MASTER INFORMATIQUE

UNIVERSITÉ CÔTE D'AZUR

TER Report

Analyzing the best strategies and developing AIs for the 7 Wonders game

Submitted by:
Zaïd Burkutally
Maya Medjad
Samuele Marino
Vincent Laubry

Master M1 Informatique
at Université Côte d'Azur
16 January 2021

Contents

1 Introduction **1**

2 Rules of the game **1**

3 Tasks done **3**
3.1 Guaranteed AI 3
3.1.1 About the Guaranteed AI 3
3.1.2 Performances of the Guaranteed AI 5
3.2 Ambitious AI 9
3.2.1 About the Ambitious AI 9
3.2.2 Performances of the Ambitious AI 10
3.3 Conclusion 12

List of Figures

1 Flowchart of Easy AI to play a card 2
2 Flowchart of Guaranteed AI to play a card 4
3 Performance of Guaranteed AI compared to the Random AI 5
4 Performance of Guaranteed AI compared to Easy AI 6
5 Average winning difference of AI by the strategy chosen for Age 1. 6
6 Comparing the strategies used by the AI. 8
7 Comparing the strategies used by the AI using RSM for first age. 9
8 Performance of Ambitious AI compared to the Random AI 11
9 Performance of Ambitious AI compared to the Easy AI 11
10 Performance of Ambitious AI compared to a Random AI and an inefficient Guaranteed AI 11
11 Performance of Ambitious AI compared to a Easy AI and a Guaranteed AI . . . 12
12 Performance of Ambitious AI compared to an Easy AI and an inefficient Guaranteed AI 12

1 Introduction

The TER Team Rocket group consists of four students: Zaïd Burkutally, Maya Medjad, Samuele Marino and Vincent Laubry who worked on the Travail étude et recherche (TER), a continued version of the Projet de développement which was to develop the 7 wonders game.

7 wonders is both a board game and a card game which can be played by 3 to 7 players using three decks of cards featuring depictions of ancient civilizations, military conflicts, and commercial activity.

The goal of the TER project is to produce competing AIs, compare the strategies they use to play the game and finally evaluate them in order to obtain the best strategy that wins the most. During the Projet de développement we developed a basic AI (called EasyStrategy) and a random one (called RandomStrategy), we aspired from these AIs to develop newly and more efficient ones and we compared the efficiency of the newly developed AIs to the ones of Projet de développement as reference.

2 Rules of the game

During the Project de développement we had the opportunity to understand how the game works and the different rules that governs the game. However, for its development, we had to omit some rules from the original set of rules of the game. Since the goal of TER is to focus more on the best strategies for the game, we therefore adhered to the same rules for the development of the AIs.

Some of the rules we implemented are:

- Rotation of cards
- Single reward cards
- Multiple reward cards (however rewards can only be chosen once and not every turn)
- Card sacrifice
- Trading a card or a wonder stage
- Obtaining a card by card chaining
- Trading and buying resources from a neighboring player
- Scientific, Civil, Military, Brown and Grey cards
- Three Wonder Board
- Three player game

The rules that we omitted are:

- Gold and purple cards
- Some wonder boards
- Cards giving reward types which have not been implemented

We developed an easy AI (EasyStrategy) which plays a card as follows:

1. Play a free card.
2. If there are no free cards available, play a chain card.
3. If there are no chain cards available, play a wonder card.
4. If there are no wonder cards available, play a card that can be played by trades (Exchange of resources between neighboring players).
5. If none of the above are available, sacrifice a card.

