CMP9132M-2122 ADVANCED ARTIFICIAL INTELLIGENCE ASSESSMENT-02

The AI-based approach is used to solve the Mean Arena game using Markov Decision Process in the Fully Observable and Partially Observable grid environment game.

SADANAND VITHAL GIRADDI
Student ID - 25895786

Table of Contents

1. ABSTRACT

The code is implemented on an AI-based approach for decision-making in order to control the character called Tallon and the main objective of the character “Tallon” is to survive for a long time and tries to collect bonuses by avoiding the monsters i.e., meanies and the sludge pits in the Mean Arena game. The python script code is provided to control the Tallon character and allows it to navigate the arena when the world is fully observable and also for Partially observable. Also, the code is evaluated for different sizes of the arena. The main outcome of this assignment proves that AI is more suitable for surviving longer time in the games while using the suitable Markov Decision Process methods.

2. INTRODUCTION

In Artificial Intelligence (AI), almost all commercially released computer games nowadays use artificial intelligence (AI) in some way or another. And The current scenarios of the games are controlled by scripts or using finite state machines [1]. The objective of the assignment in which it demonstrates how an AI technique will be implemented in the Mean Arena game with the help of Markov Decision Processes (MDP) and Value iteration (VI) algorithm. In this assignment am going to implement the game by using MDP and VI-based approach methods for decision making to control the character Tallon to survive for a longer time in the grid game by avoiding the meanies. And these can be tested with the help of the fully observable world is one in which we have always access to all available knowledge about the world (Tallon detects everything in the arena). The partially observable is one in which we are not having access to all the information about a world (Tallon detects the objects within a certain distance) [2]. Linking decisions together is at the heart of complex decision-making. This is especially important when one decision leads to another, and each one is influenced by the ones before it and affects the other [2]. The phrases deterministic and non-deterministic are used to describe the likelihood of one's actions having certain results. In a deterministic universe, no matter what action is taken, there is only one result for this kind of reasoning is given with the help of stochastic probability [3].

2.1 MEAN ARENA-2D GRID GAME

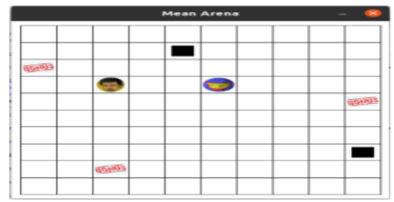


Figure 1: Mean Arena Interface

The Mean arena is a grid game, which was developed by Simon Parson. The premise of game is simple: Tallon (the yellowish icon), a solitary Blue Meanie (evil grin on a blue background), some bonuses, and two pits are depicted in the figure. The game starts with a single Meanie, three bonuses, and two pits in this setup. Every few seconds, a new Meanie arises. Tallon earns points by staying away from the Meanies and claiming bonuses. The Meanies have two distinct personalities. They move at random if they are too far away from Tallon. They move to intercept Tallon as they get closer — the simple algorithm is to reduce the x or y coordinate separating them from Tallon with each movement. This is surprisingly effective, especially when multiple Meanies are around. Tallon's default behavior is to move in the same direction as the bonuses, ignoring the Meanies and pits. This is the behavior that your solution is supposed to improve.

3. CONCEPTS

3.1 Markov Decision Process (MDP)

The Markov Decision Process is a methodology for building policies in a stochastic environment where you know the likelihood that certain outcomes will occur. To solve this assignment MDP method is used in decision making and MDPtoolbox is the library used in the python script. As MDP is the mathematical framework used for modeling the sequential decision-making task problems under some uncertainty [4]. The approach divides an environment into a number of states. They transition from one stage to the next until they reach a terminal state (i.e., a Tallon or an enemy). The control is stochastic and it is the case for our actions and MDP will be the suitable solution to solve this game. The MDP is based on the concepts of state (S), which describes the agent's present status, action (A), which affects the process' dynamics, and reward (R), which is observed for each transition between the states [5]. Non-deterministic actions are described with the help of a set of transition probabilities. And these transition probabilities are associated with a set of the possible transition that occurred between the states with the given action [7]. The main goal of this is to find the optimum policy as it maximizes the expected utility of each state. The utility is the value of each state and this helps in moving between the states with maximum expected utility [6].

3.2 OPTIMAL POLICY and UTILITY

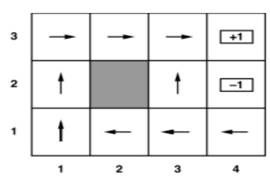


Figure 2: Generating simple Policy with the help of transition model and reward functions.

The solution is the policy in which we write it as π [2].The choice of action for each state is:

$$\pi(s) = a, \; a \in A(s).$$

In any state(s), π (s) it identifies what action should be taken. Since the actions are stochastic the policies won't give the same utility every time they generate. Normally optimum policy π *s is preferred as it generates the highest expected utility for the policies. In MDP, Optimal Policy is the one that maximizes the sum of the expected discount rewards and it is undominated in which it has no state from any policy which achieves the better sum of the expected discount rewards [9].

