Reinforcement Learning agent in Wumpus world

(April 2020)

Sahit Jain; Arizona State University, Tempe, Arizona, USA; [sjain123@asu.edu](mailto:sjain123@asu.edu)

Nabeel Khan; Arizona State University, Tempe, Arizona, USA; [nakhan8@asu.edu](mailto:nakhan8@asu.edu)

Akshay Ganesa; Arizona State University, Tempe, Az, USA; [aganes14@asu.edu](mailto:aganes14@asu.edu)

Anantha Krishnan Kumar, Arizona State University, Az, USA; [akuma218@asu.edu](mailto:akuma218@asu.edu)

ABSTRACT

Reinforcement Learning agent in Wumpus world is a toy problem domain used in Artificial intelligence: a modern approach, Prentice hall, 2009 [1] to introduce the concept of logical agents. The agents that reason about the world with the tools of formal logic is an intuitive approach that underpinned much of the earlier research in artificial intelligence, however there are problems working and using formal logic as a base for rationality such as the frame problem. We propose to implement a reinforcement learning agent that can learn to reason about its environment. Proving that complex reasoning can be understood by a learning agent allows us to solve problems that are exceedingly difficult to describe in the language of formal logic.

CCS CONCEPTS

• Hybrid logic agent • Noisy Model   • Passive and Active Reinforcement Learning

KEYWORDS

Reinforcement Learning, Wumpus world, Artificial Intelligence, Markov Decision process

ASU Reference format:

Sahit Jain, Nabeel Khan, Anantha Krishnan Kumar and Akshay Ganesa. 2020. Reinforcement Learning agent in Wumpus world. In *Proceedings of CSE 571: Artificial Intelligence. ASU, Tempe, AZ, USA, 4 pages.*

1 Introduction

## The problem

One of the important questions to study machine learning is: “What kinds of functions can be understood efficiently from noisy, and data which is not perfect?” [2]

Reinforcement Learning, also known as a semi-supervised learning model in machine learning, is a technique to make an agent to be able to take actions and interconnect with an environment to maximize the total number of rewards. It is usually modeled as a (MDP) [Markov Decision Process](https://en.wikipedia.org/wiki/Markov_decision_process).

A close up of a logo

Description automatically generatedThe Wumpus world is a well-known toy problem in artificial intelligence made popular by the reference book of Russell and Norvig [1]. We assume that the time is converted into discrete, and that the agent can take only one action per time step. It can move the grid by moving in the four cardinal directions. For the project, the Wumpus world was our standard problem for studying reinforcement learning algorithms. Also, these algorithms are standard to be applied to different problems environment. We assume that each action has a possibility to be noisy in plan traces. [3]

**2.** **Technical Approach**

One of the most interesting application can be detected to see the discrepancy when sequential data are on radar. [4] In the Reinforcement Learning framework, we define an environment, which highlights the information that can be used by the agent that allow to take some actions. In the Wumpus world, environment is semi observable as locations of the holes, the Wumpus and the treasure are not known to the agent. Though, the environment provides some signals to the agent:

* The agent is adjacent to Wumpus, receives a smell signal
* The agent is adjacent to the hole, receives a breeze signal
* The location of the agent deterministically determined by the initial position and the actions

At every time step, agent can select one of 8 different actions:

* Up, Down, Left, Right, FlashUp, FlashDown, FlashLeft, FlashRight.
* Agent will perceive the **stench** if he is in the room adjacent to the Wumpus. (Not diagonally).
* Agent will perceive **breeze** if he is in the room directly adjacent to the Pit.
* Agent will perceive the **glitter** in the room where the gold is present.
* The agent will perceive the **bump** if walks into a wall.
* When the Wumpus is shot, it emits a horrible **scream** which can be perceived anywhere across the cave.
* The percepts are represented as five element list, which will have different indicators for each sensor.
* Example if agent perceives stench, breeze, glitter, but no bump, and no scream then it can be represented as:  
  **[Stench, Breeze, Glitter, None, None]**.

**Characterization:**

* **Partially Observable:** knows only local perceptions
* **Deterministic:** outcome is precisely specified
* **Sequential:** subsequent level of actions
* **Static:** Wumpus, pits are immobile
* **Discrete:** discrete environment
* **Single-agent:** knowledge-based only agent

**PEAS Description** **for the Wumpus World**:

* Performance measures:

Agent gets the gold. It returns safe. It dies. It uses arrow at each move of the agent.

