

Hunt The Wumpus Game

Can Eldem

September 12, 2013

Abstract

This report explains a Artificial Intelligence project written with Java programming language. Project aims to simulate a game called Hunt the Wumpus.

Contents

1	Literature Review	3
2	Design	3
3	Vocabulary	5
4	Testing	6
5	Examples and Results	7
6	Conclusion	12

1 Literature Review

Hunt the Wumpus is an adventure type of game created by Gregory Yob in 1972. Main Object of the game is get treasure and find exit without killed by wumpus. When game developed by Gregory it was written in BASIC programming language and game was text based. By time regarding developments in programming languages and operating systems Wumpus game is re-written in 1975 .[1, 2] Game was including more complex features and missions .”In Snark(another version of Wumpus game), one must enter the radius of a circle around a central grid point to be informed whether the snark is inside or outside, while Mugwump tells only how far in a direct line the mugwump is hiding from each guess, leaving the player to puzzle out the direction for herself.”[1, 3].In addition to these varieties there were other versions of wumpus game such as with möbius strip or with limited arrows and super bats . In 1980 , in the light of technological developments game was developed with graphical user interface . General structure of game was same only graphical developments had been made.[1, 2].Further developments of game made in graphical interface field .

2 Design

The game is designed to play by human without any artificial intelligence. In order to achieve this task , object design made for hold information as much as possible and small objects and methods created for each action so that it could be used in complex methods in code .

All characters in game has considered as objects and methods are their actions in the game .In addition to characters all rooms (blocks) designed as an object which hold information about percepts and characters (wumpus,agent,treasure) thus blocks hold information about exit and whether block is empty or not in case of nothing in that block. Holding whether block is empty or not helps AI to interpret information with less afford(in terms of cost) . (Fig-1)

