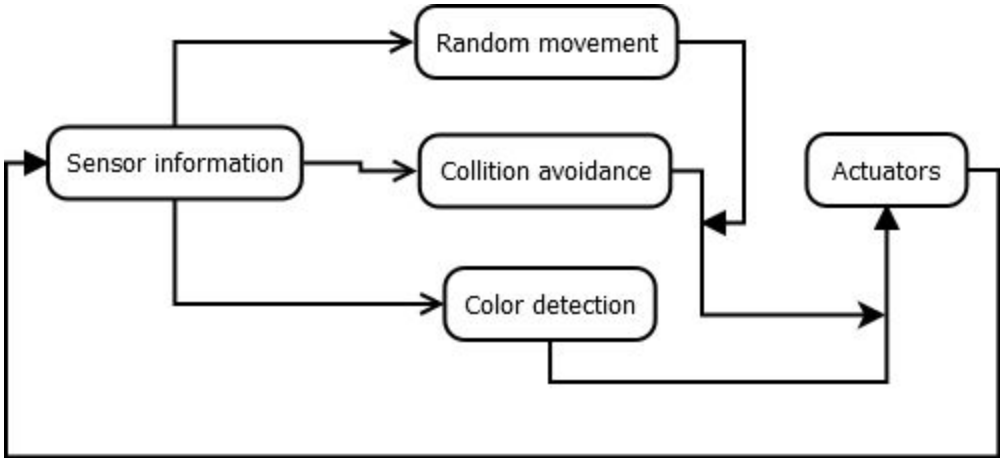
Lab: Programming a Reactive Agent

So for our reactive agent we decided to do a little story-telling. We needed a robot that would be able to enter a mine shaft and find the minerals for us. Once it found a mineral a beep would tell us to take the minerals. The process continued until there were no more minerals. So translated to the behaviors, we had the following. When there is a current (Robot is on), the robot would move on randomly to explore the environment in which it finds itself. The second behavior is that whenever it found an obstacle, it moved back in order to avoid obstacles.The third behavior was that whenever a black object (mineral) was found, the robot would beep.We chose these behaviors because they could be ordered in a reactive agent in a cascade kind of way, subsumption formally. It also helped us to develop an easier to test and show environment that could show the behaviors clearly and how the hierarchy was being followed.



The subsumption style architecture that we used is very straightforward. Obviously it has a hierarchy in how the behaviors would prioritize. Of the top, there was the random movement behavior, composed by 5 movements: go forward, turn both sides with both wheels (each side represents one movement), and turn both sides with just one wheel. This was to happen whenever there was current. After that, it comes the collision avoidance, which would make the robot go backwards if there was an obstacle to avoid or if the bumper sensor detects collision (in case of a failure in the previous sensor); after this, the random movement behavior would take over again. On the bottom, we find the color detection. This would stop the robot every time a black color was found with the sensor. This behavior is able to stop the other two and take over the actions, until the color was removed. Then the robot would continue with the other behaviors.

Reactive agents are very helpful whenever you have a very strict hierarchy of actions to follow in determined environments. While a reactive agent is still considered a somewhat intelligent system, that can adapt to its surroundings. It still lacks of some very desirable aspects, as having states, and being able to compare different actions for the same situation and calculating the best outcome. Of course they can complete complex tasks, they can take on almost any task, given that the instructions given to them can be explicit and very complete. This meaning that the more situations and actions for them that can be programmed into the agent, the better it will work on the task.