Title of the project: Elastic Monte Carlo Tree Search for *Kill the King* and *XO game*

Group - G14

Members

Ganga Snehith	-S20230010087	-SEC1
Mandapati Deekshith Kumar	-S20230010142	-SEC2
Kurnoothala Rajitha	-S20230010133	-SEC3

Abstract:

This project addresses the challenge of efficient decision-making in strategy games by proposing Elastic Monte Carlo Tree Search (MCTS), a method that dynamically applies and removes state abstraction to improve AI performance. The approach leverages approximate MDP homomorphism to group similar game states, implements unit ordering to reduce action space complexity in multi-agent environments, and applies abstraction for a controlled number of iterations (α_ABS) before reverting to standard search. This elastic behavior balances search efficiency with decision quality.

The method was evaluated by implementing three MCTS variants—Standard, Random Grouping, and Elastic—in the "Kill The King" and “XO” strategy games. Testing involved various unit compositions and board configurations through command-line simulations, with results measured in terms of win rates, node counts, and compression rates. Graphs illustrated reduced node exploration, consistent action space complexity, the evolving ratio of ground to abstract nodes, and overall algorithm efficiency. Elastic MCTS outperformed the other variants with compression rates up to 161:1 while maintaining decision quality, thus demonstrating a domain-independent and scalable solution for handling increasing complexity in strategy games.

Plane of Work:

- **Objective:**
  Develop and evaluate Elastic MCTS for strategy games by implementing automatic state abstraction without the need for domain-specific knowledge, aiming to reduce the search space while preserving decision quality.
- **Step 1: Algorithm Analysis**
  - Study the principles of MCTS and approximate MDP homomorphism for state abstraction.
  - Identify challenges in applying state abstraction within the combinatorial action spaces of strategy games.

    - Develop the concept of “elastic” abstraction that dynamically applies and removes groupings.

- **Step 2: Algorithm Implementation & Evaluation**
    - **Implementation:**
        - Develop three MCTS variants (Standard, Random Grouping, and Elastic) within the Kill The King (KTK) and XO game environment.
        - Build a game environment featuring multiple unit types (King, Warrior, Archer, Healer), configurable board sizes, and unit compositions with turn-based gameplay.
        - Integrate key components such as unit ordering (MCTS_u), dynamic node grouping/ungrouping based on the iteration threshold (α_ABS), and compression rate tracking.
    - **Analysis:**
        - Execute simulations to document turn-by-turn decision-making metrics, win rates, node counts, and compression rates.
        - Assess algorithm efficiency (nodes per choice) and determine the optimal balance of abstraction.
- **Step 3: Applications & Challenges**
    - **Applications:**
        1. Strategy games with extensive combinatorial action spaces.
        2. Multi-agent planning problems that require efficient search techniques.
        3. Games with similar unit types where state abstraction provides benefits.
    - **Challenges:**
        - Tuning abstraction parameters (η_R and η_T) for optimal performance.
        - Balancing improvements in compression rate with overall decision quality.
        - Determining the optimal abstraction threshold (α_ABS) for different game configurations.
        - Efficiently computing approximate MDP homomorphism in real time.
        - Mitigating the imperfections caused by lossy state abstraction.

Reference paper:

- • L. Xu, A. Dockhorn, and D. Perez-Liebana, "Elastic Monte Carlo Tree Search," IEEE Transactions on Games, vol. 15, no. 4, pp. 527-537, Dec. 2023, doi: 10.1109/TG.2023.3282351.