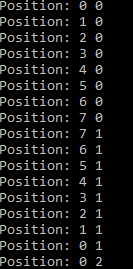
**310CT Intelligent Agents Assignment: Part 2**

The main course of this agent is to traverse the whole area and check for rocks and for water in them. If there is any water found the robot stores the rock and goes back to the base to drop the rock. I started designing the agent by writing pseudo code representing the basic structure of the plans, flow and the actions that where needed in a consequential order. From that structure I began building each plan and making it work with the previous one in order to make the debugging of the whole code easier.

The design of my rover agent is a consequential development of plans that are required to complete the task as follows:

1. Ability to move and traverse the given area completely
2. Check each position for a rock
3. Pick up rock
4. Check the rock for water
5. If not water present, drop the rock
6. If there is a rock >>
7. store it in the robot and >>
8. Return to base

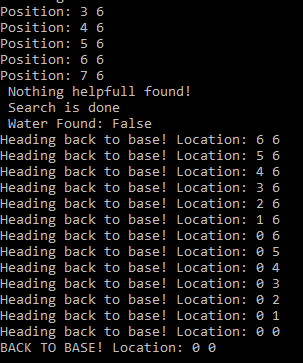
Throughout the development of some of those plans a necessity for additional FACTs, apart from the rocks, appeared. During the plan for traversing the area the robot position was needed. It was initially based on coordinates (0, 0), but for the ease of the robot’s movement it became (-1,0) in order for the first step to cover the first coordinate. When the plan for checking water was build, the other plans needed to have the information it discovered. Therefore, the water finding plan needed a variable to update when it found a rock. Finally, the robot needed to store the rock, so a fact was created for that as well(storage). Furthermore, when I was almost done with the agent I realised that some of the plans where better off when combined. I thought of combining ‘pick up rock’ and ‘check for water” as the plan for picking up the rock pretty much just called the plan for checking for water. Then I realized that that in a real-life scenario these plans will include different complex algorithms that shouldn’t be put together. That’s why some of the plan as non-efficient as they look must be present. In addition, I have some extra features. When a rock is stored it is removed from the facts and the location from where the stone is picked up is printed as inventory.

**Test Results**

The movement of the robot is based on the (x,y)

coordinate system in a matter that:

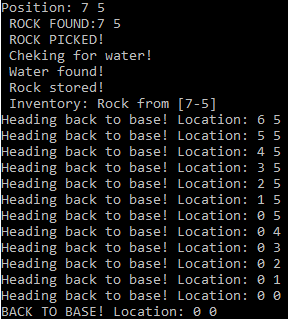
1. It increments ‘x’ until it reaches the end of the grid
2. Increase ‘y’ in order to move to the next roll
3. Decrease ‘x’ until it reaches the end of the grid
4. Increase ‘y’ in order to move to the next roll
5. Repeat until it has traversed the whole area



<<<<In case that the robot doesn’t find

any rocks with water in them

it goes back to the base (0,0).



When the robot finds a rock: >>>>

* It picks it up
* Checks for water
* If water is found >
* Store the rock in inventory
* Go back to base