Workshop 3: Wumpus World Agent

Cesar Augusto Pulido Cuervo - 20222020048

Cristian David Romero Gil - 20222020138

Universidad Distrital Francisco José de Caldas

Workshop 3: Wumpus World Agent

Machine Learning Implementation

We propose the development of a Deep Q-Network (DQN)-based agent to solve a partially observable and hazardous environment inspired by the Wumpus World. The agent receives structured observations that include position, perceptual inputs (breeze, stench, glitter), orientation, and two chaos-aware features—entropy and a Lyapunov-inspired parameter—capturing behavioral uncertainty.

The learning architecture leverages PyTorch to define a multi-layered Q-network. The input vector has eight features and outputs Q-values for six discrete actions. These include movement (four directions), grabbing gold, and shooting an arrow. The environment is implemented using Gymnasium, providing a granular reward system was designed to reinforce adaptive behavior:

- Penalties are assigned for entering dangerous zones or taking unnecesary steps.
- Positive rewards are given for collecting gold and returning safely.
- Additional shaping rewards are included to encourage safe and exploratory actions.

The entire agent-environment system is compatible with vectorized environments and experience replay, making it suitable for scalable reinforcement learning experimentation.

A structured reward system is defined to reinforce intelligent behavior and penalize inefficiency or failure. Rewards and penalties are assigned for collecting gold, dying, using the arrow, or moving without purpose. This setup allows the agent to learn optimal strategies through trial and error over multiple episodes.

Event / Action	**Reward**	**Rationale**
Move to empty cell	-1	Penalizes unnecessary movement
Fall into a pit	-1000	Severe penalty for death
Encounter the Wumpus	-1000	Severe penalty for death
Grab the gold	$+1000$	Key reward for success
Return to start with gold	$+2000$	Final reward for goal completion
Shoot and kill Wumpus	$+500$	Incentivizes strategic aggression
Shoot and miss	-50	Penalizes wasted action
Redundant action / loop	-1	Discourages inefficient behavior

Cybernetic Feedback Loop Integration

The agent is governed by a feedback-based architecture inspired by cybernetic principles. At each time step, the agent receives perceptual input, selects an action based on its policy, receives a reward, and updates its model. This perception–action–evaluation–adjustment cycle forms a classic closed-loop control system.

To make this feedback loop more expressive, we introduce two chaos-related metrics into the agent's observation space:

- *Entropy*, measuring uncertainty in action selection.
- A $Lyapunov-inspired\ variable$, representing internal instability over time.

Rather than calculating a true Lyapunov exponent—which is complex and unnecessary for our scale—we simulate sensitivity to prior decisions using a logistic map Strogatz (2015):

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n) \tag{1}$$

This simple non-linear equation is well known for producing complex, chaotic

behavior depending on the value of r. When $r \in (3.57, 4)$, the system becomes highly sensitive to initial conditions and generates unpredictable—but deterministic—sequences.

In this project, we set $r = 3.89$ because it places the system firmly in the chaotic regime, introducing a controlled form of pseudo-randomness. This allows the agent to model internal fluctuations, helping simulate unstable states such as exploration uncertainty or learning volatility without external noise Strogatz (2015).

The variable derived from this map serves as a synthetic signal representing the agent's behavioral "instability." It may be used to modulate learning dynamics or exploration rate and is intended to capture the idea of nonlinear adaptation inspired by Lyapunov instability, even though it is not a true Lyapunov exponent. Sprott (2003).

This addition enhances the feedback loop by embedding a form of chaotic self-regulation into the agent's internal state, aligning with cybernetic principles of adaptive control in complex environments.

Agent Testing and Evaluation

We have a test environment specifically designed for evaluating the agent in the Wumpus World environment. To thoroughly assess the agent's behavior and learning performance, we define a set of test case scenarios, each designed to challenge different aspects of the agent's decision-making and adaptability. These scenarios vary in complexity, layout, and pits distribution, providing a primary benchmark for evaluating success rate and learning efficiency.

