To GNN or to LLM? That is a Risk(y) Question

Andri Vidarsson
andriv@stanford.edu

Herman Brunborg
brunborg@stanford.edu

Anna Lyubarskaja
annalyu@stanford.edu

Abstract

This paper explores automated decision-making strategies for the strategy board game *Risk*. We introduce a custom-built game environment and API that support human vs. human, human vs. agent, and agent vs. agent matches. Within this framework, we implement and evaluate multiple automated strategies, including a rule-based heuristic agent, a graph neural network (GNN)-driven reinforcement learning (RL) agent, and a language model (LLM)-driven agent. By comparing these distinct approaches, our platform highlights the strengths and limitations of various methodologies in complex, multi-player strategic settings. We find that using GNN is a promising approach that is able to learn and improve, while an LLM without fine-tuning performs poorly while being more expensive to run.

1 Introduction

Risk is a popular board game in which players compete for global domination by controlling and defending territories on a world map [1]. Although its rules are conceptually straightforward, spanning phases for troop placement, attacking neighboring territories, and fortifying positions, the complexity of the game is extensive. This complexity arises from a vast combinatorial state space, uncertainty in the immediate outcomes of combat actions, and the unpredictable nature of future moves and hidden resources held by other players.

Reinforcement Learning (RL) has demonstrated remarkable success in a wide range of decision-making tasks, including complex games [2]. By interacting with an environment, receiving rewards as feedback, and iteratively refining their policies, RL agents have achieved superhuman performance in games such as Chess and Go, showcasing their capacity to handle intricate state and action spaces. Recently, RL has also been applied to complex, multi-player board games like *Diplomacy*, which share similarities with *Risk* in requiring strategic decision-making, long-term planning, and adaptation to other players' actions. Research has shown that integrating language modeling with strategic reasoning can enable human-level play in *Diplomacy* [3], while human-regularized RL and planning techniques have proven effective in mastering the no-press variant of the game [4]. These developments highlight the growing potential of advanced RL methods in navigating challenging, multi-agent environments.

In this work, we have developed a custom *Risk* environment and a corresponding application programming interface (API) that supports both human players (via a command-line interface) and a variety of autonomous agents. Using this flexible framework, we implemented four distinct types of agents: a random-move agent, a heuristic-driven rule-based agent, a graph neural network-based reinforcement learning agent trained through self-play against other agents, and an LLM-based agent leveraging language model reasoning. This platform provides a practical setting to explore and compare various AI strategies in a complex, strategic board game environment.

Our code is accessible at: github.com/hermabr/risk-rl.

2 Game Rules and Environment Setup

In a standard game of *Risk*, players aim to expand control by occupying territories and maneuvering troops. Each turn typically involves three main phases:

1. **Reinforcement:** At the start of a player's turn, they gain new troops determined by the number of territories they control and any continent-wide bonuses. Players may also trade in sets of collected cards for additional troops. These newly acquired troops are then deployed to the player's own territories before proceeding to the attack phase.
2. **Attack:** Players may initiate combat against neighboring enemy territories, using dice rolls to determine the outcome of each skirmish.
3. **Fortification:** Players can redistribute troops between connected territories they control to strengthen their strategic positions.

In our environment, the initial drafting phase, where territories are selected before the main sequence of phases has been omitted. Instead, the game starts from a randomly assigned, valid board configuration, ensuring each player begins with the appropriate number of troops. Cards are included in the implementation, and players can acquire and later trade them in for additional troops following standard *Risk* rules.

For all-agent matches, we also include a configurable upper limit on the number of rounds. This safeguard prevents games from continuing indefinitely if the state of play reaches a stalemate-like scenario, which we will discuss in detail when describing the agents.

3 Agents

3.1 Random Agent

The random agent serves as a baseline model, employing a completely non-strategic approach to gameplay. Its decision-making process is entirely based on random selection from the set of available actions at each game state. For each phase of the game (reinforcement, attack, and fortification), the random agent follows these steps: (1) Retrieve all possible legal actions from the game environment. (2) Select an action uniformly at random from this set of legal actions. (3) Execute the selected action. This process is repeated until no more actions are available or required for the current phase.

3.2 Heuristic Agent

The heuristic agent follows a straightforward, rule-based strategy for each phase of the game, aiming to reinforce vulnerable fronts, capitalize on opportunities to secure continents, and maintain well-distributed troop strength. Although these heuristics are designed to be intuitive, some decisions may skew overly defensive or lack long-term foresight.

