

# Implementation of Min Max Algorithm as Intelligent Agent on Card Battle Game

*Silvester Dian Handy Permana*

*Informatics Study Program, Faculty of Creative Industry and Telematics,  
Universitas Trilogi  
handy@trilogi.ac.id*

## **Abstract**

*Information technology brings transformation from the physical world into the digital world. This transformation developed in various fields, especially games. In the past, games that are involving physical objects such as chess, cards, dominoes, and mahjong are popular for the publics. Card battle game is a game that pits strength between 2 cards. The game must have 2 players who will compete. However, if a player wants to practice before the match or wants to play alone, he needs Non Player Character (NPC). The NPC will be the opponent in card battle games. In order for NPCs to be able to fight players, a special algorithm is needed to make the NPCs compete with players. The algorithm that can be implemented into the NPC is the Min Max Algorithm. This algorithm is a responsive algorithm which can count every step of the player. The results of this study are expected to provide suitable opponents for players who want to practice or compete in Card Battle Game on their own.*

**Keywords:** *We would like to encourage you to list your keywords in this section*

## **1. Introduction**

Digital transformation is currently reaching the world of games. The previous version of chess, dominoes, and mahjong need the physical form of the game before players can play it. In contrast to the current conditions where there are already many digital games that are representations of those physical games used to be played. Card battle game is also undergoing a transformation into the digital world. The transformation of the game into the digital world began when computers that were once owned by the military and government, nowadays belong to many individuals. Operating system developers such as Microsoft have begun to embed digital games in each operating system installer [1].

Card battle game is a power comparison game between 2 cards that must have a strategy to win. This game is certainly done between 2 people in searching for the most powerful deck / card arrangement. In the transformation of the card battle game, developers can make games offline by taking turns for players using only one computer or online with their respective devices to compete in card battle. However, if the game has only one person and he needs to compete with someone, then the game developer have to make Non Player Character (NPC) as an opponent to compete with the player. In order to compete with players, NPCs need special algorithms to make the best strategy in defeating players. The algorithm that can be used to make the Intelligent Agent in NPC is the Min Max algorithm [2].

The Min Max algorithm is an algorithm that is able to compute cards on the opponent side and look for the best card to fight them. This algorithm is the perfection of zero-sum game theory by utilizing the backtracking algorithm. This algorithm uses decision trees and also Breadth First Search or Depth First Search [3]. If needed, this algorithm can even use a delimiting function. The application of this algorithm in NPC can turn the NPC into Intelligent Agent so that it can be an opponent who competes for players. With this research, players are expected to be able to hone their card battle skills with NPCs on computers [4].

## 2. Rudimentary

The research conducted by [5] discusses "Game of Tic-Tac-Toe: Simulation Using Min-Max Algorithm". In this study, the Min-Max algorithm is designed to simulate the game Tic Tac Toe. The Min-max algorithm is used to predict steps and win matches or at least describe the steps in the case of tic-tac-toe games. This algorithm is implemented using C programming languages. This research uses the concept of graph theory or combinational game theory in its implementation. This algorithm only computes one step further using the min-max algorithm. In an ideal scenario, of course a player must calculate all possibilities to ensure victory from the Tic-tac-toe game. Research on "Using the Conspiracy Numbers for Improving Move Selection in Minimax Game Tree Search" was conducted by [6]. In a game which involve 2 people in a match, the method of Conspiracy Number Search (CNS) is used to find the possibility of winning the match. This CNS method is less successful because it uses various opportunities that arise during the course of a match. This research conducted by (Vu, et. Al), combines CNS with the MinMax framework. This is used to provide better advice on finding the possibility of winning in a match. The disadvantage of this research is the development of expensive MinMax algorithms for CNS games. The MinMax algorithm is not as fast as using the alphabet in the calculation process or conspiracy number estimation.

