BASIS – Bisca Agent Strategy Intelligent System

Autonomous Agents and Multi-Agent Systems – G30, Alameda

José Almeida
Departamento de Engenharia Informática
Instituto Superior Técnico
Lisbon, Portugal
jose.v.almeida@tecnico.ulisboa.pt

Pedro Moço
Departamento de Engenharia Informática
Instituto Superior Técnico
Lisbon, Portugal
pedro.moco@tecnico.ulisboa.pt

Tiago Teodoro
Departamento de Engenharia Informática
Instituto Superior Técnico
Lisbon, Portugal
tiagogteodoro@tecnico.ulisboa.pt

ABSTRACT

This project introduces a comprehensive and extensible multi-agent artificial intelligence system specifically designed for playing *Bisca*, a classic trick-taking game that has been passed down through generations. *Bisca* is played with a standard 40-card deck, and the primary objective is to accumulate points by winning tricks, which are rounds of play.

Our research focuses on the development of multi-agent systems and their application to the game of *Bisca*. Our main objective is to address the challenge of creating a robust framework that stimulates intelligent gameplay, where agents employ various decision-making strategies. While our primary goal is to facilitate agent-agent gameplay, we also enable interactions between agents and human players.

To achieve these goals, our approach entails the development of multiple constructs that effectively represent the *Bisca* domain. These constructs include *Card*, *Deck*, *Player*, *Game*, and *Agent*. The *Agent* construct allows for the implementation of diverse AI-driven strategies, leveraging the game state and opponent moves to make informed choices.

The contributions of this project are manifested through the development of a practical implementation of a multi-agent system for playing *Bisca*, accommodating a maximum of six players, of which only one can be human. This implementation demonstrates the interplay between traditional human players and intelligent agents in strategic card games. Additionally, this project serves as a foundation for future advancements in AI-driven card games. The modules and concepts developed herein can be reused in other game contexts. Consequently, this work has the potential to provide valuable insights for the development of AI-driven games beyond the scope of *Bisca*.

Lastly, this project sheds light on the wide range of applications for multi-agent systems and competitive AI agents in recreational activities. It highlights the possibilities of integrating advanced AI technologies into leisurely pursuits, emphasizing the potential for enhanced gameplay experiences and new dimensions of strategic engagement.

1. INTRODUCTION

The objective of this project is to develop a sophisticated multi-agent system designed to play Bisca with intelligence and employ diverse strategies for decision-making, leveraging information from the game-state and opponent moves. By implementing this system, we aim to facilitate a comparative analysis of different intelligent agent approaches and draw insightful conclusions regarding their behaviors within a dynamic environment where their decisions are interdependent.

Moreover, our focus extends beyond agent-agent gameplay. We also aim to provide a platform for agent-human interactions, offering valuable insights into the dynamics between traditional human players and intelligent agents in strategic card games. Our goal is to assess and compare the performance of various intelligent agent approaches from the perspective of an unsuspecting human player. This assessment involves measuring key metrics related to the agents' strategic quality, as well as evaluating the human player's ability to recognize and respond to the agents' strategies.

This project represents a significant contribution to the field of AI-driven card games by delivering a practical implementation of a multi-agent system for playing Bisca. The framework we develop serves as a robust foundation for further advancements and exploration within this domain, opening doors for future research and development endeavors.

1.1. The game of *Bisca*

Bisca is a trick-taking card game that has been played for generations and is popular in various parts of the world. The game is typically played with a standard deck of 40 cards, excluding the eights, nines and tens. It is commonly enjoyed by two to six players, although variations exist which can accommodate more players.

The objective of Bisca is to accumulate points by winning tricks, which are individual rounds of play. A trick consists of each player playing one card from their hand, and the player who plays the highest-ranked card of the led suit wins the trick and leads the next one.

At the beginning of the game, the dealer distributes a set number of cards to each player and places a card in the center of the table facing up – the trump. The number of cards dealt may vary depending on the number of players participating. The remaining cards form the stockpile, which is placed face-down in the center of the playing area.

The gameplay progresses with players taking turns clockwise. The first player to play a card in each trick is called the leader. Once all the players have played a card, the trick is won by the player who manages one of the following plays.

1. The highest-ranked card of the trump suit, if any were played;
2. The highest-ranked card of the led suit.

The winner of the trick collects the played cards and places them face-down in front of them, forming a won-tricks pile.