Card Usage At the beginning of a turn, if the agent is eligible to trade in cards for additional troops, it always does so without attempting to delay in hopes of acquiring a more favorable combination later. This ensures the agent maximizes immediate reinforcements whenever possible.

Reinforcement Phase During the reinforcement phase, the agent deploys its available troops one at a time. Before placing each troop, it selects the territory facing the greatest threat, determined by the total number of enemy troops on adjacent territories relative to the territory's troop count. This approach focuses on reinforcing the most vulnerable positions; however, we observed that this strategy might be too defensive, potentially limiting opportunities for more aggressive expansion.

Attack Phase When selecting territories to attack, the agent prioritizes targets based on a two-tiered criterion. First, it checks whether winning an adjacent territory would secure a continent. Among these opportunities, it chooses the option that has the maximum positive troop difference (its own troops minus the defenders'). If no continent-securing moves are available, it ranks remaining potential attacks solely by troop difference. The agent attempts attacks in order of priority, using the maximum allowed dice rolls (up to three) and continues to launch attacks until it either reaches a predefined limit on the number of attacks per turn (conditioned on its total troop count) or finds no beneficial engagements (i.e., no positive troop differences remain).

Fortification Phase In the fortification phase, the agent redistributes troops one at a time. It moves a troop from the territory with the highest troop surplus (including landlocked territories treated as having effectively infinite surplus) to the territory with the greatest need (lowest troop difference, possibly negative). If multiple source territories are equally suitable, it chooses the one with the largest number of troops. The agent continues redistributing troops until a predefined limit (based on total troop count) is reached or until the next move would require relocating troops from a territory that has already received reinforcements in the current fortification phase.

3.3 Graph Neural Network Agent

The GNN agent attempts to enhance the heuristic-based strategy by utilizing a learned model specifically for the attack phase. While it retains the heuristic logic for the reinforcement and fortification phases, the GNN enables more sophisticated and adaptive decision-making during attacks by leveraging the structural information of the game board.

3.3.1 Model Architecture

The GNN agent utilizes a compact neural network with 22,017 trainable parameters. The architecture consists of two Graph Convolutional Network (GCN) layers [5], followed by a feedforward layer that outputs a probability distribution over possible attack actions. Each index in the output vector corresponds to a specific attack move, defined by the tuple (attack from territory, attack to territory, number of troops to attack with, up to three). Additionally, one of the action indices is designated as the *end attack option*, allowing the agent to choose to stop making attacks and move to the next phase.

To execute an action, the selected index from the model's output is decoded into its corresponding attack move, which is then carried out by calling the game API. This decoding process allows the learned policy to interact with the game environment, enabling the agent to translate its outputs into gameplay actions. To ensure only valid actions are selected, invalid actions are masked during action selection.

3.3.2 Input Representation - Game State Encoding

The GNN model processes the game state through two main inputs:

- **Edge List Array:** This array encodes the adjacency structure of the game board, representing connections between territories. See Figure 3 in the Appendix for the adjacency structure.
- **Node Features:** Each territory is represented by a feature vector that evolves with the game state, capturing both the player's and opponents' attributes. The features are calculated from the perspective of the current player, whose turn it is at each time step. The features include troop counts, the number of bordering territories, and troop differences, which are dynamically updated to align with the active player's perspective on the board. Strategic indicators such as the ability to attack, the potential to secure a continent by conquering a territory, and the current game round provide additional context for decision-making. This perspective enables the model to effectively evaluate both offensive opportunities and defensive priorities. Numerical features are standardized for consistency, while binary indicators and border counts remain unscaled.

3.3.3 Training Procedure

The GNN agent is trained using the Monte Carlo Policy Gradient method, specifically the REINFORCE algorithm [6]. The training setup involves games with five players: two reinforcement learning agents, two random agents, and one heuristic agent. Training progresses through iterative gameplay, where the agent learns to optimize its attack strategies based on the outcomes of previous actions.

