**Ali ELABRIDI**

**Assignment 1**

**Agents.**

1. Simple reflex agent: an agent that turns off the light of a certain room if door is closed.

This agent responds only to the state of the door assuming that the room is used only if the main door is constantly opened to welcome people.

1. A robot playing poker in which it needs to remember what were the previous move or bets of the players when X cards were displayed on the table and choose an adequate action for the X+1 newly displayed cards.
2. A robot cleaner that has as a goal to always to keep the house clean either by doing the dishes or cleaning the table or the floor. It has a constant objective or goal which is keeping the dishes, floor, and table clean all time.
3. A good example of utility-based agents is a automated taxi driver that has as a primary goal to get the passenger from point A to B. however, it has multiple constraint such as choosing the safest, shortest and smoothest route to get that person where it has to struggle to combine between all these criteria to find the best possible route to reach the destination.
4. Learning agents: a robot developed to explore a new planet such as march in which it has a set of predefined rules such as always keep exploring, testing, and analyzing and will adjust its activities according and improve them according to the learning gotten from the previous exploration such as avoiding lands where there we found high viscosity, so that the robot will not get stuck in it.

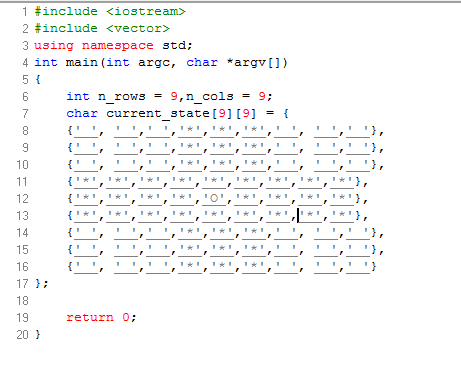
**Environments**

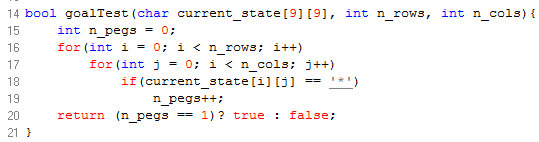
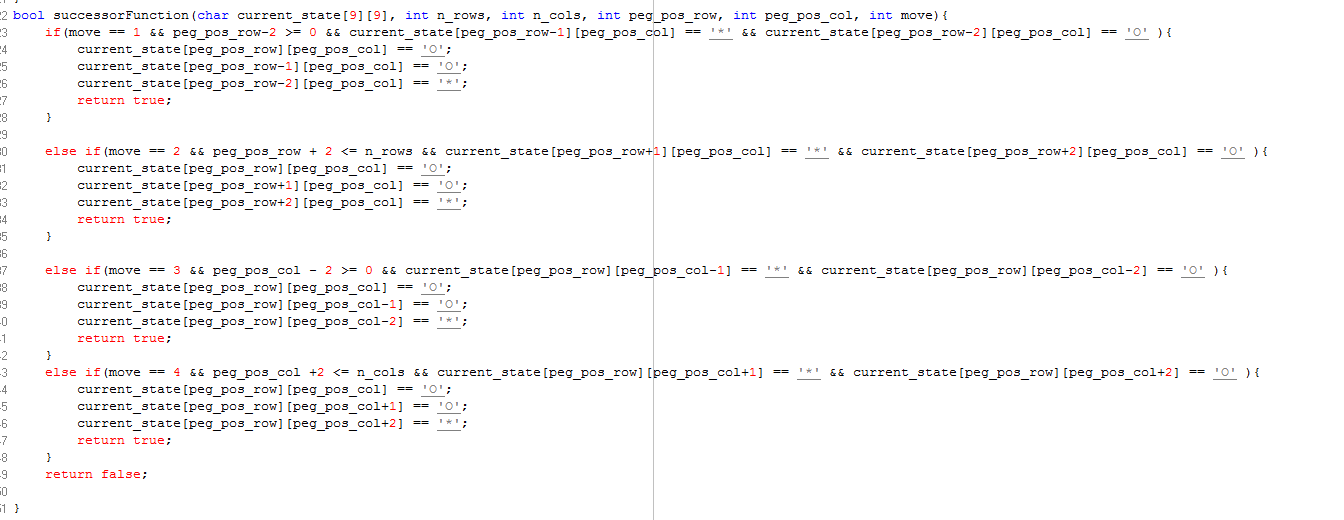
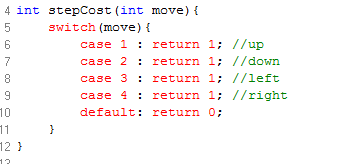
1. Partially observable environment: a satellite that monitors the weather around the world. It is partially observable because of the shape of the planet earth and the orbit of satellite that restrict the monitoring to at most half of the planet.
2. Stochastic environment: a table of black jack in which you don’t know the next card that will be dealt by the dealer, but you have a pretty much good idea of the probabilities of whether to keep going giving your cards on hand or by counting cards.
3. Sequential: a table of poker in which you need to base your bet with previous bets of your opponents to see whether it is worth it or not to continue betting.
4. Discrete: a line of production that have an image processing system for checking the quality of products. It has a finite number of state whether the current product analyzed is ready for sell or bad.
5. Static: the pegs logic game is a good example of a static example. In which there is no change to the board while you are trying to figure out your next move to reach the final goal.
6. Multi-agent cooperative: sending a troop of army robots to war. It is a multi-agent cooperative environments because the robots need to communicate with each other and reach their mutual share goal: winning the war.
7. Unknown environment: sending a robot to a planet in galaxy that is billion years away from ours in which our laws of physics as we know or the type of gravity or physical elements have never been encountered before by humans to try to develop an agent program that would deal with it in a systematic way.