For any MDP, there is a policy that says π:S :>that is optimal. And policy π is called stationary as it does not change the function as a time and is known as deterministic because the agent always takes the same action no matter what state it is in, s ∈ S.

Utilities are calculated using the average reward, the discounted rewards, and the proper policies too. As proper policies end up in the terminal states by having the finite expected utility [2]. The Utilities calculated for Additive rewards is:

Ur([S0,S1,…,Sn]) = R(S0) + R(S1) +….+ R(Sn), where R(S) is rewards [2].

3.3 MOTION MODEL

The motion model helps in generating the expected utility of each state. In this task, the AI moved in the intended direction 95% of the time and perpendicular to the desired direction 0.5% of the time. Half of the time, go left and half of the time, go right.

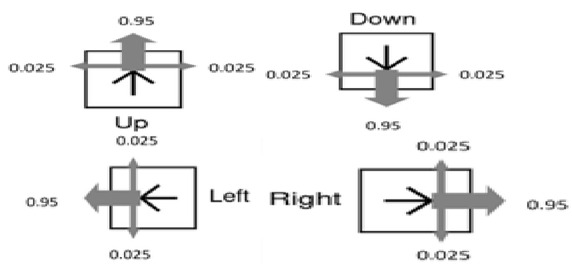


Figure 3: Motion model for the actions.

A transition model may be used to describe each action, and because the actions are stochastic, each can be represented as follows: P(s'|s,a) where a is an action that takes the AI from the present state to the new state(s'). The task has been divided into two steps, one is for Fully Observable and the other is for the Partially observable environment. A policy is a comprehensive map that connects states and activities [7]. A policy is similar to a plan in that it is created ahead of time, but it is not a series of acts like a plan. The policy was only employed when the AI had enough knowledge to produce a policy i.e. The AI has to observe at least one bonus, meanie, and a pit. In a partially observable environment as the Tallon detects the objects within a certain distance, the policy will be inactive for a long proportion of time and forces the AI to discover the bonuses by avoiding the pits and meanies. The above method helps the AI to switch between the two states one allowed the AI to successfully navigate to bonuses while avoiding meanies and pits coming in and out of visibility limit, and the other allowed it to move with a policy that told it the best action for every situation the AI was in.

3.4 REWARDS

The policies are used to direct AI to give its best actions for every state of the grid in the game. The below table demonstrates the reward matrix used in this task and also demonstrates the positive reward and the negative reward they are -1 and +1. For example, we can notice the positive rewards from the below table i.e., in the (8,9) state and also in the (7,2) states and negative reward is in the (2,4) and (4,1) respective states. The non-terminal states hold the reward of -0.04.

The reward matrix shape (State(s), Action(a)), and the reward function R(s), specifies the vectors which are equivalent to the states, and for each state there will be one expected utility. The reward matrices can be generated with the help of Bellman equations for computing the expected utilities for each state in the grid.

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s'|s,a) U(s')$$

Figure 4: Bellman equation. where γ is the discount factor.

STATES

	0	1	2	3	4	5	6	7	8	9
0	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
1	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
2	-0.04	-0.04	-0.04	-0.04	-1	-0.04	-0.04	-0.04	-0.04	-0.04
3	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
4	-0.04	-1	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
5	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
6	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
7	-0.04	-0.04	1	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
8	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	1
9	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04

Table 1: Reward values are taken for this task.

3.5 VALUE-ITERATION ALGORITHM

In determining the optimal policy, the algorithm should be used for learning and behavior [9]. In MDP, the environments should be used. The common algorithms used to determine the optimal policy are Policy iteration and value iteration [1].

$$U(s) \leftarrow R(s) + \gamma \max_{a \in A(S)} \sum_{s'} P(s'|s,a) U_{copy}(s')$$

Figure 5: Value iteration algorithm equation [2].
Where State(S), reward(R(s)), A(s) is the set of actions, and the transition model, P(s'|s,a).

Some steps should be followed like a straightforward plan. For every potential condition a series of actions is needed i.e., the policy. The above algorithm helps to get the policy values of each state with their actions.

4. FULLY OBSERVABLE AND PARTIALLY OBSERVABLE WITH THE EVALUATION RESULTS.

In the fully observable environment, Tallon detects everything in the game arena. And for a Partially observable environment, Tallon cannot detect everything in the environment. It has the capability of seeing the objects within a certain distance. As the Tallon movement is non-deterministic. As a result, they do not always succeed in moving in the desired direction.

From the implemented code, we can see the movement of Tallon surviving as long as possible by avoiding the meanies. Here, the Tallon is evaluated in different sizes of the Arena such as 10*10 grid, 8*8 grid, and 6*6 grid and found the position of the Tallon in each state and actions with the expected utility and their policy with the reward array. The reward array and the probability array are made using the function code by creating and updating the array each time by calling the function and returning the values of the reward array and giving the values of the optimal policy. From the below table, In fully observable i.e., setting the partial visibility to FALSE, It makes the Tallon goes to the bonuses location and collect it. but when the partial visibility is TRUE, it can only sense some cells within a certain distance from the current cell and tries to collect the bonus when it is near. When the bonus is far away from the Tallon it just makes the random move to find it and takes some time, during that meanies might grab the Tallon before reaching the bonus.

