CMP9132M Advanced Artificial Intelligence Assessment Item 2

Student Number: 25766099

Name: Xumin Gao

1. Problem statement and solution

a) Fully observable situation

According to the problem statements, when the dungeon is fully observable and Tallon's movement is non-deterministic, it is a Markov decision process (MDPs) problem (Poole, 2010). We need to calculate the optimum policy π^* with the highest expected utility to make a choice of Tallon's action for every state. And we choose Bellman equation (Bellman, 1957) to solve it. The Bellman equation as are shown in Equation (1):

$$U(s) = R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U(s') \qquad (1)$$

We can use the Bellman equation to calculate the utility of each state. However, it will be a huge computation. So, we use value iteration (Poole, 2010) to solve this problem, which starts with arbitrary values for states, then applies the Bellman update to all the states until the values of states converge. The value iteration are shown in Equation (2):

$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a) U_i(s') \qquad (2)$$

Based on the theoretical knowledge above, the MDPs model is constructed through using the python library mdptoolbox which we used on our workshops. There are two key parts of the MDPs. One is that using the motion model of Tallon to build the transition model. Another is to set the reward (also namely cost) for every states. In this task, setting the reward of blank grids as -0.04, the reward of Bonuses as $+10$, the reward of Meanies and Pits as -10. The motion model and rewards distribution are shown in Fig. 1 and Fig. 2:

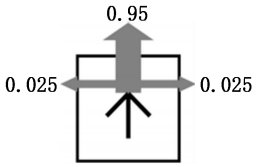

Fig. 1 The motion model of Tallon

Tallon	−0. 04	Meanie −10
Bonuse +10	−0. 04	−0. 04
−0. 04	Pit −10	−0. 04

Fig. 2 The rewards distribution

b) Partially observable situation

When the dungeon is partially observable, Tallon can only see relevant objects within a certain distance. We use partially observable MDPs (POMDPs) (Poole, 2010) to solve it. The basic solution process is as below:

1) We construct the MDPs model based on the work of **a)**.

2) Using Tallon's perception (namely sensor model) combined with Meanie's transition model to calculate Meanie's belief states through Equation (3). Then obtaining the largest belief state and

regarding the largest belief state as the current actual state of Meanies. At the same time, the next predicted state of Meanies can be obtained. This process is actually that filters and predicts the state of Meanies, which can track the movement of Meanies.

$$b'(s') = \alpha P(e|s') \sum_{s} P(s' \mid s,a) b(s) \qquad (3)$$

Here, it is a simple explanation of belief states for Meanies in Fig. 3:

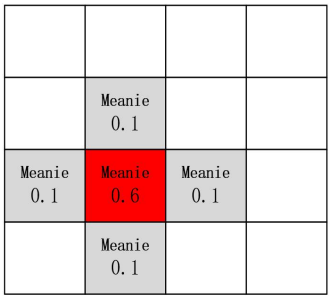

Fig. 3 The belief states for Meanie

Because of noise in the sensor model, when Tallon percepts a Meanie, it only has a belief about where the Meanie are in. So, using probability distribution over Meanie's possible states, As Fig. 3, the probability that the Meanie are in the red grid is 0.6, while it may be in each gray grid with 0.1.

3) According to the current filtered state and the next predicted state of Meanies, setting the corresponding reward value of MDPs model of step 1) to -10 and -5 respectively.

4) Running MDPs model of step 1) , getting the optimum policy and making a action for Tallon. If no any objects is found within Tallon's partially observable range, Tallon moves randomly.

5) Looping steps 2) - 4). And keep updating the belief states of Meanies.

2. Additional strategies

In order to increase Tallon's probability of winning and get more scores, I add some additional strategies in the reward distribution of MDPs model as below:

Meanie −10	−5.04	−2.54	−1.71	−0.04
−5.04	−3.58	−2.28	−0.04	Tallon
−2.54	−2.28	−0.04	Bonuse +10	−0.04
−1.71	−0.04	−0.04	−0.04	−0.04
−0.04	Pit −10	−2.04	Pit −10	Bonuse +6

Fig. 4 Additional strategies for reward distribution

1) In order to make Tallon avoid Meanies in advance, calculating the distance D between blank grids and Meanies by using Euclidean distance formula. When $D <= 3$, increasing the reward of

corresponding blank grids through the principle $\text{cost} + ((-1) / D) * 5$. The closer the blank grids are to Meanies, the more cost will be allocated. For this strategy, it can seen as the gray grids in Fig. 4.