The game continues with players leading tricks and trying to win as many as possible. When the stockpile is depleted, and all players have played their remaining cards, the round ends. The accumulated points from the won tricks are tallied, considering the values assigned to certain cards. The specific point values for each card may vary depending on the region or variation of *Bisca* being played.

Typically, the game is played in a series of rounds until a predetermined point threshold is reached. The player or team with the highest accumulated points at the end of the game is declared the winner.

Bisca is a game that combines elements of strategy, card valuation, and predicting opponents' moves. It requires careful consideration of the cards played, the remaining cards in hand, and the overall game dynamics. The ability to assess the strengths and weaknesses of the opponents' hands and to strategically play cards to maximize point accumulation is crucial for success in *Bisca*.

1.2. *Bisca* Implementation

For our implementation we considered the "*Bisca de 3*", which is a variation of *Bisca* where each player holds three cards in their hand at the beginning of each trick. Additionally, we considered the following assignments of values to cards, is typically used in Portugal, with respect to the Spanish 40-card playing deck:

- Queen – 2
- Jack – 3
- King – 4
- Seven – 10
- Ace – 11

According to *Bisca* rules, number 2 cards of all suits are removed from the deck, in order to deal with the specific case in which $n \bmod 40 \neq 0$, where n is the number of players in the game, which pertains to the cases of $n = 3$ and $n = 5$. For all other allowed cases of n, this is not required.

1.3. Multi-Agent Systems

Multi-Agent Systems (MAS) represent a significant area of research in the field of Artificial Intelligence (AI) that focuses on the study of intelligent agents interacting within complex environments. These systems consist of multiple autonomous agents, each possessing its own goals, knowledge, and decision-making capabilities, and are designed to collaborate, compete, or both in pursuit of common or individual objectives.

The application of MAS in various domains has gained significant attention, particularly in strategic games such as card games. Card games provide an interesting and challenging platform to study MAS, as they require players to make strategic decisions, adapt to uncertain and dynamic environments, and interact with other players to achieve desired outcomes.

MAS in card games offer several advantages over traditional single-agent approaches. Firstly, they allow for the modeling of complex interactions and social dynamics that arise when multiple agents with distinct strategies and behaviors coexist. This enables a more realistic representation of real-world scenarios and human-like interactions. Secondly, MAS provides a platform to study and develop intelligent agents capable of adapting and evolving their strategies based on the actions and behaviors of other agents, leading to more dynamic and challenging gameplay. Lastly, the integration of MAS in card games promotes the exploration of cooperative and competitive strategies, fostering the development of advanced AI techniques and contributing to the field's overall progress.

The integration of MAS in card games has led to significant advancements in AI-driven gameplay. Researchers have explored various approaches to designing intelligent agents, such as rule-based systems, machine learning algorithms, and deep reinforcement learning models. These agents can learn from past experiences, analyze game states, predict opponents' moves, and strategize accordingly. By incorporating MAS into card games, developers aim to create challenging opponents for human players, facilitate skill development, and enable new avenues for AI-assisted gameplay.

1.4. Related Work

Multiple card games have been the subject of several studies in the field of multi-agent systems. Previous work has explored the development of intelligent agents using different heuristics, such as rule-based strategies, Monte Carlo Tree Search, and reinforcement learning[1]. These studies have shown promising results in terms of the performance and adaptability of the developed agents. Moreover, decision theory and game theory[2] have also been applied to analyze the strategic interactions between agents in card games. Specifically, the concept of Nash equilibrium has been used to study the equilibrium solutions in card games.

We built upon these studies taking into consideration our time constraints. We have implemented a case study for a multi-agent system that plays *Bisca* (a game which has not been extensively studied) using rule-based strategies, the most adequate form of intelligence considering the time constraints.

2. APPROACH

In this study, we adopt an approach that centers around the design and implementation of the multi-agent platform specifically tailored for the game of Bisca.

2.1. Environment

The environment is dynamic, in such a way that each agent's decisions impact all the other agents' decisions and outcomes.

We can divide the environment and provide a definition as follows.

- **Game**: The game of *Bisca*, a trick-taking game, is chosen as the target environment, in a format which entails only one player per team (as opposed to *Bisca* variants which feature duos of players), and a variable number of total players.
- **Rules**: The rules of *Bisca*, including card rankings, trick-taking/round mechanics, and point calculations will be implemented to create an accurate representation of the game environment.

