

Playing Sueca with Autonomous Agents

Frederico Silva
Instituto Superior Técnico
Lisbon, Portugal
fredrico.m.silva@tecnico.ulisboa.pt

Pedro Mateus
Instituto Superior Técnico
Lisbon, Portugal
pedro.r.mateus@tecnico.ulisboa.pt

Guilherme Leitão
Instituto Superior Técnico
Lisbon, Portugal
guilhermeleitao0202@tecnico.ulisboa.pt

ABSTRACT

This project introduces a multi-agent artificial intelligence system specifically designed for playing *Sueca*, a traditional card game relatively popular in Portugal. The game is played with a trimmed 40-card deck, and the primary objective is to work with a partner and accumulate points as a team.

Our research focuses on the development of multi-agent systems and their application to the game of *Sueca*. Our main objective is to create a framework that simulates intelligent gameplay, where agents make use of several decision-making strategies. While our primary goal is to simulate agent-agent gameplay, we've also added agent-human interaction to allow for human players to participate in the game against the agents.

To achieve these goals, our technical approach is based on the development of 4 key constructs that effectively represent the game's domain. These constructs are: **Card**, **Player**, **Game**, and **Team**. The **Player** construct is the one that allows for the implementation of diverse strategies, leveraging the game state and opponent/partner moves to make informed and logical choices.

The contributions of this project mainly consist on the development of a practical implementation of a multi-agent system for playing the game of *Sueca*. This implementation mostly demonstrates the interplay between intelligent agents in strategic card games (with a twist of some human interplay as well).

Finally, the project aims to shed 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 multi-agent system with the aim of playing *Sueca* intelligently and employing multiple decision-making strategies, using information from the game-state and opponent/partner moves. After implementing this framework, we aim to facilitate a comparative analysis on different intelligent agent approaches and take conclusions regarding their effectiveness.

This project proposes to serve as a basic framework of a practical implementation of a multi-agent system for playing *Sueca*.

1.1 The rules of Sueca

Sueca is a card game that is popular in various countries, especially in Portugal and Brazil. The game is typically played with a standard deck of 40 cards, excluding the **eights**, **nines**, and **tens** (and **jokers**). *Sueca* is typically played by four individuals, playing in teams of two, but variations exist for different numbers of players, although these are not going to be considered in this project.

The objective of *Sueca* is to accumulate points by winning individual rounds of play. A round consists of each player playing one card from their hand, and the player who plays the highest-ranked card of the led/trump suit wins the round and leads the next one. The led suit is simply the suit of the first card played in that round.

At the beginning of the game, the dealer (who we can assume is one of the four players chosen at random) shuffles the deck and distributes 10 cards to each player, resulting in each player holding their entire hand for the round. Unlike similar games such as *Bisca*, for example, there is no stockpile, and the trump suit is determined by the last card dealt, which is shown to all players before being added to the dealer's hand.

Gameplay progresses with players taking turns clockwise. The first player to play a card in each round is called the leader. The round is won by the player who manages one of the following plays:

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

The winner of the round collects the played cards and places them face-down in front of them.

The game continues with players leading rounds and trying to win as many as possible. Once all cards have been played, the round ends. The accumulated points from the won rounds are tallied, considering the values assigned to each card:

- **Aces** - 11 points;
- **Sevens** - 10 points;
- **Kings** - 4 points;
- **Jacks** - 3 points;
- **Queens** - 2 points;
- **Sixes/Fives/Fours/Threes/Twos** - 0 points.

Keep in mind that the ranking of cards is equal to their values as ordered above, with the cases of the **Sixes-Twos** being assigned a rank of most to least in the order presented in relation to each other.

At the end of the game, the team with the highest accumulated points is declared the winner.

Sueca 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 opponent and partner's hands and to strategically play cards to maximize point accumulation is crucial for success in this game.

1.2 Related Work

We can find a variety of research on the development of intelligent agents for playing card games. What caught our eye, however, was the work of [Isabel Nogueira Ferreira 2015], who developed a multi-agent system for playing the card game *Sueca*. From here we took inspiration for the development of our agents and the overall multi-agent system.

The development of intelligent agents for playing card games is a well-established field in artificial intelligence. For the conceptual (i.e non-code related) development of our agents we take inspiration from the work of [Almeida et al. 2023], who developed a multi-agent system for playing the card game *Bisca*. The agents in their system were also designed to play the game with different strategies, such as randomly selecting cards, applying the greedy strategy, minimizing point loss, etc.