Reward Structure The reward structure is exclusively tied to the results of individual attacks and does not extend to other phases of the game. The specific rewards are defined as follows:

- **Lost Troop Penalty:** A small negative reward is assigned for every troop lost during an attack, encouraging the agent to minimize unnecessary losses.
- **Inaction Penalty:** A large negative reward is given if the agent moves to the next phase without making any attacks during a round. This penalty incentivizes the agent to remain active and engage in offensive actions.
- **Territory Conquest Reward:** The agent receives a reward for conquering a territory. This reward evolves with the number of rounds, following an exponential decay function. This design ensures that the model prioritizes actions that can lead to larger rewards in the later stages of the game.
- **Strategic Rewards:** Large positive rewards are provided for significant achievements, including securing an entire continent, eliminating an opponent, and ultimately winning the game.

Performance evaluations are conducted at regular intervals of every 200 games. During evaluation, the agent is tested in two configurations:

1. One RL agent competing against four random agents.
2. One RL agent competing against three random agents and one heuristic agent.

3.4 LLM agent

Finally, we also consider an LLM agent which queries an LLM to make each decision. This involves prompting the LLM with the current (observable) state of the game, and giving it a discrete set of choice to choose from. Moreover, as the LLM tends to exhibit a bias towards the choice that is offered first, we also repeat prompting multiple times, randomizing the order of the choices offered. Similarly to the Graph Neural Network Agent, we run this in two regimes.

- One is against four random agents who all make decisions uniformly at random.
- The other is against four players who follow the heuristic setup described in Section 3.2.

We use LLAMA 3.1 8B [7] as our LLM as it provides a good tradeoff between computational feasibility and performance. See the appendix for an example of a prompt for the LLM.

4 Results and Conclusions

4.1 Heuristic Agent

We evaluated the performance of the heuristic agent by simulating 1,000 games in which a single heuristic agent competed against four random agents. The heuristic agent secured a victory in 61% of the games, demonstrating a clear improvement over a random strategy. Additionally, 34% of the games concluded in ties, defined as scenarios where the maximum number of rounds was reached with the heuristic agent still remaining among the active players.

These results demonstrate that the heuristic approach effectively outperforms random strategies, achieving a significantly higher win rate. However, the high tie rate indicates that the heuristic may be too defensive, limiting opportunities for aggressive expansion. Adjusting the strategy to balance defensive and offensive actions could help reduce ties and improve overall performance.

4.2 Graph Neural Network Agent

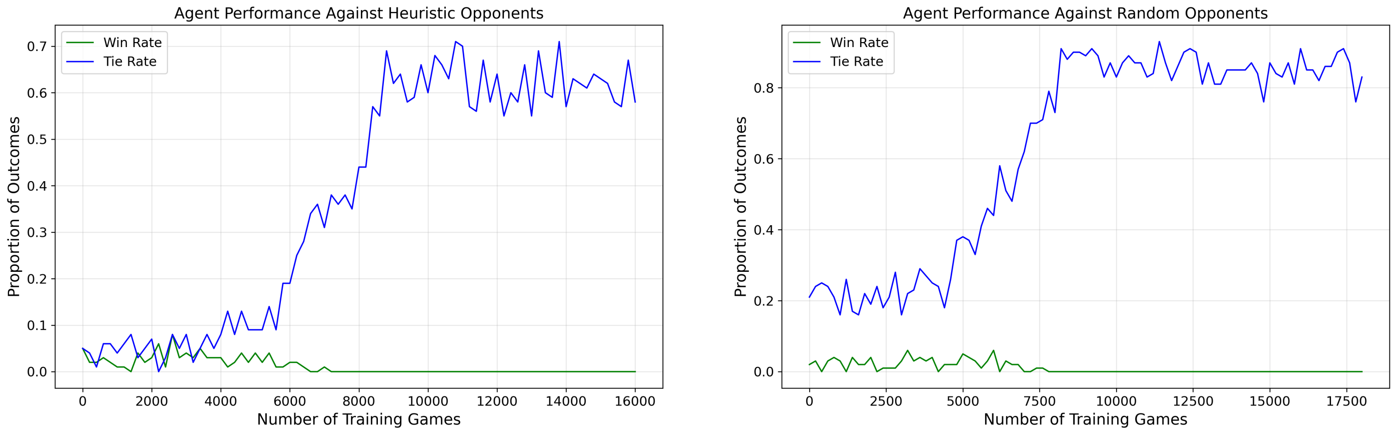


Figure 1: Agent Performance Across Different Opponents. The left plot shows performance against a mixture of random and heuristic opponents, while the right plot shows performance against random opponents.

