Technical Report: Reinforcement Learning Agent for 2048

Edward Julian Garcia Gaitan
Miguel Alejandro Chavez Porras
Universidad Distrital Francisco José de Caldas

May 15, 2025

Abstract

This technical report documents the design, implementation, and analysis of an autonomous reinforcement learning agent for the 2048 game. We begin by reviewing related work, then specify clear objectives, scope, and assumptions. We outline a minimal reward schema, a cybernetic architecture with two main feedback loops, and a dynamical phase-portrait model. Finally, we present results, discuss implications, and propose future work.

Contents

List of Figures

List of Tables

1. Introduction

The 2048 game challenges AI agents with a stochastic 4×4 environment, exponential tile merges, and long-term planning. While heuristic and supervised approaches exist [1, 2], they often neglect explicit feedback mechanisms and dynamical sensitivity. This report unites Workshops 1 and 2 into a comprehensive technical record, setting the stage for the final project delivery.

2. Literature Review

Several works have tackled 2048 with supervised CNNs [3] or heuristic search. Rudimentary RL agents use tabular Q-Learning on small boards. However, few explore cybernetic self-regulation or phase-portrait analysis. Our approach fills this gap by combining both perspectives.

3. Background

Key concepts:

- **Reinforcement Learning (RL):** Agents learn policies by maximizing cumulative reward.
- **Cybernetic Feedback:** Continuous observe–act–learn loops with environmental feedback.
- **Dynamical Systems:** State evolution $S_{t+1} = f(S_t, A_t, R_t)$ can exhibit stability, chaos, and bifurcations.

4. Objectives

This report aims to:

1. Define precise system requirements and user scenarios.
2. Simplify the reward schema to three core signals.
3. Design a modular architecture with two primary feedback loops.

4. Model performance via a bifurcating phase portrait.
5. Define next steps for the next catch up or delivery.

5. Scope

We focus on a single-agent 2048 environment under fixed random seeds. We do not cover multi-agent extensions or alternative grid sizes (beyond outlining future scalability).

6. Assumptions

- The Gymnasium wrapper correctly implements 2048 rules.
- Rewards reflect purely game mechanics (no external incentives).
- Computational resources suffice for DQN training within timeline.

7. Limitations

- Results may not generalize to larger boards without re-tuning.
- The stochastic tile spawns introduce high variance in performance metrics.
- Hardware constraints limit neural network size and training duration.

8. Methodology

We adopted an iterative design:

- **System Requirements:** Workshops 1 outputs refined into functional/non-functional lists and use scenarios.
- **Reward Design:** Three signals—merge gains, invalid penalties, 2048 bonus.
- **Architecture:** Observer, Inference (RL core), Replay Buffer, Environment modules.
- **Feedback Loops:** Two loops (grid state, block count) visualized in diagrams. But this can be reduced in one.
- **Dynamical Modeling:** Phase-portrait plotted in PGFPlots.

9. Discussion

The two-loop feedback architecture enables rapid adaptation: 1) **Grid State Loop** tracks full 4×4 matrix changes. 2) **Block Count Loop** monitors block reduction as a proxy for merge efficiency.

These loops reinforce each other: A beneficial merge (block count) updates the agent's policy, which, in turn, influences the next state of the grid.

