VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY

University of Science

Faculty of Information Technology

CS420 - Artificial Intelligence

Project 02: Wumpus World

by

21125071 Ho Viet Bao Long

21125081 Nguyen Si Minh

21125097 Nguyen Dinh Triet

Winter, 2023

Contents

1 Overview

1.1 Introduction

- This report is written to project 02: Wumpus World of the course CS420 at University of Science.
- There are totally 4 chapters in this report. First, we make an overview of the project by specifying the project scope, assigning the tasks for the team's members, describing the environment of the project, and the tutorial for using the program. Second, we demonstrate the main concepts of the Algorithm. Third, we illustrate the experiment by describing the test case and its results. In this chapter, we also making analysis of the algorithm metrics in terms of score and time complexity. Last, the discussion and conclusion section summary the project and define further works for inprovement.
- Github: `https://github.com/trietng/WumpusWorld.git`
- Backup project's drive link: `https://drive.google.com/drive/folders/1Rh-PvHFay9-Zqw3-FvuVi`
`b79k-0E?usp=sharing`
- Video Demo: `https://youtu.be/HMREB8WQa90`

1.2 Tasks Assignments

- Table 1.1 shows task assignments for each member in the project.
- Every team's member had taken part in all of the discussions to find out the solution for the problem. The task assignments is made based on coding responsibility.

	Ho Viet Bao Long	Nguyen Si Minh	Nguyen Dinh Triet
SAT solver	x		
Search algorithm			x
UI		x	
Report: Algorithm Specification			x
Report: Experiment	x		
Report: Introduction and Environment		x	

Table 1.1: Tasks Assignment

- Table 1.2 describes the tasks purpose and makes self-evaluation for each task.

	Description	Self Evaluation (/100%)	Comment
SAT solver	Including DPLL algorithm, Knowledge Base management	100%	
Search algorithm	Including update the game logic for each motion of the agent to make decision	100%	
UI	Illustration the agent process	100%	
Report: Algorithm Specifications	Description the main concepts of the algorithms	100%	
Test case and Report: Experiment	Design test case and make analysis	100%	
Report: Introduction, Environment, and video demo	Making introduction and define environment of the game.	100%	

Table 1.2: Tasks description and Self-evaluation

1.3 Environment

The purpose of the project is to design and implement a logical agent that navigates through the Wumpus World, a partially-observable environment. The primary objective is to locate the goal and safely exit the cave. During the journy, the agent may need to kill the wumpus the pursue success.

The PEAS (Performance measure, Environment, Actuators, Sensors) for the Wumpus World agent includes:

Performance Measure:

- The agent's performance is measured by the amount of gold it collects, each gold has 1000 points.
- The agent's performance decreases if it falls into a pit or gets killed by the Wumpus, if the agent is killed, the agent loses 10000 points
- The agent's performance increases if it successfully exits the cave, the agent will get 10 points.
- The agent's performance decreases if it shoots the arrows. Each arrow shot results in a loss of 100 points.

Environment:

- The environment is a 10x10 grid of rooms in an underground cave.
- Each room may contain a pit, a Wumpus, or gold.
- The agent starts at any room and always faces to the right.
- The exit is at room (1, 1).
- The environment is partially observable because the agent can only perceive its immediate surroundings (the four-neighborhood of the current room).
- There is at least one path that the agent can take to exit the cave safely. Specifically, there should be a routine that not ambiguous at all (for example, it should not have a boundary of Pit around the exit room).

Actuators:

- The agent can move forward, backward, turn left and right by 90 degrees.
- The agent can shoot the arrow in the direction it is facing

Sensors:

- The agent can perceive a breeze if it is in a room adjacent to a pit.
- The agent can perceive a stench if it is in a room adjacent to a Wumpus.
- The agent can perceive a scream if it has successfully killed the Wumpus.

Tutorial:

- cd WumpusWorld-master
- cd wumpusworld
- run main.py

2 Algorithm

2.1 SAT solver

- In this project, we use DPLL as the main technique for solving the SAT problem in the Knowledge Base of agent while it exploring the map. We use Unit Propagration and Pure Literal Elimination to prune the clauses before using backtracking to verify the satisfiability of the clause.
- The clause is stored in a dictionary. The negative symbol is denote as <symbol>: 1, otherwise, <symbol>: 0. For example, 'W': 1 means there is no Wumpus is this room; 'P': 0 means there is a Pit in the room.
- The solver works like this: When the agent enters a stench (or breeze) room, it assumes that the nearby rooms might have a wumpus (or pit). Then, it uses DPLL to check if the wumpus (or pit) can be in this room or not. If the DPLL algorithm returns False, which mean it is the assignment is unsatisfiable, then the agent can be sure that the room doesn't have a wumpus (or pit).