The results, illustrated in Figure 1, demonstrate that using a graph neural network for this problem is a promising approach. The model shows clear signs of learning and improvement as the number of training games increases, highlighting the potential of our encoding scheme to effectively capture the structural and dynamic aspects of the game state.

However, the model appears to be optimizing primarily for ties rather than outright victories. A tie is defined as a scenario where the game reaches the maximum number of rounds and the RL player has not been eliminated. This behavior is likely a result of the current reward structure, which focuses exclusively on the outcomes of individual attacks. To address this limitation, the reward structure could be adjusted to incentivize broader strategic goals. For example, rewards could be distributed for maintaining control of a territory or continent over multiple turns. Such modifications would encourage the model to prioritize actions that contribute to long-term success rather than merely avoiding losses.

4.3 LLM Agent

Simulating the game using an LLM for many turns turned out to be more expensive than we had expected, so we focus our analysis on the first 100 turns. We summarize the survival rate in Table 1. Unsurprisingly, the LLM agent is more likely to lose to heuristic agents in the first 100 turns than to random agents.

	Random Opponents	Heuristic Opponents
Total games	10	10
Survived past turn 100	9	7
Games where LLM died:		
Game ID (Turn)	5 (24)	0 (49), 3 (87), 4 (18)

Table 1: Performance of LLM against Random and Heuristic Opponents

One interesting trend we noticed which may explain the poor performance of the LLM approach is its disproportional bias towards the choice which is mentioned first. In Figure 2, we see that perhaps as a sign of "overwhelmingness", the LLM will always have an unreasonable bias towards the first option (recall that we shuffle all the options and send the prompt multiple times, so there is no systematic trend). This suggests that an LLM approach would benefit from further fine-tuning to prevent this type of decision making. This might also suggest that a more symmetric approach would be useful. Lastly, the poor performance might be due to the graph-structure of Risk, an area where AI is known to struggle [8].

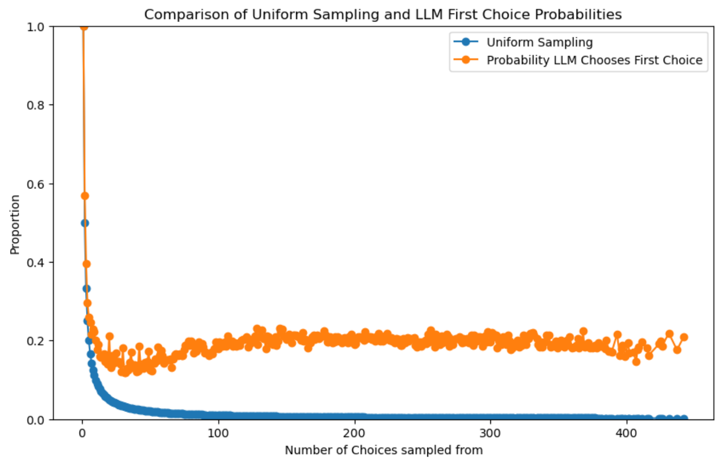


Figure 2: The likelihood that the LLM and random strategy will pick the first option among its possible actions. The LLM never chooses the first given option with less than about 0.2 probability, even as the number of options grows and even though the order of the actions is random.

5 Contributions

- Andri contributed to the development of the game API, developed the heuristic-based agent, and developed the GNN agent.
- Herman contributed to the development of the game API and developed the random-based agent and the LLM agent
- Anna contributed to the project writeup and the analysis of performance of the LLM model.

References

[1] "Risk board game," Hasbro, originally developed by Albert Lamorisse and first published by Parker Brothers (1957).

[2] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[3] M. F. A. R. D. T. (FAIR), A. Bakhtin, N. Brown, E. Dinan, G. Farina, C. Flaherty, D. Fried, A. Goff, J. Gray, H. Hu, A. P. Jacob, M. Komeili, K. Konath, M. Kwon, A. Lerer, M. Lewis, A. H. Miller, S. Mitts, A. Renduchintala, S. Roller, D. Rowe, W. Shi, J. Spisak, A. Wei, D. Wu, H. Zhang, and M. Zijlstra, "Human-level play in the game of diplomacy by combining language models with strategic reasoning," *Science*, vol. 378, no. 6623, pp. 1067–1074, 2022.

