[[1]](#footnote-1)

The Enforcers: Reinforcement Learning in the Wumpus World Environment

Jordan Miller, Guangyu Nie, Abdul Rahman Zindani, Aakarsh Reddy Duvvuru

*Abstract*— This is a team project for Arizona State University’s Artificial Intelligence course 571. The project was to create a reinforcement learning agent to be deployed inside the Wumpus world. Reinforcement learning allows the agent to explore its environment and learn while executing actions. The agent can access prior knowledge about a state when faced with the same situation in a later episode. These situations are stored in q-table. The agent then can play the Wumpus Game and perform the best actions for each state due to playing the game before and learning from prior experiences.

*Index Terms*— Wumpus, Wumpus world, artificial intelligence, reinforcement learning

# INTRODUCTION

T

HIS project is using reinforcement learning in the Wumpus world environment. This was project number five listed on the project choices and the team had four members. The team choose this project to allow the team members to gain a new skillset for many of them were new to the topic area.

The Wumpus world is a simple problem used in artificial intelligence to examine how a knowledge-based agent will react in an environment. The goal of the game is for the agent to collect the gold then climb out of the cave while avoiding the Wumpus agent. If the agent enters a location where a Wumpus is, the agent will parish and the game will end. However, if the agent senses there is a Wumpus in that location before it enters it, the agent can shoot the Wumpus and the Wumpus will parish making that location safe for our agent. If the agent’s sensors were wrong and there was no Wumpus in that location, then our agent loses points from the overall score. The agent must also avoid falling into a pit while navigating the environment or else our agent will parish and the game ends.

The agent can navigate the environment with sensors. The sensors are able to detect if there is a pit in front of the agent, if there is a breeze, or stench. The breeze indicates there is gold nearby and the stretch indicates there is a Wumpus nearby. Due the way the environment is laid out, when the agent detects a breeze or stench that means that the gold or Wumpus can be in one of the adjacent squares to the north, south, left, or right. This is why shooting the arrow is essential for the agent to discover the location of the Wumpus.

Reinforcement learning allows the agent to explore its environment and learn while executing actions. It is assumed that the agent does not know the environment prior to starting. The benefit of this technique is that it allows the agent in the environment to access prior knowledge about a state when faced with the same situation in a later episode. These situations are stored in q-table. However, each time the agent learns in the environment there is a small probability that the action the agent is trying to take will not happen. This is accounted for in probabilistic action outcomes.

The agent is a q-learning agent that compares the expected utility values for each possible state from its location. The agent performs ***active learning*** that requires the agent to learn what actions to perform while playing the game. Q-learning agents are not able to look ahead which is why the table is so important to allow the agent to remember states and look up e­­­

In this project, the Wumpus world was simplified to make the problem manageable. The agent in the project cannot detect breezes and stenches. The agent can detect gold, pits, and the Wumpus in this scenario.

# technical Approach

## Environment

The environment was constructed to be simple. The basic environment consists of a 4x4 grid containing a Wumpus and 3 pits, and gold. The locations of these, as well as the starting location of the agent are randomized for every run.

The agent can move either vertically or horizontally, one location at a time. When the agent approaches a location adjacent to either a Wumpus or a pit, the agent may detect that the respective obstacle is nearby. The agent can use this information to move around the obstacle. For any action that the agent makes, there is 40% chance of a different, random decision being made instead of the optimal action.

Additionally, if the agent detects a Wumpus, the agent can choose to shoot the Wumpus with an arrow. This action would make the location safe for the agent to travel to as long as a pit does not exist in the safe location. In order to kill the Wumpus successfully, the agent must be facing the proper direction when shooting the arrow. The agent must also have at least one remaining arrow in order to kill the Wumpus.

The environment for the Wumpus world within the game can be viewed as a grid. “W” represents the Wumpus, “0” represents the pits, “G” represents the gold, and the “X” is the agent’s current location. All other locations are represented with a “.”.

## Rewards

There are two ways in which the agent could possibly be killed. The agent could either fall down a pit or be eaten by a Wumpus. For this to occur, the agent would have to travel to the location of a pit or a living Wumpus. The agent’s death ends the game and results in a reward of -1000 points.

When approaching a Wumpus, the agent may detect that the Wumpus is nearby and choose to shoot the Wumpus. The agent must aim an arrow towards the Wumpus in order to kill it. If the agent shoots an arrow and misses the Wumpus, the reward is -10 points. Alternatively, if the agent is successful in killing the Wumpus, the reward is +10 points.

Other than the death of the agent, the other way to end the game is to win by successfully finding the gold on the map and picking it up. By finding and retrieving the gold, the agent receives +100 points and successfully wins the game.

Moving in any direction or not moving at all will lead to a reward of -1 points for every turn (unless the above reward requirements are fulfilled).

## Q-Learning

The actions stored inside the q-table are moving up, moving down, moving left, moving right, shooting the arrow up, shooting the arrow down, shooting the arrow to the right, and shooting the arrow to the left. While the agent still has arrows, the agent may choose to shoot, otherwise, the agent will not shoot.

The q-table is a matrix that stores information regarding which actions should be performed depending on the agent’s state and sensory information. As the agent progresses through the game, if the agent encounters a situation that is stored in the q-table, it can use this information to select the best course of action. Once this action has been made and the reward for the action is received, the q-table will update to reflect this received reward.

If the current state of the agent does not exist within the q-table, then the state must be added. This is done by checking all the different actions that the agent could perform in the current state and determining which would provide the best reward. This information, along with the received reward is stored in the q-table for the next time the agent encounters this situation.

## Learning-Agent

The learning agent file was developed to look at states and add them to the q-learning table. If the state had not been visited, it was added to the table.

## Testing

## The test file was designed to run the tests for the reinforcement learning algorithm. The agent moved through and environment learning states and their rewards. The agent learns until the values converge. Once the agent is done learning, it plays the game with the best moves each location.

# Results

# Conclusion

# Team Effectiveness

All team members worked on this project. Writing of the paper was split between Jordan Miller and Abdul Zindani. Guangyu Nie developed the file ‘Q-learning.py’ and ran the tests for the project. Aakarash Reddy developed the files ‘test.py’ and ‘learning-agent.py’ in addition with adding parts to the ‘env.py’ file. Jordan Miller developed the ‘env.py’ file. All team members contributed to this project and communicated during the process to ensure this project was completed.

# References

1. Completed 01 May 2020. This work is a part of a course for Arizona State University graduate program.

   Jordan Miller [jlmill41@asu.edu](mailto:jlmill41@asu.edu)

   Guangyu Nie jeremy550959742@gmail.com

   Abdul Rahman Zindani azindani@asu.edu

   Aakarash Reddy Duvvuru aduvvuru@asu.edu [↑](#footnote-ref-1)