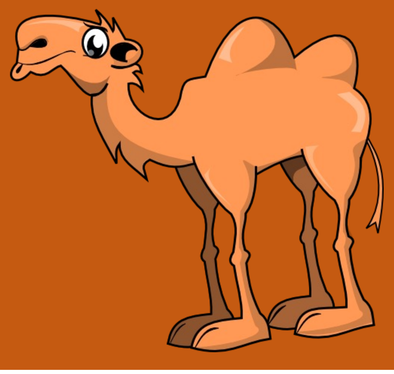


Implementation of an Intelligent Agent for the Camel Up! Board Game

Alec Gironda & Luke Lorenz

CSCI 0311 – Fall 2022
Department of Computer Science

Project Goal

The goal of this project is to develop an intelligent agent who can dethrone us as the ultimate Camel Up player. Our agent should be able to learn and adapt to different strategies over time and be able to make informed decisions based on the current game state.

Background

- Classic board games such as chess, checkers, Connect Four, Go, and Jeopardy have been used as benchmarks for AI progress.
- Modern dynamic board games present many advanced attributes, such as varying number of players, stochasticity, unknown information, and various levels of strategic depth and luck.
- Camel Up contains a restricted decision space, but many probabilistic events making it a novel challenge for AI development.
- In Camel Up, 2-8 players bet on five racing camels, trying to figure out which camel will win the race around the pyramid. The earlier you place your bet, the more you can win—assuming you guess correctly.
- While a probabilistic AI for Camel Up has been developed, we sought to combine a probabilistic heuristic with reinforcement learning for holistically better play. This offers an opportunity to push the envelope of novel AI in dynamic board game situations.

Methods

Our final model is an innovative combination of a neural network (NN) and a pseudo-expectimax agent.

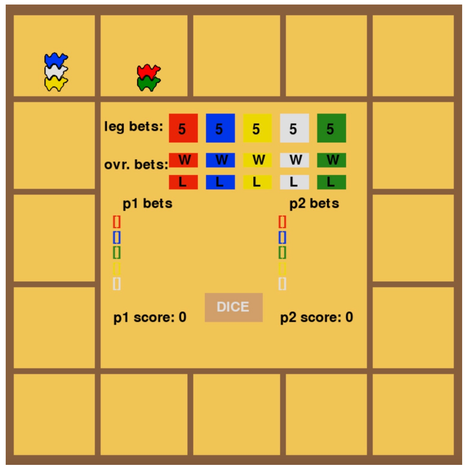
Our initial approach towards building our model focused solely on reinforcement learning. We started by pitting two random players against each other and simulating 1000 games. During those games, we stored key value pairs of the game state nodes visited, and the money earned at the end of the game. From there, we used the stored nodes as training for our model, with labels of the money earned at the end of the game. We continued to make new players in the same fashion, pitting our newest players against previous iterations, beginning with a random model and eventually simulating against a 12th generation player.

After limited success here, we decided to calculate the expected values (EVs) of leg bets as a heuristic in choosing moves. We calculated these EVs by considering all the permutations of dice that hadn't been rolled yet, and all the possible numbers that could be rolled on those (3 sided) dice. With n dice left, there are $n! \cdot 3^n$ possible ways to roll numbers on the dice. We simulated all of the camels' movement for each of these possibilities and calculated each camel bet's expected value with Equation 1.

We created our intelligent players to take the bet with the highest EV greater than 1, and if there were no bets with an EV greater than 1, the player would roll or make an overall winner or loser bet, for the EV of rolling is 1 and the EV of making an overall bet is often even less than that. This helped to decrease the decision space for the NN by reducing the number of possible children to choose from, reducing the role of our neural network to selecting overall race winners and losers.

We then developed our final model (SmartPlayer), using a NN trained from simulated games between a random player and our probabilistic agent. SmartPlayer relied on this neural network in situations where EV<1.

