THE ANDREW & ERNA VITERBI FACULTY OF ELECTRICAL ENGINEERING

CRML — Control Robotics and Machine Learning Laboratory

TECHNION Israel Institute of Technology

Artificial Intelligence Planning for Solving Blocks World with Robots

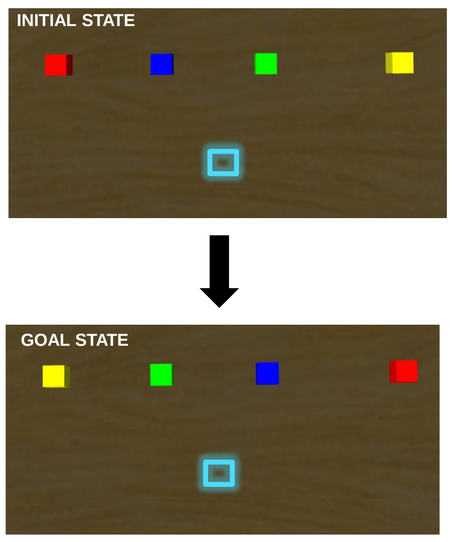
Matan Kleiner and Yuval Snir, Supervised by Ronen Nir

Introduction

The project's goal is to solve a blocs world problem with autonomous agent using Sense-Think-Act methodology.

The Problem

The blocks world problem we solve in the project is made of four blocks in different colors which we want to move from the initial state to the goal state using one empty spot.



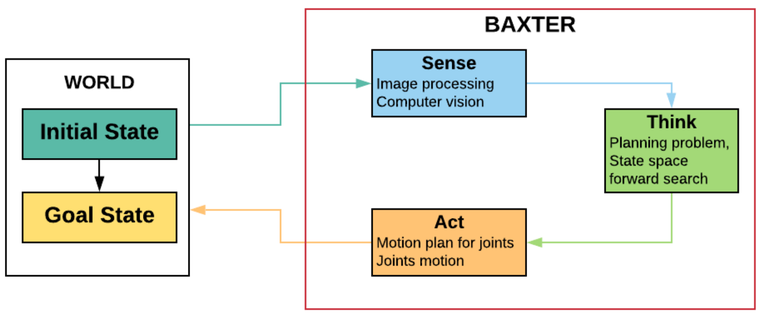
An example for initial and goal state

Sense-Think-Act

Sense-Think-Act methodology is a robot paradigm that set the stages of solution:

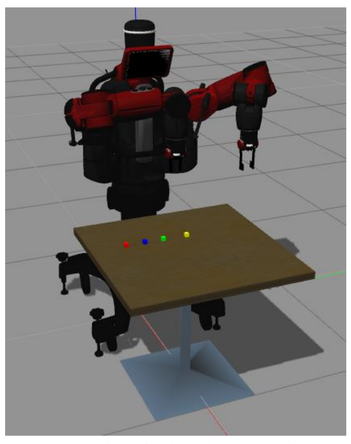
- Sense – use the autonomous agent's sensors and cameras to deduct the world's state.
- Think – calculate an action plan based on the initial state the agent sensed in the previous stage.
- Act – execute the action plan using the agent's mechanics limbs and end effectors.

The Sense-Think-Act diagram of the project:



Work Environment

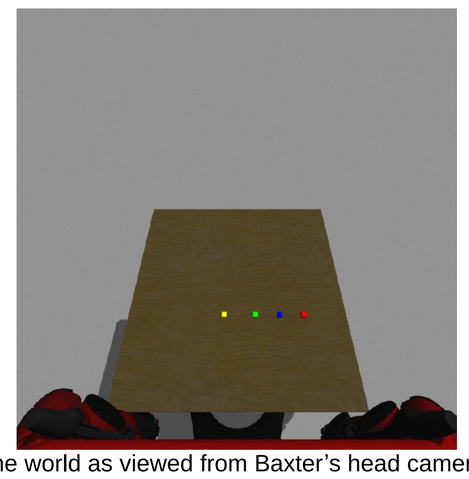
In this project we will work with ROS (Robot Operating System) and the simulation of Baxter robot in the Gazebo simulator.



