



Wumpus World: A simulation using Gama Platform and BDI agents

Christian Faccio

University of Alicante
christianfaccio@outlook.it

Abstract. The Wumpus World is a classic problem in artificial intelligence that involves an agent navigating a grid-based environment filled with hazards and treasures. This report presents a simulation of the Wumpus World using the Gama Platform, a powerful tool for building spatially explicit agent-based models. The simulation aims to demonstrate the capabilities of the Gama Platform in modeling complex environments and agent behaviors, as well as to explore strategies for effective navigation and decision-making in uncertain conditions.

1 Introduction

In the context of Multi-Agent Systems (MAS), the Wumpus World can help to understand the dynamics of basic agent interactions within a shared environment. In this case, specifically, we have a simple environment composed of a grid with variable dimensions, a player with the objective of collecting all the gold pieces in the board without falling into a pit or the wumpus and the wumpus and the pits themselves as static agents that represent dangers for the player. The player can move in the four cardinal neighboring cells, and his perceptions are activated once he walks into a cell. The perceptions include: *glitter* (neighboring cells of the gold), *breeze* (neighboring cells of a pit) and *odor* (neighboring cells of the wumpus).

2 Challenges

I decided to implement two different types of players: one which follows a naive and intuitive approach, which is to always move to the precedently visited cell every time he perceives breeze or odor, and another one which follows a more exploratory approach, which is to stochastically move to any neighboring cell once he perceives breeze or odor, with a higher weight for the precedently visited cell. It is intuitive, thus, that the first player will always perform optimally, but

may not find all the gold if the environment is particularly complex, while the second player will be able to explore more of the environment, but may fall into a pit or the wumpus more often.

3 Implementation

Following the BDI architecture, the player navigates the environment with a *belief*, a *desire* and an *intention* at each time step. As for my implementation, for the beliefs the player either

- wants to patrol;
- feels glitter;
- feels breeze;
- feels odor.

For the desires and intentions instead, the player either

- patrols;
- collects gold;
- escapes danger (pit/wumpus);
- chooses a location.

The plans the player has to follow are essentially three:

Random move This is executed whenever the player has the intention to patrol. The player simply moves to a random neighboring cell.

Get gold This is activated whenever the player has the intention to collect gold. In my implementation, the player is smart and knows that if he feels glitter it means that for sure there is a gold cell in one of his neighboring cells, and so he search for all the neighboring cells until he finds it and collects it.

Escape danger This plan is executed whenever the player has the intention to escape the danger (pit or wumpus). In my implementation, there are two versions:

- **Naive player:** the player always moves to the precedently visited cell, in this way he will never fall into a pit or the wumpus, but he may not be able to explore the whole environment and collect all the gold.
- **Exploratory player:** the player stochastically chooses one of the neighboring cells, with a higher weight for the precedently visited cell. In this way, the player will be able to explore more of the environment and potentially collect all the gold, but he may also fall into a pit or the wumpus.

4 Analysis

Different are the analysis one can carry on for this game. As an example, one can analyze the success rate for different weights in the exploratory player, or analyze the cost in terms of total number of moves increasing the grid size. I started from the naive player and for him there is only one analysis I could have done: see if increasing the grid size and keeping everything else fixed the number of total moves increases or not. As expected, it increases linearly, as shown in Fig. 1.

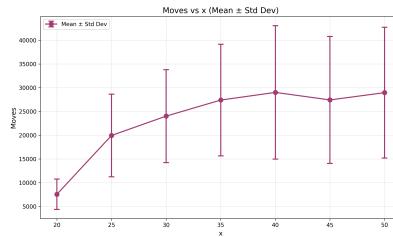


Fig. 1. Naive Player

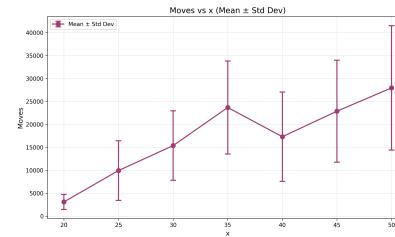


Fig. 2. Exploratory Player

I capped the total number of moves to 10000 so that the simulation would not run forever in case the player was not able to collect all the gold. Moreover, I ran 80 simulations for the values of grid size from 20 to 100 with step 1 (always keeping a square for simplicity).