2.2. Agents

This study focuses on the development of a multi-agent platform for playing *Bisca*, where each implemented agent operates as a Rational Agent. The Rational Agents in this context engage in strategic reasoning and decision-making processes. Their actions are guided by a utility function, which aims to maximize the expected reward or outcome for the agent. Moreover, these agents take into consideration the decisions made by other agents, recognizing that these decisions have a direct impact on the state of the environment.

In the context of *Bisca*, the desired outcome for each agent is defined as winning the trick. Consequently, the utility function employed by each agent is designed to prioritize achieving this outcome. By incorporating game-theoretic principles, such as strategic reasoning and utility maximization, the multi-agent platform strives to emulate human-like decision-making processes and enhance the overall gameplay experience.

As such we can define this system as a Normal Form Game:

- **Players**: The set N consists of n players involved in the game, each indexed by i. In the case of the *Bisca* game, the players can represent the intelligent agents participating in the gameplay.
- **Action Sets**: The set A is defined as the Cartesian product of $A_1 \times \cdots \times A_n$, where A_i represents the finite set of actions available to player/agent i. In *Bisca*, each player has a set of actions representing the cards in their hand that they can play in a given trick.
- **Action Profile**: An action profile refers to a specific combination of actions taken by all players. It is denoted by the vector $a = (a_1, \ldots, a_n) \in A$. In *Bisca*, an action profile represents the cards played by each player in a particular trick.
- **Utility Function**: The utility function $u = (u_1, \ldots, u_n)$ assigns a real-valued utility or payoff to each player in the game. Here, $u_i : A \longmapsto R$ represents the utility function for player i, mapping the action profiles to real numbers. In *Bisca*, the utility function determines the rewards or scores obtained by each player based on the outcome of the trick (e.g., winning the trick).

The implemented agents in this system primarily fall into the category of purely reactive agents, with one exception being the *GreedyCountingAgent*. The *GreedyCountingAgent* differs from the other agents as it maintains an internal state by keeping track of the counts of previously played cards. While this agent deviates from being purely reactive, the remaining agents make decisions solely based on the current state of the environment.

The distinction lies in the fact that the reactive agents, excluding the *GreedyCountingAgent*, do not utilize or rely on any stored information or internal state beyond the immediate observation of the game state. Their decision-making process is purely reactive, where the actions they take are determined solely by the current state of the environment, without considering any historical or past information.

The following section delves deeper into the details of the implemented player types, including the *Human* player and agents.

Human

The *Human* player in this system awaits input from a user and computes the entry based on the provided input. It acts as an interface between the human player and the game, facilitating their participation in the gameplay. The decisions made by the *Human* are dependent on the choices made by the user, allowing for interactive and engaging gameplay experiences.

RandomAgent

The *RandomAgent* operates by randomly selecting a card from all the available possibilities. It does not consider any specific strategies or game state information when making its decisions. The selection of a random card introduces an element of unpredictability to its gameplay, adding diversity and variability to the agent's actions.

SimpleGreedyAgent

The *SimpleGreedyAgent* follows a straightforward strategy of always choosing the highest-ranked card available. In other words, it selects the card that yields the highest point return. By prioritizing the highest-ranked cards, this agent aims to maximize its chances of winning tricks and accumulating points.

MinimizePointLossGreedyAgent

The *MinimizePointLossGreedyAgent* employs a strategy focused on minimizing the potential loss of points. If it is the first player in a round, it plays the card with the highest rank. However, if it is not the first player, it assesses the current best card in play and attempts to select a card that can defeat it. It evaluates the cards based on their potential as trump cards or the number of points they offer. If no card can win, the agent evaluates the cards that are likely to lose and chooses the one that would result in the least loss of points and trump cards.

MPLGreedyTrumpSaveAgent

In a manner similar to that of *MinimizePointLossGreedyAgent*, *MPLGreedyTrumpSaveAgent* prioritizes minimizing point loss. When playing first in a round, it plays the card with the highest rank. If it is not the first player, it aims to choose a card that can defeat the current best card in play, taking into consideration whether it is a trump card or its point value. This agent favors saving trump cards for later if possible, potentially sacrificing immediate wins for future strategic advantages.

MPLGreedyTrumpBasedAgent

MPLGreedyTrumpBasedAgent places a high value on trump cards and consistently attempts to play them if possible. It only deviates from playing trump cards if there are none available or if playing a trump card does not guarantee winning the current round. In such cases, it prioritizes selecting the least valuable card. This agent demonstrates a greediness towards playing trump cards and emphasizes their strategic significance in gameplay.

