Department of Computer Science

CSCI 298 Project Proposal

Title: Foosball Simulation with Monte Carlo and Reinforcement Learning

Units: 3

Student:

Advisor:

Foosball Simulation with Monte Carlo and Reinforcement Learning

1. Introduction

The intersection of simulation and reinforcement learning (RL) presents a promising avenue for advancing artificial intelligence, particularly in game strategies and control systems. One intriguing platform for exploring and refining RL techniques is the strategic and dynamic tabletop game of foosball. By simulating foosball games and leveraging Monte Carlo methods alongside RL, researchers can develop and test strategies in a controlled virtual environment before applying these insights to real-world scenarios.

Foosball, a game that balances simplicity with strategic complexity, provides an excellent opportunity to study complex decision-making and optimize strategies. Monte Carlo simulations are particularly well-suited for this, as they can handle complex probabilistic outcomes. These simulations allow RL algorithms to explore a wide range of potential tactics and strategies by creating diverse game settings and outcomes. By simulating numerous game iterations, we can assess the effectiveness of different approaches and determine which strategies maximize winning potential and adaptability across various game situations.

Reinforcement learning further enhances this process by enabling agents to learn optimal strategies through trial and error. Engaging with the simulated foosball field, RL algorithms can develop complex strategies driven by feedback and incentives. Over time, these strategies converge on performances that surpass those based on heuristics or rigid rules. This combination of Monte Carlo simulations and reinforcement learning not only improves strategy development but also provides a robust framework for navigating the inherent uncertainties and challenges of competitive gaming environments.

1.1 Research aim and scope

The objective of this study is to create and refine foosball strategies using reinforcement learning and Monte Carlo simulations. The performance of these strategies will be evaluated in simulated environments. The scope includes modeling foosball matches, designing strategies based on these advanced techniques, and evaluating their adaptability in various scenarios.

1.2 Research objectives

- To create and implement intricate simulations of foosball games, providing a controlled environment for developing and testing strategies.
- To generate a variety of data for strategy optimization by using Monte Carlo simulations to explore a broad range of game circumstances and outcomes.
- To enhance foosball techniques through trial and error in the simulated environment by developing and employing reinforcement learning algorithms.
- To evaluate the effectiveness and flexibility of the proposed techniques in both simulated games and actual foosball settings.
- To improve overall game strategies by comparing the performance of the learned strategies with existing methods, highlighting their advantages and disadvantages.

2. Related work

Reference 1

Title: Scaling multi-agent reinforcement learning to full 11 versus 11 simulated robotic football

Concept

They propose several enhancements to the conventional single-agent PPO training algorithm to facilitate its adaptation to a multi-agent environment. These enhancements include the following: (1) utilizing a policy and critic system with an 'attention mechanism' that scales with the number of agents; (2) distributing tasks among agents to enable faster processing through batching; and (3) employing league opponents, surrounding adversaries, and freezing the opposing group when necessary.

Problem defined

- Self-play methods, while beneficial for generating strategies without external influence, can struggle to converge to optimal strategies in large and complex environments.

Solution

- Define structured game mechanics and reward functions that guide agents toward more resilient behaviors to address convergence issues in self-play within vast and complicated settings. Additionally, agents can progressively master difficult environments by employing curriculum-based or hierarchical learning approaches.

Reference 2

Title: On the Positional Single Error Correction and Double Error Detection in Racetrack Memories

Concept

In this paper, the authors introduce a new positional code for single and double error detection (P-SECDED) in racetrack memory using a single read head. Specifically, this is a postamble-based methodology where a carefully chosen bit pattern, deposited on the track, is combined with a VT-encoded codeword. The paper examines the challenges of the postamble technique and establishes a standard for postambles. Additionally, they propose a method to utilize this postamble selection such that all single-bit faults and most two-bit errors can be corrected.

Problem defined

- A detailed investigation of compatibility and potential impacts on system performance and reliability is essential when integrating P-SECDED into existing memory architectures and designs.

Solution

- Conduct detailed Monte Carlo simulations across a range of scenarios and system environments to evaluate the effects of P-SECDED on system performance and reliability. This approach allows for an in-depth examination of compatibility and the identification of any issues prior to practical implementation.

