BoxRobots

I. Overview

The purpose of this project is to simulate the behaviour of robots with SMA. The robots have to look for box in an area, the claiming area, and then carry this box to another area, the release area. The robot has only four directions available (forward, backward, left and right). The environment where the robot behave is composed of wall which forbid the robot to go through. The two zones are separated by a wall and only some corridor exists to give a way between the two areas. The problem here is that corridor have only a width for one robot. In consequence the robots have to cooperate else they will be blocked.

II. Hypotheses

First, all our robots are the same and knows that the other robot are like them, they also know where they have to go. The robots can have two states, they are carrying a box or not and the other robots have this information. Finally the environment is in constant evolution, in consequence it is not fixed.

III. Nominal Behaviour

The first behaviour implemented is the nominal behaviour, the robots will not cooperate and they will just want to achieve their goal no matter what, in consequence they blocked each other very quickly.

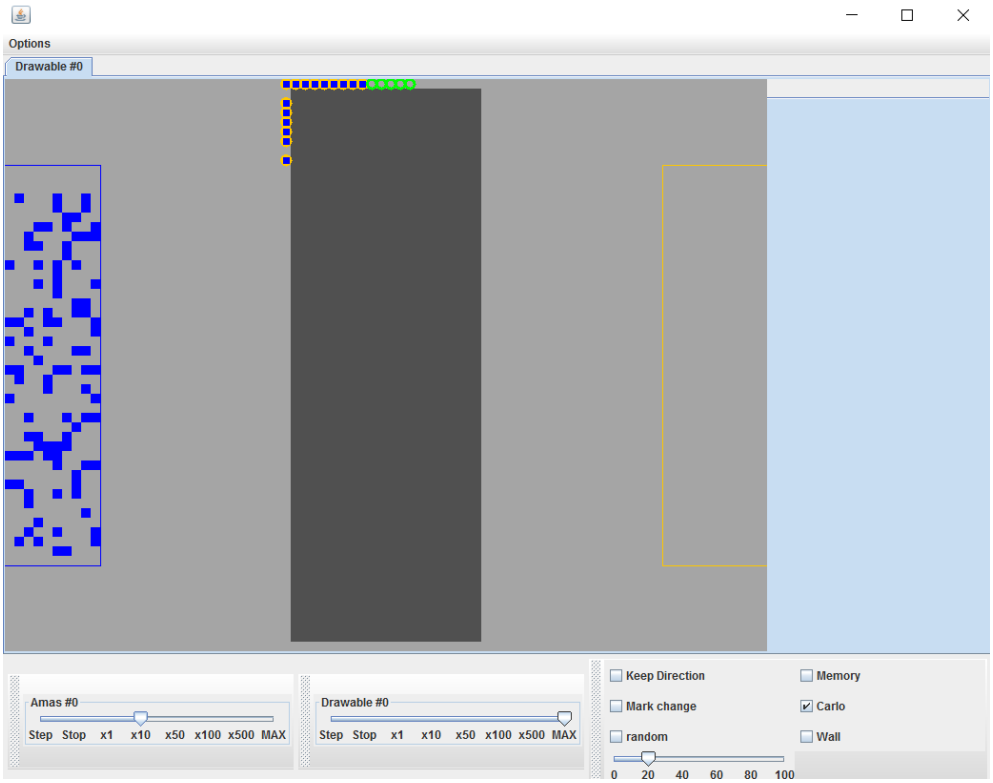

Figure 1: Nominal behaviour

IV. Random behaviour

With the random behaviour, the robot will choose a random direction when an obstacle is met. The random direction is calculated with the Monte Carlo method. The next table shows how the probability are decided.

Direction	Obstacle	Distance to objective zone	Value
Toward/Right/Left	No	Higher	15
Toward/Right/Left	No	Middle	10
Toward/Right/Left	No	Lower	1
Toward/Right/Left	Yes	NA	0
Backward	No	NA	0

Table 1 : Table of random behaviour

We can add to this chart that if all the probability are at 0, the robot will test if backward is free. In this case it will go there.