GreedyCountingAgent

The *GreedyCountingAgent* utilizes a counting mechanism to keep track of the cards that have already been played. When playing first in a round, it assesses the probability of losing the round versus gaining points and chooses the highest possible combination of both factors. When playing last, it selects the card that offers the highest point return, aiming to save trump cards for future rounds. If it is one of the middle players, it selects a card that can yield points while considering the risk of losing the round. This agent's decision-making process combines card counting, risk evaluation, and point-maximizing strategies.

2.3. System Architecture

The system adopts a modular architecture, to promote flexibility and extensibility. In general, the system architecture focuses on separation of concerns, using a design pattern similar to MVVM to separate UI code from the internal game-engine and agent-logic.

This separation between UI and engine, functioning as two completely independent systems, allowed us to build a simulation framework running on top of the engine, which allows us to retrieve important metrics by simulating large batches of games.

The engine is composed mainly by the following abstractions.

- **Card**: Includes simple attributes that describe a traditional card such as rank, suit, value and points.
- **Deck**: Features logic to initialize the set of cards, shuffle the cards and draw a card.
- **Player**: The generic player representation. Includes attributes such as the player's identification, current hand of cards, logic for playing a card, drawing a card from the deck, calculating scores and all generic auxiliary methods used in the decision-making process.
- **Agent**: All classes that inherit from **Player**, including human agents and strategy-based AI agents. Includes additional internal logic for making intelligent decisions based on the game state, opponent moves and any predetermined strategies.
- **Game**: Responsible for handling game logic. Initializing a deck, registering players, managing turns, determining winners and updating the game state. Features a game-loop, which iterates through rounds and player-specific turns. In essence, a round proceeds as follows.
    1. Players draw, starting from the previous round winner proceeding in a clockwise manner.
    2. Each player takes a turn playing a card, after which a winner is determined based on the highest same-suit card value or the highest trump-card value.
    3. The winner receives the played cards, and the next round starts, with this process being repeated until all cards are played.

The graphical user-interface was developed by making use of the *pygame_cards* library for *Python* and provides an interactive and intuitive platform for a human player to engage with the game and/or observe agent gameplay.

2.4. Human Interaction

The case of human-interaction in order to extend the platform to real users required particular attention. Given that, internally, a *Human* player requires input from the user in order to choose a card from the player's hand, this would create an unwanted dependency from the engine to the UI. Given this situation, our solution was to implement the *Human* player in such a way that the interface's consumer (i.e. any program using the engine, e.g. the UI) can register input-handling callback functions, which are consulted by the *Human* player when input is required for the given player. This integrates seamlessly with the rest of the engine, as the registered input-handler is called in the scope of *action*, given that *Human* is treated as any other *Player* object.

In the case of a simple engine-interface consumer (e.g. *example.py*), the registered input-handler can be something as simple as a function which uses *Python*'s *input* function to request a card-index based on the player's hand. On the other hand, in the case of the UI, the registered input-handler is a much more complex function which manipulates *PyGame* components to gather the user's card-choice from a mouse-click. This exact duality justifies our interest in the proposed solution, as this architecture can handle complex scenarios just as well as simple ones.

2.5. Display Modes

The program provides the option to be executed with two distinct user interfaces (UI): the *PyGame* graphical user interface (GUI), which offers an interactive and captivating experience, and the simulation framework, a command-line tool specifically designed for executing extensive sets of games and extracting data for subsequent statistical analysis of outcomes. These alternative interfaces cater to different usage scenarios and allow users to choose between a visually immersive gameplay environment or a more streamlined, data-driven simulation approach.

3. EMPIRICAL EVALUATION

3.1. Motivation

In order to assess the effectiveness and performance of the multi-agent system at playing the game of *Bisca*, an empirical evaluation has been conducted. Through this process, we aim to validate the project's objectives by measuring key metrics related to gameplay quality, agent decision-making and player experience. With this goal, the following metrics were proposed.

