



The Adventure World: A Knowledge-Based Agent

Ethan Axl S. Burayag, Ezra Jeonadab G. Del Rosario, Elisha Jeremy C. Ong
De La Salle University - Manila



Introduction : The Adventure World

***The Adventure World** is a modified adaptation of the Wumpus World as described in Russell and Norvig. The Adventure World will showcase a knowledge-based agent written in Prolog to demonstrate representation and reasoning. The world has a game-like nature consisting of an $n \times n$ grid. In the grid, some tiles may contain either a Pit or a Gold coin. However, the location of the pits and gold coins are hidden from the player. The goal of the Prolog agent is to guide the player by tagging tiles as safe, unsafe, or unknown. It creates inferences based on inputs the user gives as they explore the grid. The game ends when the player either falls into a pit or grabs two gold coins as he arrives home. The agent's effectiveness is judged based on whether it can tag tiles correctly.*

The Inference Engine

The inference engine was implemented in Prolog, a logic programming language. The knowledge was represented as a collection of facts and rules. The facts recorded were formed as dynamic complex terms with one or two arity, usually denoting the position in the grid where it applies. Initially, the knowledge base starts empty and will only add the initial facts once the player starts playing. The initial facts are the home position and tiles inferred as safe, though unexplored. The inference was made through a base rule specified in the specifications.

Further rules are generated given the facts asserted in the knowledge base. The knowledge base continues to be built as the game progresses. New facts are asserted, which contain the type of tile the player landed on and its position. The engine guides the user by marking tiles depending on inferred and discovered safety. This aims to aid the player's decision-making in the game by presenting the inferences made. Inferred terms may also be retracted in the knowledge base, given new information.