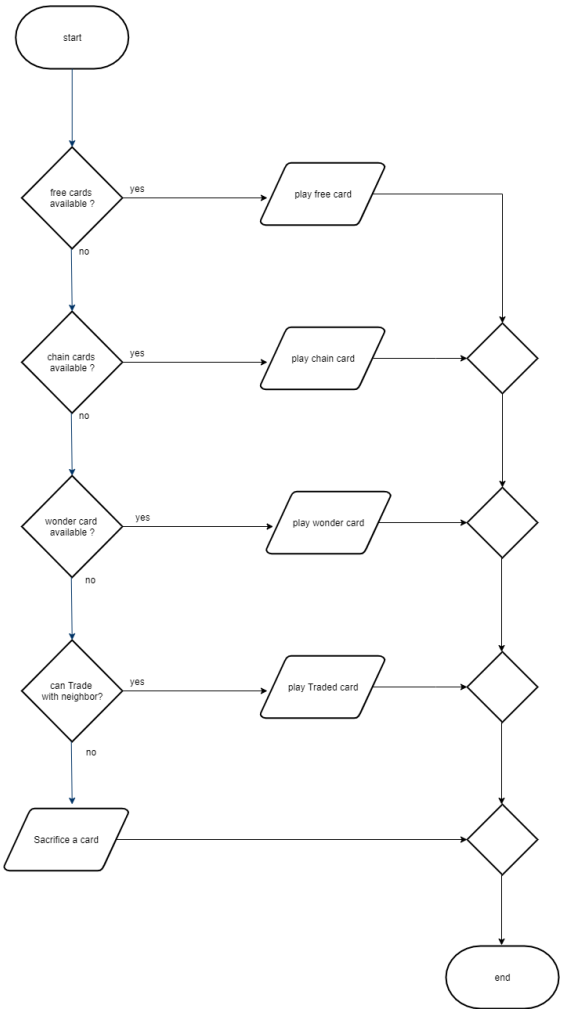

Figure 1: Flowchart of Easy AI to play a card

In order to avoid code duplication; having the same block of code in each strategy class, we made a parent class called PlayStrategy which contained the common code and each strategy inherited from the PlayStrategy class.

3 Tasks done

3.1 Guaranteed AI

3.1.1 About the Guaranteed AI

The guaranteed AI uses strategies by composition (multiple strategies for same or different age), the AI chooses the action to do (Card to play or sacrifice or trade) for each age depending on the strategy that has been chosen and it can also change strategies depending on the context to play more efficiently.

The AI can have different strategies for each age since for each age the objectives to be cleared are different, this is vital to designate a winner at the end. The strategies to be used for each age are stored in a list called StrategyPriority. For example:

1. **Age 1:** The AI will first optimize resources as they are the most important for this age. Once it has reached a certain quota of resources, it will then change its strategy to select other types of cards based on the StrategyPriority list. Similarly, if an AI cannot use the resource strategy, it will adapt to choose the best card to play from the StrategyPriority list fed to him.

2. **Age 2:** For the second age, two types of cards (strategies) will be designated, the AI will then play cards based on the strategies in its list of priorities.

3. **Age 3:** Finally, for the third age, the AI will also choose between two types of cards that are in the list of strategies.

In order to have an optimal AI, the card types to be chosen for age 2 and age 3 have been studied to obtain the best combination that yields the most points to make the AI the winner.

A strategy for blocking a player was meant to be developed but we decided to omit it because the AI would practically never use it. It always had other cards to play and it's better to play a card that can add more resources or victory points to its inventory than to block a player.

We then adapted the EasyStrategy in order to play a card according each strategy we implemented, for instance, if the AI had the ScientificStrategy, it would choose to play as follows:

1. Play a chain card.

2. If no chain cards are available it will then play a card for its wonder.

3. If no wonder cards are available, it will then choose to play a green card.

4. If no green cards are available, it will then choose to play a free card.

5. If no free cards could be played, it will then choose to play a card by using trades.

6. However, if none of the previous choices are possible, it will then sacrifice a card.

We chose this sequence as the set of rules for playing a card as a wonder card or a chain card doesn't happen very often. Therefore, it is better to play a chain card or a wonder card first as we will lose this opportunity in the next turn due to the card rotation. In addition, we can only play the wonder three times and having chain cards is rare and this doesn't stop the AI from playing the strategy it was initially set to.
We also did statistics and found that this was the best combination of playing a card (we found that with this sequence the AI won 3% more on 1000 games).

