Assignment 1: 2APL short

# 1. Team member information

|  |  |  |
| --- | --- | --- |
| No. | Full Name | ID |
| 1 | Jappie Klooster | 5771803 |
| 2 | Bram Zijlstra | 5774284 |
| 3 | Yunlu Chen | 5684218 |
| 4 | Deliang Wu | 5622182 |

# 2. Implementation

## 2.1 Avoid obstacle

When agents walk in the block world, they may encounter some obstacles, including stones and other agents. When the next block where the agents attend to access has already occupied by such obstacles, then the execution of such plan will fail. So, we use PRRULE to fix such failed plan to help the agents get the ability to avoid obstacles. The detailed method is illustrated in Figure 1.



Figure. Plan repair for avoiding obstacle

## 2.2 Avoid blind commitment

If the destination where the agents want to go is occupied by a stone or surrounded by stones, then the agents can't achieve his commitment. If the agent is blind commitment agent, then he will consist try to achieve such commitment and ignore other tasks. Now, in our current version, we just simply record how many times the agents have tried to achieve his commitment, if the counter is larger than a specific number, then, the agents should think that the goal is impossible to achieve and they will drop such goal.

Of course, there are some smarter ways to avoid blind commitment. For instance, directly detect the location of destination whether has been occupied or surrounded by stones. If yes, directly drop such goal.

## 2.3 supply information to the nearest agent

Clean bombs agents: barry1 and barry2.

Find bombs agent: hally.

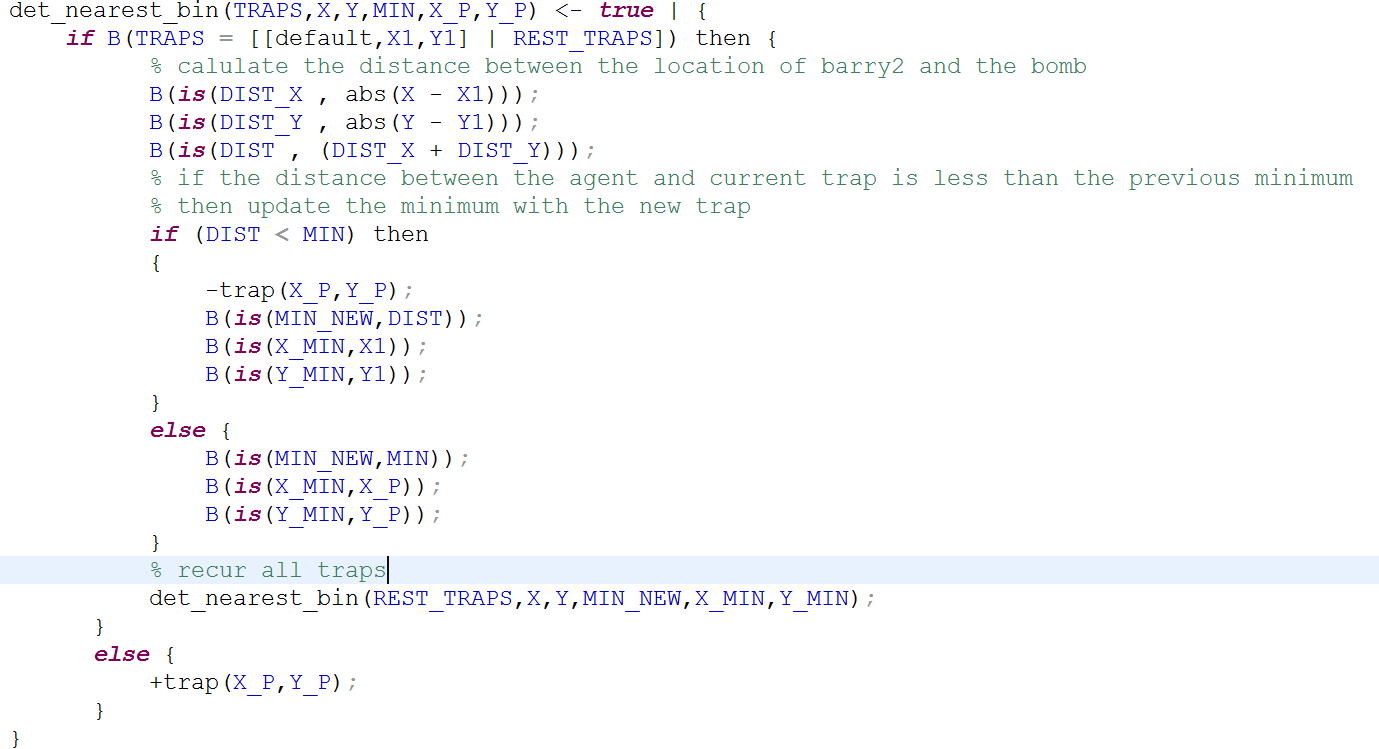
Process for barry1 and barry2: send messages to inform hally whether they are idle or busy, the messages only be sent when the status was changed for barry1 or barry2. For instance, turn into busy from idle or turn into idle from busy.

Process for hally: receive the messages from barry1 and barry2. If she finds that if and only if one clean bombs agent is idle, then send the new bomb location to the idle agent. If both clean bomb agents are idle or busy, then detection the distances between agents and the bomb, and send the bomb location message to the agent nearest to the bomb.

## 2.4 Drop bombs to the nearest trap

Every time, after the agents pick up a bomb, they will re-sense all the traps in the block world, and calculate the distance between his current location and each trap location, and then find out the nearest trap.

we developed a pcrule named *det\_nearest\_trap* to help the agents detect the nearest trap.



## 2.5 Check trap and bombs at every step

In my opinion, I think it is more reasonable for barry1 and barry2 to detect the nearest trap once each time when they pick up a bomb. Because in other time, they don't need to update the trap information.

For helping hally to detect bomb at every step, we just simply move the plan of detection the bombs into the pcrule *goto*(X,Y,M) and name it as *hally\_goto(X,Y,M)*.