When the size of the grid is decreased to 8*8 and 6*6, we can notice that Tallon dies early this is because the movement of Tallon is slightly restricted with the respawning of the meanies. And when the size of the grid goes on increasing the Tallon tends to survive as long as possible by avoiding the meanies and collecting the bonuses. When the meanies get close to the Tallon, it first tries to get away from them before collecting the bonuses.

Partial Visibility = FALSE: FULLY OBSERVABLE

For 10*10 grid

Tallon Position	Moving Direction	Final Score
54	SOUTH	13
61	WEST	5
60	NORTH	21
55	SOUTH	2
69	SOUTH	5
51	EAST	3

For 8*8 grid

Tallon Position	Moving Direction	Final Score
27	EAST	4
70	EAST	4
21	EAST	13
61	WEST	11
37	NORTH	3
20	NORTH	8

For 6*6 grid

Tallon Position	Moving Direction	Final Score
50	EAST	3
50	WEST	5
2	WEST	13
21	EAST	1
23	WEST	2
15	NORTH	1

Partial Visibility = TRUE: PARTIALLY OBSERV.

For 10*10 grid

Tallon Position	Moving Direction	Final Score
86	SOUTH	19
80	SOUTH	13
12	NORTH	16
86	EAST	16
90	SOUTH	19
66	WEST	17

For 8*8 grid

Tallon Position	Moving Direction	Final Score
70	SOUTH	1
61	WEST	1
46	NORTH	14
14	EAST	4
20	NORTH	2
52	EAST	1

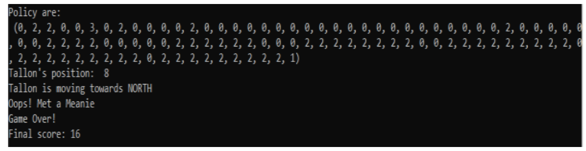
For 6* grid

Tallon Position	Moving Direction	Final Score
86	NORTH	11
11	NORTH	1
10	NORTH	0
15	EAST	2
23	WEST	1
13	NORTH	11

Table 2: Final scores of the Tallon at its positions and the directions for various grid sizes such as 10*10, 8*8, and 6*6 grid map.

RESULT

With the help of the MDP process and Value iteration algorithm used for the Fully observable and partially observable case. The Tallon moves and updates the new Optimum policy for the best possible actions. The Tallon position is determined exactly in the grid by knowing its exact position number and also detecting the exact direction in which the Tallon is moving. Also, the policy values of each action are stated with this, probability of the reward and array is found in which the Tallon moves in North, South, West, and East direction.



```
Policy are:
 (0, 2, 2, 0, 0, 3, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0
, 0, 0, 2, 2, 2, 2, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0
, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1)
Tallon's position:  8
Tallon is moving towards NORTH
Oops! Met a Meanie
Game Over!
Final score: 16
```

Figure 5: Results of the policy values when the Tallon position is at one of the states.

CONCLUSION

An Artificial intelligence method is implemented in order to control the action for the character Tallon. And from this assessment, Tallon tries to survive for a longer time to collect the bonus as much as possible by avoiding the monsters called meanies. When the respawning of meanies becomes more for every time, meanies speed also increases comparatively more than the Tallon movement speed, and later Tallon dies. And from this, I can say that the MDP process is one of the best methods for Fully observable and Partially observable environments when this method is comparatively compared with other methods in game theory such as minimax and maxmin methods.

REFERENCES

[1] Hartley, T., Mehdi, Q. and Gough, N., 2004. Applying Markov Decision Processes to 2D real time games.

[2] Simon Parsons, 2022. Lecture 9: Markov Decision Processes https://blackboard.lincoln.ac.uk/webapps/blackboard/content/listContent.jsp?course_id=_166752_1&content_id=_5852533_1

[3] Florescu, I., 2014. *Probability and stochastic processes*. John Wiley & Sons.

[4] Bonet, B., 2002, June. An e-optimal grid-based algorithm for partially observable Markov decision processes. In *Proc. of the 19th Int. Conf. on Machine Learning (ICML-02)*.

[5] Garcia, F. and Rachelson, E., 2013. Markov decision processes. *Markov Decision Processes in Artificial Intelligence*, pp.1-38.

[6] Pashenkova, E., Rish, I. and Dechter, R., 1996, April. Value iteration and policy iteration algorithms for Markov decision problem. In *AAAI'96: Workshop on Structural Issues in Planning and Temporal Reasoning*.

[7] Russell, S. and Norvig, P., 2002. Artificial intelligence: a modern approach.

[8] Peng, S., 1992. A generalized dynamic programming principle and Hamilton-Jacobi-Bellman equation. *Stochastics: An International Journal of Probability and Stochastic Processes*, *38*(2), pp.119-134.

[9] Littman, M.L., 1994. Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994* (pp. 157-163). Morgan Kaufmann.