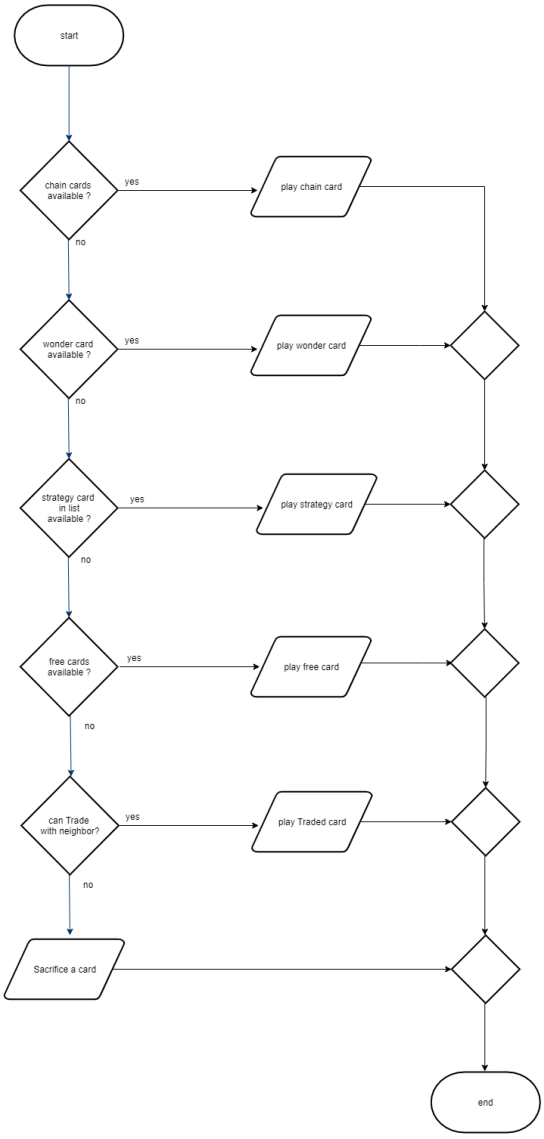

Figure 2: Flowchart of Guaranteed AI to play a card

In the first version of our guaranteed AI, only one strategy was possible for each age and the AI would switch between strategies at the end of each age. This strategy swap was performed in the class Game, however this violated the SOLID and GRASP principles as the strategy change is not the responsibility of the Game class. It also hindered the development of more complex AI.

Our second version consisted of multiple strategies per age, this time the responsibility was also removed from the Game class to a new class called MultipleStrategies. MultipleStrategies is a class that inherits from the same interface (IStrategy) as all the other strategies, however some conditions were added to check when and which strategy the AI needs to switch to.

The function chooseAction of the class MultipleStrategies loops through all the strategies

from the list strategies of the AI, filters them by the current age of the game (for instance, the list of strategies will contain many strategies and each strategies will be assigned an age to be used, therefore, the function chooseAction will filter the list to obtain only strategies to be used for this age).
Once it has all the strategies to be used for the current age, it will get the best card that can be played for each strategy that can be played. However, a check was introduced to verify if the AI reached the maximum state; for instance, if the AI is using the ResourceStrategy, the maximum state is reached when the AI has at least one resource of five types of resources. This means that the AI reached the limit for this strategy and in this case, the card to be played would be added to another list called maxReachedList and the name of the strategy would be added to a list called maxAction.
Moreover, if a strategy returns the action of sacrifice, it would be omitted from the list as the action sacrifice would not be a constructive action for the AI. Otherwise the action would be added to a list called actionLi.
Then all the strategies in the maxReachedList (strategies that the AI has reached maximum state) will then be shifted to the end of the priority list and the next strategy in the list will be used. However, all the cards that can be played for the strategies where the AI has reached maximum state is then appended to the list of action (actionLi) as it is better to play a card even if the AI has reached maximum state than to sacrifice a card.
Finally we add an action of sacrificing a card to the end of the list and the action to be played by the AI is the first one in the actionLi.
Therefore the order of the cards the AI will play is a PLAY action for the first strategy in its list given that it has not reached the maximum state, if it cannot perform the PLAY action, it will play the a card where it has reached the maximum state. Otherwise it will sacrifice a card.