Reference 3

Title: An Empirical Study on Google Research Football Multi-agent Scenarios

Concept

This investigation addresses the gap by providing a hyperparameter setup and a population-based MARL training pipeline for a multi-agent football scenario. The setup outperforms the bot with difficulty level 1.0 from scratch and runs through 2 million steps. These experiments establish a baseline for the expected outcomes of the "state-of-the-art multi-agent reinforcement learning" technique, independent proximal policy optimization (IPPO), where each agent independently optimizes its own policy across various training structures.

Problem defined

- Without substantial modification, the strategies developed may not be effective in other contexts because they are highly tailored to the specific dynamics of the Google Research Football environment.

Solution

- Utilize Monte Carlo simulations to evaluate and modify the methods in a variety of simulated settings and scenarios, enabling the identification and improvement of strategies for broader application. This approach helps generalize strategies beyond the specific dynamics of the Google Research Football setting.

Reference 4

Title: Toward Comprehensive Shifting Fault Tolerance for Domain-Wall Memories With PIETT

Concept

To achieve better misalignment correction than the current state of the art, they propose PIETT (Pinning, Insertion, Erasure, and Translation-fault Tolerance). PIETT combines derived error correction with multi-domain access to identify and fix at least three misalignment issues following a random modification process. The study also describes the rates of misalignment and pinning faults in DWM nanowires, highlighting that pinning faults are a significant concern. PIETT is the first approach to address pinning errors in random access DWMs along with misalignment correction. The novel PIETT "Transverse Access Points (TAPs)" are introduced, which utilize a unique write access method that allows for the storage of shift distance detection codes and the setting/resetting of multiple domains in a single intrinsic action.

Problem defined

- The existing model may not sufficiently explanation for variables like infection oscillations, mechanical strain, and added ecological impacts, which might have an antagonistic effect on PIETT's presentation in real-world executions.

Solution

- Iteratively enhance PIETT by training agents in simulations that vary temperature, mechanical strain, and other environmental conditions using Proximal Policy Optimization (PPO). This iterative process makes the model more robust to these factors and more suitable for real-world conditions.

Reference 5

Title: KIcker: An Industrial Drive and Control Foosball System automated with Deep Reinforcement Learning

Concept

This study contributes by employing simulation-based Deep Reinforcement Learning to automate a complex industrial-like process. By applying the rules learned in simulation to the real system, the presented workflow demonstrates that "Deep Reinforcement Learning" can be applied to physical systems built from industrial-grade components and control mechanisms. These systems are designed to replicate the characteristics of real manufacturing processes. Training in a virtual environment is essential for safe Reinforcement Learning and benefits from increased training speed, enabled by domain randomization.

Problem defined

- While virtual training enhances safety during the learning phase, it is crucial to ensure that the learned rules can function safely and reliably in unpredictable real-world circumstances.

Solution

- By training agents in a variety of controlled simulated situations, Proximal Policy Optimization (PPO) combined with Monte Carlo simulations can enhance safety and reliability. PPO stabilizes the policy, while Monte Carlo simulations expose agents to a range of real-world scenarios, ensuring that the rules are reliable and safe in various situations.

2.1 Overall problem statement

The Previous approaches to this problem only dealt with the Foosball Simulation environment; however, these approaches also had the following extra issues:

- **Difficulty of Strategy Convergence:** While self-play techniques are useful for developing strategies independently of external influences, they may struggle to converge on optimal strategies in a vast and complex environment.
- **Effects of Reliability and System Performance:** When integrating P-SECDED into existing memory systems and architectures, a careful examination of compatibility and impacts on system performance and reliability is required.
- **Challenges with Generalization:** The strategies developed may not be applicable in other settings without significant modifications, as they are highly tailored to the specific characteristics of the Google Research Football environment. PIETT's performance in real-world implementations may suffer if factors like temperature changes, mechanical strain, and other environmental effects are not adequately considered.
- **Practical Safety and Dependability:** Although virtual training enhances safety during the learning process, it is essential to ensure that the learned rules can be reliably and safely applied in unpredictable real-world situations.

