A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer screen

Description automatically generatedA screenshot of a computer screen

Description automatically generated

* Initially I am setting a random number for row and column. If it’s an obstacle or Start or End then the program would re generate the random numbers for rows and columns.
* I call method pickNextMove, which helps to chose 95% of time the optimal side and 5% times its select random number.
* pickNextMove in turn calls bestMove which helps to select best move out of the four. If the values are equal, then program would select random number.
* Now, program decides where the Robot has to move. 90% of the time it moves to it’s North. 10% of the time it moves to it’s left and 10% of the time to its right.
* After knowing the next move, I calculate Q’(s’,a’) of the next state and calculate reward for North=-1, East or West =-2 or South =-3. After getting those values, Q(s,a) is calculated .
* Trial is incremented only if robot falls in to start or pit.

Still, I am not getting optimal moves on the last column as the Robot don’t go much into those block [(1,6),(2,6),(3,6),(4,6),(5,6)]