2) Some blank grids are close to multiple Meanies or Pits at the same time. These areas are very dangerous for Tallon. According to the number of Meanies or Pits close to blank grids, different costs are allocated to blank grids through the principle $\max(\text{cost} + (-n), -8)$ (n represents the number of Meanies or Pits close to a blank grid). When a blank grid closes to more Meanies or Pits, The more costs are allocated. For this strategy, it can seen as the blank grid with cost -2.04 in Fig. 4.

3) Some Bonuses are close to Pits or Boundaries, some Bonuses are not. If Tallon collects Bonuses with close to Pits or Boundaries firstly, it is easy to get stuck and die. In this case, it would be wise to collect Bonuses with not close to pits or Boundaries firstly, and then collect other Bonuses. So, using the principle $4 + (4 / n)$ to allocate reward for every Bonuses, n represents the number of Pits or Boundaries close to a Bonuse. If a Bonuse close to more Pits or Boundaries, it will be allocated less reward. For this strategy, it can seen as the green grids in Fig. 4.

3. Evaluation and conclusions

In this part, I set the program of game.py to run 50 times every time. I use the winning rate (When Tallon collected all of Bonuses in each task, it was counted as a winning) and average score (Tallon got an average score for many tasks) to evaluate the performance of MDPs and POMDPs. Evaluable variables includes different methods (MDPs, POMDPs), size of the grid, different numbers of bonuses or pits, and how quickly the Meanies spawn. The evaluation results are shown in Table 1 - Table 4 (**see Appendix A**). Please note that when one of variables changes, other variables remain unchanged which are consistent with default config.py file.

According to the evaluation results, we can draw following conclusions:

- As can be seen from Table 1, when the size of the grid increases, there are more blank grids for Tallon to avoid Meanies or Pits, and Meanies will not approach Tallon in a short time. So Tallon can collect Bonuses more easily, the winning rate and average score for MDPs are also increasing. On the contrary, the winning rate of POMDPs decreases and the average score increase slightly. This is because, Bonuses become more scattered, while Tallon's perception range is limited. In this case, Tallon is not easy to find Bonuses and then be killed by the Meanies in the process of looking for Bonuses.

- As can be seen from Table 2, when the number of Bonuses increases, the difficulty of completing the task increases, so the winning rate decreases, although the average score increases.

- As can be seen from Table 3, when the number of Pits increases, normally, this prevents Tallon from collecting bonuses, so the winning rate and the average score for MDPs will decreases. However, it is different for POMDPs. When increasing the appropriate amount of Pits, Tallon can see more objects in the map, which is more conducive to collecting Bonuses. At this time, the winning rate and the average score increase, but when the number of Pits is too large, Tallon is easy to fall into pits, and the winning rate and the average score decrease.

- As can be seen from Table 4, when Meanies spawn faster, more Meanies will be generated in the map, which is extremely unfavorable to Tallon, so the winning rate and the average score decrease.

References

Bellman, R., 1957. *A Markovian decision process*. Journal of mathematics and mechanics, pp.679-684.

Poole, D.L. and Mackworth, A.K., 2010. *Artificial Intelligence: foundations of computational agents*. Cambridge University Press.

Appendix A

Table 1. Evaluation result in different sizes of the grid using MDPs and POMDPs

Method	Sizes of the grid	Winning rate	Average score
MDPs	10x10	0.56	19
MDPs	15x15	0.66	27
MDPs	20x20	0.70	30
POMDPs	10x10	0.34	18
POMDPs	15x15	0.20	21
POMDPs	20x20	0.12	20

Table 2. Evaluation result in different numbers of bonuses using MDPs and POMDPs

Method	numbers of bonuses	Winning rate	Average score
MDPs	1	0.78	13
MDPs	5	0.08	32
MDPs	10	0.02	50
POMDPs	1	0.58	13
POMDPs	5	0.06	32
POMDPs	10	0	49

Table 3. Evaluation result in different numbers of Pits using MDPs and POMDPs

Method	numbers of Pits	Winning rate	Average score
MDPs	1	0.62	22
MDPs	5	0.54	20
MDPs	10	0.48	18
POMDPs	1	0.36	19
POMDPs	5	0.56	21
POMDPs	10	0.30	18

Table 4. Evaluation result in how often adding a Meanie using MDPs and POMDPs

Method	How often adding a Meanie	Winning rate	Average score
MDPs	1	0.14	12
MDPs	5	0.56	19
MDPs	10	0.68	24
POMDPs	1	0.08	10
POMDPs	5	0.34	18
POMDPs	10	0.38	22