* Environment:

A cave with *16(4×4)* rooms, adjacent not diagonally to the Wumpus are stinking, adjacent not diagonally to the pit are breezy

* Actuators:

They allow to perform some actions such as Move, forward, turn right, Shoot, Grab, Turn left, and Release

* Sensors:

They help the agent in sensing from the environment.

**3. Results**

We presented a new approach for learning programs from noisy datasets of arbitrary size. We instantiated our approach to two important noise settings: the setting where we can place a bound on the noise and the setting where the dataset contains unbounded noise. We showed that the second setting leads to a new way of performing approximate empirical risk minimization over hypotheses classes formed by discrete search spaces. We then illustrated how to instantiate the different noise settings for building practical synthesizers that can deal with noisy datasets. [5]

Here we assume that the first 90 games that the agent plays are very random and after that it uses the reinforcement learning to make a proper decision.

Uses an NXN grid with

1. Agent
2. Wumpus
3. 3 pits
4. Treasure/Gold
5. Agent has 3 bullets initially.

Input contains Number of games and Number of iterations:

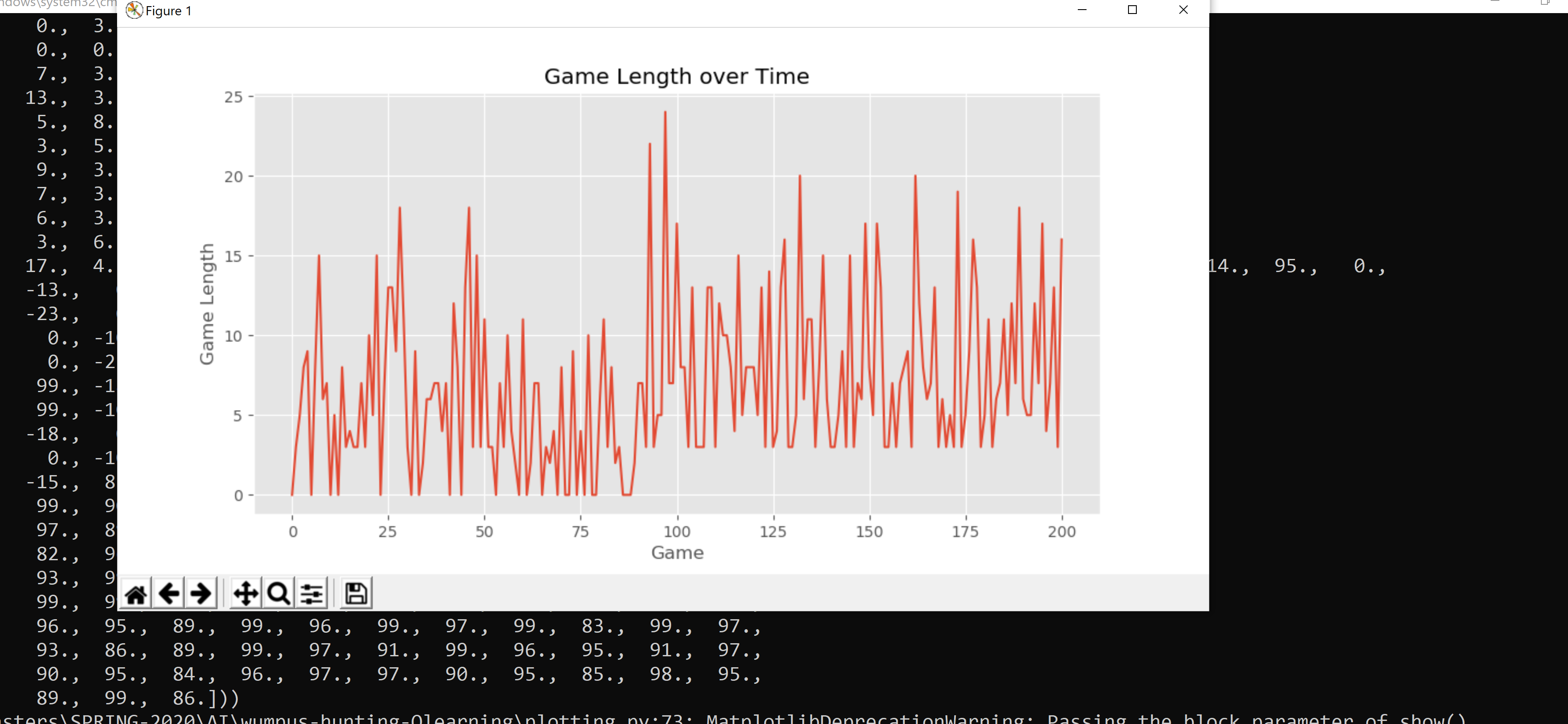
python environment.py --ngames 150 --niter 40 --gridsize 4 4 --numwumpus 1 --numholes 3 --bullets 3