1. **Agent win-rate**: Calculated by comparing the number of games won by the agents against the total number of games played. A higher win-rate indicates the effectiveness of the agent in playing *Bisca*.
2. **Point accumulation**: Refers to the average number of points accumulated by the agents during a game of *Bisca*. Winning tricks/rounds is crucial for victory, as such, this metric assesses an agent's effectiveness in strategically winning tricks which carry higher point values. This directly indicates better decision-making and skill in maximizing points.
3. **Comparison with human players**: By recording win-rate and point accumulation metrics for human players, and comparing them with the performance of agents, we can assess (1) how well agents do against human players and (2) how human-like agent gameplay is.
4. **Player perception**: Collecting feedback and ratings from human players who have interacted with the system will allow us to evaluate indicators such as (1) perceived challenge, (2) fairness, and (3) enjoyment in playing against agents.

Measuring these metrics allows us to provide quantitative and qualitative insights into the capabilities and effectiveness of the multi-agent system in playing *Bisca*, while also generating insight as to whether or not interaction with the platform is enjoyable from a recreational point-of-view.

3.2. Research Questions

In an effort to guide the empirical evaluation of the multi-agent system when it comes to playing *Bisca*, the following questions have been set as goals.

1. How effective are the agents in playing *Bisca* when compared to each other? This question aims to compare the performance of different agents implemented in the system by analyzing win-rate and point accumulation metrics.
2. How do the agents perform against human players in *Bisca*? We can compare the win-rate and point accumulation metrics between humans and agents in order to assess how well the agents fare against human players.
3. How does the performance of the agents vary with different player configurations in *Bisca*? In the scope of this goal, we'll analyze the win-rate and point-accumulation metrics for each agent under different player configurations (i.e. 2 players, 3 players, 4 players, 5 players, 6 players), aiming to identify trends or patterns in the agents' performance given different game dynamics.
4. What is the perceived challenge, fairness, and enjoyment of playing against the agents? By gathering qualitative feedback from human players who have interacted with the system, such as ratings and opinions on factors such as challenge, fairness, and enjoyment, we can assess the player perception of playing against the agents and the overall experience of the multi-agent platform.

3.3. Data Collection

In order to gather necessary data for empirical analysis, a series of simulations were conducted using a simulation framework we developed for the multi-agent system. The simulations were performed in as controlled as possible of an environment, though it is difficult to ensure consistency given the stochastic nature of the game of *Bisca*, as well as the stochastic nature of some of the components used in various agents (e.g. *RandomAgent*, tie-breaks in other agents).

The simulations were conducted with all possible agent configurations, including 2 players, 3 players, 4 players, 5 players and 6 players, for all combinations of players. In order to achieve some level of statistical confidence regarding agent behavior, despite being quite resource-intensive, a sample size of 100000 simulations (per-combination, per-player group size) has been chosen.

It is important to note that the simulation framework does not take *Human* players into account. With regard to the previously defined categories of metrics, the following section overviews the data collected.

During simulations, two main sets of metrics were captured on a per-agent basis.

1. **Agent win-rate:** The number of games won, lost, and tied by each agent, along with the subsequent calculation of win-rate, as a metric of effectiveness.

2. **Point accumulation**: For each game and each trick, the point accumulation of each agent was tracked, providing insights into the agents' decision-making process and strategy.

Regarding human-related data, a single main set of information was collected.

1. **Player perception**: Feedback and ratings were collected from the human participants who interacted with the system, allowing us to assess the recreational value of interacting with the multi-agent system.

3.4. Data Analysis

Comparative analysis of agents

In this section, we analyze the results obtained from running simulations with different player group sizes and agent combinations. To evaluate performance of each player type, we consider several metrics, including wins, losses, draws, average points per-game, highest game turnover and average points per-trick. These metrics provide insights into the effectiveness and efficiency of the different player types in the game.

To obtain these metrics, we conducted a large number of simulations, as mentioned in **Data Collection**. In each simulation, we randomly assigned players to different agent types and had them play the game. After each simulation, we recorded the required metrics.

Once all simulations for a given group-size were completed, we aggregated the results for the group-size, calculating the total value for each metric, for each player within that player configuration. This process is repeated for a group-sizes because agents may have different performance levels based on the number of players in the game. Additionally, by performing this aggregation process, we're able to make comparisons between different group sizes. Tables 1 through 5 represent the performance indicators calculated for each group-size. The automation for the described process can be found in *analysis.py*, and the data used is present in *report_analysis.json*.