**Agents**

1. The representation of the state of the game would be a 2D array that has N\*M dimension: N being the maximum number of rows and M maximum number of columns for every board variant at the sole exception of the triangular shape that will need to be rotated by 45 degrees to be accommodated by the 2D array. The holes would be represented by a (o) and pigs by a (\*) and the empty slots in the 2D array would be having empty or a space as content.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 |  |  |  | \* | \* | \* |  |  |  |
| 1 |  |  |  | \* | \* | \* |  |  |  |
| 2 |  |  |  | \* | \* | \* |  |  |  |
| 3 | \* | \* | \* | \* | \* | \* | \* | \* | \* |
| 4 | \* | \* | \* | \* | O | \* | \* | \* | \* |
| 5 | \* | \* | \* | \* | \* | \* | \* | \* | \* |
| 6 |  |  |  | \* | \* | \* |  |  |  |
| 7 |  |  |  | \* | \* | \* |  |  |  |
| 8 |  |  |  | \* | \* | \* |  |  |  |

1. The operators of the game are: move up, down, right, and left a peg if and only if there is a hole after the pig adjacent to it.
2. 

* 
* 
* 

4.

1. The branching factor can be very complex to know depending on the current state that we are in. for example, in the initial state we can move four pegs either up, down,left, or right. However, in the middle of the game or for a well sparse current state in which there 2/3 of the pigs present and 1/3 of holes in between each pegs we could do the 4 moves for every one of those 2/3 pigs (give or take… because move may overlap) and therefore for a game of 44 pegs we could have 2/3 of it doing all possible move (again give or take because of the edge) therefore we could have 44\*2/3\*4 branching factors at some points in the game. We could minimize it by applying symmetry on the move. For example, do the same move from the 4 angles of view of the game, and try to work it from there till we reach a state in which there is no that much of branching factor, and then play the moves independently from the 4 angles.
2. One obvious method of search to use in our game is the Depth first search as it provides backtracking to find the combination of move to reach the goal state, or back up if it ends up to be no a good idea to go from a state to another as it produces two sparse pegs than one of them cannot be removed. Moreover, we could apply the local search as we could be chunked the game in 4 or 5 pieces and try to figure out what is the method that provides the maximum number of holes and then combine the results.