3.1.2 Performances of the Guaranteed AI

We tested the performance of the guaranteed AI against 3 different opponents on 1000 games:

1. **Random AI**: which choses every card to play randomly.
   We observed that against the Guaranteed AI, the Random AI did not win any of the games. This seems logical as the Random AI only executed actions randomly whereas the Guaranteed AI choose the best action to be played.

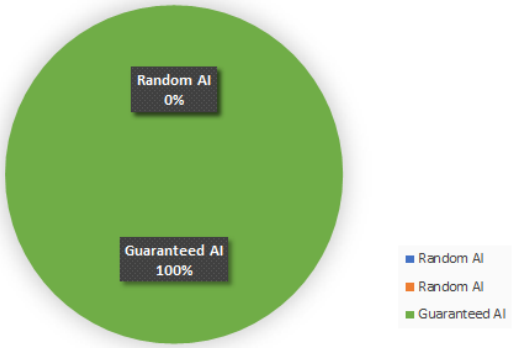

Figure 3: Performance of Guaranteed AI compared to the Random AI

2. **Easy AI**: which uses the strategy described in Rules of the Game section to play a card.

When the Guaranteed AI played against the Easy AI, it won 784 games out of 1000, had 4 draw between the Guaranteed AI and Easy AI. This proves that the Easy AI is better than the Random AI but against the Guaranteed AI it is less effective.

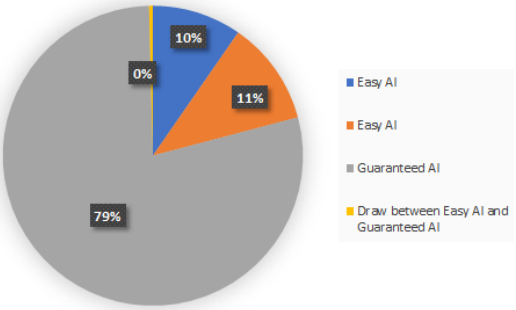

Figure 4: Performance of Guaranteed AI compared to Easy AI

3. **Guaranteed AI** : Finally the Guaranteed AI playing against itself.
   The Guaranteed AI can have $({}^4P_2 \times 2!) = 24$ possibilities of strategies for each age and the total number of possibilities for the three ages is $({}^4P_2 \times 2! \times 3) = 72$ possibilities. We selected the strategies that are expected to fulfill the objectives of each age more rapidly (for example: it is better to start the first age with resources as a priority than having military as a priority).

   After analyzing the statistics we collected, we observed that using the Resource strategy for the first age is usually a good choice as it allows the AI to build as much resources and types of resources as possible to be able to play more cards for the second and third ageand the AI would win on average 446 games.

   We observed that using Scientific strategy and Military strategy for the first age allowed the AI to win around 463 games.

   However, using the Civil strategy decreased the number of winnings to around 168 games.

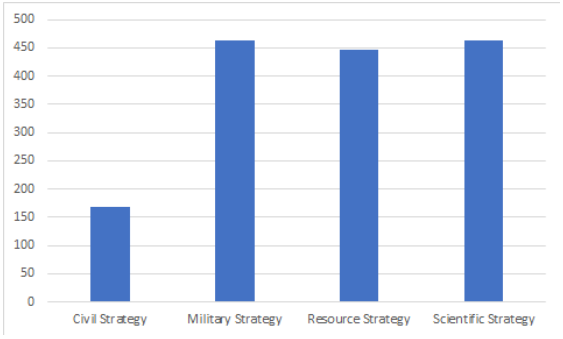

Figure 5: Average winning difference of AI by the strategy chosen for Age 1.

For the second age, two types of cards (strategies) will be designated, the AI will then play cards based on the strategies in its list of priorities.

Finally, for the third age, the AI will also choose between two types of cards that are in the list of strategies.

