Hex Game

Salome Christiani
FH Technikum Wien
Master AI & Game Engineering
Wien, AT
ai24m057@technikum-wien.at

Nikolaus Rieder
FH Technikum Wien
Master AI & Game Engineering
Wien, AT
TODO@technikum-wien.at

***Abstract*—This paper presents the development and evaluation of agents for playing the Hex board game. In a first step, two heuristic, rule-based agents were crafted to provide strong, interpretable baselines; these agents rely on explicit pattern-matching for tactics such as bridging, blocking, and chain extension. They then served as opponents during reinforcement learning experiments. The main objective was to train a Proximal Policy Optimization (PPO) agent that outperforms these rule-based baselines and field our best agent in a class-wide 7x7 Hex board game competition against agents from other teams. While the PPO agent successfully learned to beat random and our rule-based opponents, mastering complex strategies and adapting to stronger opponents remained a challenge: it struggled with defensive awareness as the RL agent sees offence and defence through the same reward lens and thus finds it difficult to decide when to sacrifice progress for an urgent block. Additionally, the RL agent lacked multi-turn strategic planning, often reacting only to the current board state without maintaining continuity in its intended strategies across multiple moves. This work lays the foundation for future improvements through hybrid learning approaches and enhanced evaluation pipelines. The project also included the integration of an interactive visualization framework supporting multiple play modes (human vs. machine, machine vs. machine).**

***Index Terms*—rule-based, ppo, reinforcement learning**

I. Introduction

Hex is a deterministic, turn-based connection game with a rich space of possible states, making it a compelling environment for reinforcement learning (RL). Our project aimed to explore two approaches to building agents for Hex:

- Rule-based agents inspired by human reasoning and heuristics.
- A learning-based PPO agent trained using self-play and various opponent configurations.

This two-fold approach allowed for both interpretable baselines and adaptive learning, which we evaluated in different scenarios.

II. Game Engine and Visualisation

The project included a custom hexPosition class handling the game rules, win conditions, and board representation. A modular visualization framework allowed humans to play against agents, observe matches between agents, and export videos for evaluation. It supported modes like:

- Human vs. Rule-Based Agent
- PPO Agent vs. Rule-Based Agent
- PPO Agent vs. Random/Scripted/Past Agents

The GUI was helpful for debugging agent behavior and qualitatively assessing learned strategies.

TODO (insert graphics)

III. Rule-Based Agent

Rule-based agents were developed as training opponents for the PPO agent, giving it exposure to more structured and challenging play than random agents could provide. This helped accelerate early learning, ensuring that the reinforcement learner encountered realistic board dynamics from the start.

A. Architecture and Strategy Framework

Each rule-based agent followed a common structure: multiple predefined strategies were executed in parallel, each suggesting a move based on the current board state. These suggestions were then combined via a weighted voting mechanism, where weights were adapted depending on the game phase (early, mid, or late). Strategies could be tuned or prioritized differently in versions v3 and v4.

The following strategies were implemented:

- **Immediate Win Detection**
  Checks whether placing a stone in a given cell would result in an immediate win. If such a move exists, it is selected immediately.
- **Block Opponent Win**
  Detects whether the opponent has a winning move available and blocks it, preventing an immediate loss.
- **Forcing Win**
  Identifies moves that don't directly win but guarantee a win in the next turn, regardless of how the opponent responds.
- **Extend Own Chain**
  Finds the agent's most advanced chain (using breadth-first search) and attempts to extend it toward its goal edge.
- **Break Opponent Bridge**
  Detects typical bridging patterns in the opponent's formation and inserts a blocking move to disrupt their connectivity.
- **Protect Own Chain from Cut**
  Looks for weak points where the agent's own chain

could be severed and reinforces them by placing protective stones.

- **Create Double Threat**
  Suggests moves that create two simultaneous threats so that the opponent cannot defend against both in one turn.
- **Shortest Connection Path (via Dijkstra)**
  Uses a Dijkstra-based heuristic to compute the shortest potential connection between the agent's start and goal edges, then selects a move that advances along that path.
- **Take Center**
  Prioritizes occupying the central cell (or the nearest available cell to it) early in the game, where flexibility is highest.
- **Advance Toward Goal**
  Suggests moves that push the agent's chain generally closer to the target edge, even if not connected directly.
- **Block Aligned Opponent Path**
  Detects lines or near-connections formed by the opponent along their win axis and attempts to block or disrupt their directionality.
- **Mild Block Threat**
  Identifies subtle threats by the opponent (such as loose connections) and preemptively interferes before they escalate.

Each strategy outputs a suggested move or `None`. Only legal moves from the current `action_set` are considered. If multiple strategies propose the same move, their weights are summed, increasing its chance of being selected.