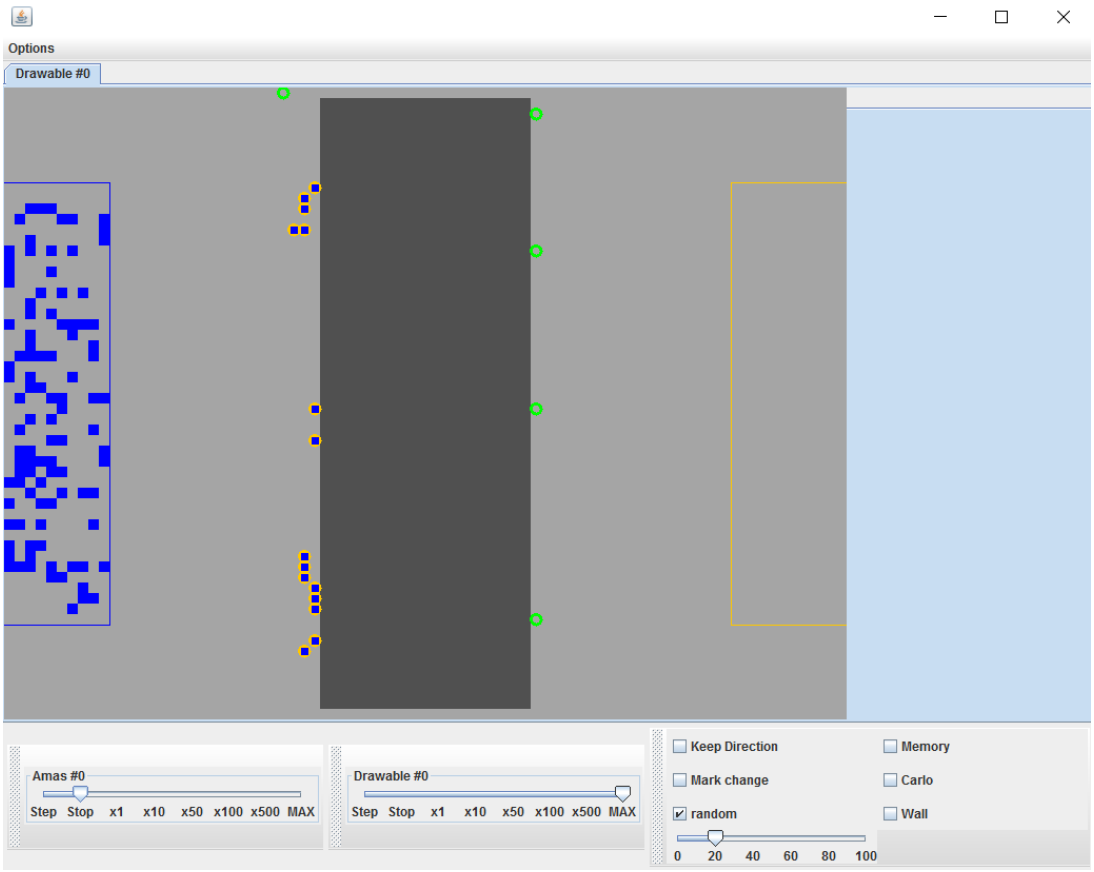

Figure 2: Random behaviour

V. Random behaviour and keep of the last direction

The last behaviour is better because some of the robots achieve their goal but it takes a long time and some of the robots never achieve it. In consequence we have decided that the robot will keep the last direction it use if the toward direction is not available and it will also keep which were its last area. With that addition, the robot will not be inclined to go to the last area and will be inclined to go in the same direction as before.

Direction	Obstacle	Last direction	Distance to objective zone	Value
Toward/Right/Left	No	Yes	Higher	25
Toward/Right/Left	No	No	Middle	10
Toward/Right/Left	No	Not me	NA	0
Toward/Right/Left	Yes	NA	NA	0
Last area	No	NA	NA	0