The test case scenarios are:

1. Default random placement

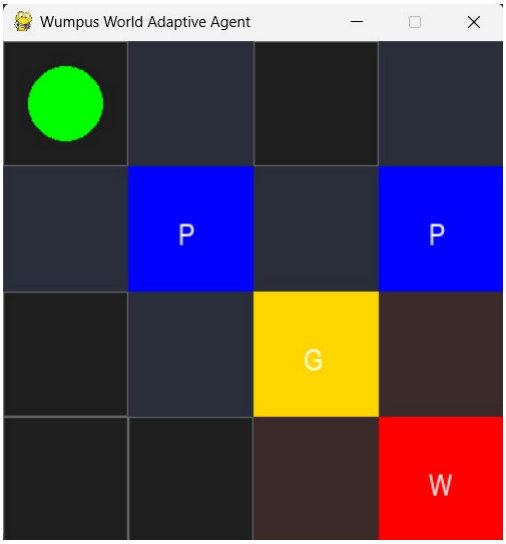

Figure 1. Scenario 1

2. Pits and Wumpus in corners

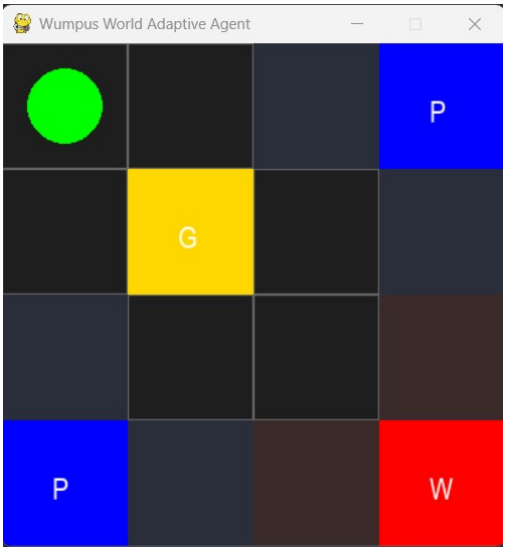

Figure 2. Scenario 2

3. All elements together

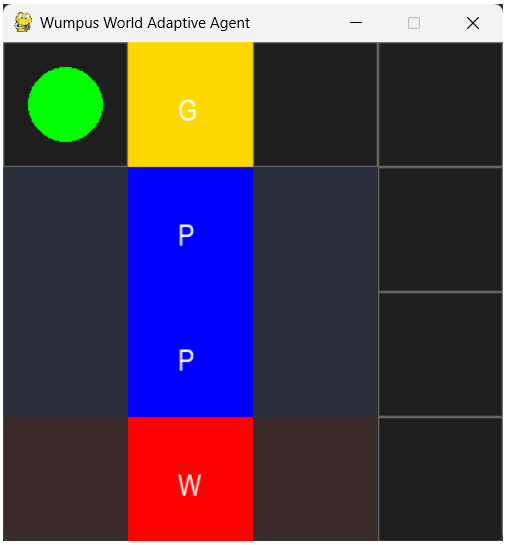

Figure 3. Scenario 3

4. Random placement in a 6x6 matrix

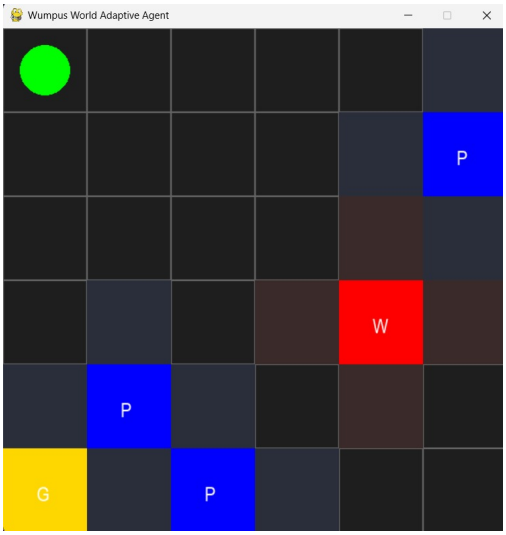

Figure 4. Scenario 4

We compare the agent's performance using two key evaluation metrics: average reward and victory rate, both measured every 5 episodes. These metrics provide complementary insights into the learning progress and strategic effectiveness of the agent.