3. Proposed method

This streamlined approach, consisting of three main procedures and associated methods, ensures a focused and effective testing strategy by enhancing the Foosball Simulation with Monte Carlo simulations and Reinforcement Learning.

- Environment setup
- Monte Carlo Simulations
- Proximal Policy Optimization

A. Environmental setup

The game mechanics are first defined, after which we build the virtual foosball table. To do this, the table completes with the ball, players, and field must be digitally recreated. We designate the action space as having all conceivable moves that the players may perform, and the state space as having the locations and velocities of the ball and players. Ensuring realistic simulations requires precise modeling of the game's physics, including ball movement and player interactions. The incentive system, which offers rewards for goals scored and penalties for goals given up, is also described in detail. This system lays the foundation for the growth of strategic play.

Game Environment:

- **Dimensionality:** The game will be modeled in 2D to simplify the simulation while maintaining the essential elements of foosball gameplay. This includes a top-down view of the table where the players and ball are represented as 2D objects.
- **Game Physics:** Accurate simulation of game physics is crucial for realism. This involves:
  - **Ball Movement:** The ball's motion will be governed by Newtonian physics, including principles such as inertia, friction, and collisions. The ball will bounce off the walls and players with realistic responses based on the angle and speed of impact.
  - **Player Movements:** Players will be modeled with constraints that mimic human control. They will have rotational and translational motions that allow them to kick, block, and intercept the ball.
  - **Collisions:** Collision detection and response algorithms will handle interactions between the ball, players, and table edges to ensure smooth and realistic gameplay.

Game Mechanics:

- **Player Control:** Each player will have a set of controls that allow them to move their rods horizontally and rotate them to kick the ball. The control mechanism will be designed to emulate real-world foosball controls.

- **Scoring System:** The game will include a scoring system where goals are counted, and scores are displayed. The system will keep track of which team scores and update the game state accordingly.
- **Game Rules:** The simulation will enforce standard foosball rules, such as no spinning of the rods, out-of-bounds handling, and goal counts. These rules ensure the game remains fair and adheres to traditional gameplay.