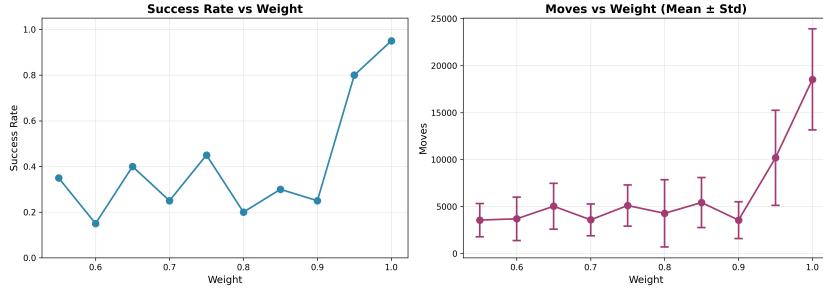
For the exploratory player instead, many more analysis could have been carried on. The first and most intuitive is the same as before, but this time showing also the total winnings and losses. For this I selected a weight for the previous cell of 0.7, dividing uniformly the rest between the other three neighboring cells. The results are shown in Fig. 2.

As expected, the number of moves increases but on lower grids it is more stable. Moreover, the player is able to win more in lower grids, again as expected.

Another interesting analysis is to see the effect of varying the weight of the previous cell on the success rate of the player. One should expect that increasing the weight should increase the success rate (since it approaches the naive player), but at the same time the average number of moves should also increase. The results are shown in Fig. 3.

5 Conclusions

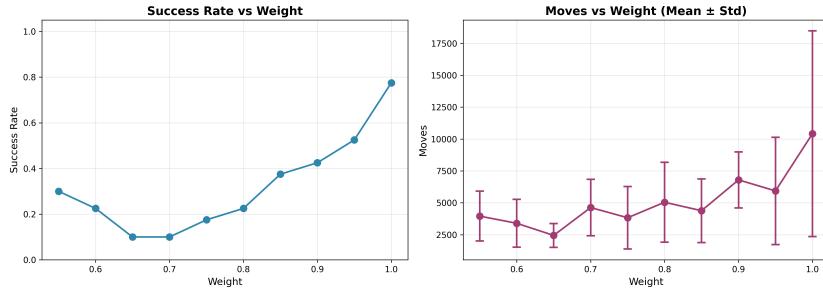
This report presented a simulation of the Wumpus World using the Gama Platform and a BDI architecture for the player, implementing two different types of players. The results confirmed the expected behaviors of both players, highlighting some inefficiencies and trade offs with respect to the choice of policy.

**Fig. 3.** Success rate vs Weight

Even if this game is quite simple, it represents an opportunity to test and learn about agent-based modeling and the BDI architecture, focusing on the dynamics created by a single dynamic agent interacting with static agents in a shared environment. Finally, further improvements are possible, such as implementing a memory concept for the player which could help him remember past paths and navigate more intelligently in the environment, or make the wumpus a dynamic agent as well, which would increase the level of difficulty and complexity of the game.

6 Extra

Since it was appealing to see how a dynamic wumpus would have affected the game, I implemented it as well. In this implementation, the wumpus moves randomly, and does not perceive the player (since also him moves randomly and so there is not a rational way for which even when perceiving him, the wumpus could follow him). Staying with the naive player, I would expect that the player loses more often, since even with the safe strategy of always going back to the previous cell, this time the wumpus could have moved into his cell. The results are shown in Fig. 4.

**Fig. 4.** Cycle vs Grid Size with Dynamic Wumpus