# Mid-Status Report: AI-Powered Tower Defense

CS6660 Introduction to AI - Term Project

Authors: Islam Murtazaev, Leonel Mainsah Ngu, Raymond Frimpong Amoateng

## 1. Research Question

How can AI techniques (A pathfinding and reinforcement learning) be used to optimize defender deployment strategies in a tower defense game?

* **Evolution**: Our initial focus on A\*-informed heuristics pivoted to a hybrid approach: *A pathfinding*\* for intelligent soldier movement, and **Reinforcement Learning (RL)** for optimal strategic placement. This combination offers a more robust, generalizable, and adaptable solution for dynamic environments.

## 2. Introduction & Problem Importance

Tower defense games present complex challenges in spatial reasoning and resource allocation. Our project, **"Winterfell Tower Defense,"** requires an AI agent to place 10 mobile defenders (footmen and archers) to survive 5 waves of 300 enemies. Unlike traditional tower defense, our units are **mobile defenders** that use A\* to actively pursue enemies, making the placement strategy critical and highly dynamic.

The problem's importance lies in optimizing decision-making in real-time strategy environments. We leverage RL to learn placement strategies that outperform hand-crafted heuristics, providing a superior solution for coordinating mobile units in a complex, dynamic state space.

## 3. Related Literature & Approach

Our methodology is grounded in recent hybrid AI research:

* Jaramillo-Martínez et al. (2024) validated that RL algorithms can achieve superior efficiency in path planning compared to classic A\*, supporting our pivot to an RL-based deployment strategy.  
  (Reference: <https://www.mdpi.com/2076-3417/14/17/7654> )
* Pfeiffer et al. (2018) showed that combining learning with prior knowledge significantly improves sample efficiency, which informed our decision to simplify the reward function for our PPO agent.  
  (Reference:<https://arxiv.org/abs/1805.07095> )
* Liu et al. (2021) (mini-AlphaStar) demonstrated RL's capability in complex real-time strategy games with large state spaces, validating our choice of RL for a game environment defined by a 32x32 grid.  
  (Reference: <https://www.emergentmind.com/papers/2104.06890> )

**AI Techniques Used So Far:**

* **A\* Pathfinding:** Implemented for soldier navigation (Objective 1).
* **PPO (Proximal Policy Optimization):** Used for placement strategy learning (Objective 2).
* **Gymnasium Environment:** Custom environment for PPO training, using a flattened **1,027-value Observation Space** and a **2,048-action Discrete Space**.

**How Our Work Differs:** Our approach uniquely combines A\* pathfinding with RL for **mobile units**. Previous RL tower defense work largely assumed static units. We are learning placement strategies that anticipate the dynamic movement of intelligent, pathfinding soldiers, not just static towers.

## 4. Development Results: Achievements, Challenges, and Workarounds

### Achievements

**A\* Implementation:** Successfully integrated A\* into soldier AI with a 20-pixel grid resolution.

**Strategic Placement:** RL agent discovers complex strategies like creating overlapping detection zones and balancing unit composition.

**Quantitative Evaluation:** Established clear performance metrics:

**Random Policy Baseline**: 0-5% win rate.

**Well-Trained Agent (1M+ steps)**: 40-60% win rate, +100 to +400 average reward.

### Challenges and Workarounds

* **Training Speed:** Initial slowness was overcome by implementing **fast simulation mode** and **parallel environments**, resulting in a **20-40x speedup**.
* **Observation/Action Complexity:** Flattened the observation space to a single 1,027-value array and tuned PPO's entropy bonus to encourage efficient exploration of the 2,048-action space.
* **Reward Shaping:** Refined the reward structure (e.g., +10 per enemy killed, -5 per castle damage, +500 victory bonus) to provide clearer learning signals for the agent.
* **Methodology Pivot:** Successfully transitioned the placement logic from a hand-crafted heuristic to the more effective PPO-based RL learning.

## Next Steps

* Continue training agents to 2M+ timesteps to achieve higher win rates.
* Experiment with different reward structures and hyperparameters.
* Analyze learned placement patterns to extract strategic insights.
* Compare with alternative RL algorithms (A2C, DQN).
* Document final results and prepare comprehensive evaluation.

## 6. GitHub Repository

**Repository Link**: <https://github.com/IslaMurtazaev/tower-defense-ai-agent>