Agent	Wins	Losses	W/L Ratio	Draws	Avg. Points/ game	Highest game turnover	Avg. points/ trick
GreedyCountingAgent	**331739**	**158190**	**2.10**	10071	**64.83**	**118**	**6.11**
MPLGreedyTrumpSaveAgent	327970	161953	2.03	**10077**	59.64	118	5.95
MinimizePointLossGreedyAgent	323719	166247	1.95	10034	59.24	116	5.96
MPLGreedyTrumpBasedAgent	273625	216121	0.17	10254	55.17	113	5.87
SimpleGreedyAgent	112881	380374	0.30	6745	41.15	117	5.38
RandomAgent	103197	390246	0.26	6557	40.74	110	5.35

Table 1 – Aggregated metrics for a group-size of 2, sorted by most wins.

Agent	Wins	Losses	W/L Ratio	Draws	Avg. Points/ game	Highest game turnover	Avg. points/ trick
MPLGreedyTrumpSaveAgent	**432508**	**554749**	**0.78**	**12743**	41.34	114	9.88
MinimizePointLossGreedyAgent	415170	572440	0.73	12390	40.44	**116**	9.84
GreedyCountingAgent	410670	577349	0.71	11981	**42.59**	114	**10.64**
MPLGreedyTrumpBasedAgent	352232	635911	0.55	11857	36.07	117	9.47
RandomAgent	188182	803622	0.23	8196	32.87	118	10.05
SimpleGreedyAgent	169242	823075	0.21	7683	30.10	110	9.17

Table 2 – Aggregated metrics for a group-size of 3, sorted by most wins.

Agent	Wins	Losses	W/L Ratio	Draws	Avg. Points/ game	Highest game turnover	Avg. points/ trick
MPLGreedyTrumpSaveAgent	**305726**	**682424**	**0.45**	**11850**	**30.75**	108	12.13
GreedyCountingAgent	302573	685616	0,44	11811	30.70	110	**12.15**
MinimizePointLossGreedyAgent	294523	693807	0.42	11670	30.30	108	12.09
MPLGreedyTrumpBasedAgent	254169	734798	0.35	11033	28.26	104	11.60
RandomAgent	166030	825667	0.2	8303	25.25	**116**	11.90
SimpleGreedyAgent	146004	846180	0.17	7816	25.24	102	11.34

Table 3 – Aggregated metrics for a group-size of 4, sorted by most wins.

Agent	Wins	Losses	W/L Ratio	Draws	Avg. Points/ game	Highest game turnover	Avg. points/ trick
MPLGreedyTrumpSaveAgent	**113564**	**381460**	**0.30**	**4976**	25.37	111	15.26
GreedyCountingAgent	112723	382302	0.29	4975	**25.45**	110	**15.25**
MinimizePointLossGreedyAgent	110164	384747	0.29	5089	25.13	104	15.23
MPLGreedyTrumpBasedAgent	100016	395256	0.25	4728	24.36	**114**	14.91
RandomAgent	84522	411516	0.21	3962	22.55	110	**15.64**
SimpleGreedyAgent	65438	430876	0.15	3686	19.69	113	14.21

Table 4 – Aggregated metrics for a group-size of 5, sorted by most wins.

Agent	Wins	Losses	W/L Ratio	Draws	Avg. Points/ game	Highest game turnover	Avg. points/ trick
MPLGreedyTrumpSaveAgent	**18129**	**81088**	**0.22**	783	**21.07**	**120**	20
RandomAgent	17795	81437	0.22	768	20.30	111	**21.26**
GreedyCountingAgent	17308	81907	0.21	785	20.69	116	20.51
MinimizePointLossGreedyAgent	17296	81891	0.21	**813**	20.80	112	19.89
MPLGreedyTrumpBasedAgent	15988	83225	0.19	787	20.54	116	19.64
SimpleGreedyAgent	11241	88151	0.13	608	16.59	115	18.63

Table 5 – Aggregated metrics for a group-size of 6, sorted by most wins.

Considering the data obtained from the simulations, we delve further into the discovered trends.

Across all player group sizes, the *MPLGreedyTrumpSaveAgent* consistently displayed competitive performance, achieving high numbers of wins and maintaining favorable win/loss ratios. *GreedyCountingAgent* also consistently demonstrated strong performance, particularly in terms of average points per game and game turnover. This matches our expectation, as *MLPGreedyTrumpSaveAgent* uses one of the most common strategies employed by human players, to save trumps for final round, and *GreedyCountingAgent* has a vague notion of other player's cards, being able to compute the objectively-best card from it's point-of-view.