Before the TER we had a feeling that using the Resource strategy for the first age, Scientific strategy for second age and Scientific strategy for the third age would yield better results in terms of winnings. Indeed, after analyzing the statistics we collected, we observed that using the Resource strategy for the first age is usually a good choice as it allows the AI to build as much resources and types of resources as possible to be able to play more cards for the second and third age.

We defined the following one letter code to represent a list of strategies as a means to decrease repetition of sentences:

(a) **R = Resource strategy, Scientific strategy and Military strategy for the same age.**

(b) **M = Military strategy, Scientific strategy and Civil strategy for the same age.**

(c) **C = Civil strategy, Scientific strategy and Military strategy for the same age.**

(d) **S = Scientific strategy, Civil strategy and Military strategy for the same age.**

(e) **S_Inv = Scientific strategy, Military strategy and Civil strategy for the same age.**

Therefore for example, writing R1 represents having Resource strategy, Scientific strategy and Military strategy for the first age.

In order to find the optimal strategy, we studied each strategy that won the most for each age, for example we tested all the strategies for the first age, second and third age. The result we obtained is S1, M2 and R3 for third age.
However since all three ages are considered at the end of the third age and not individually, we observed that having the Resource strategy at either the first or second age increased the number of winning by about 0.05%.

Therefore we tested again different combinations of strategies to find the best and we found four strategies :

(a) S1, S2 and M3 for the first, second and third age respectively wins around 49.8% of the 1000 games (the AI wins 498 games).

(b) By having M1, R2 and S3 for each age respectively wins around 54.5% of the 1000 games (the AI wins 545 games).

(c) The AI which has S1, R2 and M3 for each age respectively wins around 49.8% of the 1000 games (the AI wins 498 games) which ties with the first option above.

(d) Having R1, M2 and M3 for each age respectively wins around 51.1% of the 1000 games (the AI wins 511 games).

We then tested the two best list of strategies above (part b and part d) with other strategies and we concluded that the best strategy was R1, M2 and M3 as it won against the strategies in part (b) and other list of strategies. Below are presented our findings :

(a) R1, M2 and M3 won 6.9% more against M1, R2 and S3.

(b) R1, M2 and M3 won 15.6% more against R1, S2 and S_Inv3.

We can observe that having R1 and M1 is always a good strategy to use which makes sense as we first collect as much resources as we need, then with Military strategy we can have the advantage and finally with Scientific strategy we collect scientific points.
This means that when using the Military Strategy, the AI will first try to have at least two more military points than its opponents. Upon reaching the maximum military points, it will switch to scientific, diversifying the points efficiently. Moreover, if an opponent would play military and be better than the AI, it would switch back and try to surpass him again.

Comparing S1, we found that there is no limit for playing scientific cards, so it will always play those cards, therefore stopping the AI to play military cards when it need them. For example, the AI may have to switch to military at a turn when there is no military card, it will be less likely for the AI to have points for both strategies. This is why RSM, MSC, MSC is better than RSM, SCM, SMC.
However, if the AI try to use Civil at the second position of the priority list (RCM, MCS, MCS), it will no longer have the same advantage. This is because civil cards do not make the AI win as much points as the scientific cards. The below figure represents the data to support this claim.
Note:
-Player 1 uses RCM, MCS and MCS
-Player 2 uses M1, R2 and S3
-Player 3 uses R1, S2 and S_Inv3

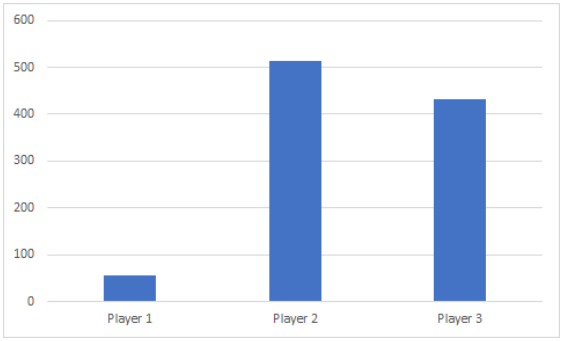

Figure 6: Comparing the strategies used by the AI.