B. Agent Versions

To iteratively improve the rule-based baseline, we developed multiple versions—most notably v3 and v4:

- **v3 Agent**
  Used adaptive weights based on game stage and bridge status.
- **v4 Agent**
  Fine-tuned those weights to favor late-game double threats and path alignment blocks.

Both share the same underlying architecture and strategy set, but differ in how strategy priorities are dynamically adjusted depending on the game phase. The goal was to simulate a human-like shift from early expansion, to mid-game structuring, to late-game precision.

1) Game Phases: The game board is monitored for the total number of placed stones, allowing the agent to switch behavior based on thresholds:

- **Opening Phase**
  Fewer than 10% of board cells filled
- **Mid Game**
  Between 10%–50% filled
- **End Game**
  More than 50% filled

2) Strategy Weight Encoding: Each version encodes relative preferences by assigning weights to each strategy depending on the phase. Each strategy has a fixed *base weight*, referred to as the *ground weight*, which reflects its general importance independent of timing.

Depending on the game phase, some strategies receive boosts or reductions depending on the phase. For example, `take_center` is heavily favored on the very first move, while `create_double_threat` and `block_aligned_opponent_path` are emphasized in the late game. Additionally, if the last move formed a bridge, `extend_own_chain` receives a temporary weight boost in the next turn.

TABLE I: Ground (base) weights for all strategies in rule-based agent `v3`.

Strategy	Ground Weight
`take_center`	1
`extend_own_chain`	3
`break_opponent_bridge`	3
`protect_own_chain_from_cut`	3
`create_double_threat`	4
`shortest_connection`	4
`make_own_bridge`	3
`mild_block_threat`	2
`advance_toward_goal`	2
`block_aligned_opponent_path`	3

TABLE II: Phase-dependent strategy weights for rule-based agent `v3`.

Strategy	First	Early	Mid	Late
`take_center`	11	1	1	1
`extend_own_chain`[a]	3	3	3	5
`break_opponent_bridge`	3	3	6	3
`protect_own_chain_from_cut`	3	3	6	3
`create_double_threat`	4	4	4	4
`shortest_connection`	4	4	4	8
`make_own_bridge`	3	6	3	3
`mild_block_threat`	2	2	12	2
`advance_toward_goal`	2	3	3	7
`block_aligned_opponent_path`	3	3	7	8

[a]Weight is further increased if the previous move formed a bridge.

3) Difference between v3 and v4 agents:

- v3 is balanced and leans toward tactical responses (e.g., cutting chains, extending own path).
- v4 places greater emphasis on late-game disruption, especially against aligned opponent chains and in recognizing multiple threats.

IV. PPO Reinforcement Learning Agent

The PPO (Proximal Policy Optimization) agent was implemented using the `MaskablePPO` algorithm from the `sb3_contrib` library. The agent was trained in a self-play environment with configurable opponents, including random, rule-based v3, and rule-based v4 agents.

A. Architecture

The agent observes the board using a 3-channel tensor representation:

- Channel 1: Player 1's stones (white)
- Channel 2: Player -1's stones (black)
- Channel 3: A constant plane indicating the current player's turn

The policy network includes:

- A custom CNN feature extractor (`HexCNN`)
- A fully connected MLP head
- An action head producing logits over the 49 board positions (7x7)

Only legal moves are allowed via action masking. Illegal actions are given a probability of zero at inference time.

B. Training Setup

Training was coordinated via `train_alg_hex.py`, dispatching to `ppo_train.py`. Key hyperparameters:

- Board size: 7x7
- PPO algorithm: `MaskablePPO`
- Parallel environments: up to 16
- Opponent mix: configurable (e.g., 50% rule-based, 50% random)
- Reward shaping: customized per experiment

Evaluation was performed regularly against fixed opponents and optionally recorded as video.

C. Model Export

Trained agents could be exported in two formats:

- TorchScript format (`policy.pt`) using `export_tscript.py`
- SB3 model checkpoint (`model.zip`) for live loading

These formats were integrated into evaluation agents such as `sb3_agent.py` and `tscript_agent.py`.

V. Experiments

We conducted a series of experiments to evaluate PPO agent performance and learning progression. Each experiment had a distinct training setup and reward scheme. Unless otherwise noted, agents were trained for 2000–12000 games.

A. Experiment 1: Fixed Side vs. Random

The agent played only as one color (e.g., white) against a random opponent. Reward shaping included:

- +1 for winning
- Penalty: $0.01\times$ number of moves used

This setup favored fast convergence but failed to generalize when switching sides.

B. Experiment 2: Alternating Sides

The agent alternated between white and black every 200 games. This improved generalization but slowed down convergence due to side-switching. The reward structure remained the same as in Experiment 1.

