RL for Tower Defense with Evolutionary Towers

Group 12
Andrew Wallace - 101210291 - andrewwallace3@cmail.carleton.ca
Mohammad Rehman - 101220514 - mohammadrehman@cmail.carleton.ca
Manal Hassan - 101263813 - manalhassa@cmail.carleton.ca
Derrick Zhang - 101232374 - derrickzhang@cmail.carleton.ca

Problem Statement

We aim to develop and train a reinforcement learning agent to play an evolving tower defense game where strategic resource allocation and adaptive tower placement are critical for success. Unlike traditional tower defense games with manual upgrades, our environment features towers that automatically evolve and increase in strength based on the number of enemies they eliminate, creating a feedback loop between decisions and long-term strategy when it comes to positioning.

The agent must learn to place two types of towers: single target and area-of-effect. The environment will be grid-based with limited budget constraints per wave. As enemy waves progress with increasing health, the agent must balance immediate defensive needs with long-term tower development, deciding not only where to place towers but also which towers to prioritize for enemy elimination to maximize evolutionary potential.

This problem presents a few challenges: immediate survival against long-term tower development, spatial reasoning for optimal tower placement on a constrained grid, resource allocation under budget constraints, and adaptation to increasingly difficult enemy waves.

The evolving tower mechanism introduces a novel aspect where early decisions significantly impact later capabilities, making this an interesting environment for a reinforcement learning algorithms that must learn both immediate responses and long-term strategic planning.

Feasibility

Reinforcement learning (RL) is well-suited for this problem because the environment is stochastic, sequential, and dynamic, with an agent's actions influencing future states.

As the enemy waves increase in difficulty, the agent must adapt in terms of where to place towers, what type of towers to place, and their strategy to maximize their chances of survival. The sequential relationship of agent action and future success is a delayed reward optimization problem that RL methods, such as Q-learning, specialize in, making RL a good fit for this problem. [2]

Furthermore, the presence of two tower types, budget constraints, enemy health, and a grid-based environment creates a large and dynamic state space, making traditional machine learning techniques infeasible. This further supports the case for reinforcement learning. [3]

On the other hand, RL alone may be infeasible. Similar work in a grid-based tower defense game has shown that, "A fully-fledged RL agent, trained to select one of the units and place it wherever desired, is hardly feasible in this case" [4]. As such, techniques beyond reinforcement learning alone may be considered in this project.

Lastly, the evolving tower mechanism further complicates the balance of immediate survival and long-term tower development, which is a direct RL use case.

Overall, RL is well-suited for this problem due to its ability to handle delayed rewards, dynamic environments, and complex sequential decision-making, all of which are essential in a successful long-term tower defense strategy.

Milestones

Initial Environment Setup (September 29 – October 13)

- Implement basic gym environment
- Submit bi-weekly progress report due (Oct 15)

Final Environment Implementation (October 14 – October 27)

- Complete custom gym environment
- Test environment with random agent
- Submit environment demo video (10 minutes)

Training and Initial Results (October 28 – November 10)

- Research relevant RL algorithms for our use case
- Start implementing RL algorithms
- Submit second bi-weekly progress report (Oct 30)

Final Algorithm Implementation (November 11 – November 24)

- Have a working RL algorithms
- Gather initial training results
- Submit third bi-weekly progress report (Nov 15)

Result Demo (November 25 – December 1)

- Gather and complete the experimental results
- Different algorithm comparison analysis
- Fourth bi-weekly progress report (Nov 30)
- Submit demo video (Dec 1)

Final Report Writing (December 2 – December 7)

- Every group member to write a section of the report
- Gather and review the final report for final draft
- Submit final report (Dec 8)

References

[1] Sindhu Padakandla. 2021. A Survey of Reinforcement Learning Algorithms for Dynamically Varying Environments. ACM Comput. Surv. 54, 6, Article 127 (July 2021), 25 pages. https://doi.org/10.1145/3459991

[2] J. Clifton, E. Laber. (2020). Q-Learning: Theory and Applications. [ONLINE]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-statistics-031219-041220

[3] Dias, A., Foleiss, J., Lopes, R.P. (2022). Reinforcement Learning in Tower Defense. In: Barbedo, I., Barroso, B., Legerén, B., Roque, L., Sousa, J.P. (eds) Videogame Sciences and Arts. VJ 2020. Communications in Computer and Information Science, vol 1531. Springer, Cham. https://doi.org/10.1007/978-3-030-95305-8_10

[4] J. Bergdahl, A. Sestini and L. Gisslén, "Reinforcement Learning for High-Level Strategic Control in Tower Defense Games," 2024 IEEE Conference on Games (CoG), Milan, Italy, 2024, pp. 1-8, doi: 10.1109/CoG60054.2024.10645621

[5] Gym library. https://gym.openai.com/. Accessed September 20 2025