We found that the AI lose almost every time, the only way to overcome this is to use RSM for the first age and this does not allow the AI to be better, the AI therefore loses the advantage of the Military and Scientific combo as shown below:
Note:
-Player 1 uses RSM, MCS and MCS
-Player 2 uses M1, R2 and S3
-Player 3 uses R1, S2 and S_Inv3

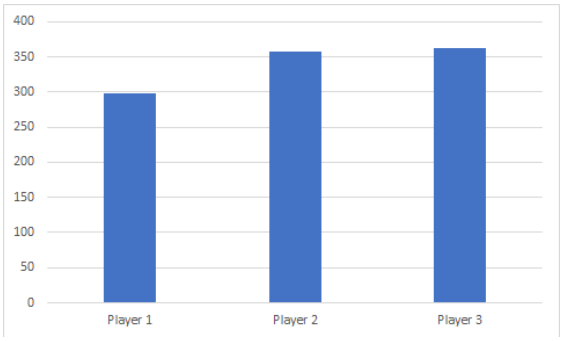

Figure 7: Comparing the strategies used by the AI using RSM for first age.

We observed that the Civil strategy is not a good strategy to use as it has very few rules. During the first and second age, blue cards are not rewarded with high number of points. Therefore if the AI has Civil as priority, instead of building Military or Scientific it will have less victory points.
For example, a way to improve this would be to use the Civil strategy only if the civil card in hand gives five points or more, and if not, it would automatically switch to other strategies.

Now we studied how the results of R1, S2, S3inv versus R1, M2, M3 could be improved. We found out that by adding M3, it allowed the AI to have equal number of winnings. The use of M3 allows the AI to use the Military and Scientific combo. If the AI was behind for Military points during the second age, it will catch up in the third age and also because it used S2 during the second age, it already has a lot of Scientific points.

We observed that using R1, M2, M3 makes the AI more or less win the same number of games as R1, S2inv, M3; 352 as compared to 355 respectively.

As a result of all our analysis of the statistics, we observed that an IA can have many possible strategies and most strategies can be used for the first age and still win most of the time. The only strategy that is not a very good choice is the Civil strategy but the strategy could be improved as defined in the above paragraph. We found out that gathering resource and then focusing on scientific cards is always a good choice.

We deducted that the Military combo (MSC) is a good choice for the second and third age from the results we obtained. This combo can also be used differently with SMC during the second age and MSC during the third age and the results are more or less the same. Those are therefore the best strategies for the guaranteed AI we developed.

We observed that our AI has its strengths and weaknesses, the priority list permitted us to do efficient strategies and modulate what we wanted the AI to do, but there were far too many combinations to check everything. Nonetheless, AI only play in a number of ways and with the priority list, it could adapt its approach to play against different combinations of the Guaranteed AIs, and still score decently.

3.2 Ambitious AI

3.2.1 About the Ambitious AI

The ambitious AI has been derived from the guaranteed AI, however this time the list of strategies is no longer limited to two card types. The AI is free to play any card type but depending on

the card chosen, the AI the priority list will be updated accordingly.
For example, if the AI chose two green cards, then green cards will become one of its priorities (the Scientific Strategy).

We planned to improve the decision-making process as efficiently as possible using the Monte Carlo algorithm. We started by having another class Game that did not distribute cards to the players so that we could set the cards as this would help us launch a game with predefined settings. However, once we had the new Game class, we noticed that we needed to refactor most of our code, and we did not have enough time and resources to do so.

Our final version of the AI can redefine its priority list every three turns, upon launching, it uses the Easy strategy for the first three turns then, it uses the classes Multiples Strategies and Advanced Multiple Strategy to update the priority list and play a card accordingly.

