Report of Assignment Two

CSI4106 – Fall 2017

Group members:

Qiguang Chu

#300042722

Liyuan Cao

#300042723

Date: Nov, 21, 2017

Introduction

Wumpus World is represented as a solving game which has 4*4 rooms. There are some pits, a gold and a Wumpus in the rooms of the world. Besides, the rooms around wumpus have stench and rooms around pits have breeze. The room which has gold is glitter. Wumpus, pits and gold conflicts each others. The start room, which is set as [1, 1], has no threats. The appearance of pits has probability of 0.2. The goal of player is to get the gold without entering pits and Wumpus and return to the start position. Player has an arrow which can be used to kill Wumpus. Payoff is the score of game. The rules are: every action costs 1; entering pits and Wumpus costs 1000, which means game over; using arrow costs 100; getting gold + 1000. When the Wumpus is killed, Wumpus and stenches will be removed and generating scream all over the rooms.

The first thing to consider is the compatibility of pits, Wumpus and gold, which means if these three elements can be set in one square. After consulting it with professor, these three things are informed not to be set in one square, because of the conflicts of adding and minus of payoff. The second issue is that if pits and Wumpus can be set with breeze or stench. Actually, in order to keep the consistent rules in every square, breeze and stench can be set with Wumpus and pits. The last thing to be consider is that the definition of death. For not be mentioned in the assignment guide, we formulate a definition of death. That is, once payoff is less than -1000, we consider that the person is dead of falling into pits or eaten by Wumpus.

Establishing Wumpus World

(a) The rooms are 4*4 states.

(b) Pick a room for Wumpus.

(c) Pick a room for gold. (Not the same as Wumpus)

(d) Except from these two rooms and start room, every room has the probability of 0.2 to have a pit. We achieve this goal by randomly choosing from 1~5. If "1" is chosen, the room has a pit.

(e) Every room has a list to represent its attributions, including breeze, stench, glitter and scream

(f) Person list is used to ascertain the position of the player

Exploration Agent 1

1. Method 1

(a) Ascertain the position of player by making a list where one "person" in there and judging the position of "person" in list. Then, find all possible actions in the grids.

(b) Find the attributions of this square and infer the surroundings around it with knowledgebase. Choose a safe place if there is a absolutely safe place to go. Choose a not unsafe place when we are uncertain the safety of squares around. Set safe orientation as the orientation of the next movement. Face is the next movement orientation.

(c) If we find that Wumpus is as the same row or column with the current position, use arrow to kill Wumpus.

(d) Execute actions until we finding a gold.

(e) During the seeking, once Payoff is less than -1000, print "Game over".

2. Result

In method 1, because of not familiar with knowledge base and logic, our project is not good. It runs so slowly and even cannot come out with results in complete maps, which made us cannot get the average payoff of 10000 random maps.

The simulation 0, which is given by professor, is shown in figure 1. The output of simulation 0 is shown in figure 2. As shown in figure 2, the payoff is 997, man moved three steps (right, up, up) and finally got the gold. The result shows that method 1 is not good at calling logic and knowledge base to solve problems. The man will be caught by Wumpus or falls into pits in many cases.

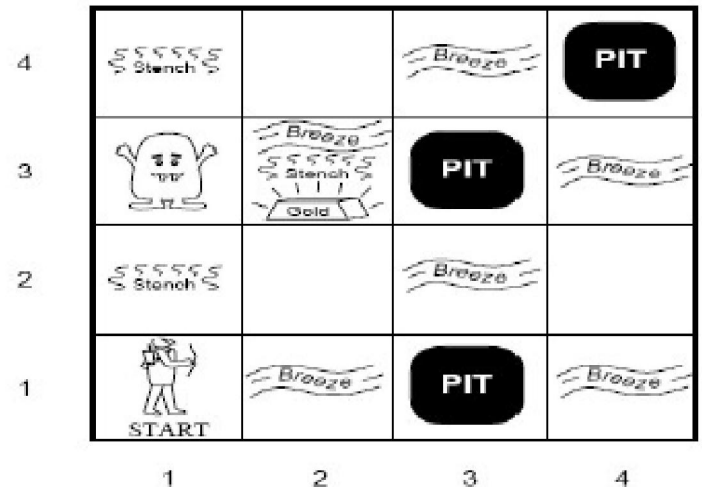

Figure 1: simulation 0

```
['stench'][]['breeze']['pit']

['wumpus']['breeze', 'stench', 'glitter']['pit']['breeze']

['stench'][]['breeze'][]

['start!']['breeze']['pit']['breeze']

right
up
up
Get gold!
997
```

Figure 2: the best output of simulation 0 by using method 1

In this assignment, we were asked to run 10 simulations randomly. But as professor said on bright space, the logic.py which was given for reference may make some procedures very slow. Random maps leads to 10 different outputs are as bellowed. By using this method, 5/10 cases are solved, which means find the gold. The other 4 cases are unsolvable. Finally, the average payoff of these ten maps is -1.5 and the average payoff of 100 random maps is 35.4. The average payoff is low because it falls into pits too frequently.