The research conducted by [7] titled "Is Minimax Really An Optimal Strategy In Games?" gives the impression that the MinMax algorithm can be used to optimize strategy in games. the presence of this algorithm carries the assumption that opponents are built to have rational and optimal actions. In strategy games, the Minimax algorithm can be a challenging algorithm during a match. The research entitled "Alpha-Beta Pruning in Mini-Max Algorithm - An Optimized Approach for a Connect-4 Game" was conducted by [8] This study compares the usual Minmax algorithm with the Minmax algorithm which is optimized with alpha beta pruning. From the results of these comparisons the Minmax algorithm combined with alpha beta pruning provides the best step in solving a case. This study uses Alpha-Beta Pruning algorithm in the Mini-Max algorithm to optimize the movement of Artificial Intelligence (AI) in online games. This algorithm provides fast computing time and has fewer iterations than Minimax in computing each step. The research conducted by [9] with the title "Research on Different Heuristics for Minimax Insight from Connect-4 Game Algorithm" studied the application of the Minmax algorithm as the opponent to compete in the Connect-4 game. This research translates the game heuristically and looks for strategies against players. This research carried out 3 functional experiments, the best searches, and several features provided in the Connect-4 game. This study found that the more in search, heuristics have a better function. When a number of features are given in a game, heuristic functions become more optimal. Heuristics can increase the depth of search which is relatively weaker and can defeat opponents.

## 3. Research Methodology

This study has several research steps illustrated in Figure 1.



**Figure 1. Research Steps**

The steps in this study are as shown below:

a. Problem Analysis

Card Battle game needs to have a goal that is to provide a challenge as the competitor in Card Battle game. The goal is to provide precise and quick counters to opponent movement.

b. Game and NPC Design

In the design, a certain programming languages that are compiled with an IDE are needed. This game has an arena to compete, various type of cards, and calculation of Health Point (HP). This game is designed for two sides, namely the player side which is the person who plays the game and the NPC side or Non Player Character as the opponent.

c. Implementation of MinMax Algorithm

This step gives the NPC its way of thinking. NPCs will be given a MinMax algorithm to fight players by computing each cards on the player side. The implementation of this algorithm calculates the strength of its opponent cards on the arena and looks for the optimal value of the NPC cards in hand against the opponent cards.

d. Validation

This step is to test the implemented MinMax algorithm in the game by using several game scenarios. The actions carried out by the NPC will be recorded through the debugger from the IDE in use.

#### 4. Result and Discussion

The MiniMax algorithm has a simple calculation but is able to provide optimal steps against enemies. The first action taken by the NPC if it gets its first turn in the study is determined randomly. The next NPC action after the opponent's turn is done will be determined based on the strength of the card summoned by the enemy. The NPC will summon a card that can fight the opponent's card. Of course, this algorithm give resistances.