2.2 Data structure

2.2.1 World

This is the initial data structure that stores the map like how it was loaded from the disk, which includes Wumpus(W), Pit(P), Agent(A) and Gold(G) with extra percepts Stench(S) and Breeze(B) updated according to the former. It uses the Wumpus World coordinate system: the

bottom, leftmost room has the coordinate of $(1, 1)$, which is also the exit point of the cave (the agent does not know this).

2.2.2 Map

The agent does not know the true size of the map (10x10), so every time it visit a new room, that room and its adjacent rooms will be added to the agent's "memory" (i.e. a dynamic 2-dimensional array). The coordinate system of this data structures has the origin $(0, 0)$ be the room where the agent starts, with the adjacent coordinates of any specific room (x, y) be $(x-1, y)$ (left), $(x+1, y)$ (right), $(x, y-1)$ (down) and $(x, y+1)$ (up). This coordinate system is also used as the identifier for each room in the knowledge base.

2.2.3 Types of room

There are 3 types of room in the Wumpus World: SAFE, UNSAFE and EXPLORED. The search algorithm will utilize this information to finish the problem.

2.3 Search algorithm

To gain the most score out of the Wumpus World, our agent obeys the following rules:

1. Each time the agent visits a room that was marked as SAFE, it would update the room's type to EXPLORED.

2. If gold was detected inside the room, pick it up.

3. If the current room (x, y) contains neither Breeze nor Stench, the agent would mark the adjacent, not EXPLORED rooms as SAFE. The knowledge base would be updated with the clause $\neg W(x-1, y) \wedge \neg P(x-1, y) \wedge \neg W(x+1, y) \wedge \neg P(x+1, y) \wedge \neg W(x, y-1) \wedge \neg P(x, y-1) \wedge \neg W(x, y+1) \wedge \neg P(x, y+1)$. Then, the agent would pick an adjacent SAFE room to visit.

4. If the current room (x, y) contains Breeze, the agent would use knowledge base and SAT

solver to infer whether an adjacent room that was yet to be visited (not EXPLORED) could possibly contain a Pit (SAFE) or not (UNSAFE). The clause $B(x, y) \Rightarrow P(x-1, y) \vee P(x+1, y) \vee P(x, y-1) \vee P(x, y+1)$ is added to the knowledge base. Then, the agent would pick a SAFE room to visit.

5. If the current room contains Stench, the agent would attempt to shoot in the directions of all rooms that were yet to be visited (not EXPLORED) until the Stench in the current room disappears. The information on whether an arrow hit a Wumpus is known by its scream. It would then pick an adjacent SAFE room to visit.

6. **Optimization:** After shooting, if the adjacent rooms are all EXPLORED or UNSAFE, then the agent will search its memory for the path to the nearest unexplored safe room by using the UCS algorithm and go there to continue the searching process. The path only includes all the room it has visited and confirmed to be safe. This is the faster alternative to the our original backtracking method.

7. If the map as it was recorded by agent does not contains any room that was yet to be EXPLORED, stop searching and find the shortest path to the exit room to climb out of the cave.

3 Experiment and Analysis

3.1 Test case 1:

- Figure 3.1 shows the input map of test case 1. This is the basic map with two wumpuses around the exit room. There are also many Golds and Pits that can challenges the agent.
- The result and analysis after the algorithm is executed is shown in Figure 3.2 and Table 3.1, correspondingly.