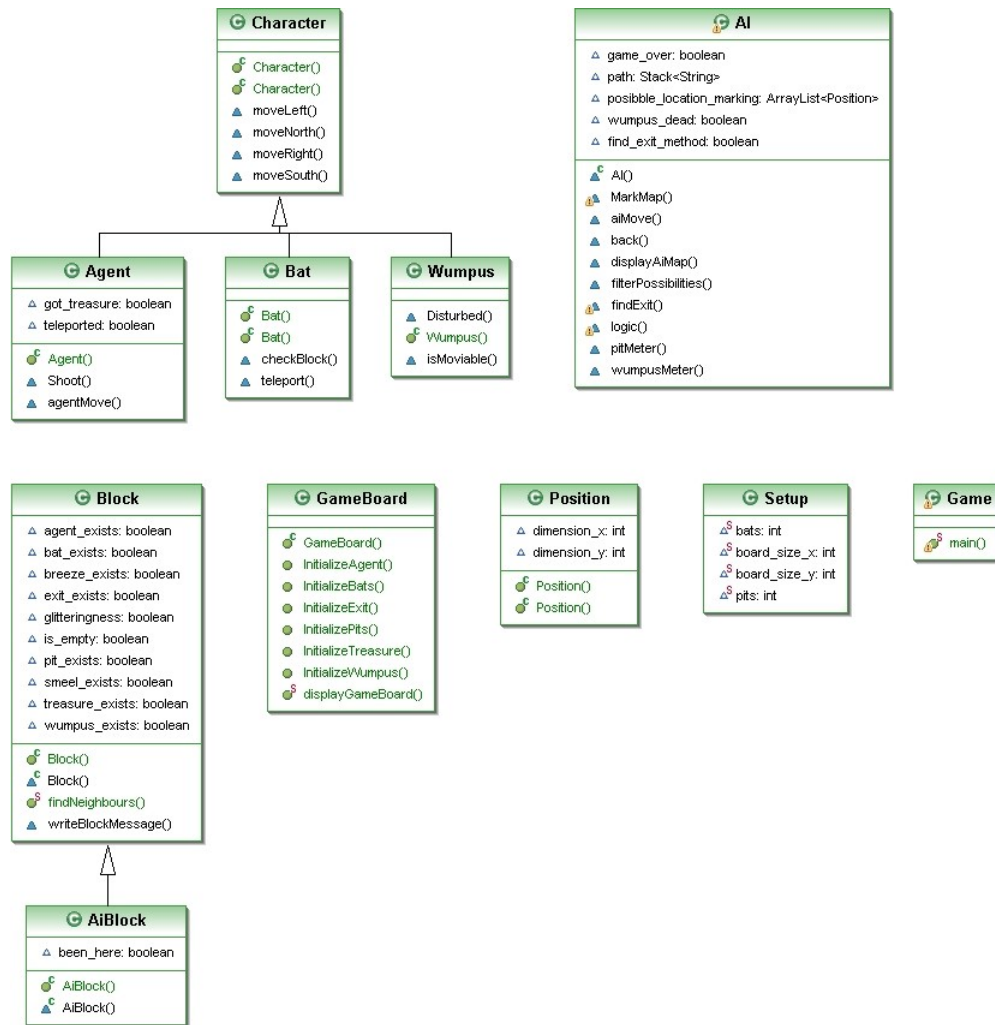


Figure 1: Class diagram of Java Project

Figure2 shows how game board is structured in two dimensional array. Every point in figure 2 shows an block object , index information is used as position information of block objects . E.g position[x][y]

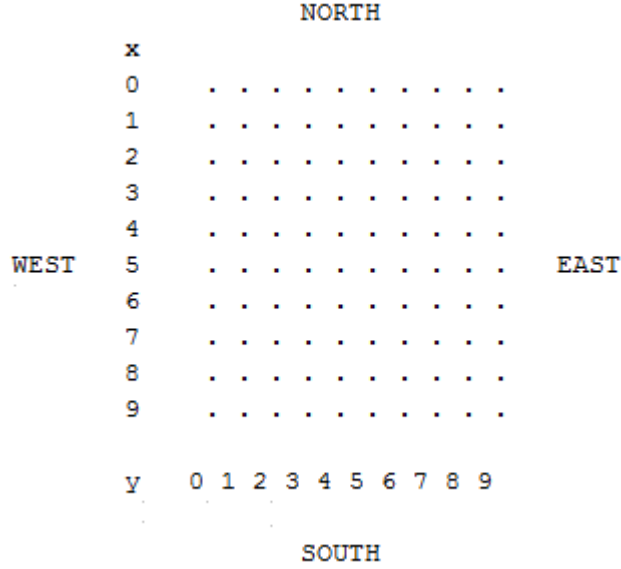


Figure 2: Game board

All character objects have their own position which will be set as index information of related block and all characters have move methods to change position. Move function designed to consider the shape of cave when they are called. AI intelligence design has been made in the light of obtained information in Russell and Norvig book chp 7. According to rules of the wumpus game agent allows to get percepts and create a knowledge base so that future decisions could be based on previous information. In order to achieve maximum percentage of success most suitable artificial design for this object is designing a learning agent. In this way artificial intelligence collects information (ASK) interpret and do specific action for that situation (TELL) then add or update new information to storage of agent. All AI design ideas implemented to Ai class. Markmap() method in Ai class gets current information of block and process to artificial agents own knowledge base map(ai map) with help of previous forecasted information. Then logic class do series of actions based on knowledge.

3 Vocabulary

Before moving to testing section. I would like to explain some of the terminology on the game map.