[4] A. Bakhtin, D. J. Wu, A. Lerer, J. Gray, A. P. Jacob, G. Farina, A. H. Miller, and N. Brown, "Mastering the game of no-press diplomacy via human-regularized reinforcement learning and planning," arXiv:2210.05492, 2022.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: http://arxiv.org/abs/1609.02907

[6] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[7] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[8] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord, "Think you have solved question answering? try arc, the ai2 reasoning challenge," *arXiv preprint arXiv:1803.05457*, 2018.

Appendix - Playing Environment

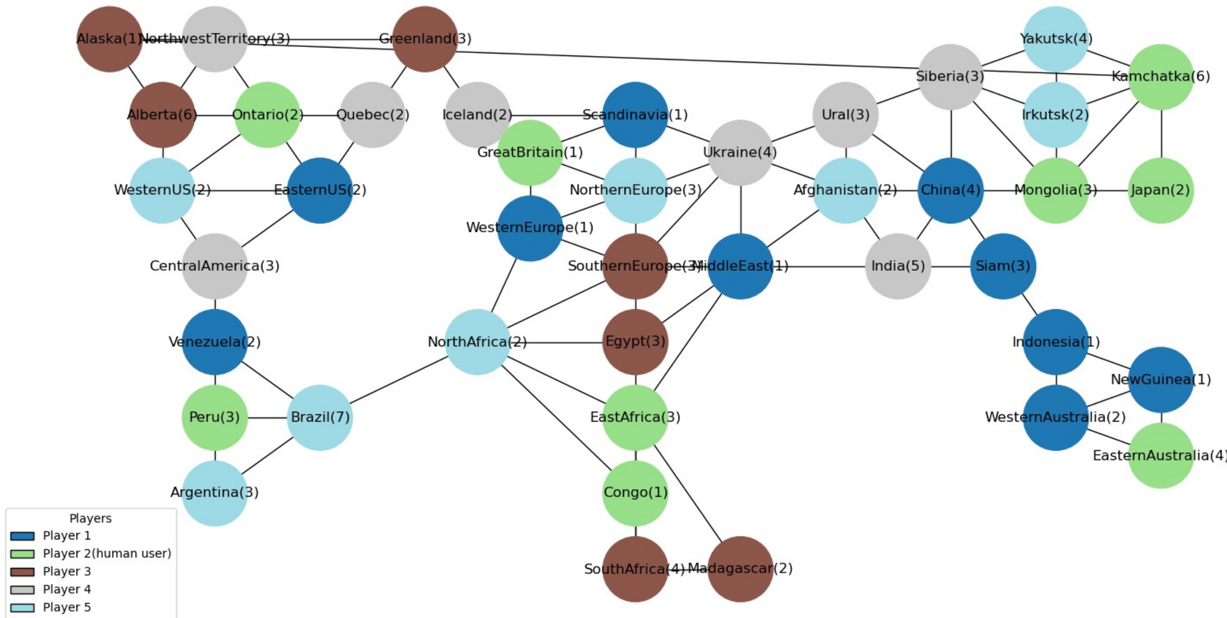


Figure 3: Dynamic visualization of the current game state in Risk, showing territories, connections, army counts, and player ownership. The graph updates in real-time based on changes in the game state when playing the game through command line.