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 *Sueca*.

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 *Sueca*, is chosen as the target environment, in a format which always entails two teams, each with two players.
- **Rules** - The rules of *Sueca*, including card rankings, round mechanics and point calculations, will be implemented to create an accurate representation of the game environment.

2.2 Agents

In the environment there will be Rational Agents. For some of these, their actions will be guided by a utility function, defined according to the selected strategy. The desired outcome for the agents is to win the game by accumulating more points than the opponent team. The utility functions used by each agent will prioritize achieving this specific outcome, through the use of different strategies.

For simplicity's sake, we've aggregated the strategies by team, meaning that each team's members will share the same strategy and can't diverge from one another. This way we'll be able to evaluate these strategies team-wise. It is noteworthy to mention that the agents are implemented to follow the rules of the game, which means that they will not be able to cheat or break the rules in any way.

We can define this system as the following *Normal Form Game* (N, A, u) :

- **Players:** The set N consists of the 4 players involved in the game, each indexed by i . In the case of *Sueca*, the intelligent agents participating in the gameplay are the players.
- **Action Sets:** The set A is defined as the Cartesian product of $A_1 \times \dots \times A_n$, where A_i represents the finite set of actions available to agent i . In the context of *Sueca*, each player's set of actions correspond to the cards in their hand that they can play in a certain round. On a similar note, an action profile refers to a specific combination of actions taken by all players. It is denoted by the vector $\mathbf{a} = (a_1, \dots, a_n) \in A$. In *Sueca*, an action profile represents the cards played by each player in a given round.
- **Utility Function:** The utility function $\mathbf{u} = (u_1, \dots, u_n)$ assigns a real-value utility (in this case) to each player in the game. Here, $u_i : A \rightarrow \mathbb{R}$ represents the utility function for player i , mapping the action profiles to real numbers. In *Sueca*, the utility function determines the scores obtained by each player based on the outcome of the round (e.g., winning or losing the round).

The following sections aim to summarize the different agent strategies implemented, ordered by complexity.

Random Agent

This is the simplest agent, as it randomly selects a card to play from its hand. This agent serves as a baseline to evaluate the performance of more sophisticated strategies.

Greedy Agent

The Greedy agent uses a straight forward strategy of always playing the highest ranked card available, it doesn't consider the cards that have previously been played in round.

MaximizeRoundsWon Agent

The MaximizeRoundsWon Agent aims to win as many rounds as possible. If it's the first turn, the agent plays the highest ranked card within its hand. Subsequently, the following strategic approach is adopted: if the agent's team is leading the round, it plays the lowest card that still ensures a win; if its team is behind, it plays the lowest card that can potentially win the round, preserving better cards for future rounds. This strategy aims to maintain a strong hand towards the game's end, ensuring that the agent can win crucial rounds later in the game when it might matter most.

MaximizePointsWon Agent

The MaximizePointsWon Agent is designed to maximize the points won in each round. If it's the first turn, the agent plays the highest card within its hand. For subsequent turns, the agent follows this strategy: if the agent's team is leading the round, it plays a card that maximizes the round's point total; otherwise, it

plays the highest card that can potentially win the round, if such a card exists. This strategy focuses on securing the highest possible points from each round.

Cooperative Agent

The Cooperative agent maintains a belief of its partner's cards and updates it each time a card is played. If it's the first player of the round it tries to select a suit that maximizes the possibility of earning points by evaluating its partner's cards and choosing a card from the suit that will lead to earning the biggest amount of points. In case it's the second player of the round, it will check if the team will be able to win the round, based on the its cards, the cards that have been played so far and the partner's belief, and play according to this information. In other cases it behaves just like the **MaximizePointsWon** agent.

DeckPredictor Agent

This is the most complex agent, as it uses a probabilistic approach similar to the **Cooperative Agent**'s to keep track of the likelihood of each card being in **every player**'s hand (friend or foe). For every (playable) card in its hand, it simulates every possible round outcome and returns the utility value for that card. This value is calculated by summing the probability of each possible outcome multiplied by the points that would be won/lost in that outcome (adding the points won by the agent's team or subtracting the points won by the opponent team). The agent then plays the card with the highest utility value.

Logically, we can easily foresee an immediate issue, which is that in the first rounds, the agent will immediately play the trump cards, as they are the most likely to win. This is a problem because the agent will not have any trump cards left for the later rounds, where they are more likely to be needed. To mitigate this, we've added a simple heuristic that, for initial rounds, and if the agent is the leader, we will decrease the trump cards' utility value by a considerable amount, so that the agent will avoid playing them.