On the other hand, *SimpleGreedyAgent* consistently performed poorly as, as expected, given the naïve strategy of always using the highest-valued card at hand. *RandomAgent*'s strategy is also far from performant, as it consistently appears at the lower-end of the leaderboard, These agents had lower win rates, unfavorable win/loss ratios and lower average points per game, suggesting less effective gameplay strategies compared to the top-performing agents.

It's important to note that, as the group size increased, the win/loss ratios generally decreased for all agents. This indicates that the game becomes more challenging as the number of players increases, which would require more intelligent agents to adjust accordingly.

Qualitative analysis from user feedback

The qualitative user feedback collected from participants who interacted with our platform provides valuable insights into their experiences and perceptions. As such, we present an overview of the feedback, highlighting key findings and observations.

Player	Perceived Challenge	Fairness	Enjoyment	Detected Strategy	Bisca Experience
Player 1	2	1	5	1	0
Player 2	1	1	3	1	2
Player 3	2	5	5	1	2
Player 4	3	1	4	1	2
Player 5	3	2	4	4	4
Player 6	2	1	3	3	4
Player 7	4	2	5	1	4
Player 8	5	3	3	2	2
Player 9	3	1	4	3	2
Player 10	4	2	4	3	2
Player 11	1	1	2	5	5
Player 12	1	1	2	5	5
Player 13	2	1	3	4	5
Player 14	2	1	3	5	5
Player 15	4	1	5	3	5
Player 16	3	1	4	4	2
Player 17	5	2	4	1	2

Table 6 – Overview of collected user feedback, each property is rated from 0 to 5.

It is important to note that each participant played, at least once, against each of the agents available in the platform.

Participants were asked to evaluate the perceived challenge of the platform. The average rating for perceived challenge was around 3, indicating a moderate level of challenge overall, with some players finding the platform relatively easier while others considered it more challenging.

The players' experience levels in Bisca ranged from 0 (inexperienced) to 5 (expert), which may have influenced their perception of the platform's difficulty. For example, Player 1, who had limited experience, had a different perspective compared to Player 15, who exhibited a higher level of expertise. This variation in Bisca experience influenced their perceptions and feedback on different aspects of the platform.

Overall, the participants reported a positive level of enjoyment while engaging with the platform. The average enjoyment rating was 3.7, indicating a generally satisfying experience.

Additionally, we asked participants to identify and provide feedback on whether or not they understood the strategies employed by the agents they encountered from interacting with them. The average rating for strategy detection was 2.7, though this metric is very clearly separated. Players with higher experience levels in Bisca displayed a deeper understanding of the strategies employed by the agents they faced, while others recognized primary strategies to varying extents.

3.5. Discussion

The empirical evaluation of the multi-agent platform provided valuable insights into its effectiveness, performance and player-experience. In this section, we aim to discuss the findings and interpretation of our evaluation, in the context of the research questions proposed in section **3.2.**

How do different agents perform in the game of *Bisca*?

Our evaluation revealed distinct performance patterns among the different agents. *MPLGreedyTrumpSaveAgent* consistently achieved high numbers of wins and favorable win/loss ratios. This agent's strategy of saving trumps for the final round aligned with human player strategies, contributing to its effectiveness.

GreedyCountingAgent also demonstrated strong performance, particularly in terms of average points per game and game turnover. Its ability to compute the objectively best card based on available information provided an advantage in decision-making.

On the other hand, *SimpleGreedyAgent* and *RandomAgent* consistently performed poorly. *The SimpleGreedyAgent's* strategy of always using the highest-valued card at hand proved to be naive and ineffective in maximizing points and winning games. The RandomAgent's random decision-making approach resulted in suboptimal gameplay strategies. These findings emphasize the importance of intelligent decision-making and strategic planning for better performance in Bisca, as agents employing well-thought out strategies very clearly overpower other agents.

How does the performance of the agents vary with different player group-sizes?

Our evaluation demonstrated that the performance of the agents varied with different player group sizes. As the number of players increased, the win/loss ratios for all agents decreased. This suggests that the game becomes more challenging with a higher number of players, requiring more intelligent agents to adapt to the complexities. The performance difference among the agents became less pronounced in larger player groups, indicating the need for further research and development of agents that can effectively handle these scenarios and devise better strategies for highly-populated games.

How do players perceive the gameplay experience provided by the multi-agent system?