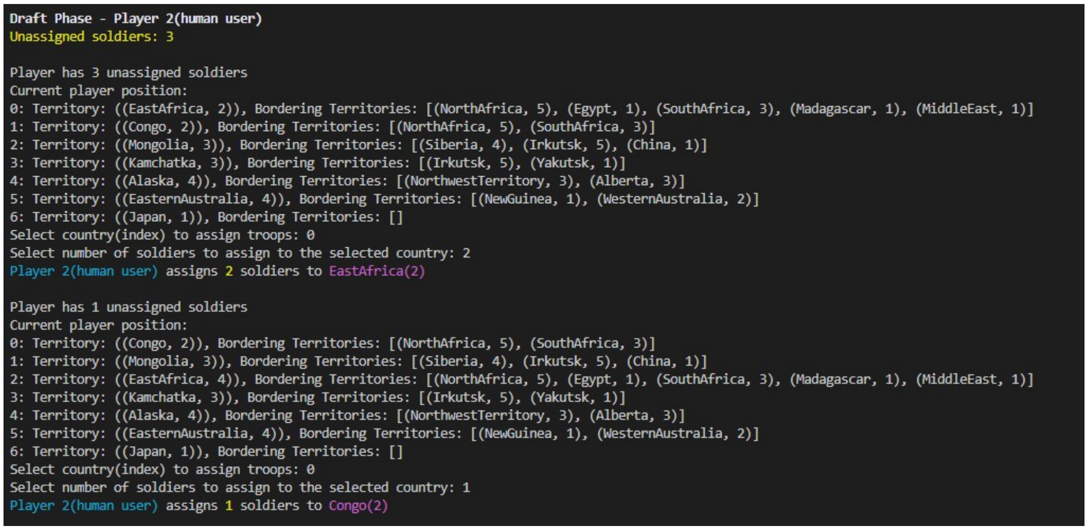
Next, we show the interface when selecting actions for the human user during different phases of the game. For each phase, the action options are ranked, the best action according to the heuristic strategy is displayed first, the heuristic based agents always select that first action(with some constraints such as a limited number of actions per round).

```
Attack Phase - Player 2(human user)

Attack options:
0: (Kamchatka, 6) -> (Alaska, 1)
1: (Kamchatka, 6) -> (Irkutsk, 2)
2: (EasternAustralia, 4) -> (NewGuinea, 1)
3: (EasternAustralia, 4) -> (WesternAustralia, 2)
4: (Kamchatka, 6) -> (Yakutsk, 4)
5: (EastAfrica, 3) -> (MiddleEast, 1)
6: (Mongolia, 3) -> (Irkutsk, 2)
7: (Peru, 3) -> (Venezuela, 2)
8: (EastAfrica, 3) -> (NorthAfrica, 2)
9: (EastAfrica, 3) -> (Madagascar, 2)
10: (Mongolia, 3) -> (Siberia, 3)
11: (Ontario, 2) -> (Quebec, 2)
12: (Ontario, 2) -> (EasternUS, 2)
13: (Ontario, 2) -> (WesternUS, 2)
14: (Peru, 3) -> (Argentina, 3)
15: (EastAfrica, 3) -> (Egypt, 3)
16: (Mongolia, 3) -> (China, 4)
17: (Ontario, 2) -> (NorthwestTerritory, 3)
18: (EastAfrica, 3) -> (SouthAfrica, 4)
19: (Ontario, 2) -> (Alberta, 6)
20: (Peru, 3) -> (Brazil, 7)

Select attack option(or -1 to skip): 0
Select number of soldiers(maximum 3): 3

Battle: Kamchatka(6) -> Alaska(1), attacking soldiers: 3
Attacker rolls: [4, 3, 1]
Defender rolls: [3]
Defender loses 1 soldiers
Attacker loses 0 soldiers
Alaska(0) has been conquered!
```

