Study and research work report

AIs for the game of 7 Wonders

Produced by
Loïc Filippi
Alessandro Pepegna
Meryem Boufalah
Anastasiia Kozlova
Chaeyeon Shim

Master Informatique
at Université Côte d'Azur
First semester of the year 2020 - 2021

Abstract

In this report, we present and examine various AIs for the game of "7 Wonders" and discuss their respective performance at winning the game. After a brief description of the game, we will go through two main AIs crafted with different aims, expose the algorithmic component and testing their skill in different settings.

Contents

1 About the game

1.1 Description

7 Wonders is a strategy card game where each player randomly receives a "Wonder board", each depicting one of the Seven Wonders of the Ancient World. The game is played over three ages, and each player is randomly dealt 7 cards at the start of each age. The player selects a card to play and passes the remaining cards to the next player. By means of different mechanics, players earn victory points. The player with the most victory points wins.

1.2 Version

As part of the development project, we have managed to implement the Seven Wonders' first version rules with two exceptions: the resources granted by the Wonder can be sold and purchased between players and the cards offering a choice between two resources don't allow the player to use them to play a card although they can still sell them to other players.

2 Other AIs

7 Wonders is a game played among 3 to 7 players. For this reason, and to allow us to put our two AIs against different opponents, we implemented other strategies:

- First card : chooses the first card in hand and builds it if possible, sells it otherwise.
- Random card : chooses a random card in hand and builds it if possible, sells it otherwise.
- "Dumb" : picks the first card in hand and sells it.
- Wonder : builds a Wonder step if possible, if not, tries to play a card, else discards a random card.

3 Rule-based AI

3.1 Aim

The first AI seeks to achieve a solid level of play while keeping the technique simple.

3.2 Algorithm

We set a priority list for the AI to follow. If a condition is not met, the AI simply goes to the next rule and sees if it can apply it. The order is as follows:

- The civil card with the greatest victory points reward is played
- A military card is played only if the AI is not the only leader in military and the cards allows the AI to become the (or one of the) leading military player(s)
- In the third age, the card with the best immediate victory points reward
- The science card granting the most victory points
- A card providing a single resource that is lacking to the AI

- A card providing 2 or more resource types
- A card the AI can not play in order to build a step of his Wonder
- A random card is played if possible, else a random card is sold

3.3 Performance

This AI was taken from Monte-Carlo Tree Search for the Game of "7 Wonders" by Denis Robilliard, Cyril Fonlupt and Fabien Teytaud. The priority order was rearranged and allowed the AI to build the steps of its Wonder.

At first it was considered to test every possible arrangement for the 7 first rules to find the best one, but after realizing it amounted to $7! = 5040$ different arrangements, another method was selected. We opted for a "trial and error" approach in order to identify which leading rule was inducing the most wins in 10 000 games. We came to the conclusion that playing the card granting the best immediate score was not the best choice in the third age. On the other hand, playing the civil cards and, secondly, military cards, showed high potential.

Eventually, we set for an order which seemed to be the best arrangement possible.

Rule based	Stupid	FirstCard	Random	Wonder	Rule based
89%	0%	11%	-	-	-
98,6%	6%	-	8%	-	-
73%	0%	-	-	27%	-
54,2%	0%	-	-	-	45,8%

Table 1: Each line presents the results of 1000 games between different combinations of AIs, "-" meaning the AI didn't participate

We observe that the rule-based AI shows convincing results against other AIs.

4 Monte-Carlo AI

4.1 Aim

For the second AI, the goal was to surpass the AI previously seen by using a more advanced technique.

4.2 Algorithm

The purpose of the Monte-Carlo algorithm is to choose the promising move given a game state. The general principle of the algorithm is to simulate the continuation of the game N times for each possible combination of cards and actions for the player at the current state of the game. The continuation of the game is done using a base strategy, while the strategies of the other players rest unchanged.

Eventually, we choose the combination of card and action which leads to the best result depending on the evaluation approach. We implemented two different approaches:

- The chosen combination allows the player to win the maximum number of times, games are simulated for all the ages up to the end (later referred as "MC Victory")

- The chosen combination allows to get the maximum score, games are simulated for all the ages up to the end (later referred as "MC Score")

4.3 Performance

Table 2 shows the results for the maximum score approach, while Table 3 shows the results for the most wins approach.

MC Victory	Stupid	FirstCard	Random	Wonder	Rule based	MC Victory	MC Score
88,9%	2,3%	8,8%	-	-	-	-	-
91,8%	5,9%	-	2,3%	-	-	-	-
77,3%	2%	-	-	20,7%	-	-	-
64,6%	0,4%	-	-	-	35%	-	-
50,9%	5,6%	-	-	-	-	43,5%	-
44,8%	1,7%	-	-	-	-	-	53,3%

Table 2: Each line presents the results of 1000 games between different combinations of AIs, "-" meaning the AI didn't participate

MC Score	Stupid	FirstCard	Random	Wonder	Rule based	MC Victory	MC Score
92,1 %	0,1 %	7,8 %	-	-	-	-	-
97,9 %	0,8 %	-	1,3 %	-	-	-	-
78,4 %	0%	-	-	21,6 %	-	-	-
63,5 %	0%	-	-	-	36,5%	-	-
53,3 %	1,7 %	-	-	-	-	44,8 %	-
50,3 %	0,2 %	-	-	-	-	-	49,5%

Table 3: Each line presents the results of 1000 games between different combinations of AIs, "-" meaning the AI didn't participate

Rule based	MC Victory	MC Score
34,7 %	25,2 %	40,1 %

Table 4: Winrates on 1000 games

Stupid	FirstCard	Random	Wonder	Rule based	MC Victory	MC Score
0%	4,3 %	0,1%	10,9%	43,4%	13,2%	28,1%

Table 5: Winrates on 1000 games

Both approaches of the Monte-Carlo AI win the most games in each setting, including when they are facing the rule-based AI. Contrary to what might have been expected, the results demonstrated by the Victory variant are less satisfactory than those demonstrated by the Score variant. The difference between the two approaches could be explained by the fact that de Victory variant is given less data to work with, while the Score variant can be more nuanced thanks to score measurements, unlike the Victory variant who deals in a binary ”winning or losing” fashion.

5 Conclusion

In this project we implemented two AIs that showed relatively good results. Moreover, just as we expected, our Monte-Carlo ambitious AI has proved to be the best in games against all of our AIs, and rule-based AI outperforms all the robots except the Monte-Carlo.

While we are satisfied with the results obtained, we see room for improvement. In particular, we could have opted for a bigger number of games simulated in Monte-Carlo. However, the time of execution for more than 1000 simulations remains too long, so improving our performances and refactoring could potentially help. As for the rule-based AI, we vigorously experimented with the order of preferences, therefore the results we obtained seem to be the best possible.

Furthermore, considering playing against real experienced players, we could rather opt for a combination of all existing strategies, preferably adapt the decisions depending on a current situation, possibly with training or prediction of the other players’ moves.