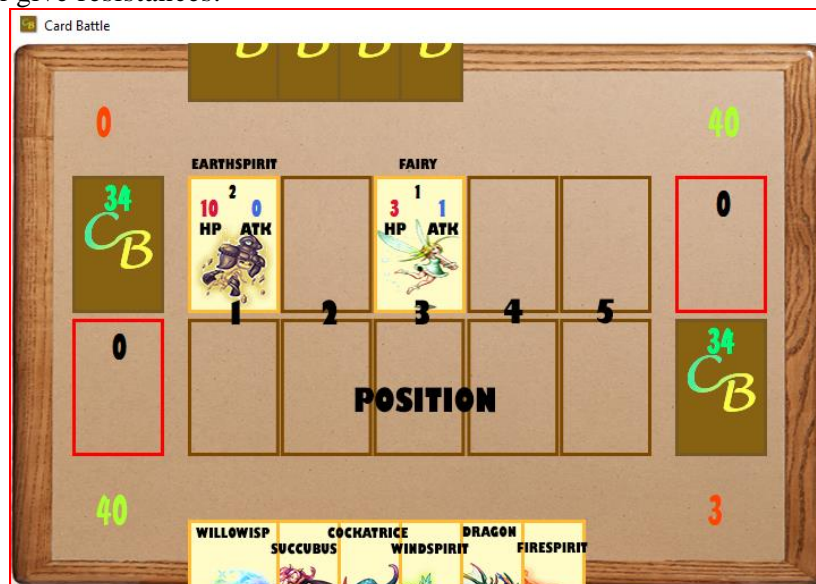


Figure 2. Preview of Card Battle using the MiniMax Algorithm

The case in Figure 2 will be used to explain how the MiniMax algorithm works. In Figure 2 the MiniMax NPC is the card holder on the bottom screen and the opponent of the MiniMax NPC is the player on the top screen. MiniMax NPC has 6

cards in hand and 3 SP points. The strength of the cards in the hand of the MiniMax NPC from left to right are as follows:

1. *Willowisp* (HP 5, ATK 0, Cost 1)
2. *Succubus* (HP 5, ATK 5, Cost 3)
3. *Cockatrice* (HP 8, ATK 3, Cost 3)
4. *Windspirit* (HP 6, ATK 2, Cost 2)
5. *Dragon* (HP 19, ATK 7, Cost 7)
6. *Firespirit* (HP 4, ATK 3, Cost 2)

Based on the conditions above, MiniMax NPC will find out in advance the card that can be issued. The card needs to have less Cost than SP (Summon Point) that is owned by MiniMax NPC. Cards with the name Dragon cannot be issued because they have a cost of more than 3. After the card selection is complete, the NPC will see information from the cards owned by the opponent. Based on this information, MiniMax NPC will issue a card that can maximize the damage given to the opponent and can minimize the damage given to the NPC in the next turn. Table 1 below describes the calculation based on the above conditions:

**Table 1. Computations of MiniMax Algorithm scenario**

No	Area 1	Area 2	Area 3	Area 4	Area 5	Damage	HP Left
1	Willowisp	-	Windspirit	-	-	33.3%	91.65%
2	Windspirit	-	Willowisp	-	-	10%	90%
3	Willowisp	-	Firespirit	-	-	50%	100%
4	Firespirit	-	Willowisp	-	-	15%	90%
5	Succubus	-	-	-	-	25%	50%
6	-	-	Succubus	-	-	50%	100%
7	Cockatrice	-	-	-	-	15%	50%
8	-	-	Cockatrice	-	-	50%	100%
9	Willowisp	-	-	-	-	0%	50%
10	-	-	Willowisp	-	-	0%	40%
11	Firespirit	-	-	-	-	15%	50%
12	-	-	Firespirit	-	-	50%	100%
13	Windspirit	-	-	-	-	10%	50%
14	-	-	Windspirit	-	-	33.3%	87.5%

Damage is the percentage of attacks given by MiniMax NPCs to their opponents. Damage can be calculated by dividing the number of ATK (Attack) of each MiniMax NPC card by the number of HP (Health Point) that is owned by each opponent's card, then divided by the number of cards contained in the opponent's arena. The results of the calculation are multiplied by 100% in order to make it as a percentage. The percentage damage formula is as follows:

$$Damage = \frac{\sum_{i=0}^n pATK_i / eHP_i}{n} 100\% \quad (1)$$

Explanations:

- eHP is the HP of the i-card owned by the player
- pATK is the ATK of the i-card owned by the NPC
- n is the total of cards owned by player on the arena

HP Left is the estimation of the percentage of HP left owned by a card from the MiniMax NPC side when the player attacks the MiniMax NPC with a card that is still in the opponent side of the arena. Remaining HP can be calculated by dividing the number of ATK from each player's card by the number of HPs that are owned by each MiniMax NPC card, then divided by the number of cards in the MiniMax NPC arena. The results of the calculation are made percent by multiplying by 100%. The HP Left percentage formula is as follows:

$$HP\ Left = \frac{\sum_{i=0}^n eATK_i / pHP_i}{n} 100\% \quad (2)$$

Explanations:

- eATK is the ATK of the i-card owned by the player
- pHP is the HP of the i-card owned by the NPC
- n is the total of cards owned by NPC on the arena

The numbers in Table 1 represent the options available at the turn of MiniMax NPC. The MiniMax NPC will choose the option with the highest Remaining Damage and HP value. In this case the MiniMax NPC will choose option numbers 3, 6, 8, and 12 because it has the same value level of 50% damage and 100% HP remaining. After that, MiniMax NPC will randomly select one option from the four available options.