method 1	Average payoff	Win times
10 random maps	-1.5	5/10
100 random maps	35.4	52/100

Figure 3: outputs of random maps by using method 1

```
['breeze', 'pit']['breeze', 'stench']['pit']['breeze']

['breeze', 'stench', 'pit']['breeze', 'wumpus']['breeze', 'glitter', 'stench']['pit']

['breeze', 'wumpus']['breeze', 'stench', 'pit']['breeze', 'pit']['breeze']

['start!', 'stench']['breeze', 'glitter']['breeze']['pit']

Get gold!
999

['stench']['wumpus']['breeze', 'stench'][]

['breeze']['breeze', 'stench']['pit']['breeze']

['pit']['breeze']['breeze', 'glitter'][]

['start!', 'breeze'][][][]

up
Fall into pit
Game over!
-1001

['pit']['breeze']['stench'][]

['breeze']['stench']['wumpus']['breeze', 'stench']

[][]['breeze', 'stench']['pit', 'breeze']

['start!']['glitter', 'breeze']['pit', 'breeze']['pit', 'breeze']

right
Get gold!
999
```

```
[][][]['stench']

['breeze']['breeze']['stench']['wumpus']

['breeze', 'pit']['breeze', 'pit']['breeze']['glitter', 'stench']

['breeze', 'start!']['breeze'][][]

up
Fall into pit
Game over!
-1001

['pit', 'stench']['breeze']['glitter', 'breeze'][]

['wumpus', 'breeze']['breeze', 'stench']['pit']['breeze']

['breeze', 'stench']['pit']['breeze']['breeze']

['start!']['breeze']['pit', 'breeze']['pit', 'breeze']

up
up
Caught by Wumpus
Game over!
-1002

[][][][]

['breeze']['glitter'][]['breeze']

['pit']['breeze']['breeze', 'stench']['pit']

['breeze', 'start!']['stench']['wumpus']['breeze', 'stench']

up
Fall into pit
Game over!
-1001

[]['breeze'][][]

['breeze', 'glitter']['stench', 'pit']['breeze'][]

['stench']['breeze', 'wumpus']['stench', 'breeze'][]

['start!']['stench', 'breeze']['pit']['breeze']

up
up
Get gold!
998

['wumpus']['breeze', 'stench']['pit']['breeze']

['glitter', 'stench', 'breeze']['breeze', 'pit']['breeze'][]

['breeze']['breeze', 'pit']['breeze'][]

['start!']['breeze'][][]

up
up
Get gold!
998
```

```
['stench']['wumpus']['stench'][]

['breeze']['stench'][][]

['pit']['glitter', 'breeze']['breeze'][]

['start!', 'breeze']['breeze']['pit']['breeze']

right
up
Get gold!
998


['pit', 'breeze']['pit', 'breeze']['breeze'][]

['pit', 'breeze']['breeze', 'stench']['pit']['breeze']

['glitter', 'breeze', 'stench']['wumpus']['breeze', 'stench'][]

['start!'][]['stench'][][]

right
up
Caught by Wumpus
Game over!
-1002
```

Exploration Agent 2

1. Method 2

This method is to run safe and unsure way for each square. As we know, sometimes based on our lack knowledge base, safe ways may not be inferred for each square. So, we remove unsafe actions from unsure list and choose one action in it if there is no safe action available, which means the person may return to the former square. Because of complexity, this method costs much time and cannot come out 10000 times. So we just put the result of simulation 0 and five random maps in report. To improve this, we improved Knowledge base and got method 3, which is much better than method 2. Finally, we choose method 3 as our final algorithm.

2. Result

Figure 4: output of simulation 0 by using method 2

```
['stench'][]['breeze']['pit']

['wumpus']['breeze', 'stench', 'glitter']['pit']['breeze']

['stench'][]['breeze'][]

['start!']['breeze']['pit']['breeze']

right
left
up
Scream!
up
right
Get gold!
985
```

Figure 5: five outputs of random maps by using method 2

```
['stench']['wumpus']['breeze', 'stench'][]

['breeze']['breeze', 'stench']['pit']['breeze']

['pit']['breeze']['breeze', 'glitter'][]

['start!', 'breeze'][][][]

up
Fall into pit
Game over!
-1001
```

```
[]['breeze']['pit']['breeze']

[][]['breeze']['breeze']

[]['stench', 'glitter']['breeze']['pit']

['stench', 'start!']['wumpus', 'breeze']['stench', 'pit']['breeze']

up
right
Get gold!
998
```

```
[][]['breeze']['breeze']

[]['breeze']['breeze', 'pit']['breeze', 'pit']

['glitter'][]['breeze']['breeze', 'stench', 'pit']

['start!'][]['stench']['wumpus']

right
left
up
Get gold!
997
```

```
[]['glitter', 'breeze'][][]

['breeze']['stench', 'pit']['breeze'][]

['stench']['wumpus', 'breeze']['stench'][]

['start!', 'breeze']['stench', 'pit']['breeze'][]

up
down
right
Fall into pit
Game over!
-1013

['breeze', 'pit']['breeze'][][]

['breeze', 'pit']['breeze', 'pit']['breeze'][]

['breeze', 'glitter']['breeze'][]['stench']

['start!'][]['stench']['wumpus']

right
left
up
Get gold!
997
```

Exploration Agent 3

3. Method 3

By researching former methods, we find avoiding caught by Wumpus and falling into pits is more helpful to increase average because once you dead, the payoff minuses 1000 marks and worse than stand still. So we only run in the start square and never take risks to get the gold.