When a game is launched, for each card the player must choose, it will either call Multiple Strategies if it's a guaranteed AI or Advanced Multiple Strategy if it's an ambitious AI.
For the ambitious AI it will then check if the turn is correct for it to reevaluate its priority list, if not, it will be redirected to Multiple Strategies and the AI will play as a guaranteed AI.
However, if it is the turn to reevaluate the priority list, it will start the reevaluation process as follows:

1. For each card type played, count the number of cards of each type of card played and store them in a list.

2. Sort the list in descending order of number of cards played with respect to type of card to obtain the most played card type.

3. Assign the most played card type as the first priority in the priority list for all ages.

Upon verifying statistics of the games, we observed that the Resource strategy must only be used as a priority during the first age otherwise, for the remaining ages, we will have too many resource cards and it would always be the first priority.
It should also be said that the AI starts the game with Easy strategy, as without a starting strategy it will not be able to play and have cards to be able to re-evaluate the priority list.

The Ambitious AI is unfortunately not optimized enough to be able to make the best decision as it lacks direction in the early game. It cannot always play the type of cards it has already played. This is a good starting point that should have permitted us to implement more things to make this AI much more independent and efficient from a predefined priority list. It was supposed to be improved by the Monte Carlo algorithm, otherwise it cannot be better than the Guaranteed AI.

3.2.2 Performances of the Ambitious AI

We first tested the Ambitious AI against the Random AI over 1000 games and we observed that the first Ambitious AI wins 464 games, the second Ambitious AI wins 507 games, there are 29 games where both Ambitious AIs are tied and the Random AI does not win any game. The results are shown in the figure below.

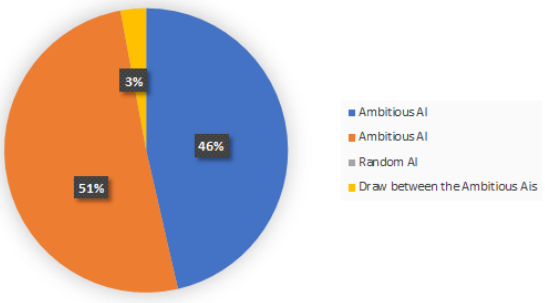

Figure 8: Performance of Ambitious AI compared to the Random AI

We tested the Ambitious AI against the Easy AI and we observed that the Ambitious AI wins more games compared to the Easy AI. We launched the 1000 games with two Ambitious AIs and an Easy AI and found that the Ambitious AI wins 329 and 395 games whereas the Easy AI wins 276 times. The figure 9 below illustrate the results of the 1000 games.

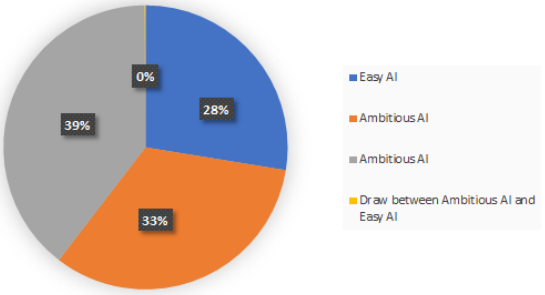

Figure 9: Performance of Ambitious AI compared to the Easy AI

We also ran simulations on an inefficient Guaranteed AI (the Guaranteed AI uses Scientific strategy and Civil Strategy for all three ages), a Random AI and an Ambitious AI. The Guaranteed AI wins 521 games, the Ambitious AI wins 479 games and the Random AI does not win any game, there is only one draw between the Guaranteed AI and the Ambitious AI. The below figure 10 illustrates the results we obtained.