Table 2 : Table of keep direction

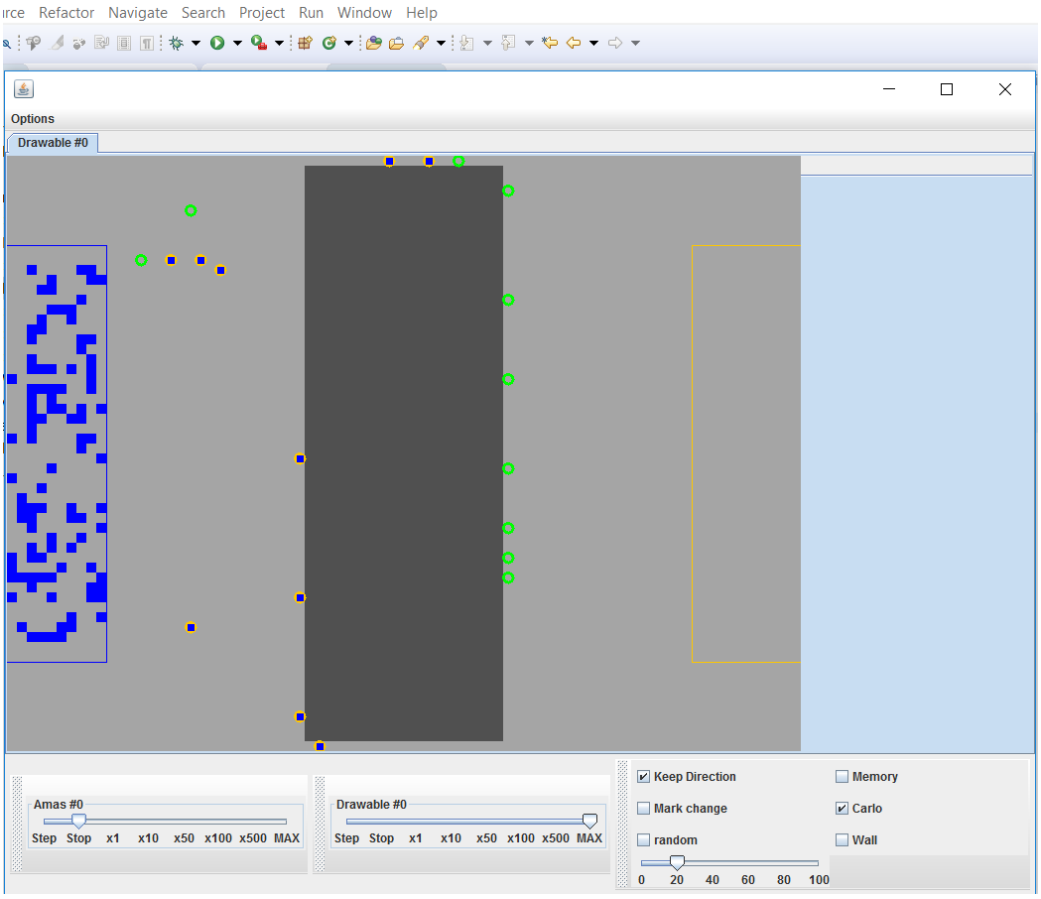

Figure 3 : Keeping of the last direction

VI. Cooperative behaviour

We still have a problem when two robots meet each other in a corridor and they are not in the same state. In this case, we gave a priority order for the robots, the one who is the closer to its objective will continue and the other will return in a zone in an area where two robots can be. In addition of this, if a robot meet another robot in the same state as him but it is currently going back then it will also go back. If the robots is in a deadlock it will also go back as if the obstacle was a robot.

With this behaviour, the robot are no longer blocked in a deadlock and let the other robot going through.

VII. Memory behaviour

Now we want to optimize the behaviour of the robots. In consequence, they have to keep some memory of their previous problem. All the robots have a personal memory for each state of the robot with a limited space. Each time a robot mark an area, this area will be added to the memory, if it is full the older area will be remove. With that they will mark an area in different cases:

- It has to go back.
    - If it is in a deadlock, it will only mark the first area where another possibility than going back.
- It has to change the direction whereas the toward area is not free

- When the robot has the priority against another robot it will mark the area which is at the out of the corridor.

The purpose of marked the area is to influence the robot decision. When the robot sees a marked area within is vision range multiply by 3, the probability for Monte Carlo will change. The modification of the probabilities is done by adding a value to the other three probability instead of reducing the first one.

1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	1	1	1	1	1
1	1	1	1	2	3	2	1	1	1	1
1	1	1	2	3	4	3	2	1	1	1
1	1	2	3	4	5	4	3	2	1	1
1	2	3	4	5	Robot	5	4	3	2	1
1	1	2	3	4	5	4	3	2	1	1
1	1	1	2	3	4	3	2	1	1	1
1	1	1	1	2	3	2	1	1	1	1
1	1	1	1	1	2	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1

Table 3: Modification of probability

Each direction is influenced by the values between two lines. The green case corresponds to the area shared between two directions. The last area will still stay at 0 to prevent the robot to go there. In result the robot will have new probabilities and will use the Monte Carlo method.