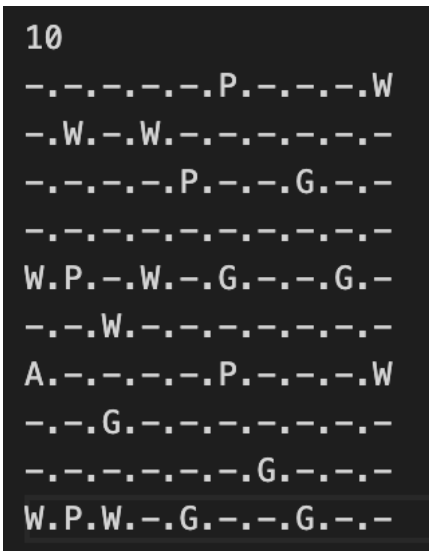

Figure 3.1: Input file Map 1.txt

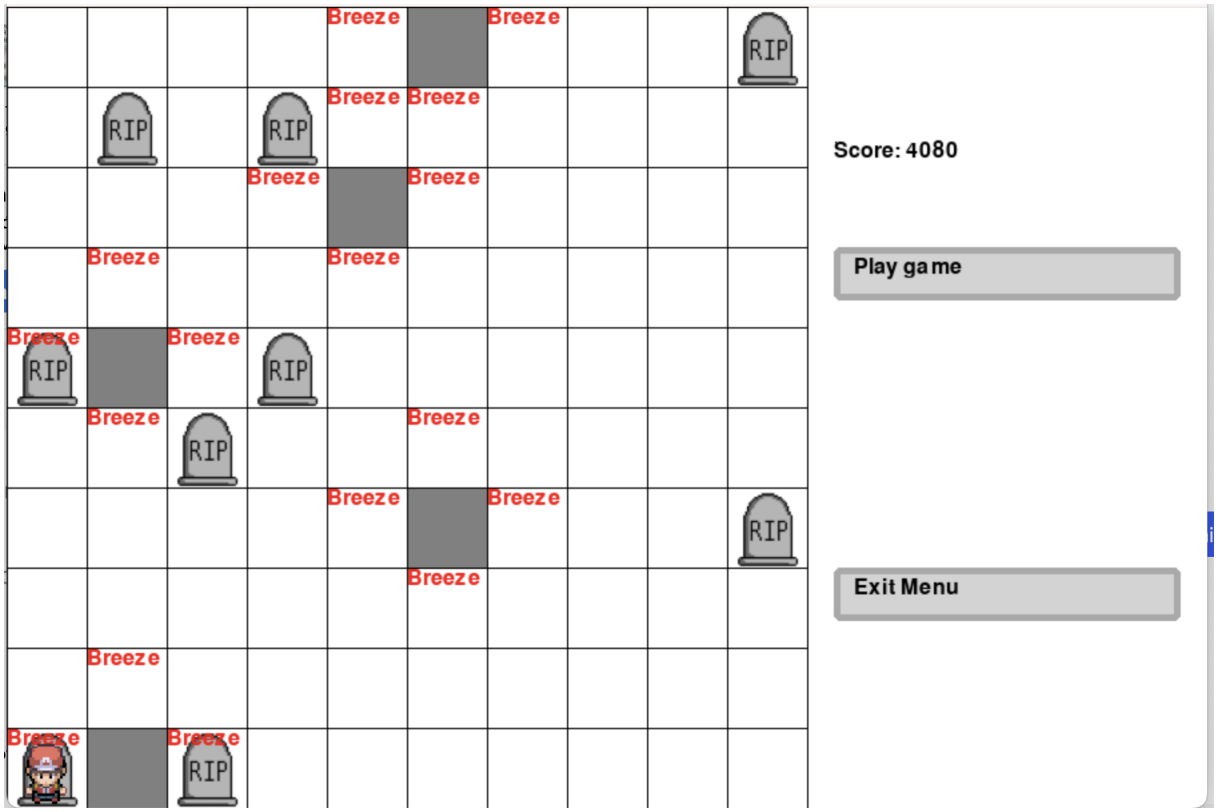

Figure 3.2: The final state of the Wumpus World for map1

Score	Execution time (s)
4080	0.229

Table 3.1: Score and execution time for map1

3.2 Test case 2:

- Figure 3.3 shows the input map of test case 2. This map is more challenged than test case 1, which has many wumpus around the exit door. Moreover, there are also numerous of wumpus surround numbers of gold.