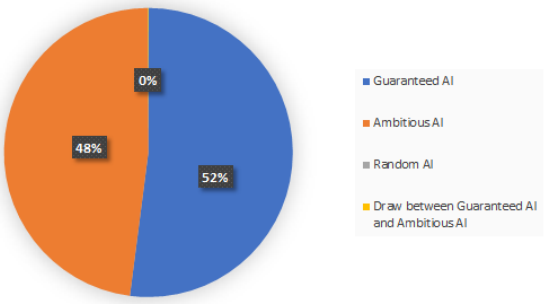

Figure 10: Performance of Ambitious AI compared to a Random AI and an inefficient Guaranteed AI

We then tested the Ambitious AI against the Easy and Guaranteed AI (having Resources strategy and Scientific strategy for the first age, Scientific strategy and Civil strategy for both the second and third age) and the results are presented graphically in the below figure below. The Guaranteed AI wins 717 games whereas the Ambitious AI wins 113 games and the Easy AI wins 97 games. We also observed that there have been 10 draws between the Ambitious AI and the Guaranteed AI. This proves that the Ambitious AI is better than the Easy AI but the Guaranteed AI wins much more than the Ambitious AI.

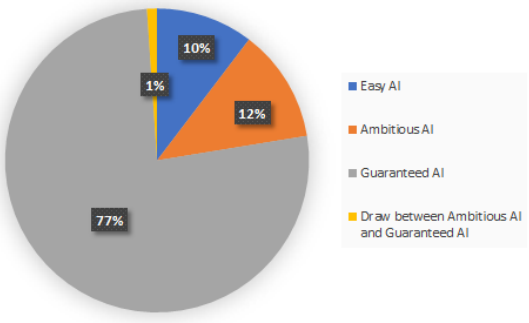

Figure 11: Performance of Ambitious AI compared to a Easy AI and a Guaranteed AI

We also tested the Ambitious AI against the Easy AI and a less efficient Guaranteed AI (the Guaranteed AI uses Scientific strategy and Civil Strategy for all three ages) and we found that the Ambitious AI wins almost the same number of times as the Guaranteed AI. The Easy AI wins 241 games, the Ambitious AI wins 358 games, the Guaranteed AI wins 401 games and there are 8 games that are tied between the Ambitious AI and the Guaranteed AI. The results are illustrated in the figure 12 below.

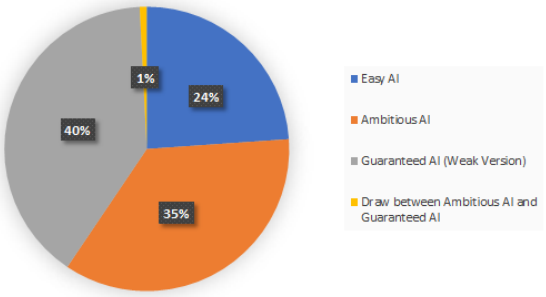

Figure 12: Performance of Ambitious AI compared to an Easy AI and an inefficient Guaranteed AI

We observe that the Ambitious AI is not more efficient than the Guaranteed AI, but it is better the Easy AI as it is not limited to play the first playable card and the Ambitious AI is also derived from the Guaranteed AI, as a result, the rules implemented for each strategy is therefore inherited.

3.3 Conclusion

As a result of our work, we concluded that the Guaranteed AI was about as efficient as we expected by analyzing the statistics obtained for each of the strategies we developed. We could not really compare it to online platform such as Board Game Arena as there is no general statistics, only forum post about players giving advice on the game. Some approaches were found to be better for the AI as it cannot really adapt to what happens throughout the game. It would need a large number of rules to make decisions like a human player as a human player adapts to the game; for instance, when an opponent plays too much scientific cards,it will diversify its game to win.

Bearing in mind the idea of adapting, we had the idea of our Ambitious AI, we wanted an AI that is able to adapt better to changes during the game. However we could not make it more efficient as it was supposed to be improved via a Monte Carlo algorithm, however even though it was not fully optimized, it was better than the basic AI (Easy Strategy). After analyzing the performances and statistics of the Guaranteed AI, we had several improvements that could be made to the Ambitious AI and this would have been more easier to do with the Monte Carlo algorithm. The Guaranteed AI fulfilled all the tasks we planned, but there is still room for improvement.

As future work for this project, it would be to implement the Monte Carlo algorithm to optimize the decision making process of cards to be played and optimizing the strategies that have already been developed. This would greatly improve the score of the Ambitious AI as compared to the Guaranteed AI.