B. Monte Carlo Simulations

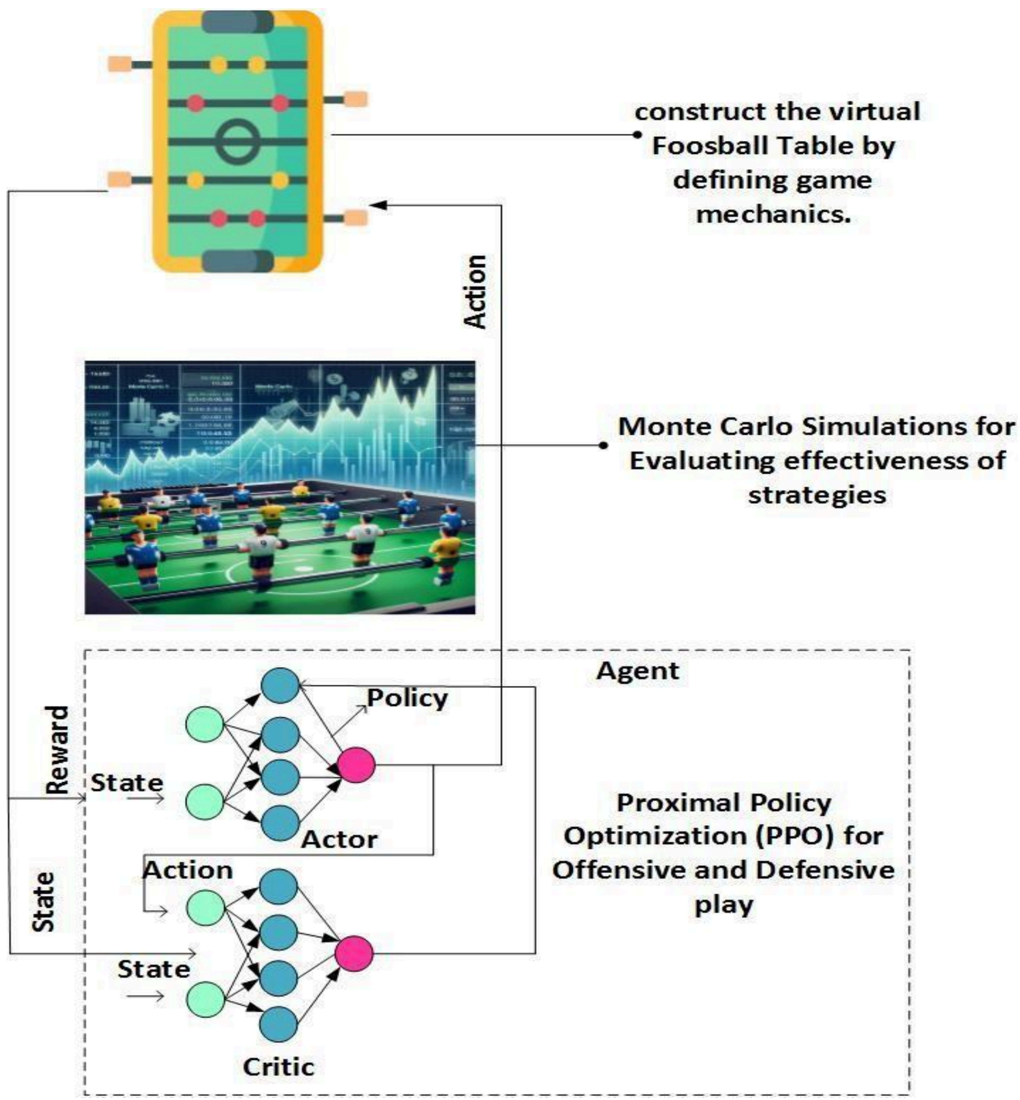
After that, we run Monte Carlo simulations to evaluate the effectiveness of different strategies. We establish the initial conditions for each game and determine the total number of games to be played in the simulations. The goal of each simulation is to play a game using randomly selected actions from the defined action space. These simulations generate game states and rewards, which are tracked and analyzed. By conducting numerous simulations and collecting numerical data, we provide a robust evaluation framework for strategy development, allowing us to identify which strategies perform better and under what conditions.

C. Proximal Policy Optimization Implementation

Next, we apply Proximal Policy Optimization (PPO) to enhance both offensive and defensive performance following the Monte Carlo simulations. The PPO algorithm determines the optimal sequence of actions using the data generated by the Monte Carlo simulations. The system selects actions and adjusts its policy based on the rewards it receives, refining its performance through continuous interaction with the simulated environment. This process allows the network to continuously learn and adapt its decision-making capabilities. PPO aims to develop both offensive and defensive strategies to prevent the opposition from scoring, thereby improving overall gameplay.

Several metrics are used to evaluate the system's performance, including:

- Win rate
- Goal differential
- Strategic adaptability



The proposed architecture

Research highlights

- Defines the rules of the game and creates a virtual foosball table with realistic physics, including player actions, ball motion, and reward structures.

- Conducts simulations using randomly selected actions to analyze different strategies, collecting game states and rewards to evaluate the effectiveness of each strategy.
- Establishes a robust evaluation framework by analyzing data from Monte Carlo simulations to determine how various techniques perform in different scenarios.
- Enhances decision-making and strategy refinement by using simulation data to train a network, improving both offensive and defensive strategies through PPO implementation.
- Improves overall gaming performance in both offensive and defensive domains by continuously adjusting strategies based on simulation input using PPO.

Literature Study Report

Reference 6

Title: Hierarchical control of multi-agent reinforcement learning team in real-time strategy (RTS) games

Concept

This study presents a hierarchical architecture for command and control, consisting of several low-level and one high-level reinforcement learning agents working in a dynamic environment. The high-level commander agent gives instructions to the lower-level unit agents, who can then make independent decisions thanks to this hierarchical arrangement.

Limitation

- The integration of adaptive doctrines and managing the increased complexity of the gaming environment are formidable obstacles to overcome in order to extend the hierarchical model to a fully functional bot capable of playing the complete RTS game.