- The result and analysis after the algorithm is executed is shown in Figure 3.4 and Table 3.2, correspondingly.

```
10
G.W.-.-.-.-.-.-.W.G
W.-.P.P.-.-.P.-.W.G
-.-.-.-.-.W.-.-.W.W
P.-.-.-.-.-.-.-.-.G
-.-.W.-.W.-.P.-.-.-
-.-.-.-.-.-.-.-.-.-
-.G.-.-.-.W.-.-.-.-
W.W.P.-.W.A.W.-.P.-
-.W.W.-.-.W.-.-.-.-
-.W.W.-.-.-.-.-.-.-
```

Figure 3.3: Input file map2.txt

Score	Execution time (s)
550	0.247

Table 3.2: Score and execution time for map2

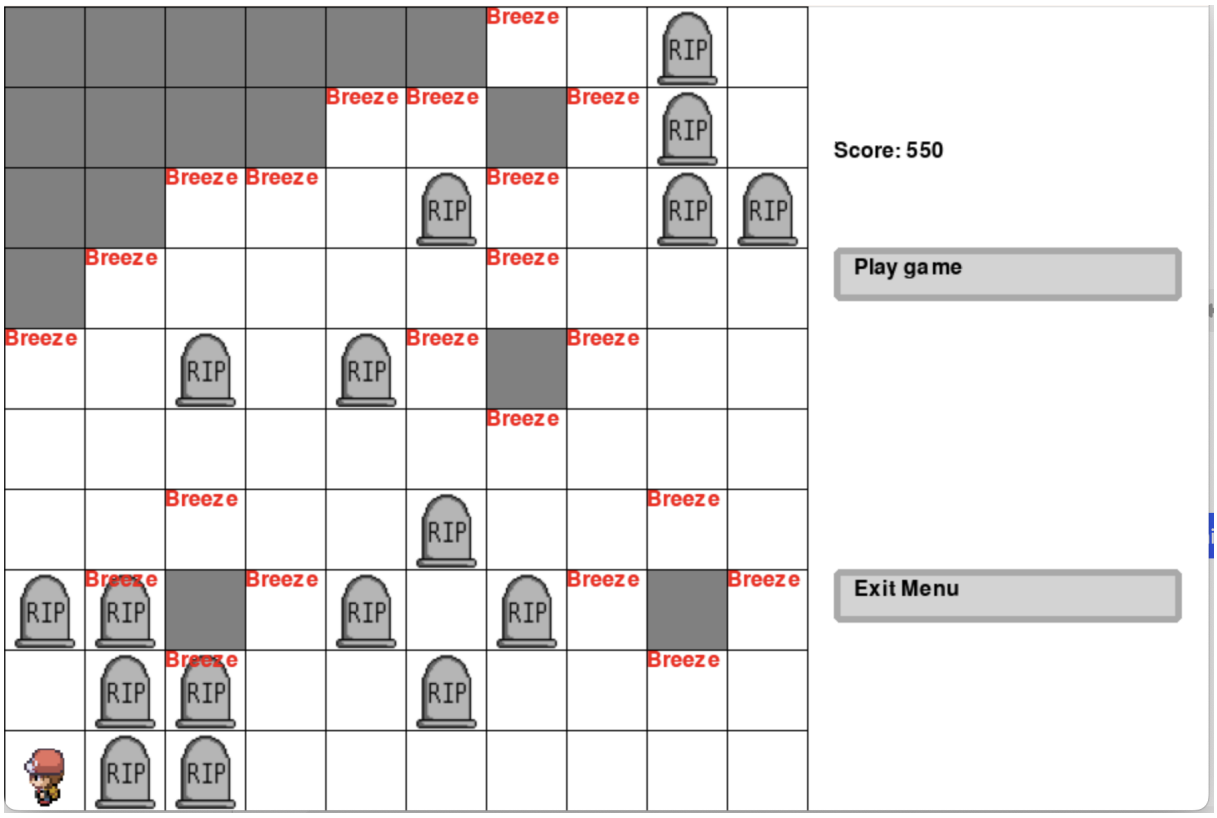

Figure 3.4: The final state of the Wumpus World for map2

3.3 Test case 3:

- Figure 3.5 shows the input map of test case 3. In this map, there is a Pit beside the started point of the agent. It requires the agent to find out the safe room to start the trip. In this case, the safe room can be recognized by the scream of the wumpus after be shoot.
- The result and analysis after the algorithm is executed is shown in Figure 3.6 and Table 3.3, correspondingly.