2.3 System Architecture

The system architecture is divided into four main abstractions:

- **Card** - Represents a card in the game of *Sueca*. Each card is defined by a *suit* - or 'naipe' in portuguese - and a *rank*, and can be compared to other cards (via an *order* value). Each card also has a *value* assigned to it by the game rules.
- **Player** - Represents a generic player in the game of *Sueca*. Each player has a *name*, a *hand* of cards, and a *team* that they belong to. This class also includes basic methods for facilitating the player's actions. From this abstraction we can directly derive the simpler agents, such as the **RandomPlayer**, the **GreedyPlayer**, the **MaximizePointsWonPlayer** and the **MaximizeRoundsWonPlayer**, that will implement the respective strategies.

Furthermore, there is one more abstraction that derives from the *Player* class, which is the *BeliefPlayer*. This class is used

to represent the **CooperativePlayer** and **PredictorPlayer**, which maintains a belief of the other players' cards and updates it each time a card is played. This belief is then used to make more complex and informed decisions.

- **Team** - Represents a team in the game of *Sueca*. Each team has two *players*, can accumulate points won in each round to form a *score* and also maintains information regarding the *initialpoints* its players had at the beginning of the game for further evaluation purposes. This class also includes methods for facilitating the posterior gathering of game related information for evaluation.
- **Game** - Represents a game of *Sueca*. It is responsible for handling game logic, like initializing a deck, registering players, managing turns, determining winners and updating the game state. It features a game-loop, which iterates through rounds and player-specific turns. It has a *trump* card, two specific *teams*, an *order of players* and certain *game information* associated with it. This class also includes methods for facilitating the game's progression such as round winning logic.

In essence, the game proceeds as follows:

- 1 The game initializes by shuffling the deck and dealing cards to each player, choosing the trump card as the last card dealt and determining the player order just as if the players were sat around a table;
- 2 The game proceeds with each player taking turns to play a card from their hand, with the first player being the leader of the round;
- 3 The game determines the winner of the round and updates the game state accordingly;
- 4 It then rotates the player order and repeats the process until all cards have been played;
- 5 Finally, the game calculates the points won by each team and updates the game state accordingly.

2.4 Human Interaction

Interaction with the user is used by an actual human player. The user is able to interact with the game through the command line interface, and can play against the other agents. He is able to play cards from its hand, and the game will proceed as usual. This is useful for testing the agents' strategies and for providing a more interactive experience for the user. At last, it is assumed that the human player will play according to the rules of the game, and will not cheat or break the rules in any way.

This feature is not considered in the comparative analysis, and is only used as more of a curiosity and a way to observe the agents in action in real-time.

This logic was inserted directly in the *Game* class, where the round-loop was modified to check if the current player is a *human* player. If so, the game will wait for the user to input a card, and then proceed as usual. This way, the player can play against the other agents, and the game will proceed as if the human player was just another agent.

3 EMPIRICAL EVALUATION

In order to assess the performance of the multi-agent system at playing the game of *Sueca*, we need to conduct an evaluation process. Through this, we aim to validate the project’s objectives by measuring key metrics related to gameplay quality and agent decision-making.

With this in mind, we will use the following metrics:

- 1 **W/R** - The win-rate ratio of each strategy, calculated as the number of games won by that strategy’s team divided by the total number of games played (remember that the strategies are employed team-wise). The higher the win-rate, the better that strategy is for playing the game;
- 2 **Avg. P/G** - The average number of points accumulated by each strategy’s team per game. This metric will serve to validate the *Win – rate*, in the sense that a strategy that accumulates more points per game should also win more games. Also, even if a strategy doesn’t win a game, it can still be considered semi-successful if it accumulates a high number of points;
- 3 **Points conv.** - The difference between the final points accumulated by each team and the initial points they had at the beginning of the game. For example, if a team starts with a total of 40 points and ends with a total of 90 points, it had a *points conversion* of +50. A strategy that can convert a high percentage of the initial points into final points is considered effective;

Measuring these metrics allows us to provide valuable insights into the capabilities and effectiveness of the multi-agent system in playing *Sueca*, as well as the performance of the individual strategies. The results of the evaluation process will be used to draw further conclusions regarding the system’s performance and the effectiveness of the different strategies employed.