Reference 7

Title: Multi-Agent Deep Reinforcement Learning using Attentive Graph Neural Architectures for Real-Time Strategy Games

Concept

This study proposes an algorithm called QMIX, based on multi-agent deep reinforcement learning (MADRL), with distributed MADRL at its core. In addition to QMIX-based distributed computation, the study introduces a new preprocessing technique called state classification for representing graph attention. Relationships between agents are identified using graph-based self-attention techniques. The study provides an attention policy for classified state graphs

(CSGA-policy) based on these methods.

Limitation

- Despite the algorithm's promising results in StarCraft II, its performance and efficacy in different real-world circumstances and other RTS games may vary, necessitating further validation and modification.

Reference 8

Title: Game Adaptation by Using Reinforcement Learning Over Meta Games

Concept

This study proposes a "Multi-Agent System training" paradigm for the game main manager, where it competes in the base game against a variety of different agent opponents, each with unique behaviors and skill levels. The experiments are conducted in both single-player and multiplayer versions within an adaptive grid-world environment.

Limitation

- Need to verify the deep learning work, which is essential for continuous and large-state contexts

Reference 9

Title: Data-Driven Inverse Reinforcement Learning Control for Linear Multiplayer Games

Concept

This paper introduces a model-based inverse reinforcement learning strategy repetition framework with three steps: 1) a “policy evaluation step” (using Lyapunov functions to reconstruct cost matrices); 2) a “state-reward weight improvement step” (using inverse optimal control, or IOC); and 3) a “policy development phase” (using an optimal controller). Additionally, the paper develops an online data-driven off-policy inverse reinforcement learning method based on the model-based policy iteration approach, without requiring any prior knowledge of expert control gains or system dynamics. A thorough analysis of the algorithms' convergence and stability is provided.

Limitation

- There is a need to extend the approach to multi-agent graphical games or time-varying systems.

Reference 10

Title: Innovation theater in corporate venturing units: Cultural design as a (de)legitimizing mechanism

Concept

This study develops theories on "theatrical cultural design in corporate venturing units," demonstrated through fun aesthetics (beanbags and foosball tables), youthful behavior (construction with Lego blocks and casual attire), and startup terminology. The authors create a series of claims showing how corporate venturing units may be (de)legitimized by theatrical cultural design, emphasizing three distinct functions: the attention-direction purpose, the societal classification purpose, and the diverting purpose.

Limitation

- Evaluating how cultural design and Innovation Theater affect legitimacy and resilience can be challenging, making it difficult to measure the framework's efficacy and validate it empirically.

Reference 11

Title: Game of Drones: Multi-UAV Pursuit-Evasion Game With Online Motion Planning by Deep Reinforcement Learning

Concept

This study presents the "pursuit-evasion scenario (PES)" framework, enabling quadcopter managers to operate and cooperate within the surrounding environment, powered by a physics engine. A vectorized implementation of the "multi-agent deep deterministic policy gradient (MADDPG)" training forms the basis for the "coronal bidirectionally coordinated with target prediction network (CBC-TP Net)" to ensure the effectiveness of the swarm strategy in the pursuit-evasion task.

Limitation

- There is more work needed to teach six-DOF multiquadcopter agents to use DRL techniques to attack more flexible targets, and to enhance techniques for three-dimensional space missions and more complex scenarios.

Reference 12

Title: A Framework for the Classification and Evaluation of Game Mechanics for Virtual Reality Games

Concept

This study presents a categorization of collaboration mechanics based on various measures, focusing on the diversity of VR gaming and coining the term "interaction mechanics." Based on this taxonomy, the authors identify several characteristics related to targets, tasks, and tools that might affect the quality of interaction mechanics.

Limitation

- Due to its concentration on a small number of specific mechanics (choose and place, shoot, and slice), the framework and test application may not be fully applicable to other VR game genres.

4. Project Deliverables

1. Monte Carlo Simulations

- Initial simulations with data collection and analysis.

2. Preliminary and Refined Strategies

- Development and refinement of strategies based on simulation data.

3. Evaluation Reports

- Intermediate and final evaluation of strategy performance.