APPENDIX

STATES

	0	1	2	3	4	5	6	7	8	9
0	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
1	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
2	-0.04	-0.04	-0.04	-0.04	-1	-0.04	-0.04	-0.04	-0.04	-0.04
3	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
4	-0.04	-1	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
5	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
6	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
7	-0.04	-0.04	1	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04
8	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	1
9	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04	-0.04

Table 1: Reward values are taken for this task.

EVALUATION RESULTS

When PARTIAL VISIBILITY parameter is set to **FALSE:**

For 10*10 grid map

1. Tallon position: 54
   Tallon is moving towards SOUTH
   Final Score = 13

```
Policy are:
 (2, 2, 2, 2, 2, 0, 0, 0, 3, 3, 2, 2, 2, 2, 2, 0, 0, 0, 3, 0, 2, 2, 2, 2, 2, 2, 0, 3, 3, 3, 2, 2, 2, 2, 2, 2, 0, 3, 3,
, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 3, 0, 0, 1, 1, 2, 1, 1, 1, 1, 3, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 3,
, 1, 1, 2, 2, 1, 1, 1, 1, 3, 3, 1, 2, 2, 2, 2, 1, 1, 1, 3, 3)
Tallon's position:  54
Tallon is moving towards SOUTH
Oops! Met a Meanie
Game Over!
Final score: 13
```

2. Tallon position: 61
   Tallon is moving towards WEST
   Final Score = 5

```
Policy are:
 (0, 0, 0, 0, 0, 0, 3, 3, 3, 3, 0, 0, 0, 0, 0, 0, 0, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 3, 3, 0, 0, 0, 3, 0, 0, 0, 0, 3, 3, 3
, 0, 0, 3, 0, 0, 0, 0, 3, 3, 3, 0, 0, 1, 0, 0, 0, 3, 1, 3, 0, 0, 3, 0, 2, 0, 0, 3, 0, 2, 0, 0, 0, 0, 0, 3, 3, 3, 0, 0, 3
, 2, 0, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3)
Tallon's position:  61
Tallon is moving towards WEST
Oops! Met a Meanie
Game Over!
Final score: 5
```

3. Tallon position: 60
Tallon is moving towards NORTH
Final Score = 21

```
Policy are:
 (0, 0, 0, 0, 0, 3, 3, 3, 3, 0, 0, 0, 0, 0, 0, 0, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 0, 3, 0, 0
, 2, 2, 0, 0, 0, 3, 3, 3, 3, 0, 2, 1, 0, 0, 0, 3, 3, 3, 3, 3, 0, 0, 2, 0, 3, 3, 3, 3, 3, 3, 0, 0, 0, 0, 1, 3, 3, 3, 3, 3
, 0, 0, 0, 3, 0, 2, 0, 0, 3, 3, 2, 2, 2, 0, 3, 3, 3, 3, 3, 3)
Tallon's position:  60
Tallon is moving towards NORTH
Oops! Met a Meanie
Oops! Met a Meanie
Oops! Met a Meanie
Game Over!
Final score: 21
```

4. Tallon position: 55
Tallon is moving towards SOUTH
Final Score = 2

```
Policy are:
 (0, 0, 3, 3, 3, 3, 3, 3, 3, 3, 0, 0, 3, 3, 3, 3, 3, 3, 3, 3, 0, 3, 3, 3, 3, 3, 3, 3, 3, 3, 0, 3, 3, 3, 3, 3, 1, 3, 3, 3
, 1, 3, 3, 3, 3, 3, 0, 2, 1, 1, 1, 1, 3, 3, 3, 1, 3, 1, 1, 1, 1, 1, 1, 3, 3, 0, 2, 0, 1, 1, 1, 1, 1, 1, 3, 0, 3, 3, 3, 3
, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3)
Tallon's position:  55
Tallon is moving towards SOUTH
Oops! Met a Meanie
Game Over!
Final score: 2
```

5. Tallon position: 69
Tallon is moving towards SOUTH
Final Score = 5

```
Policy are:
 (0, 2, 2, 2, 0, 0, 0, 3, 3, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 3
, 0, 2, 2, 0, 0, 0, 0, 3, 3, 3, 2, 2, 2, 2, 0, 0, 3, 3, 1, 1, 2, 2, 2, 2, 2, 0, 3, 3, 0, 1, 2, 2, 2, 2, 1, 1, 3, 3, 0, 0
, 1, 2, 2, 1, 1, 1, 1, 3, 3, 3, 2, 2, 1, 1, 3, 0, 2, 1, 3, 3)
Tallon's position:  69
Tallon is moving towards SOUTH
Oops! Met a Meanie
Game Over!
Final score: 5
```

6. Tallon position: 51
Tallon is moving towards EAST
Final Score = 3

```
Policy are:
 (2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 2, 2, 2, 1, 0, 0, 3, 2, 2, 2, 2, 1, 3, 0, 0, 3, 3
, 2, 2, 2, 2, 2, 2, 0, 1, 1, 3, 2, 2, 1, 2, 2, 2, 2, 1, 1, 1, 1, 3, 0, 2, 2, 2, 2, 1, 1, 1, 2, 2, 0, 2, 2, 2, 1, 1, 1, 1
, 2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1)
Tallon's position:  51
Tallon is moving towards EAST
Oops! Met a Meanie
Game Over!
Final score: 3
```