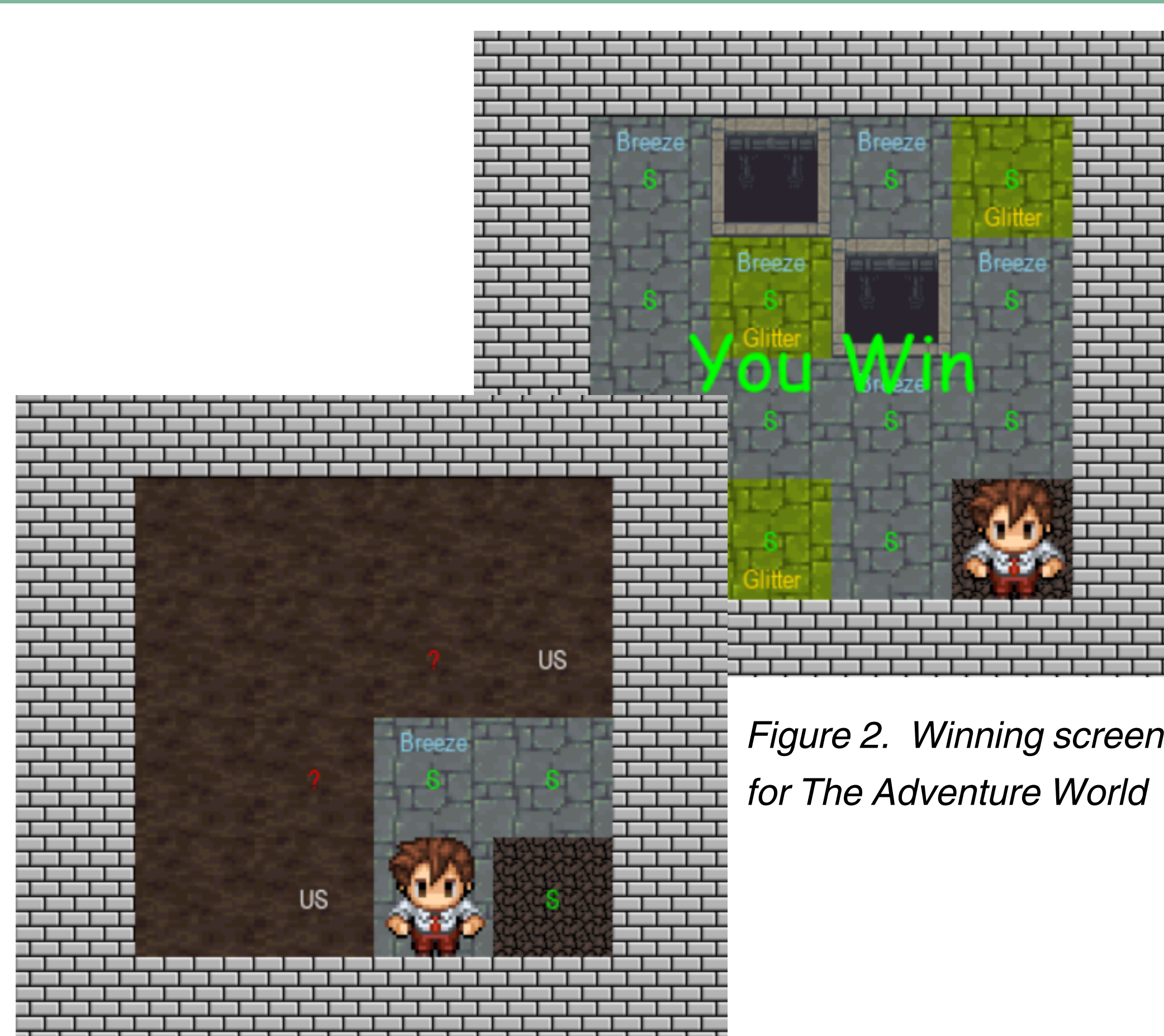


Figure 1. Cell tags as determined by the agent

Figure 2. Winning screen for The Adventure World

Evaluation and Performance

The Adventure World logic and GUI were tested and evaluated with five unique, hand-made maps. All five maps followed the specifications accordingly, with $n \times n$ tiles, $n-2$ pits, and $n-1$ gold pieces. A combination of 4×4 to 6×6 maps was tested, with edge cases added to evaluate the program's performance properly. Tests were done by hand on MS Paint, then done in Prolog to check, and then with the GUI to check a third time. This ensured that the program met the expectations of both the programmers and the specifications. Overall, the testing of the maps showed significant successes, with most of the features working as intended on the first map we tested. However, some minor setbacks were found with edge cases, which will be further explained in the challenges section.

Strenghts and Weaknesses

One strength of the program was its reasoning, which was made with the end goal in mind, allowing for many issues to be solved quickly or by simply rearranging the order of the program. The weaknesses of the program came from the edge cases that were found, giving the program a more challenging time in figuring out what it should tell the player. However, these issues were solved after a tedious evaluation.

Challenges

There were four main challenges that arose during the creation of the program. Figuring out the facts and rules, implementing the rules in prolog, creating the GUI, and connecting prolog and python. Figuring out the facts and the rules were done by brainstorming as a group and with continuous effort, the programmers were able to determine the rules as a group and were able to begin from there. Implementing the rules and creating the GUI went hand in hand, with each being given a framework to start, research was done for both python and prolog, allowing the programmers to create and implement the program properly. Lastly connecting the GUI was a challenge as it was a new concept to the programmers, but with enough research, trial and error, and working at it step by step, the programmers were able to overcome the challenges.

Conclusion

The Adventure World was a simple yet strenuous problem to solve. The world in itself requires little thinking to understand. However, it shows its complexity once you dive deeper into dissecting the knowledge and rules to represent it. The first step taken mainly was developing the knowledge space in the Prolog agent. This step also involved the most challenges. The biggest one is deciding how to accept new facts or derive given inferences based on rules. Decisions and revisions were made to ensure the agent would tag tiles as accurately as possible. And then came the GUI, which was implemented in Python using Pygame. Given proper research, the user interface was properly made. The challenges were mainly encountered in developing the connection between Prolog and Python. The UI was made to facilitate the tell-and-ask process while the agent handled logical reasoning. In the end, the program was able to guide the user throughout the game achieving the goal specified.