C. Experiment 3: Phased Side Training + Self-Play

The agent played the first 1500 games as white and the next 1500 as black, initially against a random opponent. Then, it trained against the best model of the previous phase. This setup aimed to gradually improve robustness and adaptivity.

D. Experiment 4: Curriculum Learning via Board Size

The agent started training on a 3x3 board and progressively moved to 7x7, increasing the board size every 250 games. This curriculum approach helped the agent discover high-level strategies in simpler environments first.

E. Experiment 5: Two-Agent Self-Play

Two PPO agents were initialized and trained by playing against each other for 12,000 games. They alternated in learning every 100 games and switched sides every 50 games. A broad reward structure was used, including:

- +10 for winning, -10 for losing
- +3 for blocking, +2.5 for connecting
- Penalties for missing clear blocks
- Entropy bonus to prevent determinism

F. Experiment 6: Phased Best-vs-Best Training

The agent trained in 12 sequential phases of 1000 games each. In every new phase, the agent played against the best version from the previous phase. Every fourth phase included random opponents to avoid overfitting. Rewards:

- $20 + 5 \cdot \left(\frac{7}{\text{moves}}\right)^2$ for winning
- +2 for connecting, +1 for blocking
- -30 for losing

At the end, all 12 agents were evaluated in round-robin to identify the strongest.

VI. Results and Discussion

A. Performance Against Random Opponents

All PPO agents reached high win rates against random agents. In Experiment 1, the fixed-side agent achieved over 90% win rate as white and around 87% as black. However, when forced to switch sides, it often failed to adapt and played as if it were still on the original side. This confirmed that training exclusively on one perspective hinders generalization.

B. Effect of Alternating Sides and Self-Play

Experiment 2 improved generalization, but introduced instability due to frequent side-switching. Experiment 3's phased approach helped maintain continuity, and training against one's own previous best (self-play) in Experiments 3 and 6 led to modest improvements in adaptability.

C. Curriculum Learning and Two-Agent Play

Experiment 4 demonstrated the benefits of curriculum learning: the agent learned faster when starting on small boards and gradually increasing complexity. Experiment 5, which featured two agents learning together in self-play, showed richer, more varied gameplay and slightly higher resilience when facing novel opponents.

D. Best-vs-Best Phase Training Results

Experiment 6 yielded the most robust agents. Phased competition against previous bests created a natural curriculum. However, final round-robin evaluation revealed that performance peaked mid-way through training. Later agents tended to overfit to the last opponent, losing general adaptability.

E. Observed Weaknesses

Despite some success, PPO agents consistently showed two major weaknesses:

1) **Lack of Defensive Awareness:** Agents rarely blocked straightforward winning threats from the opponent unless such moves aligned with their own immediate strategy.
2) **No Multi-Turn Planning:** Unlike rule-based agents which simulate continuity, PPO agents appeared reactive: they optimized for the current board state only and failed to follow through on long-term strategies unless guided by dense rewards.

Manual test games confirmed that PPO agents often ignored near-wins by the opponent and failed to exploit guaranteed winning positions. Visual inspection also showed repeated deterministic openers and suboptimal mid-game responses.

VII. Conclusion and Future Work

This project explored the design of both rule-based and reinforcement learning agents for the game of Hex. We implemented a modular framework with a visual interface, allowing easy evaluation of agents in human and agent-vs-agent modes.

Our rule-based agents provided interpretable and reliable performance using weighted heuristic strategies, with dynamic tuning across game phases. These agents also served as structured opponents for the PPO agent during training.

We trained multiple PPO agents using self-play, curriculum learning, and phased best-vs-best setups. While agents learned to win consistently against random players, they struggled to match the strategic depth and awareness of rule-based opponents.

Key Takeaways

- Rule-based agents offered stable baselines and accelerated PPO learning.
- Curriculum learning (board-size and opponent difficulty) helped early convergence.
- Dense and informative rewards were essential for meaningful PPO behavior.
- Reinforcement learning agents lacked strategic continuity and defensive reasoning.

Future Work

To bridge the gap between tactical awareness and long-term planning, future efforts could include:

- Hybrid models that combine rule-based priors with learned policy gradients.
- Incorporation of look-ahead mechanisms or Monte Carlo Tree Search.
- Enhanced reward shaping based on threat detection or plan consistency.
- Evaluation via Elo-style agent tournaments or curriculum-leveled tests.

The codebase and evaluation infrastructure built during this project lay the foundation for further work on interpretable and competitive Hex agents.

Acknowledgment

We would like to express our sincere gratitude to our instructor, **De Oliveira Gomes Rosana**, for their valuable guidance, insightful feedback, and continuous support throughout this project. Their expertise and encouragement played a crucial role in shaping both our technical approach and understanding of reinforcement learning in strategic games.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, "Title of paper if known," unpublished.

[5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.