The qualitative analysis of user feedback provided insights into the players' experiences and perceptions. Participants generally found the platform moderately challenging, with variations based on their individual experience levels in Bisca. The enjoyment ratings were generally positive, indicating that the multi-agent system provided a satisfying and engaging experience for players. However, there were differences in the participants' ability to detect the strategies employed by the agents. Players with higher experience levels in Bisca displayed a deeper understanding of the agents' strategies, while others recognized primary strategies to varying extents. This finding suggests that the agents' strategies could be further improved to make them more transparent and understandable for all players, regardless of their experience level, in order to increase the platform's recreational value.

Other observations

An interesting observation from our evaluation is the presence of an inherent bias in the game of Bisca when the starting positions of players are not shuffled during the simulation process. We noticed that the last player consistently had an advantage and tended to come out on top, regardless of the type of agent being used. This finding raises intriguing questions about the game's design and dynamics. It suggests that the order in which players take their turns may have a significant impact on the outcome, potentially favoring the player who goes last. Further investigation is required to explore this phenomenon and understand its underlying causes. This observation adds an additional layer of complexity to the game and highlights the importance of considering the fairness and balance of turn-based games like Bisca.

3.6. Limitations

This section discusses several limitations encountered during the empirical analysis, highlighting factors that may have influenced the data collection process and the interpretability of the results.

Limited data for human players

Gathering a significant amount of data for human players posed a challenge compared to simulating batches of games with agents. Human participants may have limited availability or interest in participating in numerous game sessions, leading to a relatively smaller sample size for human player data. As a result, the analysis focuses more heavily on data collected from agent simulations, potentially misrepresenting values of agent performance vs. human players, limiting the representativeness of human player performance.

Dependency on user experience

The data collected from human players is inherently dependent on their individual experience and familiarity with the game of Bisca. Player performance and decision-making can be influenced by factors such as their level of expertise, strategic understanding, and overall skill in playing the game. Participants with extensive experience may exhibit more nuanced gameplay and achieve higher win-rates or point accumulations, while novice players may struggle to optimize their strategies. This dependence on user experience introduces a potential confounding factor that may impact the reliability and comparability of the results, making it important to interpret the findings with caution.

Inconsistency due to stochastic nature of *Bisca*

The stochastic nature of the game of Bisca contributes to inherent variability and uncertainty in the outcomes of individual games. The game involves elements of chance and randomness, including the shuffling of cards and the distribution of hands. Despite conducting a significant number of simulations or game sessions, the collected data may exhibit inherent inconsistency due to the influence of these random factors. Performance metrics of both agents and human players may fluctuate across different games, making it challenging to draw definitive conclusions or generalize the findings to all possible scenarios.

The existence of these limitations is acknowledged as looking deeper into them can provide insights into potential biases influencing the empirical analysis. Some of the limitations can be attributed to the inevitable nature of the game, but by recognizing them we can better interpret and generalize the obtained results, which in turn allows us to provide a more comprehensive assessment of the project's validity.

4. CONCLUSION

In this paper, we presented a multi-agent platform for playing the game of Bisca and conducted an empirical evaluation to assess its effectiveness and performance. Our evaluation focused on key metrics related to gameplay quality, agent decision-making, and player experience. Through the analysis of these metrics and the collection of qualitative feedback from human players, we gained valuable insights into the capabilities and user perception of the multi-agent system.

Key findings are the competitive performance of rule-based agents (*MPLGreedyTrumpBasedAgent*, *GreedyCountingAgent*) over other considered agents and the variation of agent performance and distribution of wins and losses as player-count varies.

Additionally, qualitative analysis of user feedback allowed us to infer that users feel a moderate level of challenge and overall satisfaction with the platform, indicating a positive recreational experience.

Overall, the *BASIS* multi-agent platform for playing Bisca shows promise as a recreational tool and delivers a foundation for future developments in the field of multi-agent systems and game playing.

5. REFERENCES

1. Barros, P., Tanevska, A., & Sciutti, A. (2021). Learning from Learners: Adapting Reinforcement Learning Agents to be Competitive in a Card Game. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 2716-2723). Milan, Italy. doi: 10.1109/ICPR48806.2021.9412807.
2. Parsons, S., & Wooldridge, M. (2002). Game Theory and Decision Theory in Multi-Agent Systems. Autonomous Agents and Multi-Agent Systems, 5, 243-254. doi: 10.1023/A:1015575522401.