Figure 4: Attack phase

```
Fortify Phase - Player 2(human user)

Fortify options (ranked):
0: Move from Japan to Alaska | Dest Troop Diff: -3 | Dest Army Size: 2 | Origin Army Size: 3
1: Move from Kamchatka to Alaska | Dest Troop Diff: -3 | Dest Army Size: 3 | Origin Army Size: 3
2: Move from Mongolia to Alaska | Dest Troop Diff: -3 | Dest Army Size: 3 | Origin Army Size: 3
3: Move from EastAfrica to Congo | Dest Troop Diff: -3 | Dest Army Size: 3 | Origin Army Size: 1
4: Move from Japan to Kamchatka | Dest Troop Diff: -1 | Dest Army Size: 2 | Origin Army Size: 3
5: Move from Japan to Mongolia | Dest Troop Diff: -1 | Dest Army Size: 2 | Origin Army Size: 3
6: Move from Kamchatka to Mongolia | Dest Troop Diff: -1 | Dest Army Size: 3 | Origin Army Size: 3
7: Move from Mongolia to Kamchatka | Dest Troop Diff: -1 | Dest Army Size: 3 | Origin Army Size: 3
8: Move from Alaska to Kamchatka | Dest Troop Diff: -1 | Dest Army Size: 3 | Origin Army Size: 3
9: Move from Alaska to Mongolia | Dest Troop Diff: -1 | Dest Army Size: 3 | Origin Army Size: 3
10: Move from Kamchatka to Japan | Dest Troop Diff: inf | Dest Army Size: 3 | Origin Army Size: 2
11: Move from Mongolia to Japan | Dest Troop Diff: inf | Dest Army Size: 3 | Origin Army Size: 2
12: Move from Alaska to Japan | Dest Troop Diff: inf | Dest Army Size: 3 | Origin Army Size: 2

Select an option to fortify (or -1 to skip): 0
Select move (1 to 1): 1
Player 2(human user) fortifies 1 from Japan(2) to Alaska(3)

Fortify options (ranked):
0: Move from EastAfrica to Congo | Dest Troop Diff: -3 | Dest Army Size: 3 | Origin Army Size: 1
1: Move from Kamchatka to Alaska | Dest Troop Diff: -2 | Dest Army Size: 3 | Origin Army Size: 4
2: Move from Mongolia to Alaska | Dest Troop Diff: -2 | Dest Army Size: 3 | Origin Army Size: 4
3: Move from Kamchatka to Mongolia | Dest Troop Diff: -1 | Dest Army Size: 3 | Origin Army Size: 3
4: Move from Mongolia to Kamchatka | Dest Troop Diff: -1 | Dest Army Size: 3 | Origin Army Size: 3
5: Move from Alaska to Kamchatka | Dest Troop Diff: -1 | Dest Army Size: 4 | Origin Army Size: 3
6: Move from Alaska to Mongolia | Dest Troop Diff: -1 | Dest Army Size: 4 | Origin Army Size: 3
7: Move from Kamchatka to Japan | Dest Troop Diff: inf | Dest Army Size: 3 | Origin Army Size: 1
8: Move from Mongolia to Japan | Dest Troop Diff: inf | Dest Army Size: 3 | Origin Army Size: 1
9: Move from Alaska to Japan | Dest Troop Diff: inf | Dest Army Size: 4 | Origin Army Size: 1

Select an option to fortify (or -1 to skip): 0
Select move (1 to 2): 1
Player 2(human user) fortifies 1 from EastAfrica(3) to Congo(1)
```

Figure 5: Fortify phase



```
Draft Phase - Player 2(human user)
Unassigned soldiers: 3

Player has 3 unassigned soldiers
Current player position:
0: Territory: ((EastAfrica, 2)), Bordering Territories: [(NorthAfrica, 5), (Egypt, 1), (SouthAfrica, 3), (Madagascar, 1), (MiddleEast, 1)]
1: Territory: ((Congo, 2)), Bordering Territories: [(NorthAfrica, 5), (SouthAfrica, 3)]
2: Territory: ((Mongolia, 3)), Bordering Territories: [(Siberia, 4), (Irkutsk, 5), (China, 1)]
3: Territory: ((Kamchatka, 3)), Bordering Territories: [(Irkutsk, 5), (Yakutsk, 1)]
4: Territory: ((Alaska, 4)), Bordering Territories: [(NorthwestTerritory, 3), (Alberta, 3)]
5: Territory: ((EasternAustralia, 4)), Bordering Territories: [(NewGuinea, 1), (WesternAustralia, 2)]
6: Territory: ((Japan, 1)), Bordering Territories: []
Select country(index) to assign troops: 0
Select number of soldiers to assign to the selected country: 2
Player 2(human user) assigns 2 soldiers to EastAfrica(2)

Player has 1 unassigned soldiers
Current player position:
0: Territory: ((Congo, 2)), Bordering Territories: [(NorthAfrica, 5), (SouthAfrica, 3)]
1: Territory: ((Mongolia, 3)), Bordering Territories: [(Siberia, 4), (Irkutsk, 5), (China, 1)]
2: Territory: ((EastAfrica, 4)), Bordering Territories: [(NorthAfrica, 5), (Egypt, 1), (SouthAfrica, 3), (Madagascar, 1), (MiddleEast, 1)]
3: Territory: ((Kamchatka, 3)), Bordering Territories: [(Irkutsk, 5), (Yakutsk, 1)]
4: Territory: ((Alaska, 4)), Bordering Territories: [(NorthwestTerritory, 3), (Alberta, 3)]
5: Territory: ((EasternAustralia, 4)), Bordering Territories: [(NewGuinea, 1), (WesternAustralia, 2)]
6: Territory: ((Japan, 1)), Bordering Territories: []
Select country(index) to assign troops: 0
Select number of soldiers to assign to the selected country: 1
Player 2(human user) assigns 1 soldiers to Congo(2)
```