For 8*8 grid map

1. Tallon position: 27
   Tallon is moving towards EAST
   Final Score = 4

```
Policy are:
 (2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 1, 1, 3, 0, 2, 0, 2, 2, 1, 1, 0, 0, 0, 1, 2, 2, 2, 2, 2, 0, 0, 1, 2, 2, 2, 2, 2, 2, 2, 1
, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1)
Tallon's position:  27
Tallon is moving towards EAST
Oops! Met a Meanie
Game Over!
Final score: 4
```

2. Tallon position: 70
   Tallon is moving towards EAST
   Final Score = 4

```
Policy are:
 (2, 2, 2, 2, 2, 0, 0, 0, 0, 2, 2, 2, 0, 0, 0, 0, 0, 2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 2, 2, 2, 0, 0, 2, 2, 2, 2, 2, 2, 0, 0
, 1, 2, 2, 2, 2, 1, 0, 3, 0, 2, 1, 1, 3, 0, 2, 1, 2, 2, 1, 1, 1, 2, 2, 1)
Tallon's position:  70
looking for bonuses
Oops! Met a Meanie
Game Over!
Final score: 4
```

3. Tallon position: 21
   Tallon is moving towards EAST
   Final Score = 13

```
Policy are:
 (0, 0, 2, 2, 2, 2, 0, 0, 0, 0, 1, 0, 0, 2, 0, 0, 0, 3, 0, 2, 0, 2, 0, 0, 2, 2, 0, 2, 2, 2, 0, 3, 2, 2, 2, 2, 2, 2, 1, 1
, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 2, 1, 1, 1)
Tallon's position:  21
Tallon is moving towards EAST
Oops! Met a Meanie
Game Over!
Final score: 13
```

4. Tallon position: 61
   Tallon is moving towards WEST
   Final Score = 11

```
Policy are:
 (0, 0, 3, 3, 3, 3, 3, 3, 2, 0, 3, 3, 3, 3, 3, 3, 1, 1, 1, 3, 3, 3, 3, 3, 1, 1, 1, 1, 3, 3, 3, 1, 1, 1, 1, 1, 1, 3, 1, 1
, 1, 0, 2, 1, 1, 1, 1, 1, 0, 2, 2, 1, 1, 1, 1, 1, 2, 2, 1, 1, 1, 3, 3, 1)
Tallon's position:  61
Tallon is moving towards WEST
Arghhhhh! Fell in a pit
Game Over!
Final score: 11
```

5. Tallon position: 37
   Tallon is moving towards NORTH
   Final Score = 3

```
Policy are:
 (0, 0, 0, 0, 3, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 2, 2, 2, 0, 0, 0, 3, 2, 2, 2, 2, 2, 0, 0, 1
, 2, 2, 2, 2, 2, 2, 0, 0, 2, 2, 2, 2, 2, 1, 1, 0, 2, 2, 2, 2, 2, 1, 1, 3)
Tallon's position:  37
Tallon is moving towards NORTH
Oops! Met a Meanie
Game Over!
Final score: 3
```

6. Tallon position: 20
   Tallon is moving towards NORTH
   Final Score = 8

```
Policy are:
 (0, 2, 0, 0, 0, 3, 3, 3, 0, 1, 0, 0, 0, 0, 3, 3, 0, 0, 2, 0, 0, 3, 1, 1, 0, 0, 0, 3, 3, 3, 0, 0, 0, 3, 3, 3, 3, 3, 0, 0
, 1, 3, 3, 3, 3, 3, 3, 3, 1, 1, 3, 3, 3, 3, 3, 3, 1, 1, 3, 3, 3, 3, 3, 3)
Tallon's position:  20
Tallon is moving towards NORTH
Oops! Met a Meanie
Game Over!
Final score: 8
```

<u>For 6*6 grid map</u>

1. Tallon position: 50
   Looking for the bonuses
   Final Score = 3

```
Policy are:
 (0, 0, 0, 0, 3, 0, 2, 0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 1, 2, 0, 0, 3, 3, 0, 2, 0, 3, 3, 3, 0, 0, 2, 1, 3, 3)
Tallon's position:  50
looking for bonuses
Oops! Met a Meanie
Game Over!
Final score: 3
```

2. Tallon position: 50
   Looking for the bonuses
   Final Score = 5

```
Policy are:
 (2, 0, 0, 0, 0, 3, 2, 0, 0, 0, 0, 1, 2, 2, 0, 0, 3, 0, 1, 2, 2, 2, 2, 0, 0, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1)
Tallon's position:  50
looking for bonuses
Oops! Met a Meanie
Game Over!
Final score: 5
```