3.1 Research Questions

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

- 1 **How effective are the agents at winning the game?** This question aims to evaluate the ultimate goal of each agent, which is to win the game by accumulating more points than the opponent team;
- 2 **How predominant is each agent’s game?** This question aims to evaluate the performance of each agent in terms of the average number of points accumulated per game. This metric will provide insights into the efficiency of each agent in accumulating points;
- 3 **How well do the agents do, given the points they begin with?** This question aims to evaluate the effectiveness of each agent in converting the initial points they have at the

beginning of the game into final points. This way we can see the agents that can better leverage the initial points they have to win the game.

3.2 Data Collection

In order to gather necessary data for the empirical analysis, a series of simulations were conducted using a simulation framework we developed for the multi-agent system. However, it is difficult to ensure consistency given the stochastic nature of the game of *Sueca* (as well as any game of chance), as well as the stochastic nature of some of the strategies (namely the *RandomAgent*).

For all combinations of agent behaviours, a total of **10000** games were simulated in order to achieve a sufficient level of statistical confidence and accuracy regarding agent behavior.

It is important to note that the simulation framework does not take Human players into account.

3.3 Data Analysis

In this subsection, we analyze the results obtained from running simulations with different agent combinations. To evaluate performance of each player type, we consider several metrics explained above. 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 **from the same team** to different agent types and had them play the game. After each simulation, we recorded the required metrics.

Following are the results of the comparative analysis on the different intelligent agent approaches:

Agent Type	W/R %	Avg. P/G	Points conv.	Draws /10000
Greedy	39.7	54	-5.6	177
MaxPointsWon	27.3	48	-12.2	150
MaxRoundsWon	31.9	49	-11.1	149
Cooperative	28.5	48	-11.7	169
Predictor	26.1	46	-14.1	139
<i>Metric Average</i>	30.7	49	-10.9	157

Table 1: Random Agent related performance

Agent Type	W/R %	Avg. P/G	Points conv.	Draws /10000
Random	58.6	66	+5.6	177
MaxPointsWon	27.2	46	-13.7	145
MaxRoundsWon	28.6	47	-13.1	162
Cooperative	26.7	45	-15.3	142
Predictor	23.2	43	-17.6	123
<i>Metric Average</i>	32.9	49	-10.8	150

Table 2: Greedy Agent related performance

Agent Type	W/R %	Avg. P/G	Points conv.	Draws /10000
Random	68.1	71	+11.1	149
Greedy	71.4	73	+13.1	162
MaxPointsWon	48.2	59	-1.1	201
Cooperative	44.6	58	-1.9	185
Predictor	43.0	57	-2.5	184
<i>Metric Average</i>	55.0	64	+4.1	176

Table 3: MaximizeRoundsWon Agent related performance

Agent Type	W/R %	Avg. P/G	Points conv.	Draws /10000
Random	71.3	72	+12.2	150
Greedy	72.8	74	+13.7	145
MaxRoundsWon	51.8	61	+1.1	201
Cooperative	46.0	59	-1.1	197
Predictor	44.4	58	-1.9	199
<i>Metric Average</i>	57.3	65	+4.8	178

Table 4: MaximizePointsWon Agent related performance

Agent Type	W/R %	Avg. P/G	Points conv.	Draws /10000
Random	71.5	74	+11.7	169
Greedy	73.3	75	+15.3	142
MaxPointsWon	54	61	+1.1	197
MaxRoundsWon	55.4	62	+1.9	185
Predictor	47.3	59	-1.2	182
<i>Metric Average</i>	60.5	66	+5.8	175

Table 5: Cooperative Agent related performance

Agent Type	W/R %	Avg. P/G	Points conv.	Draws /10000
Random	73.9	74	+14.1	139
Greedy	76.8	77	+17.6	123
MaxPointsWon	55.6	62	+1.9	199
MaxRoundsWon	57.0	63	+2.5	184
Cooperative	52.7	61	+1.2	182
<i>Metric Average</i>	63.2	68	+7.4	165

Table 6: DeckPredictor Agent related performance

3.4 Discussion

The empirical evaluation of the multi-agent platform provided valuable insights into its effectiveness and performance. In this subsection, we aim to discuss the findings of our evaluation, in the context of the research questions proposed in section 3.1:

How effective are the agents at winning the game?

As we can see from the average win-rate percentages, the agents that performed the best were the **DeckPredictor** and the **Cooperative** agents, with win-rates of 63.2% and 60.5% respectively, winning the majority of the games they played. These strategies can handle better the information available at each round, resulting in a stronger decision-making process.