- N=null (no information about block)
- X=Agent been here

- W=wumpus
- P=pit
- T=treasure
- Q=exit
- E=empty

4 Testing

Trial and error testing techniques have been used in order to validate project .

```

exit location is saved
got treasure:true
north cost:0
south cost:0
west cost:0
east cost:0

*****AI board*****
X X X X X X X E N W X W N N P
X X X X X X X E W X X E N P X
X X X X X X X E N W X X P X X
X X X X X X X E N E X X X P X
X X X X X X X E N N E X X E X
X X X X X X X E N E N E X E X
X X X X X X X E E X E E X E X
X X X X X X E N N E N E X E X
X X E X X X E N N N N E X E X
X P X X X X E N N N N N E E X
X X X X X X E N N N N N N E X
X X X E X X P N N N N N N N E
E X X E X X X P N N N N N N N
E X X X X X P N N N N N N N N
E X X X X X X P N N W N N N N
wins:7
lose:3
average:336
got treasure:true

```

Figure 3: Test results on the screen

Instead of running game for each time and wait game come to an end. One loop is set in order to run game multiple times in one runtime. When code exits from the loop, statistics values displayed for that loop like in figure 3. In figure 3 game is run for 10 times and 7 of attempts end with success for this set of settings. (15X15 board 5 pits 5 bats). If there is a problem in the game such as AI agent can't decide what to do or there is a kind of code error such as null pointer exception ,testing loop does not stop or runtime environment gives an error. Code is analyzed with debugging tools and with error guessing techniques code try to be fixed.

5 Examples and Results

Initial position of AI shown in figure . When agent starts to game it marks its neighbors the map according to information of current block. In a way ai is marking its possibilities around itself and indicates that there is a treasure around itself with treasureMeter() function

```

my current location is:7,4  room is glittering...
Game board:                treasure meter is:4
* * * * T * * *          Agent moves to NORTH...
* * * * * W * *          there is nothing in this room..
* * * E * * * P
P * * * * B * *          *****AI board*****
* * * * * * * *          N N N N T N N N
* * * * * * * *          N N N N N N N N
* * * * * * * *          N N N N N N N N
* * * * * * * *          N N N N N N N N
* * * * A * * *          N N N N N N N N
                        N N N N E N N N
                        N N N E X E N N
                        N N N T X T N N

```

Figure 4: AI gathering information

In figure-5 after series of moves agent face with a position which agent surrounded by possible treasure and wumpus information .As it is indicated in figure-5 treasure meter is 3 and wumpus meter is 3 which shows that agent does not know exact position of treasure (exact position is when meter shows 1) in that case best move is going back and searching for other possibilities .

```

my current location is:1,4
Game board:
* * * * T * * *
* * * * A W * *
* * * E * * * P
P * * * * B * *
* * * * * * * *
* * * * * * * *
* * * * * * * *
* * * * * * * *

smell detected...
room is glittering...
treasure meter is:3
wumpus meter is:3
Evaluated... need to turn back.
last move:NORTH
Agent moves to SOUTH...
there is nothing in this room...
treasure meter is:0
wumpus meter is:0

*****AI board*****
N N N N W N N N
N N N W X W N N
N N N E X E N N
N N N E X E N N
N N N E X E N N
N N N E X E N N
N N N E X E N N
N N N T X T N N

```

Figure 5: There is not enough information in knowledge base

In figure-6 agent faces with exit and saves exit location .in case it finds treasure to it will its destination point to finishing the game .


```

my current location is:1,4
Game board:
* * * * T * * *
* * * * A W * *
* * * E * * * P
P * * * * B * *
* * * * * * * *
* * * * * * * *
* * * * * * * *
* * * * * * * *

smell detected...
room is glittering...
treasure meter is:3
wumpus meter is:3
Evaluated... need to turn back.
last move:NORTH
Agent moves to SOUTH...
there is nothing in this room...
treasure meter is:0
wumpus meter is:0

*****AI board*****
N N N N W N N N
N N N W X W N N
N N N E X E N N
N N N E X E N N
N N N E X E N N
N N N E X E N N
N N N E X E N N
N N N T X T N N

```

Figure 6: Agent saves the location of exit point

In figure-7 agent decreases the possibilities of pit which takes place in $x=3$ $y=0$ but since agent could not detect exact position of it turns back .

```

my current location is:3,1
Game board:
* * * * T * * *
* * * * * W * *
* * * E * * * P
P A * * * B * *
* * * * * * * *
* * * * * * * *
* * * * * * * *
* * * * * * * *

It is cold...
pit meter is:2
Evaluated... need to turn back.
last move:NORTH
Agent moves to SOUTH...
there is nothing in this room...

*****AI board*****
N E X X T N N N
N E X X X W N N
N P X X X E N N
P X X X X E N N
E X X X X E N N
N E X X X E N N
N E X X X E N N
N E X X X T N N