Metrics comparative

- **Average reward** reflects the cumulative score obtained by the agent during each episode, accounting for movement costs, penalties, and rewards. By calculating the average over 5-episode intervals, we can smooth out noisy fluctuations and observe the general trend of learning stability and efficiency over time.
- **Victory rate** measures the proportion of episodes in which the agent successfully completes the goal each 5-episode window. This metric captures the agent's ability to consistently succeed in the task, offering a clearer picture of strategic mastery and goal achievement. In this first version, we do not include metrics related to the dynamic movement of the Wumpus.

Scenario 1.

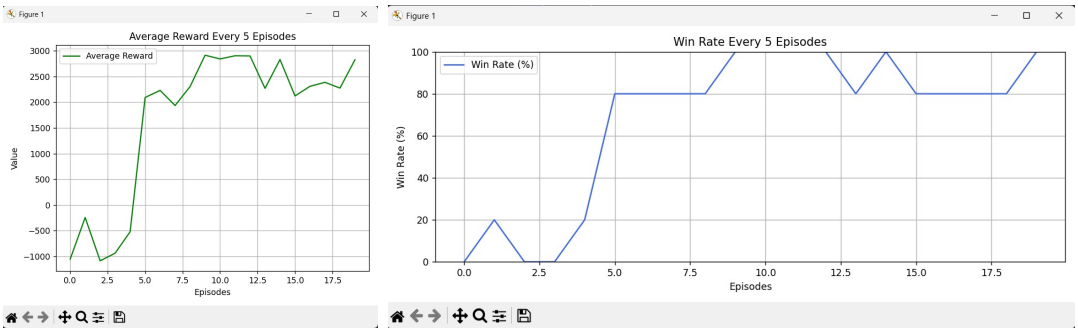

Figure 5. Scenario 1 metrics

In this scenario, the agent demonstrates a fast learning convergence around episode 25, progressively stabilizing its behavior as it accumulates experience.

Scenario 2.

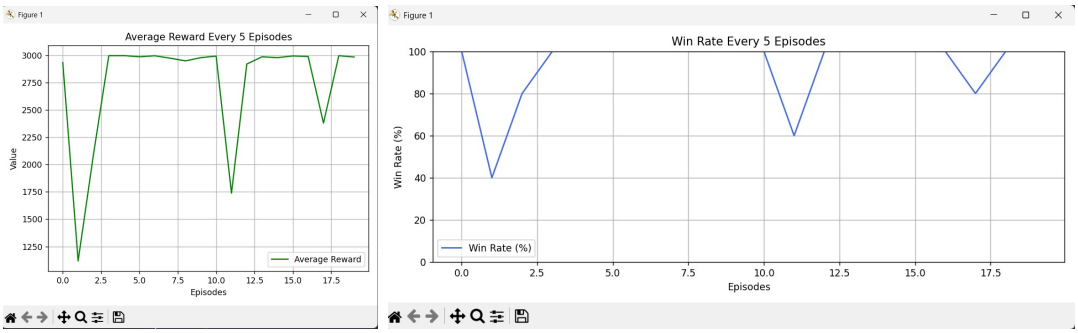

Figure 6. Scenario 2 metrics

The agent achieves rapid convergence within the first episodes due to the proximity of the gold to the starting position, which simplifies initial learning.

Scenario 3.