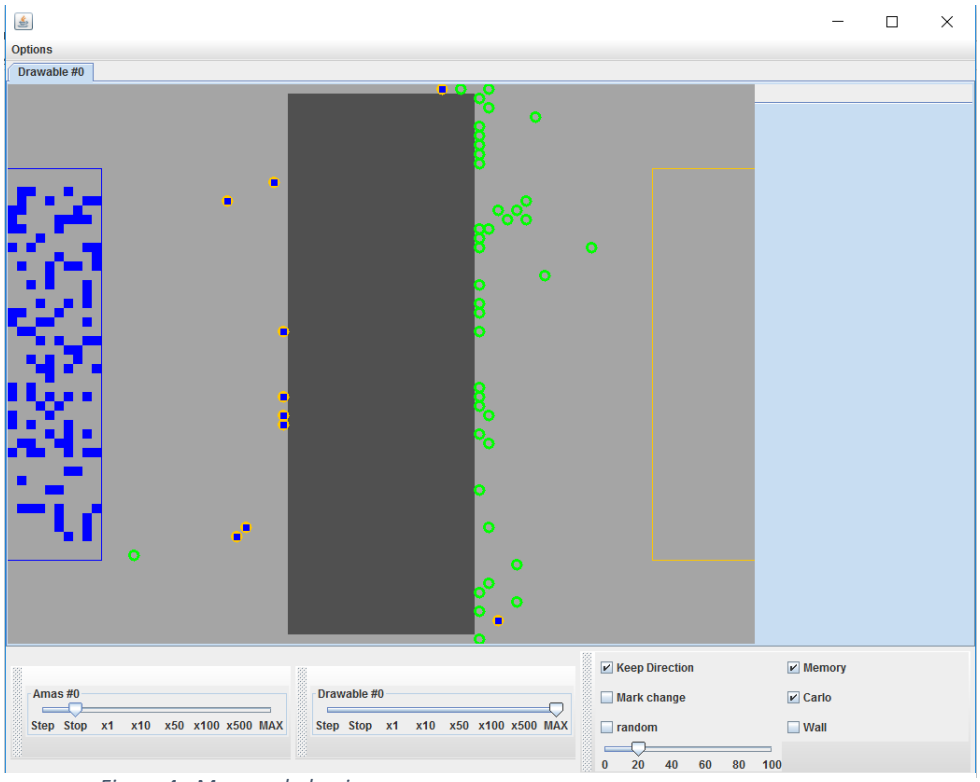

Figure 4 : Memory behaviour

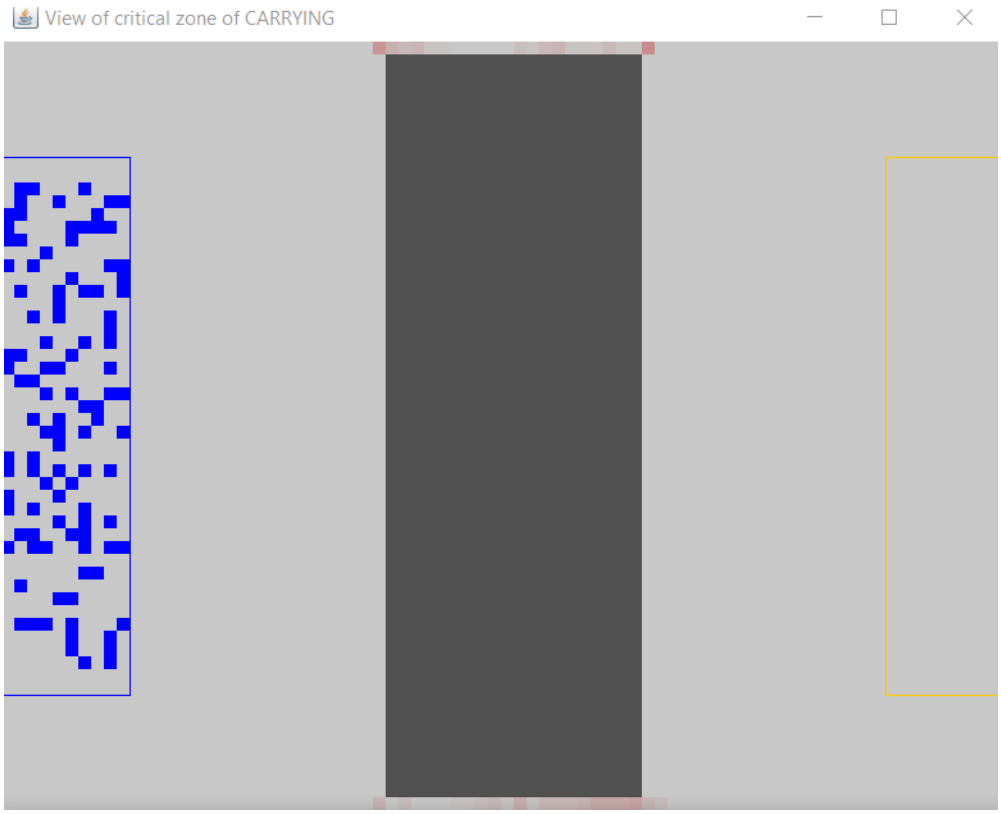