```

Figure 7: Agent turns back

In figure-8 shows that agent changes and updates its knowledge base according to current and previous information. Comparing with figure-8 possible pit in 4.0 becomes X symbol which is secure area for agent to visit .

```

my current location is:3,7
Game board:
* * * * T * * *
* * * * * W * *
* * * E * * * P
P * * * * B * A
* * * * * * * *
* * * * * * * *
* * * * * * * *
* * * * * * * *

pit meter is:3
Evaluated... need to turn back.
last move:NORTH
Agent moves to SOUTH...
there is nothing in this room...

*****AI board*****
X X X X T N N E
X X X X X W N E
X X X X X E N P
P X X X X E P X
X X X X X E E X
X X X X X E N E
X X X X X T N E

```

Figure 8: Agent updates knowledge base

AI agent moves to their neighbors even though treasure meter is two which means AI does not know about treasure's exact position and if there is any other danger. Agent programmed to act like that in order to don't lose time. If AI tries to find treasure's exact position in secure area instead of taking action it will be time consuming. Thus, whenever AI finds treasure it calculates costs to going to exit so it can direct itself to the exit. Agent tires to move exit in the places that it had been to in order to sucess in the game.

```
my current location is:0,3
Game board:
* * * * A * * *
* * * * * W * *
* * * E * * * P
P * * * * B * *
* * * * * * *
* * * * * * *
* * * * * * *
* * * * * * *

room is glittering...
treasure meter is:2
Agent moves to RIGHT...
Agent got the treasure...
exit location is:2,3
north cost:6
south cost:1
west cost:1
east cost:0
```

Figure 9: Treasure meter is two. Agent makes a random action if there is not enough information.

In this part of the report I will mention about logical mistakes of Agent that I designed.

In figure-10 show illustration about how findexit() method works .Method calculates "area" to reach destination point(exit) and tries to go by taking safe steps .However, if exit surrounded by obstacles, AI can't reach to exit .Findexit() method should draw and path to exit instead of going area of exit. Therefore, it does not work %100 efficiency.

Agent can not escape from area when its surrounded by possibilities . AI constantly changes places within safe area because all other routes indicates a risk . In this case project should include kind of variable to indicate risk in a way and if AI choke it should take risk and make an random move .

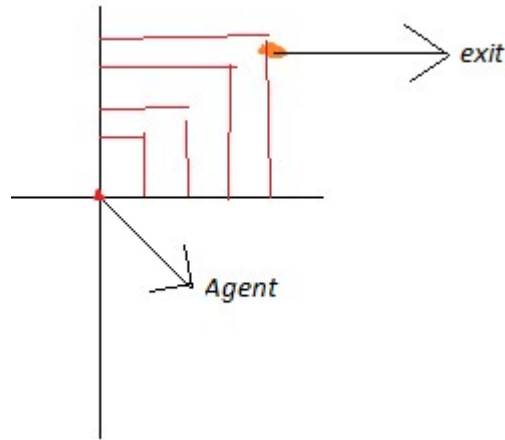


Figure 10: Agent moving towards to located exit point

6 Conclusion

Project manage to create relatively good but not perfect knowledge based agent. Thus , in order to have representative project effectively gui enhancement should be achieved .

References

- [1] Russell SJ, Norvig P. Artificial intelligence: A modern approach. Pearson Education; 2003.
- [2] Kell B, Melik J. Hunt the wumpus. http://en.wikipedia.org/wiki/Hunt_the_Wumpus. Updated 20122010.
- [3] Maher J. Hunt the wumpus part 1-2. <http://www.filfre.net/tag/hunt-the-wumpus/2011>.