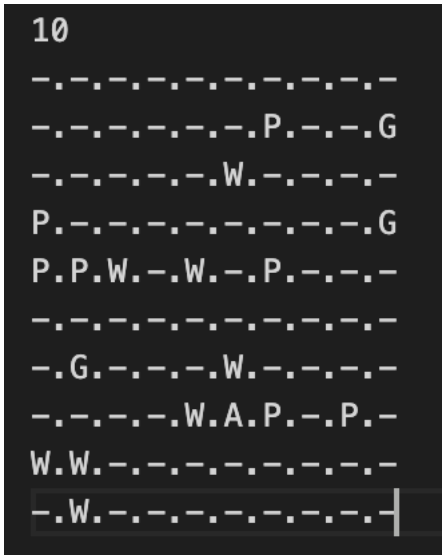

Figure 3.5: Input file map3.txt

Score	Execution time (s)
-1360	0.139

Table 3.3: Score and execution time for map3.txt

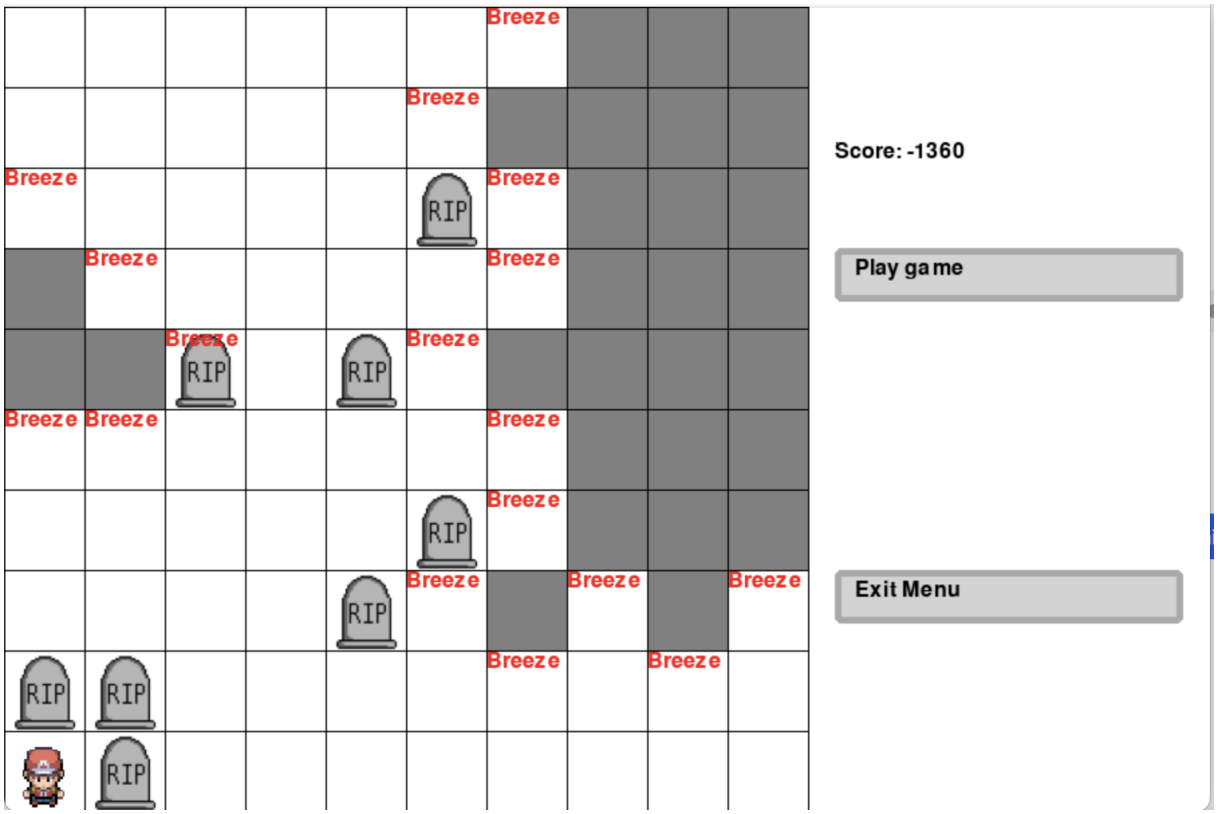

Figure 3.6: The final state of the Wumpus World for map 3

3.4 Test case 4:

- Figure 3.7 shows the input map of test case 4. In this map, wumpus and pits are not located as group, but separated around the map randomly.

- The result and analysis after the algorithm is executed is shown in Figure 3.8 and Table 3.4, correspondingly.