Figure 5: Area in the carrying memory

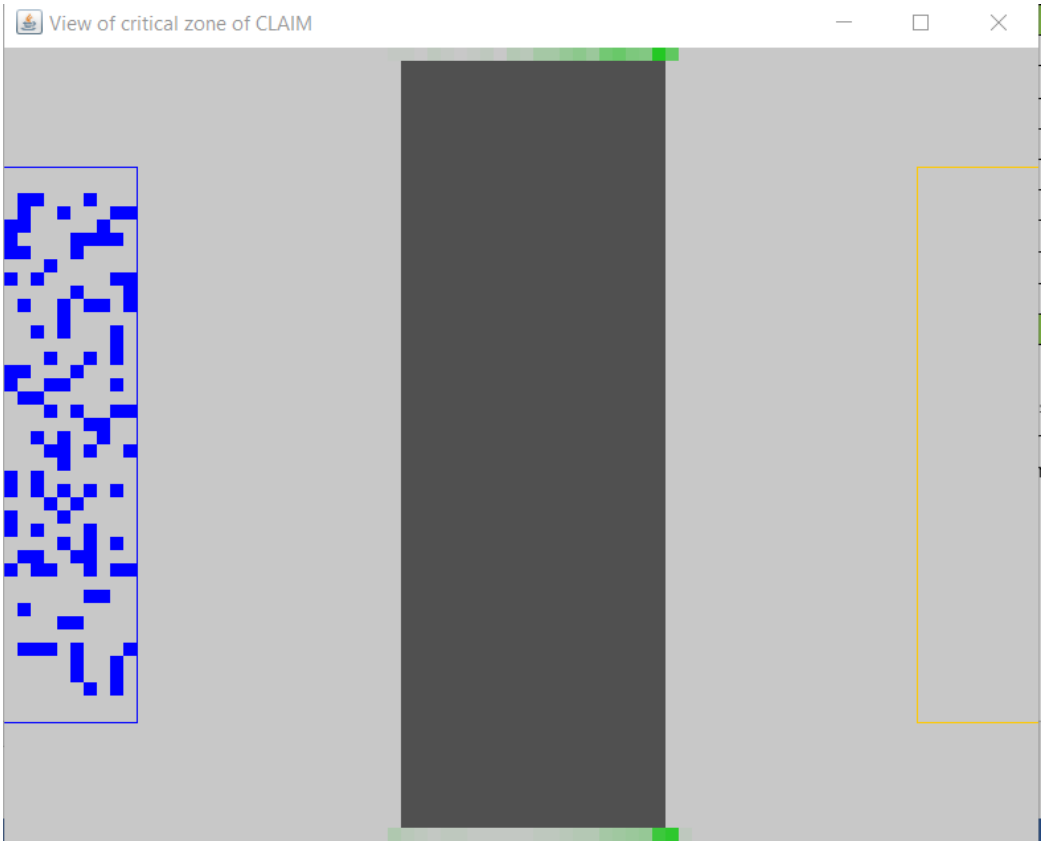

Figure 6 : Area in the release memory