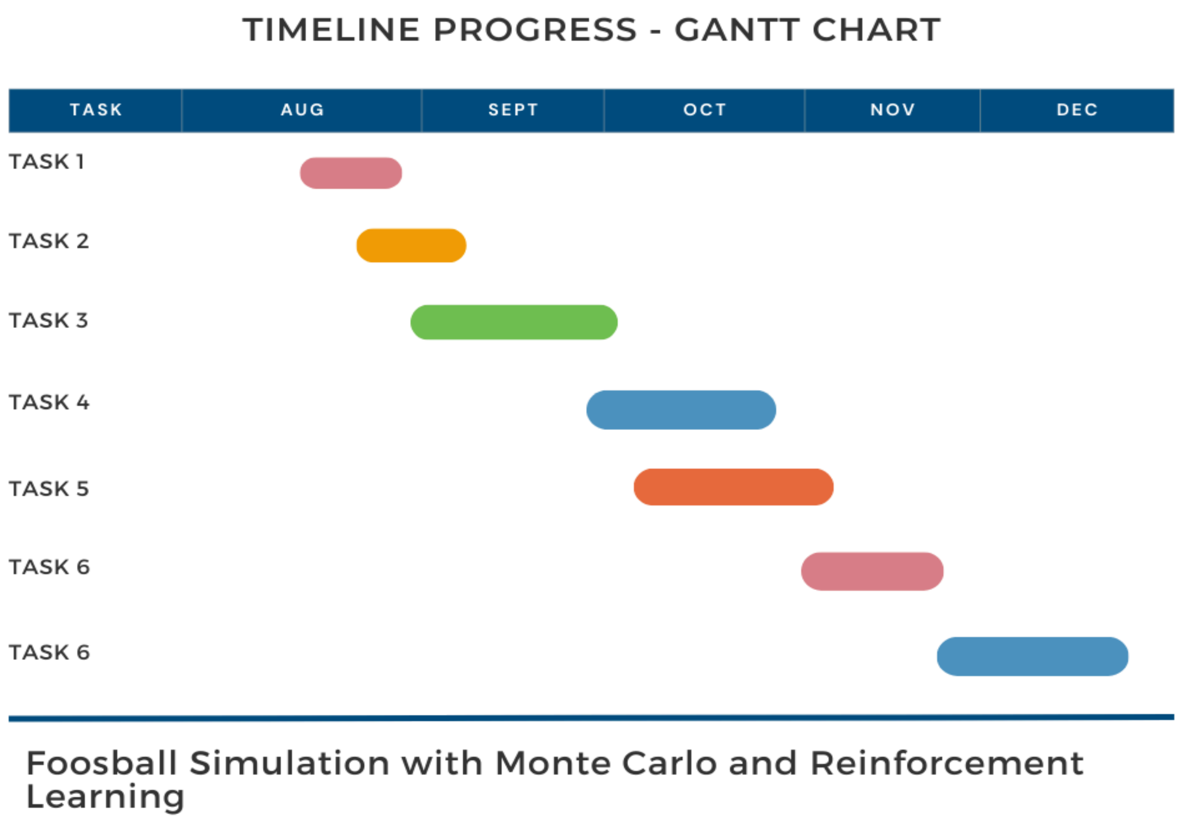
4. Final Report and Presentation

- Comprehensive final report and presentation summarizing the project outcomes.

5. Source Code and Documentation

- Complete source code and documentation for the project.

5. EXPECTED TIMELINE



Summary for Gantt Chart

1. Project Kickoff and Planning: Aug 19 - Aug 25
2. Environment Setup: Aug 26 - Sep 8
3. Monte Carlo Simulations: Sep 9 - Sep 29
4. Preliminary Strategy Development: Sep 30 - Oct 13
5. PPO Implementation: Oct 14 - Nov 3
6. Evaluation and Robustness Testing: Nov 4 - Nov 17
7. Simulation Continuation and Finalization: Nov 18 - Dec 8

Part 1: Project Kickoff and Planning
Duration: 1 week
Timeline: August 19 - August 25

- Initial meeting with the professor.
- Define game mechanics and rules.