This method is written for accelerate the solving process, which uses heuristic and rewrite the form of action.

We first replace actions with coordinates because we consider this might decrease steps and complicates of seeking. We also use heuristic to try to find the gold as quick as possible. However it still not work well sometimes

The advantages of this method have two.

For the first one, the speed is faster than former two but still has room to improve. Apart from that, the exploration is more specific, which means decrease the possibility of falling into pits and caught by Wumpus.

For the second, this algorithm only choose absolutely safe place to go and if person meet a blank board, it will remain return choice. This configuration promise the agent say “no way to go” only when it runs all safe places.

As for the start square, the agent must go even if it has risks of falling into pits. The disadvantage of this method is obvious, which is sometimes the algorithm may cannot find the best way to go, which means the average payoff will decrease a little bit.

As the algorithm goes, the knowledge base becomes bigger and bigger so that “ask if true” will spend lots of time running so that the exploration Is very slow.

4. Result

The screenshot below is a good solution of exploration. Because it has to using increasing knowledge base and inferring solutions, the algorithm run slower and slower as time goes by. So we choose resolutions which can be printed in limited time to show. .

Figure 4: a good output of simulation 0 by using method 3

```
['stench'][]['breeze']['pit']

['wumpus']['breeze', 'stench', 'glitter']['pit']['breeze']

['stench'][]['breeze'][]

['start!']['breeze']['pit']['breeze']

( 1 , 1 )
( 1 , 2 )
( 1 , 1 )
Scream!
( 2 , 1 )
( 2 , 2 )
( 2 , 3 )
Got geld!
985
```

Figure 5: ten outputs of 4*4 random maps by using method 3

```
['pit', 'breeze']['breeze']['stench']['breeze']

['pit', 'breeze']['stench', 'pit', 'breeze']['wumpus', 'breeze']['stench', 'pit']

['breeze']['glitter', 'breeze']['stench']['breeze']

['start!'][]['breeze']['pit']

( 1 , 1 )
( 2 , 1 )
( 1 , 1 )
( 1 , 2 )
( 2 , 2 )
Get gold!
996
```

```
[][][][]

[][][]['stench']

[][]['stench']['wumpus']

['start!'][]['glitter']['stench']

( 1 , 1 )
( 1 , 2 )
( 1 , 3 )
Get gold!
998
```

```
['pit']['breeze'][][]

['breeze'][][][]

[]['glitter']['stench'][]

['start!']['stench']['wumpus']['stench']

( 1 , 1 )
( 2 , 1 )
( 2 , 2 )
Get gold!
998
```

```
[][][][]

['breeze']['breeze'][][]

['pit', 'breeze']['pit', 'breeze']['glitter', 'breeze']['stench']

['start!', 'breeze']['breeze']['stench']['wumpus']

( 1 , 1 )
( 2 , 1 )
Fall into pit
Game over!
-1001
```

```
['pit']['breeze'][][]

['breeze'][][][]

[]['glitter']['stench'][]

['start!']['stench']['wumpus']['stench']

( 1 , 1 )
( 2 , 1 )
( 2 , 2 )
Get gold!
998
```

```
['breeze']['pit', 'breeze']['pit', 'breeze']['breeze']

[]['breeze']['breeze'][]

[][][]['stench']

['start!'][]['glitter', 'stench']['wumpus']

( 1 , 1 )
( 1 , 2 )
Scream!
( 1 , 3 )
Get gold!
988

['breeze']['pit']['stench', 'breeze']['pit']

[]['stench', 'breeze']['wumpus', 'breeze']['glitter', 'breeze', 'stench']

['breeze']['breeze', 'pit']['stench', 'breeze', 'pit']['breeze']

['start!']['breeze']['breeze'][]

( 1 , 1 )
( 2 , 1 )
( 1 , 1 )
( 1 , 2 )
( 1 , 1 )
( 1 , 1 )
No way to go
-5
```

The average payoff of these ten random maps is 396.2, which is much higher than the average payoff of method 1. Actually, the calculation costs lots of time and thus we use 3*3 grids to calculate average marks in 10000 times. The average mark is 421.62.

Method 1	Average payoff	**Method 3**	Average payoff
10 random maps	-1.5	10 random maps	396.2
100 random maps	35.4	10000 random maps	421.62

As we can see from the solution, the possibility of finding the gold and avoiding Wumpus and pits is much higher than former algorithm. It meets his demand of question basically.

Reference

Logic codes from AIMA-Python,

https://github.com/aimacode/aima-python/blob/master/logic.ipynb