3. Tallon position: 2
   Tallon is moving towards WEST
   Final Score = 13

```
Policy are:
 (0, 0, 3, 3, 3, 3, 0, 3, 1, 3, 3, 3, 1, 0, 0, 2, 1, 1, 1, 0, 0, 3, 1, 1, 1, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3)
Tallon's position:  2
Tallon is moving towards WEST
Oops! Met a Meanie
Game Over!
Final score: 13
```

4. Tallon position: 21
   Tallon is moving towards EAST
   Final Score = 1

```
Policy are:
 (2, 2, 2, 0, 0, 0, 2, 2, 1, 2, 0, 0, 1, 3, 0, 2, 0, 3, 0, 3, 0, 2, 1, 1, 2, 2, 0, 2, 1, 1, 2, 2, 2, 1, 1, 1)
Tallon's position:  21
Tallon is moving towards EAST
Oops! Met a Meanie
Arghhhhh! Fell in a pit
Game Over!
Final score: 1
```

5. Tallon position: 23
   Tallon is moving towards WEST
   Final Score = 2

```
Policy are:
 (2, 2, 2, 0, 0, 0, 2, 1, 1, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 3, 2, 2, 2, 2, 1, 1, 2, 2, 2, 1, 1, 1)
Tallon's position:  23
Tallon is moving towards WEST
Oops! Met a Meanie
Game Over!
Final score: 2
```

6. Tallon position: 15
   Tallon is moving towards NORTH
   Final Score = 1

```
Policy are:
 (0, 2, 0, 1, 0, 1, 0, 2, 0, 2, 1, 1, 0, 0, 3, 0, 3, 0, 0, 0, 3, 2, 0, 0, 0, 0, 3, 2, 0, 0, 0, 0, 3, 2, 0, 0)
Tallon's position:  15
Tallon is moving towards NORTH
Oops! Met a Meanie
Game Over!
Final score: 1
```

When PARTIAL VISIBILITY parameter is set to **TRUE:**

For 10*10 grid map

1. Tallon position: 86
   Tallon is moving towards SOUTH
   Final Score = 19

```
Policy are:
 (0, 0, 0, 3, 3, 0, 0, 3, 3, 0, 0, 0, 0, 3, 3, 0, 0, 0, 0, 0, 0, 0, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 2, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 3, 3, 3, 3, 0, 0, 0, 0, 3, 3, 3, 3, 3, 3, 3, 0, 0, 0, 3, 3, 3, 3, 3, 3, 3, 0, 0, 3, 1, 3, 3, 3, 3, 3, 3
, 0, 0, 3, 0, 2, 1, 1, 3, 3, 3, 2, 0, 3, 3, 3, 3, 3, 3, 3, 3)
Tallon's position:  86
Tallon is moving towards SOUTH
Oops! Met a Meanie
Game Over!
Final score: 19
```

2. Tallon position: 80
   Tallon is moving towards SOUTH
   Final Score = 13

```
Policy are:
 (0, 0, 0, 3, 3, 3, 3, 3, 3, 3, 0, 0, 0, 3, 3, 3, 3, 3, 3, 0, 0, 0, 0, 3, 3, 3, 3, 3, 3, 0, 0, 0, 3, 3, 3, 3, 3, 3, 3, 3
, 0, 3, 3, 3, 3, 3, 3, 3, 3, 3, 0, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 1, 0, 2, 1, 1, 3, 3, 3, 3, 1
, 1, 0, 1, 1, 1, 1, 3, 3, 3, 1, 1, 3, 3, 0, 2, 1, 1, 3, 3, 3)
Tallon's position:  80
Tallon is moving towards SOUTH
Oops! Met a Meanie
Game Over!
Final score: 13
```

3. Tallon position: 12
   Tallon is moving towards NORTH
   Final Score = 16

```
Policy are:
 (0, 0, 0, 3, 3, 2, 2, 0, 0, 0, 0, 0, 0, 3, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 3, 3, 3, 3, 0
, 0, 0, 0, 3, 3, 3, 3, 3, 3, 3, 0, 0, 3, 3, 3, 3, 3, 3, 3, 3, 2, 0, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 3, 3, 3, 3, 3, 3, 3, 3
, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 3, 3, 3, 3, 3, 3, 3)
Tallon's position:  12
Tallon is moving towards NORTH
Oops! Met a Meanie
Game Over!
Final score: 16
```

4. Tallon position: 86
   Tallon is moving towards EAST
   Final Score = 16

```
Policy are:
 (0, 0, 0, 3, 3, 3, 3, 3, 3, 3, 0, 0, 0, 0, 3, 3, 3, 3, 3, 0, 0, 0, 0, 0, 0, 3, 3, 3, 0, 0, 0, 0, 0, 0, 0, 3, 3, 3, 3, 0
, 0, 0, 0, 0, 3, 3, 3, 3, 3, 3, 0, 0, 0, 3, 3, 3, 3, 3, 3, 3, 0, 0, 3, 3, 3, 3, 3, 3, 1, 1, 2, 0, 3, 3, 3, 1, 3, 3, 0, 1
, 1, 1, 3, 3, 3, 0, 2, 1, 0, 0, 1, 1, 1, 3, 3, 3, 1, 1, 3, 3)
Tallon's position:  86
Tallon is moving towards EAST
free variable 'Nearest_to_Bonus' referenced before assignment in enclosing scope
looking for bonuses
Oops! Met a Meanie
Game Over!
Final score: 16
```