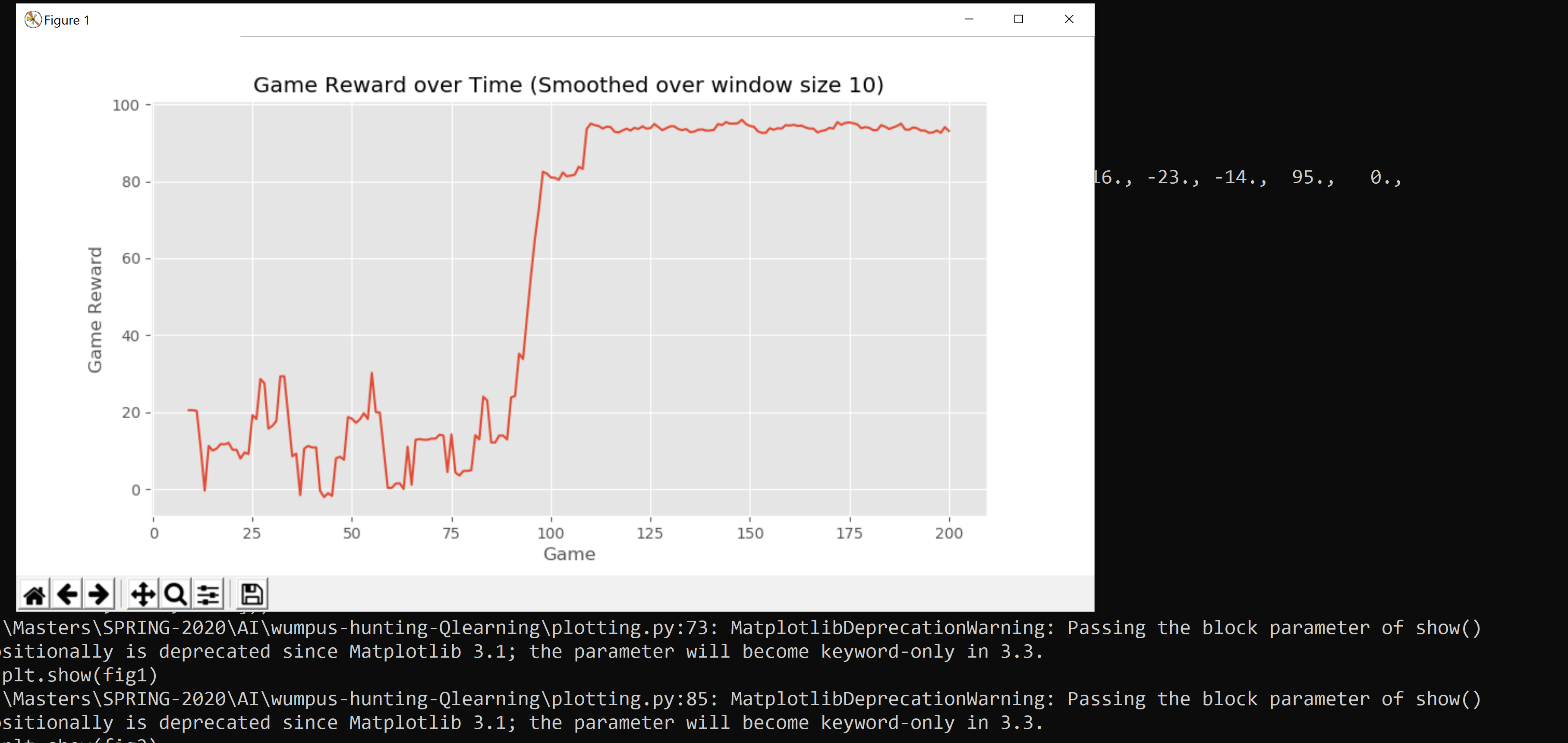
The cumulative reward obtained is 10984.0

On an average considering that we have played 125 games. Average reward per game would be 87.872.