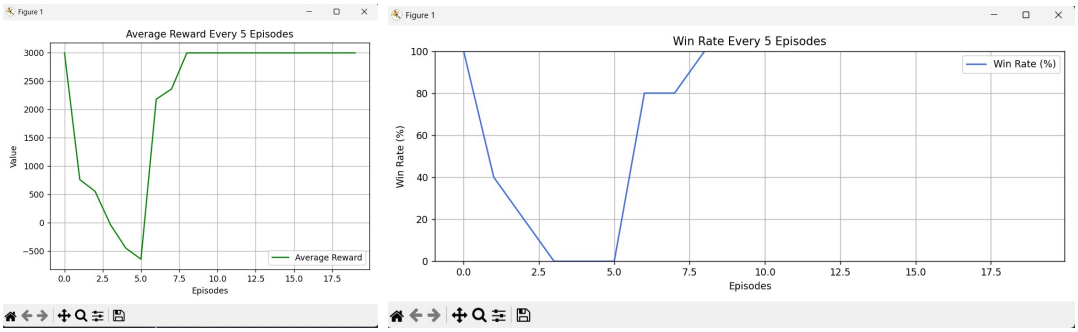

Figure 7. Scenario 3 metrics

In this particular scenario, the agent shows an initial phase with relatively easy wins due to the short distance to the gold, followed by stable convergence around episode 30 as it refines its policy.

Scenario 4.

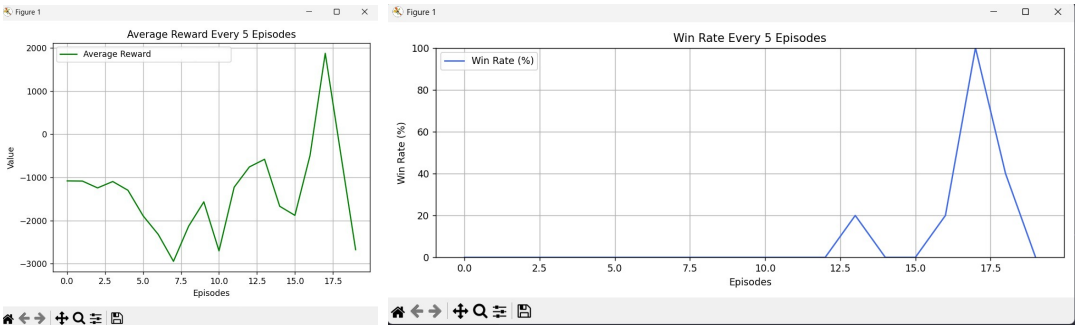

Figure 8. Scenario 4 metrics

In this scenario, the agent struggles to achieve convergence within the current number of episodes due to the increased complexity of the 6x6 environment, requiring extended training (more of 100 episodes) for policy stabilization.

Key Points

While the current implementation demonstrates acceptable performance across multiple scenarios, there are multiple aspects that can be refined to enhance both learning efficiency and system maintainability. First, fine-tuning the reward structure may lead to faster convergence in certain environments where the current incentives are not sufficiently informative or balanced. Adjusting the reward signals can help the agent better prioritize its decisions and reduce unnecessary exploration in well-learned situations. Also testing epsilon constraints for more exploration is another way to improve the agent.

Future revisions of the code should focus on improving clarity, modularity, and readability. A more organized architecture will facilitate future modifications, debugging, and the integration of additional features. Although the present work operates with a static Wumpus in most scenarios, future deliverables should incorporate dynamic elements where the Wumpus moves independently within the environment. Introducing these system dynamics will significantly increase the complexity of the problem and provide a more realistic and challenging testbed for evaluating the agent's adaptive capabilities.

References

Sprott, J. C. (2003). *Chaos and time-series analysis*. Oxford University Press.

Strogatz, S. H. (2015). *Nonlinear dynamics and chaos: With applications to physics, biology, chemistry, and engineering*. CRC Press.