5. Tallon position: 90
   Tallon is moving towards SOUTH
   Final Score = 19

```
Policy are:
 (0, 2, 2, 0, 0, 0, 0, 3, 3, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 2, 2, 1, 2, 0, 0, 0, 3, 3, 0
, 1, 3, 0, 2, 0, 0, 3, 3, 3, 3, 1, 3, 3, 0, 0, 3, 3, 3, 3, 3)
Tallon's position:  90
Tallon is moving towards SOUTH
Oops! Met a Meanie
Game Over!
Final score: 19
```

6. Tallon position: 66
   Tallon is moving towards WEST
   Final Score = 17

```
Policy are:
 (2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0
, 2, 2, 0, 0, 0, 0, 0, 2, 0, 0, 1, 2, 2, 0, 0, 0, 0, 1, 0, 0, 0, 2, 2, 2, 0, 0, 3, 0, 2, 0, 0, 2, 2, 2, 2, 2, 2, 0, 3, 3
, 2, 2, 2, 2, 2, 2, 2, 1, 3, 3, 2, 2, 2, 2, 2, 2, 1, 1, 1, 3)
Tallon's position:  66
Tallon is moving towards WEST
Oops! Met a Meanie
Game Over!
Final score: 17
```

For 8*8 grid map

1. Tallon position: 70
   Tallon is moving towards SOUTH
   Final Score = 1

```
Policy are:
 (0, 0, 0, 3, 3, 3, 3, 3, 0, 0, 0, 3, 3, 3, 3, 0, 0, 0, 0, 3, 3, 3, 3, 3, 0, 0, 3, 3, 3, 3, 3, 3, 0, 3, 3, 1, 3, 3, 3, 3
, 0, 3, 3, 0, 2, 1, 1, 1, 1, 1, 3, 3, 3, 1, 1, 1, 1, 0, 1, 3, 3, 1, 1, 1)
Tallon's position:  70
looking for bonuses
Oops! Met a Meanie
Game Over!
Final score: 1
```

2. Tallon position: 61
   Tallon is moving towards WEST
   Final Score = 1

```
Policy are:
 (2, 2, 0, 0, 0, 3, 3, 3, 2, 2, 2, 0, 3, 3, 3, 3, 2, 2, 2, 0, 3, 3, 3, 3, 2, 2, 1, 1, 1, 3, 3, 3, 1, 1, 1, 1, 1, 1, 3,
, 1, 1, 1, 1, 1, 1, 1, 1, 0, 2, 1, 1, 3, 0, 2, 1, 2, 2, 1, 1, 3, 3, 1, 1)
Tallon's position:  61
Tallon is moving towards WEST
Arghhhhh! Fell in a pit
Game Over!
Final score: 1
```

3. Tallon position: 46
   Tallon is moving towards NORTH
   Final Score = 14

```
Policy are:
 (0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 3, 0, 2, 0, 0, 2, 2, 2, 0, 0, 0, 3, 3, 2, 2, 2, 2, 0, 3, 1, 1
, 2, 2, 2, 1, 1, 3, 0, 1, 1, 2, 2, 1, 1, 3, 0, 0, 2, 2, 2, 1, 1, 1, 3, 3)
Tallon's position:  46
Tallon is moving towards NORTH
Oops! Met a Meanie
Game Over!
Final score: 14
```

4. Tallon position: 14
   Tallon is moving towards EAST
   Final Score = 4

```
Policy are:
 (0, 0, 0, 0, 0, 2, 2, 0, 0, 0, 0, 0, 3, 0, 2, 0, 0, 2, 0, 0, 3, 0, 2, 0, 2, 2, 2, 2, 0, 0, 2, 0, 2, 2, 2, 2, 2, 2, 2, 0
, 2, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 2, 1, 1)
Tallon's position:  14
Tallon is moving towards EAST
Oops! Met a Meanie
Game Over!
Final score: 4
```

5. Tallon position: 20
   Tallon is moving towards NORTH
   Final Score = 2

```
Policy are:
 (0, 0, 3, 3, 3, 3, 0, 0, 2, 0, 3, 3, 0, 2, 0, 0, 1, 1, 1, 3, 0, 3, 3, 3, 1, 0, 2, 1, 3, 3, 3, 3, 1, 0, 1, 1, 1, 3, 3,
, 2, 2, 1, 1, 1, 3, 3, 1, 2, 2, 1, 1, 1, 1, 3, 1, 2, 2, 1, 1, 1, 3, 3, 3)
Tallon's position:  20
Tallon is moving towards NORTH
Oops! Met a Meanie
Game Over!
Final score: 2
```