Figure 6: Reinforcement phase

Appendix - LLM prompt

```
You are an expert risk board game player and you are Player 3.

Regions:Afghanistan: 4 soldiers owned by Player 2
Alaska: 1 soldiers owned by Player 4
Alberta: 1 soldiers owned by Player 4
Argentina: 1 soldiers owned by Player 4
Brazil: 1 soldiers owned by Player 1
CentralAmerica: 1 soldiers owned by Player 5
China: 1 soldiers owned by Player 2
Congo: 6 soldiers owned by Player 3
EastAfrica: 1 soldiers owned by Player 1
EasternAustralia: 1 soldiers owned by Player 5
EasternUS: 1 soldiers owned by Player 1
Egypt: 1 soldiers owned by Player 1
GreatBritain: 1 soldiers owned by Player 1
Greenland: 1 soldiers owned by Player 1
Iceland: 1 soldiers owned by Player 1
India: 1 soldiers owned by Player 2
Indonesia: 3 soldiers owned by Player 5
Irkutsk: 2 soldiers owned by Player 4
Japan: 1 soldiers owned by Player 4
Kamchatka: 1 soldiers owned by Player 4
Madagascar: 1 soldiers owned by Player 1
MiddleEast: 1 soldiers owned by Player 1
Mongolia: 1 soldiers owned by Player 4
NewGuinea: 7 soldiers owned by Player 3
NorthAfrica: 1 soldiers owned by Player 3
NorthernEurope: 3 soldiers owned by Player 1
NorthwestTerritory: 1 soldiers owned by Player 4
Ontario: 1 soldiers owned by Player 1
Peru: 1 soldiers owned by Player 5
Quebec: 1 soldiers owned by Player 1
Scandinavia: 15 soldiers owned by Player 3
Siam: 1 soldiers owned by Player 5
Siberia: 1 soldiers owned by Player 2
SouthAfrica: 1 soldiers owned by Player 1
SouthernEurope: 1 soldiers owned by Player 1
Ukraine: 4 soldiers owned by Player 1
Ural: 1 soldiers owned by Player 2
```

```
Venezuela: 1 soldiers owned by Player 1
WesternAustralia: 4 soldiers owned by Player 5
WesternEurope: 2 soldiers owned by Player 3
WesternUS: 1 soldiers owned by Player 1
Yakutsk: 1 soldiers owned by Player 2

Solders per player:
Player 2: 9 soldiers
Player 4: 9 soldiers
Player 1: 22 soldiers
Player 5: 11 soldiers
Player 3: 31 soldiers

Cards on hand:
Infatry: 3 cards
Cavalry: 1 cards
Artillery: 0 cards

Current turn: Player 3

You have already conquered a country this round.
Attack options:

1) Scandinavia (Player 3) with 15 soldiers to attack Iceland (Player 1) with 1 soldiers
2) Congo (Player 3) with 6 soldiers to attack SouthAfrica (Player 1) with 1 soldiers
3) NewGuinea (Player 3) with 7 soldiers to attack EasternAustralia (Player 5) with 1 soldiers
4) Scandinavia (Player 3) with 15 soldiers to attack GreatBritain (Player 1) with 1 soldiers
5) Scandinavia (Player 3) with 15 soldiers to attack Ukraine (Player 1) with 4 soldiers
6) NewGuinea (Player 3) with 7 soldiers to attack Indonesia (Player 5) with 3 soldiers
7) WesternEurope (Player 3) with 2 soldiers to attack SouthernEurope (Player 1) with 1 soldiers
8) Congo (Player 3) with 6 soldiers to attack EastAfrica (Player 1) with 1 soldiers
9) WesternEurope (Player 3) with 2 soldiers to attack GreatBritain (Player 1) with 1 soldiers
10) WesternEurope (Player 3) with 2 soldiers to attack NorthernEurope (Player 1) with 3
    soldiers
11) Scandinavia (Player 3) with 15 soldiers to attack NorthernEurope (Player 1) with 3 soldiers
12) NewGuinea (Player 3) with 7 soldiers to attack WesternAustralia (Player 5) with 4 soldiers

Which option do you choose?
```
Listing 1: LLM Prompt Example