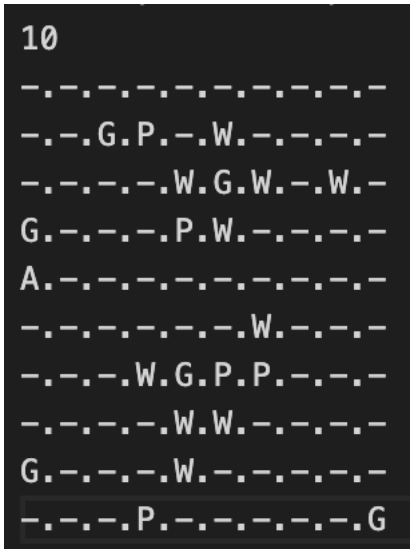

Figure 3.7: Input file map4.txt

Score	Execution time (s)
3360	0.177

Table 3.4: Score and execution time for map 4

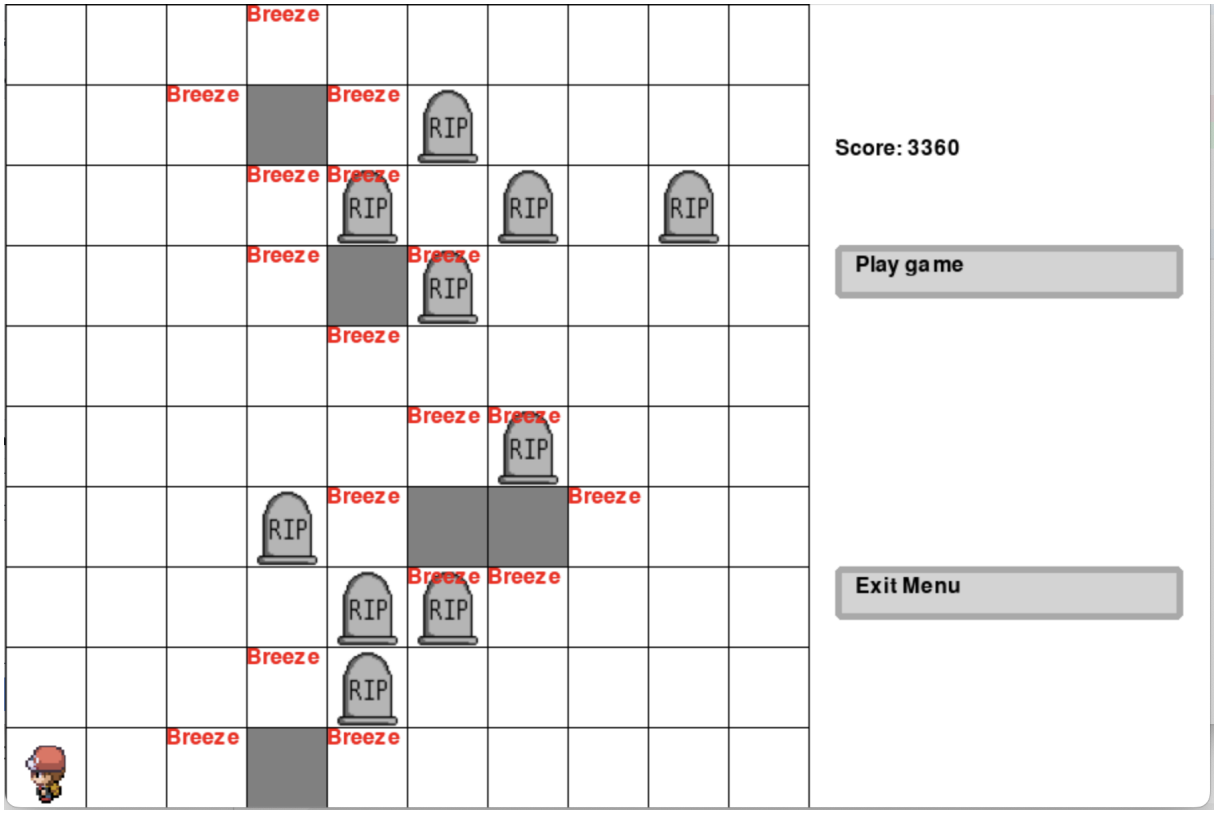

Figure 3.8: The final state of the Wumpus World for map 4

3.5 Test case 5:

- Figure 3.9 shows the input map of test case 4. There is many classes of wumpus around the exit door. Especially, there is a Pit besides the exit door, which causes the exit door's perception contains Breeze.

- The result and analysis after the algorithm is executed is shown in Figure 3.10 and Table 3.5, correspondingly.

