

will be left empty. Fix your URDF file by explicitly specifying collision geometry.
[ WARN] [1622991828.382785800]: Link right_hand_range has visual geometry but no collision geometry. Collision geometry will be left empty. Fix your URDF file by explicitly specifying collision geometry.
[ WARN] [1622991828.431535900]: Kinematics solver doesn't support #attempts anymore, but only a timeout.
Please remove the parameter '/robot_description_kinematics/left_arm/kinematics_solver_attempts' from your configuration.
[ INFO] [1622991829.470414700]: Ready to take commands for planning group right_arm.
[ INFO] [1622991829.810458200]: Ready to take commands for planning group left_arm.
[ INFO] [1622991830.010511200]: Ready to take commands for planning group both_arms.

```

Per lanciare il task manager,

```
roslaunch controller_baxter task_manager.py
```

In caso dovessi riavviare la simulazione perchè Baxter ha mosso qualche blocchetto, prima ricorda di terminare le precedenti istanze dei processi `baxter_at_home` e `controller_baxter`:

```

ps
kill <PID controller_baxter> <PID baxter_at_home>

```

Dopodichè, segui la procedura dall'avvio della simulazione in poi.

Prima esecuzione del task manager

ovviamente senza log ... provvedo ad aggiungere, altrimenti non ci si capisce nulla.

```

root@3c27404a5b91:~/SOFAR_ASSIGNMENT/controller_baxter/launch# roslaunch controller_baxter task_manager.py
[INFO] [1622992642.445602]: subscribe: unity_tf ...
[INFO] [1622992642.447582]: subscribe: unity_tf ... OK!
[INFO] [1622992642.448878]: service: controller_server ...
[INFO] [1622992642.452093]: service: controller_server ... OK!
[INFO] [1622992642.453167]: service: baxter_at_home_server ...
[INFO] [1622992642.455145]: service: baxter_at_home_server ... OK!
[INFO] [1622992642.455945]: task_manager online
[ INFO] [1622992642.959807200]: Trying to execute a trajectory
[ INFO] [1622992643.962460300]: Didn't received robot state (joint angles) with recent timestamp within 1 seconds.
Check clock synchronization if your are running ROS across multiple machines!
[ WARN] [1622992643.962539200]: Failed to fetch current robot state.
[ INFO] [1622992643.962830300]: Planning request received for MoveGroup action. Forwarding to planning pipeline.
[ INFO] [1622992643.976922300]: Planner configuration 'left_arm' will use planner 'geometric::RRTConnect'. Additional configuration parameters will be set when the planner is constructed.
[ INFO] [1622992643.978280000]: left_arm/left_arm: Starting planning with 1 states already in datastructure
[ INFO] [1622992643.989916100]: left_arm/left_arm: Created 4 states (2 start + 2 goal)
[ INFO] [1622992643.989989500]: Solution found in 0.011944 seconds
[ INFO] [1622992643.992765500]: SimpleSetup: Path simplification took 0.002655 seconds and changed from 3 to 2 states
[ INFO] [1622992643.994427100]: Plan Result:SUCCESS
[ INFO] [1622992645.073346800]: Didn't received robot state (joint angles) with recent timestamp within 1 seconds.
Check clock synchronization if your are running ROS across multiple machines!
[ WARN] [1622992645.073417500]: Failed to fetch current robot state.
[ INFO] [1622992645.073573400]: Planning request received for MoveGroup action. Forwarding to planning pipeline.
[ INFO] [1622992645.074814100]: Planner configuration 'left_arm' will use planner 'geometric::RRTConnect'. Additional configuration parameters will be set when the planner is constructed.
[ INFO] [1622992645.075002600]: left_arm/left_arm: Starting planning with 1 states already in datastructure
[ INFO] [1622992645.085802500]: left_arm/left_arm: Created 5 states (2 start + 3 goal)
[ INFO] [1622992645.085894300]: Solution found in 0.010963 seconds
[ INFO] [1622992645.089584800]: SimpleSetup: Path simplification took 0.003563 seconds and changed from 4 to 2 states
[ INFO] [1622992645.090083400]: Plan Result:SUCCESS
[ INFO] [1622992646.173644300]: Didn't received robot state (joint angles) with recent timestamp within 1 seconds.
Check clock synchronization if your are running ROS across multiple machines!
[ WARN] [1622992646.173721300]: Failed to fetch current robot state.
[ INFO] [1622992646.173860900]: Planning request received for MoveGroup action. Forwarding to planning pipeline.
[ INFO] [1622992646.175055900]: Planner configuration 'left_arm' will use planner 'geometric::RRTConnect'. Additional configuration parameters will be set when the planner is constructed.
[ INFO] [1622992646.175312400]: left_arm/left_arm: Starting planning with 1 states already in datastructure
[ INFO] [1622992646.186058700]: left_arm/left_arm: Created 4 states (2 start + 2 goal)
[ INFO] [1622992646.186141100]: Solution found in 0.010931 seconds
[ INFO] [1622992646.190369600]: SimpleSetup: Path simplification took 0.004163 seconds and changed from 3 to 2 states
[ INFO] [1622992646.190921400]: Plan Result:SUCCESS
[ INFO] [1622992647.273427500]: Didn't received robot state (joint angles) with recent timestamp within 1 seconds.
Check clock synchronization if your are running ROS across multiple machines!

```

```

[ WARN] [1622992647.273493900]: Failed to fetch current robot state.
[ INFO] [1622992647.273619800]: Planning request received for MoveGroup action. Forwarding to planning pipeline.
[ INFO] [1622992647.274841600]: Planner configuration 'left_arm' will use planner 'geometric::RRTConnect'. Additional configuration parameters will be set when the planner is constructed.
[ INFO] [1622992647.275040100]: left_arm/left_arm: Starting planning with 1 states already in datastructure
[ INFO] [1622992647.285597800]: left_arm/left_arm: Created 4 states (2 start + 2 goal)
[ INFO] [1622992647.285692400]: Solution found in 0.010725 seconds
[ INFO] [1622992647.289736900]: SimpleSetup: Path simplification took 0.003905 seconds and changed from 3 to 2 states
[ INFO] [1622992647.290422400]: Plan Result:SUCCESS
Traceback (most recent call last):
  File "/root/ros_ws/devel/lib/controller_baxter/task_manager.py", line 15, in <module>
    exec(compile(fh.read(), python_script, 'exec'), context)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 420, in <module>
    main()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 415, in main
    task_manager_body()
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 229, in task_manager_body
    task_execute(task_arm_right, task_arm_left)
  File "/root/ros_ws/src/controller_baxter/scripts/task_manager.py", line 372, in task_execute
    left_at_home = ( srv_baxter_home( req ) ).at_home
TypeError: 'NoneType' object is not callable
[INFO] [1622992647.450312]: task_manager offline

```