Game length indicates the number of steps that it took to end the game.

End of game includes falling into the pit, getting swallowed by Wumpus or capturing the gold.





The grid structure for the current scenario is(4x4).

|  |  |  |  |
| --- | --- | --- | --- |
| Hole |  |  |  |
|  |  | Agent | Hole |
|  |  | Gold |  |
|  |  | Hole | Wumpus |

We can observe that once the initial 90 games are done the agents reward for every subsequent is approximately equal its previous and next games.

The reward depends on the position of gold and the Agent. In general, we can observe that the reward obtained is high once 90 games are finished.

In our situation

* Treasure found: reward +100, game ends
* kill the Wumpus: reward +1
* Wumpus catches: reward −10 and game ends
* Fall into a hole: reward −10 and game ends.
* Nothing happens reward −1

It seems that convergence is happening after around 100 games. Execution command:

python environment.py --ngames 150 --niter 40 --gridsize 4 4 --numwumpus 1 --numholes 3 --bullets 3

None of the arguments are compulsory.

* ngames indicates- number of games
* niter indicates- number of iterations
* gridsize 4 4 indicates rowsize columnsize
* numwumpus indicates- Wumpus in grid
* numholes indicate number of holes in the grid
* bullets indicate bullets that agent possesses

Implemented the hybrid logic agent utilizing most of the individual project 3. Implemented in wumpus\_kb.py and wumpus\_planners.py. Added noise to the model by randomly taking a TurnLeft or TurnRight action 10 % of the time for every Forward action. Implemented in wumpus\_agent.py. Also made a few custom layouts to test the agent. (cust1.lay and cust2.lay)

Cust2: 974 (Without noise)

(90-10 noise)

*A screenshot of a cell phone

Description automatically generated*

A screenshot of a cell phone

Description automatically generated

**4.** **Conclusions**

We have presented an overview of Reinforcement Learning agent in Wumpus world that uses a series of noise control based in Wumpus World to implement reinforcement learning and noisy logic control. The advantages of this project include: a fun and simple software framework for team projects, an importance given on methods rather than learning a framework for each, and a hands-on approach to AI that requires reflection on why different AI approaches work.

5) **Team effectiveness**

|  |
| --- |
| **Members contribution** |

**Sahit Jain-**

1. Noisy logic Agent
2. Learning prerequisites for design
3. Studied and explored about implementation

**Anantha Krishnan Kumar-**

1. Reinforcement Learning
2. Learning prerequisites for design
3. Studied and explored about implementation

**Akshay Ganesa-**

1. Reinforcement Learning
2. Learning prerequisites for design
3. Studies and explored about implementation

**Nabeel Khan-**

* 1. Exploring and familiarizing with repository
  2. Create basic project structure for design
  3. Created report for the submission

**ACKNOWLEDGMENTS**

We would like to start by thanking Professor Yu Zhang for giving this opportunity to work on this interesting work. We got to learn a lot about how the Reinforcement Learning agent in Wumpus world works. We also gained knowledge about the steps, strategies, and patterns in this field. We would like to continue this research ahead and use the knowledge we have gained in a practical setting and industry work.

# **References**

|  |  |
| --- | --- |
| [1] | S. R. a. P. Norvig, Artificial intelligence: a modern approach, Prentice hall, 2009. |
| [2] | A. K. A. H. W. AVRIM BLUM, "Noise-Tolerant Learning, the Parity Problem, and the statistical Query Model," *ACM,* vol. 50, no. 4, p. 14, 2003. |
| [3] | H. H. Z. a. S. Kambhampati, "Action-Model Acquisition from Noisy Plan Traces," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. |
| [4] | G. I. Min-hwan Oh, "Sequential Anomaly Detection using Inverse Reinforcement," *ACM,* p. 11, 2019. |
| [5] | P. B. M. V. A. K. Veselin Raychev, "Learning Programs from Noisy Data," *ACM,* p. 14, 2016. |
| [6] | A. K. A. H. W. AVRIM BLUM, "Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model," *ACM,* vol. 50, no. 4, p. 14, 2003. |

Conference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00