The **MaximizePointsWon** and **MaximizeRoundsWon** agents also performed well, with win-rates of 57.3% and 55.0% respectively, nearing the 50% mark. The **Greedy** agent had a win-rate of 32.9%, while the **Random** agent had a win-rate of 30.7%. These last two were the ones that performed the poorest, which was expected, as they are the simplest agents, that play a card regardless the cards that have already been played, resulting in non-optimal choices.

The obtained results highlight the importance of a well-structured decision-making process is key for winning in *Sueca*, as agents that use more information clearly overpower the others.

How predominant is each agent's game?

The agents that accumulated the most points per game were the **DeckPredictor** and the **Cooperative** agents, with averages of 68 and 66 points per game.

The **MaximizeRoundsWon** and **MaximizePointsWon** agents also performed well, with averages of 65 and 64 points per game respectively.

The **Random** and **Greedy** agents had an average of 49 points per game.

This is consistent with the previous results, as the agents that accumulated the most points per game were also the ones that won the most games. Interestingly, the agents with the lowest points per game (*Random* and *Greedy*) lost by more points per game (11 points)

than the best agent (*DeckPredictor*) won per game (8 points). Similarly to the previous question, the **Random** agent accumulated the least amount of points per game, while the **DeckPredictor** agent accumulated the most, making them the least and most predominant agents in the game, respectively.

How well do the agents do, given the points they begin with?

To answer this question, we need to look at the points conversion metric. The agents that converted the most points were the **DeckPredictor** and the **Cooperative** agents, with averages of +7.4 and +5.8 points respectively.

The **MaximizePointsWon** and **MaximizeRoundsWon** agents also performed reasonably well, with averages of +4.8 and +4.1 points each.

The **Greedy** agent, however, had an average of −10.8 points, and the **Random** agent an average of −10.9 points, both losing points.

Again, we piggyback on the previous results, as the agents that converted the most points were also the ones that accumulated the most points per game. The **Random** and **Greedy** agents were the ones that converted the least amount of points, ending up losing rather than winning points, as one would expect from the lowest winning-rate agents. Conversely, the **DeckPredictor** agent converted the most points, making it the most effective agent at converting the initial points into final points.

By analyzing the collected data we can conclude that the most intelligent agents implemented (**DeckPredictor** and **Cooperative**) relay less on the hand they get, as they are able to take more points from the opponent team.

It’s amusing (and logical) to note that the points converted can actually be calculated directly from the average points per game (and vice-versa) by simply subtracting the final *avg P/G* from 60. This is because, as we converge to infinite games, the points a team begins with will eventually converge to 60, i.e., every player should begin with roughly the same number of points as the other players and thus the converted points will match the total *average points per game*.

4 CONCLUSION

In this paper, we’ve presented a multi-agent platform for playing the game of *Sueca* and conducted an empirical evaluation to assess its effectiveness and performance. Our evaluation focused on metrics related solely to agent decision-making. Through our analysis we were able to grasp the capabilities and potential of a multi-agent system for playing card games.

The **DeckPredictor** agent was the most effective at winning the game, accumulating and converting the most points per game - making the *Utility Tree* logic the most effective strategy out of the ones we’ve implemented.

The **Cooperative** agent was the second most effective, making use of a similar *Belief* track of the game state as the previous agent,

but with a simpler decision-making process.

The **MaximizePointsWon** and **MaximizeRoundsWon** agents were the third and forth most effective, respectively, with the first agent being slightly more effective than the latter agent. From this we can conclude that prioritizing points over round wins can be a more effective strategy.

The **Greedy** and the **Random** agent are the less effective, with the lowest win-rate, points per game and points conversion. These agents served as a baseline for comparison with the other agents and only basic logic was implemented in it, hence the poor performance.

In conclusion, the results of the evaluation process provided valuable insights into the performance of the different agent strategies we’ve come up with and managed to corroborate our initial prediction regarding the effectiveness of each strategy.

REFERENCES

- José Almeida, Pedro Moço, and Tiago Teodoro. 2023. BASIS: Bisca Agent Strategy Intelligent System. (2023). <https://github.com/zev4l/basis/blob/main/paper.pdf>
- Filipa Isabel Nogueira Ferreira. 2015. EMYS: a social robot that plays “Sueca”. (2015). <https://fenix.tecnico.ulisboa.pt/cursos/meic-a/dissertacao/1128253548920919>