Part 2: Environment Setup
Duration: 2 weeks
Timeline: August 26 - September 8

- Create a virtual foosball table with realistic physics.
- Define state and action spaces, and reward structures.

Part 3: Monte Carlo Simulations
Duration: 3 weeks
Timeline: September 9 - September 29

- Establish initial conditions for simulations.
- Run simulations with randomly selected actions.
- Collect and analyze data to identify effective strategies.

Part 4: Preliminary Strategy Development
Duration: 2 weeks
Timeline: September 30 - October 13

- Develop preliminary strategies based on simulation data.
- Test strategies within the simulation environment.

Part 5: Proximal Policy Optimization (PPO) Implementation
Duration: 3 weeks
Timeline: October 14 - November 3

- Integrate PPO with the simulation environment.
- Train the model using PPO and adjust hyperparameters.
- Refine strategies based on continuous feedback.

Part 6: Evaluation and Robustness Testing
Duration: 2 weeks
Timeline: November 4 - November 17

- Conduct intermediate and robustness testing under various game conditions.
- Evaluate the effectiveness of strategies.

Part 7: Simulation Continuation and Finalization
Duration: 3 weeks
Timeline: November 18 - December 8

- Try to apply strategies to real-world foosball scenarios.
- Make final adjustments and optimizations.
- Prepare and deliver the presentation.
- Prepare and deliver the final report.

References

1. Smit, A., Engelbrecht, H. A., Brink, W., & Pretorius, A. (2023). Scaling multi-agent reinforcement learning to full 11 versus 11 simulated robotic football. *Autonomous Agents and Multi-Agent Systems*, *37*(1), 20.
2. Saeed, A., Fayyaz, U. U., Tahir, A., Hong, S., & Mahmood, T. (2023). On the positional single error correction and double error detection in racetrack memories. *IEEE Access*, *11*, 18300-18310.
3. Song, Y., Jiang, H., Tian, Z., Zhang, H., Zhang, Y., Zhu, J., ... & Wang, J. (2024). An empirical study on google research football multi-agent scenarios. *Machine Intelligence Research*, *21*(3), 549-570.
4. Ollivier, S., Longofono, S., Dutta, P., Hu, J., Bhanja, S., & Jones, A. K. (2022). Toward comprehensive shifting fault tolerance for domain-wall memories with piett. *IEEE Transactions on Computers*, *72*(4), 1095-1109.
5. De Blasi, S., Klöser, S., Müller, A., Reuben, R., Sturm, F., & Zerrer, T. (2021). KIcker: an industrial drive and control foosball system automated with deep reinforcement learning. *Journal of Intelligent & Robotic Systems*, *102*(1), 20.
6. Zhou, W. J., Subagdja, B., Tan, A. H., & Ong, D. W. S. (2021). Hierarchical control of multi-agent reinforcement learning team in real-time strategy (RTS) games. *Expert Systems with Applications*, *186*, 115707.
7. Yun, W. J., Yi, S., & Kim, J. (2021, October). Multi-agent deep reinforcement learning using attentive graph neural architectures for real-time strategy games. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 2967-2972). IEEE.

8. Reis, S., Reis, L. P., & Lau, N. (2021). Game adaptation by using reinforcement learning over meta games. *Group Decision and Negotiation*, *30*(2), 321-340.
9. Lian, B., Donge, V. S., Lewis, F. L., Chai, T., & Davoudi, A. (2022). Data-driven inverse reinforcement learning control for linear multiplayer games. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(2), 2028-2041.
10. Brattström, A., & Faems, D. (2024). Innovation theater in corporate venturing units: Cultural design as a (de) legitimizing mechanism. *Journal of Product Innovation Management*.
11. Zhang, R., Zong, Q., Zhang, X., Dou, L., & Tian, B. (2022). Game of drones: Multi-UAV pursuit-evasion game with online motion planning by deep reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(10), 7900-7909.
12. Vlahovic, S., Suznjevic, M., & Skorin-Kapov, L. (2022). A framework for the classification and evaluation of game mechanics for virtual reality games. *Electronics*, *11*(18), 2946.