6. Tallon position: 52
   Tallon is moving towards EAST
   Final Score = 1

```
Policy are:
 (0, 0, 0, 3, 3, 3, 3, 3, 0, 0, 0, 3, 3, 3, 3, 3, 0, 0, 3, 1, 3, 3, 3, 3, 0, 0, 3, 0, 2, 1, 1, 3, 0, 0, 3, 0, 3, 3, 3, 3
, 0, 0, 3, 3, 3, 3, 3, 3, 2, 0, 3, 0, 2, 0, 3, 3, 1, 1, 3, 3, 3, 3, 3, 3)
Tallon's position:  52
Tallon is moving towards EAST
Oops! Met a Meanie
Game Over!
Final score: 1
```

<u>For 6*6 grid map</u>

1. Tallon position: 15
   Tallon is moving towards NORTH
   Final Score = 11

```
Policy are:
 (0, 0, 0, 0, 3, 0, 0, 0, 0, 0, 3, 3, 0, 2, 0, 0, 3, 0, 2, 2, 2, 2, 0, 0, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 1)
Tallon's position:  15
Tallon is moving towards NORTH
Oops! Met a Meanie
Game Over!
Final score: 11
```

2. Tallon position: 11
   Tallon is moving towards NORTH
   Final Score = 1

```
Policy are:
 (2, 0, 0, 3, 0, 0, 2, 2, 2, 2, 0, 0, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 1, 1, 1, 2, 2, 1, 1, 1, 0, 2, 2, 1, 1)
Tallon's position:  11
Tallon is moving towards NORTH
Arghhhhh! Fell in a pit
Game Over!
Final score: 1
```

3. Tallon position: 10
   Tallon is moving towards NORTH
   Final Score = 0

```
Policy are:
 (2, 2, 2, 0, 0, 0, 2, 2, 2, 2, 0, 0, 0, 2, 2, 2, 0, 3, 2, 2, 2, 2, 1, 1, 0, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1, 1)
Tallon's position:  10
Tallon is moving towards NORTH
Arghhhhh! Fell in a pit
Game Over!
Final score: 0
```

4. Tallon position: 15
   Tallon is moving towards EAST
   Final Score = 2

```
Policy are:
 (2, 2, 0, 0, 3, 0, 2, 2, 2, 0, 0, 3, 2, 2, 2, 2, 0, 0, 2, 2, 2, 1, 1, 0, 1, 2, 1, 1, 1, 3, 2, 2, 2, 1, 1, 3)
Tallon's position:  15
Tallon is moving towards EAST
Oops! Met a Meanie
Game Over!
Final score: 2
```

5. Tallon position: 23
Tallon is moving towards WEST
Final Score = 1

```
Policy are:
 (2, 2, 0, 0, 0, 0, 2, 1, 0, 0, 0, 0, 1, 0, 2, 0, 3, 3, 0, 2, 0, 3, 3, 3, 0, 2, 1, 1, 3, 3, 2, 2, 1, 1, 3, 3)
Tallon's position:  23
Tallon is moving towards WEST
Oops! Met a Meanie
Game Over!
Final score: 1
```

6. Tallon position: 13
Tallon is moving towards NORTH
Final Score = 11

```
Policy are:
 (2, 0, 0, 3, 3, 3, 0, 2, 0, 1, 3, 0, 0, 0, 3, 0, 2, 0, 0, 3, 3, 0, 3, 3, 1, 3, 3, 3, 3, 3, 1, 3, 3, 3, 3, 3)
Tallon's position:  13
Tallon is moving towards NORTH
Oops! Met a Meanie
Game Over!
Final score: 11
```

```
Reward array:
 [[-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-1.   -1.   -1.   -1.  ]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-1.   -1.   -1.   -1.  ]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [ 1.    1.    1.    1.  ]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]
 [-0.04 -0.04 -0.04 -0.04]]
```

FIGURE : REWARD ARRAY OF ONE ITERATION FOR 6*6 GRID MAP

Partial Visibility = FALSE: FULLY OBSERVABLE

For 10*10 grid

Tallon Position	Moving Direction	Final Score
54	SOUTH	13
61	WEST	5
60	NORTH	21
55	SOUTH	2
69	SOUTH	5
51	EAST	3

For 8*8 grid

Tallon Position	Moving Direction	Final Score
27	EAST	4
70	EAST	4
21	EAST	13
61	WEST	11
37	NORTH	3
20	NORTH	8

For 6*6 grid

Tallon Position	Moving Direction	Final Score
50	EAST	3
50	WEST	5
2	WEST	13
21	EAST	1
23	WEST	2
15	NORTH	1

Partial Visibility = TRUE: PARTIALLY OBSERVABLE

For 10*10 grid

Tallon Position	Moving Direction	Final Score
86	SOUTH	19
80	SOUTH	13
12	NORTH	16
86	EAST	16
90	SOUTH	19
66	WEST	17

For 8*8 grid

Tallon Position	Moving Direction	Final Score
70	SOUTH	1
61	WEST	1
46	NORTH	14
14	EAST	4
20	NORTH	2
52	EAST	1

For 6* grid

Tallon Position	Moving Direction	Final Score
86	NORTH	11
11	NORTH	1
10	NORTH	0
15	EAST	2
23	WEST	1
13	NORTH	11