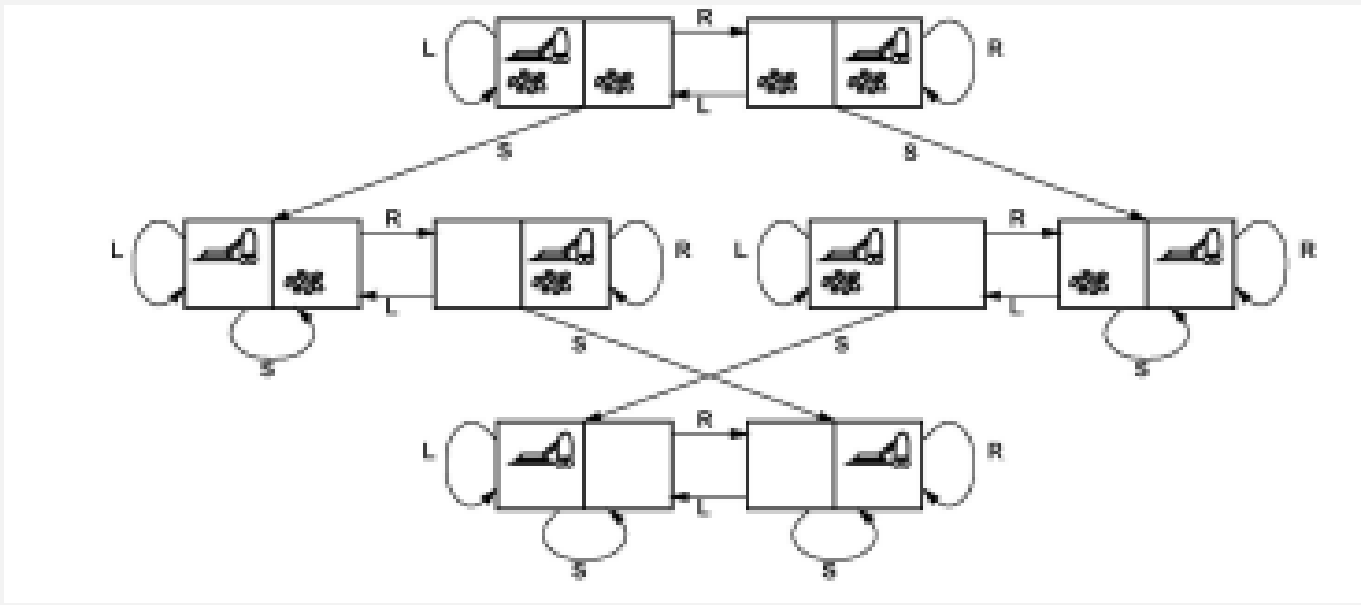
AI - Assignment 1

1.



The bot can perceive if the given block is dirty or clean and will move right or left at random. The bot is restricted from moving outside the environment.

Running the simulations for 1000 timestamps, we get the following:
(The output with the state-space search graph is given in output1.txt since it is too huge to fit in the screenshots)

```
~/Documents/ai_assigment  main !2 ?2  python q1.py
The environment is : [A:Clean, B:Clean]
Start Position:  0
Performance Score:  0

The environment is : [A:Clean, B:Dirty]
Start Position:  0
Performance Score:  1

The environment is : [A:Dirty, B:Clean]
Start Position:  0
Performance Score:  1

The environment is : [A:Dirty, B:Dirty]
Start Position:  0
Performance Score:  2

The environment is : [A:Clean, B:Clean]
Start Position:  1
Performance Score:  0

The environment is : [A:Clean, B:Dirty]
Start Position:  1
Performance Score:  1

The environment is : [A:Dirty, B:Clean]
Start Position:  1
Performance Score:  1

The environment is : [A:Dirty, B:Dirty]
Start Position:  1
Performance Score:  2
```

The outputs are the same no matter how many times you run it which proves that the saturation point is reached.

Different outputs can be obtained when the timesteps are less, say 2 or 3
Q2. Pile Game:

```
~/Documents/ai_assigment  main !2 ?1  python q2.py
Enter the number of stones in pile 1: 3
Enter the number of stones in pile 2: 3
Game State: (3, 3)
Enter the pile you want to remove from: 1
Enter the number of stones you want to remove: 1
Game State: (2, 3)
Agent has removed 1 stones from Pile 2
Game State: (2, 2)
Enter the pile you want to remove from: 1
Enter the number of stones you want to remove: 1
Game State: (1, 2)
Agent has removed 1 stones from Pile 2
Game State: (1, 1)
Enter the pile you want to remove from: 1
Enter the number of stones you want to remove: 1
Game State: (0, 1)
Agent has removed 1 stones from Pile 2
You lost!
```

In this scenario, there are 2 piles of stones and 2 players.

- A player can remove any number of stones from one pile
- The game is turn-based
- A player loses if there are no more stones left on their turn
- The initial number of stones is taken as input (State)
- The state is displayed as (number of stones in pile 1, number of stones in pile 2)

The Agent uses a MiniMax algorithm to decide the best move on each turn.
The end states (leaf nodes) are evaluated as:

1 - If player 1 wins
-1 - If player 1 loses