Figure 3.9: Input file map5.txt

Score	Execution time (s)
910	0.283

Table 3.5: Score and execution time for map5.txt

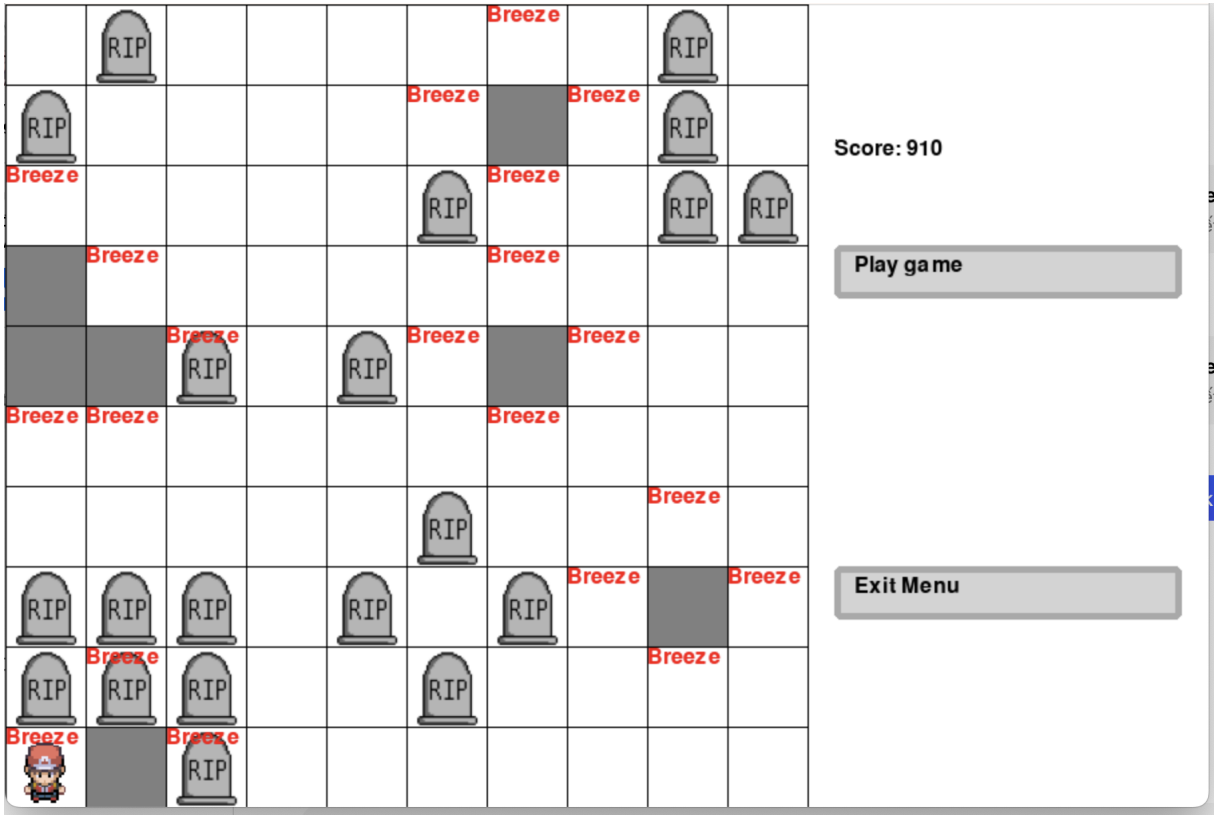

Figure 3.10: The final state of the Wumpus World for map5.txt

4 Discussion and Conclusion

While our agent aims to get the highest score out of its provided input maps, there are still other alternative strategies and search in that could be considered for this problem:

- Try to visit all ”safe” room first before shooting any room with a chance to contain Wumpus. While this might sound better on paper, it still can perform worse than our strategy if the Wumpuses are distributed far away from each other.
- Introduce randomness into decision making. When encountering a room that contains Breeze (after shooting all Wumpuses and visited all other rooms), the agent can try to make a random guess and move forward into the direction that could possibly have a pit. Doing this can help the agent cover more rooms, so it could potentially find more gold bars and improve its score. This, however, could lead to unexpected deaths of the agent, which is not desirable.

5 References

- Pygame: https://www.pygame.org/docs/

- Minisat: http://www.sista.arizona.edu/~clayton/courses/ai/projects/wumpus/docs/wumpus/_agent.html

- Background theory of Wumpus World: https://www.youtube.com/watch?v=NkP1RVPJAII