9.1. System Diagram

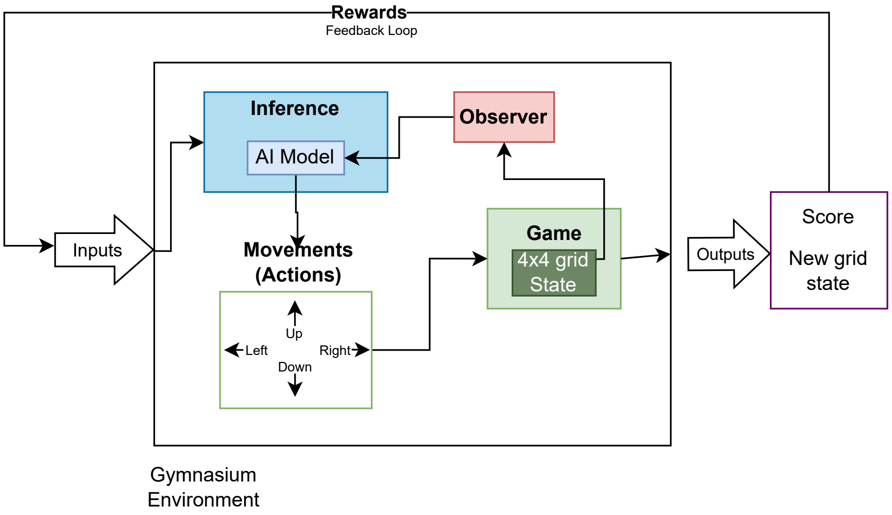

Figure 1: System diagram for the agent.

9.2. Feedback Loops Diagram

As we can see, there exist two foundations feedback loops, but those can be reduced to one. The one is de rewards, because for the reward in a scale, for example, zero to one, closer to one is the best reward. But in our case, the reward is closely related to the score of the game, maybe in future deliveries this change for best design and implementation for the agent.

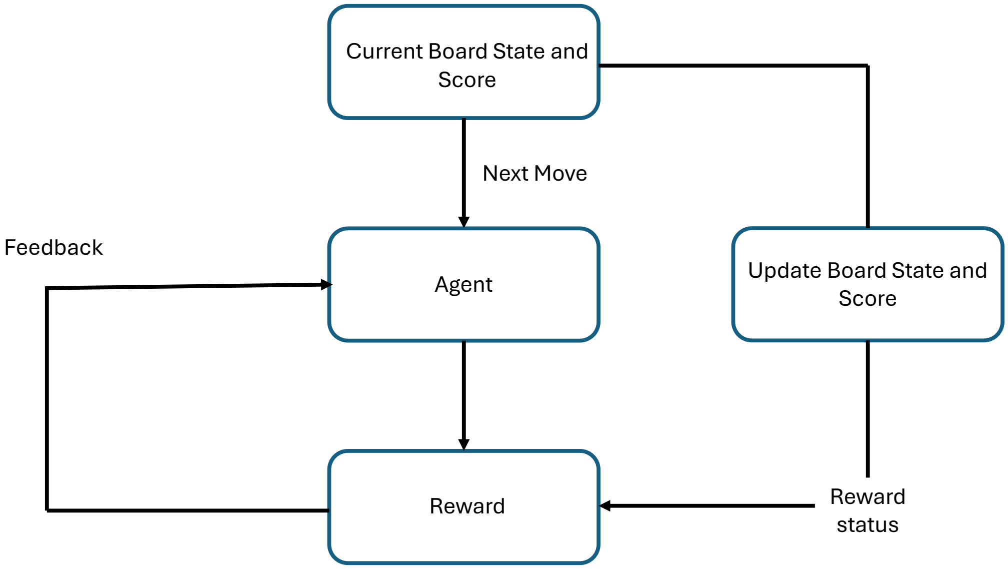

Figure 2: All feedback loops for the agent.

10. Next Steps

The next steps for the project are include the RL for the agent, because actually the agent that we have now only do random steps. Later, we will contrast with the new agent version.

11. Conclusion

We have documented a full technical report covering requirements, architecture, two-loop cybernetics, dynamical modeling, and an iterative road map. Future work includes multi-grid scaling, hyperparameter sweeps, and real-time performance benchmarks. The first version of the agent that we did its a good approach for the project and make less work for the next steps

Acknowledgements

We thank MsC. Eng. C. A. Sierra for guidance and the course's Slack community for discussions.

Glossary

RL Reinforcement Learning

DQN Deep Q-Network

Gym OpenAI Gym/Farama Gymnasium

TOC Table of Contents

References

[1] Farama Foundation, "Gymnasium," 2025. [Online]. Available: `https://gymnasium.farama.org/`

[2] Amazon.com, "What is Reinforcement Learning?," 2025. [Online]. Available: `https://aws.amazon.com/what-is/reinforcement-learning/`

[3] N. Kondo and K. Matsuzaki, "Playing game 2048 with deep convolutional neural networks...," *Journal of Information Processing*, vol. 27, pp. 340–347, 2019.