(A)



(B) EV's: [0.740, 0.533, 0.566, **1.599**, 1.06]

(C)

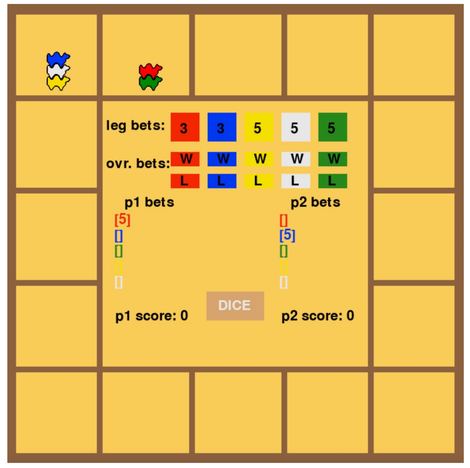


Figure 1. Playable interface for our board game. (A) shows the starting configuration of the board.(C) shows the resulting bet taken by the AI following our leg bet on the red camel based on the expected value (B) of betting on each camel.

$$EV = \frac{(camel\ wins)(leg\ bet\ payout)}{n! \cdot 3^n}$$

Equation 1. Leg bet expected value calculation

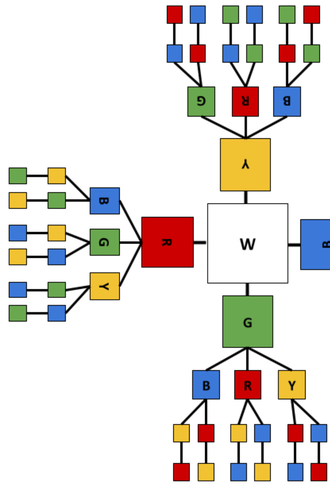


Figure 2. Possible permutations of dice when the white die is rolled first.

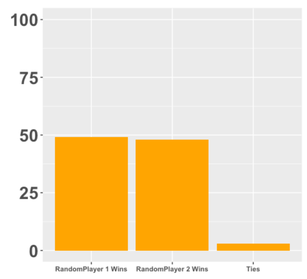


Figure 3. Result of 100 simulated 1v1 games between RandomPlayer and RandomPlayer

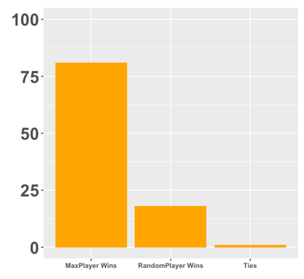


Figure 4. Result of 100 simulated 1v1 games between MaxPlayer and RandomPlayer

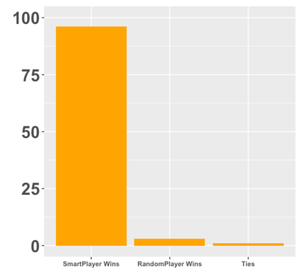


Figure 5. Result of 100 simulated 1v1 games between SmartPlayer and RandomPlayer

Methods (Cont.)

Results/Conclusions

For win/loss results, see Figures 3-5. For situational decision making, see Figure 1.

This project demonstrates the power of reinforcement learning and heuristics for rapidly learning board game strategy. Though our model achieved strong results, we believe that we only began to scratch the surface of what is possible in the realm of Camel Up. Training the model for longer, extending to 4 players, adding second place payouts, and adding oasis tiles come to mind to create our comprehensive AI. While the neural network only predicts overall winners and losers, we hope to expand the decision-making capabilities to some of the extensions mentioned above. We view this as a steppingstone for advanced machine learning techniques for even more complex board games in future research.

References & Acknowledgements

De Mesentier Silva, Fernando & Lee, Scott & Togelius, Julian & Nealen, Andy. (2017). AI-based playtesting of contemporary board games. 1-10. 10.1145/3102071.3102105.

Heyden, Cathleen. (2009). Implementing a Computer Player for Carcassonne.

Linderman, Michael. (2022). Middlebury College Artificial Intelligence Presentation "Introduction to AI"

This work was made possible by computational resources, data, and expertise provided by Professor Linderman.