## 5. Conclusion

Based on the results of the decision-making carried out by the MiniMax Algorithm above, found the following conclusions:

- a. MiniMax NPC can summon cards to fight cards on the opponent's arena. This causes the MiniMax NPC to always summon a card on the arena opposed with the opponent card. MiniMax NPC cannot issue cards in the arena where there is no opponent's card to attack LP directly from the opponent.
- b. MiniMax NPC takes action based on the highest percentage of Damage and HP Remaining.

## References

- [1] A. Marchand and T. Hennig-Thurau, "Value creation in the video game industry: Industry economics, consumer benefits, and research opportunities," *J. Interact. Mark.*, vol. 27, no. 3, pp. 141–157, 2013.
- [2] K. Choro, "Design and Development of 'Battle Drone' Computer-Based Trading Card Game (CTCG)," *Springer Int. Publ. Switz. 2017*, vol. 1, pp. 574–585, 2019.
- [3] S. D. Handy Permana, K. B. Yogha Bintoro, B. Arifitama, and A. Syahputra, "Comparative Analysis of Pathfinding Algorithms A \*, Dijkstra, and BFS on Maze Runner Game," *IJISTECH (International J. Inf. Syst. Technol.)*, vol. 1, no. 2, p. 1, 2018.
- [4] W. M. Koolen, P. L. Bartlett, A. Malek, and Y. Abbasi-Yadkori, "Minimax Time Series Prediction," *Adv. Neural Inf. Process. Syst. 28 (NIPS 2015)*, vol. 1, no. 1, pp. 1–9, 2015.
- [5] R. Garg and D. P. Nayak, "Game of Tic-Tac-Toe: Simulation Using Min-Max Algorithm," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 5, pp. 1074–1077, 2017.
- [6] Q. Vu, T. Ishitobi, J.-C. Terrillon, and H. Iida, "Using Conspiracy Numbers for Improving Move Selection in Minimax Game-Tree Search," *Proc. of the 8th Int. Conf. Agents Artif. Intell.*, vol. 2, no. Icaart, pp. 400–406, 2016.
- [7] K. Kosciuk, "Is Minimax Really an Optimal Strategy in Games?," *Zesz. Nauk. Politech. Białostockiej. Inform.*, vol. 6, no. 1, pp. 63–75, 2010.
- [8] R. Nasa, R. Didwania, S. Maji, and V. Kumar, "Alpha-Beta Pruning in Mini-Max Algorithm –An Optimized Approach for a Connect-4 Game," *Int. Res. J. Eng. Technol.*, pp. 1637–1641, 2018.
- [9] X. Kang, Y. Wang, and Y. Hu, "Research on Different Heuristics for Minimax Algorithm Insight from Connect-4 Game," *J. Intell. Learn. Syst. Appl.*, vol. 11, no. 02, pp. 15–31, 2019.

## Authors



**Silvester Dian Handy Permana** was born in Yogyakarta, Indonesia on November 26, 1990. He graduated from Universitas Atma Jaya Yogyakarta - Indonesia, with a degree in Informatics Engineering and later received a Master's degree in Information Technology from Universitas Indonesia - Indonesia.

He currently works for Universitas Trilogi as Informatics Lecturer, and have experience in lecturing for more than 4 years. His research interests include Game Application, Game Algorithm and Complexity, and Multimedia.