Baxter simulation in Gazebo

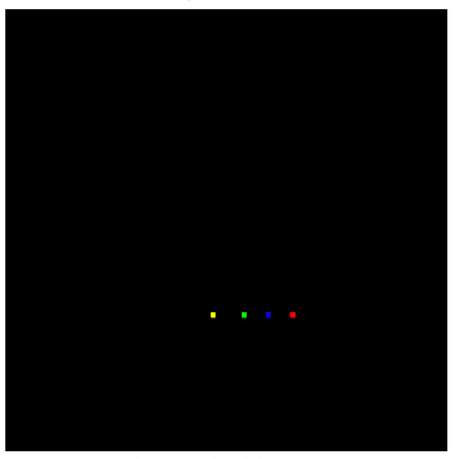
Stage 1: Sense

We use Baxter's head camera to see the world's state



The world as viewed from Baxter's head camera

Then we change the image format in order to implement computer vision tools and create a binary mask of the image.



The masked image

From the binary image we deduct the initial state and publish it via a ROS topic.

```
ubuntu@ubuntu:~$ rosrun think_and_act sense.py red blue green yellow
[INFO] [1598892214.905709, 185.148000]: red blue green yellow
[INFO] [1598892218.283236, 186.148000]: red blue green yellow
[INFO] [1598892221.320160, 187.150000]: red blue green yellow
[INFO] [1598892224.381008, 188.148000]: red blue green yellow
[INFO] [1598892227.729884, 189.151000]: red blue green yellow
```

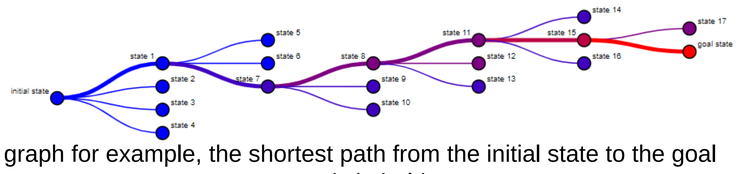
Publishing a initial state

Stage 2: Think

We write and solve a planning problem using PDDL (Planning Domain Definition Language).

We write the planning problem based on the initial state (as deduct in previous stage) and the goal state (received as input from the user). Where the only legal move form one state to another is by moving a block to an empty spot.

The planning problem solution is derived by state space forward search in a graph. In this graph every node is a possible state and every link is a legal move from one state to another.



A graph for example, the shortest path from the initial state to the goal state is in bold

We also get an action plan from the initial state to the goal state

```
(move red_cube position1 position5)
(move yellow_cube position4 position1)
(move red_cube position5 position4)
(move green_cube position3 position5)
(move blue_cube position2 position3)
(move green_cube position5 position2)
```

An action plan for example. From the initial state of red, blue, green, yellow to the goal state of yellow, green, blue, red.

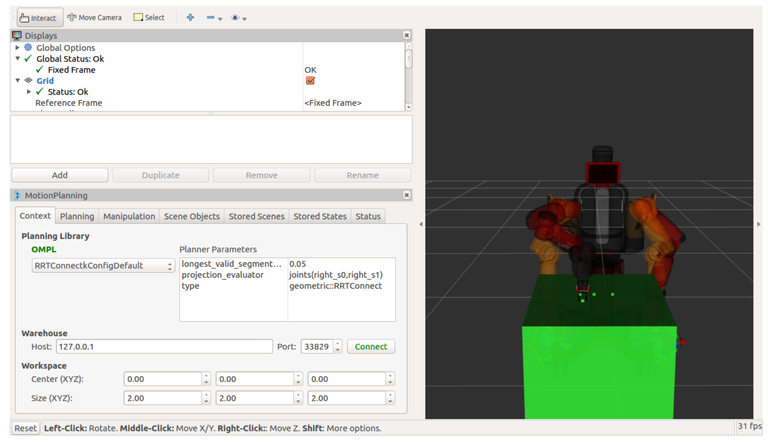
Stage 3: Think

Now we need to devise a movement plan based on the action plan we reached in the previous stage.

We do it with MoveIt! Which is the main software to plan and execute a movement plan in ROS.

The movement plan is a synchronized plan for all of Baxter's joints in order to perfectly bring the end effector (a gripper) to a desired position.

We divide the movement to two parts. The first part is to bring the end effector above the wanted position. The second part is to lower it to the wanted position. We've found that this is a good way that allow Baxter pick and place the blocks in a good robust way.



Baxter while picking a cube as seen in RViz, a visualization tool that work with MoveIt!

Results

We solved the detailed blocks world problem using Sense-Think-Act methodology as described.

For example, Baxter change the initial state of red, blue, green, yellow as shown in image 1 to the goal state of blue, red, green, yellow as shown in image 8. Meaning, change the red and blue blocks position.