

DEVELOPING GAME OF TIC-TAC-TOE USING ARTIFICIAL INTELLIGENT HEURISTIC APPROACH (MIN-MAX ALGORITHM)

Assoc Prof Dr Azlan Mohd Zain

School of Computing,
University of Technology,
Johor Bahru, Malaysia
azlanmz@utm.my

Chai Cheah Wen

School of Computing,
University of Technology,
Johor Bahru, Malaysia
cheahwen1997@live.utm.my

Goh Chiang Cheng

School of Computing,
University of Technology,
Johor Bahru, Malaysia
chiangcheng@graduate.utm.my

Lim Bao Jing

School of Computing,
University of Technology,
Johor Bahru, Malaysia
baojing@live.utm.my

Low Chia Jing

School of Computing,
University of Technology,
Johor Bahru, Malaysia
chia.jing.low@graduate.utm.my

Tan See Jou

School of Computing,
University of Technology,
Johor Bahru, Malaysia
jou1997@graduate.utm.my

Abstract: Tic-Tac-Toe is a famous game around the world that is suitable for different age groups. It's a simple and fun two-player game. Before starting the game, a 3x3 grid must be formed by using two vertical and two horizontal lines. The players can fill the nine spaces with any symbol they like but the symbol must be different for both players. However, the commonly used symbols are crosses ('X') and noughts ('O'). The winning condition for this game is placing three similar marks in horizontal, vertical or diagonal rows. The game will be over when it is a draw or either one of players successfully placing three similar marks in any horizontal, vertical or diagonal rows.

In this paper, we will talk about the development process of this game. In the development of this game, we created one artificial intelligent as the player's opponent for this game. We want to prove that by using an artificial intelligent heuristic approach which is the min-max algorithm and also the alpha-beta pruning algorithm, artificial intelligence can perform the best move in every different situation.

Keywords: Artificial intelligence, Artificial intelligence heuristic approach, Min-max algorithm

1.Introduction

The tic-tac-toe game shown in Figure 1 is a well-known simple which is played across all age groups. A 3x3 grid must be formed by using two vertical and two horizontal lines before the game starts. The players can fill the nine spaces with any 2 different symbols to differentiate both teams and the commonly used symbols are crosses ('X') and noughts ('O'). The winning condition for this game is placing three similar marks in horizontal, vertical or diagonal rows.[10]

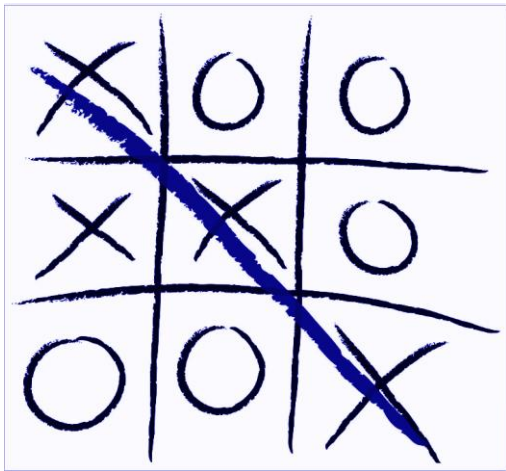


Figure 1: Tic-Tac-Toe game

As we know, there is a lot of software and tools which can be used to develop a game. In our case, we choose Visual Studio 2019 as our tool to develop this tic-tac-toe game. The reason we chose this tool to develop this game because most of us are familiar with it and this tool is easy to use. Although the game developed by

using this tool doesn't come with a fancy user interface compared to other game development software such as unity, we tried to improve the game user interface by using the graphics library which provided for this tool.

To prove that the artificial intelligent heuristic approach (min-max algorithm and alpha-beta pruning algorithm) applied in artificial intelligence can assist it in performing the best move in every different situation, we tried to find some players to compete with it. The artificial intelligence will become the opponent for players. When there are no players who can win against artificial intelligence, then it is proved that the artificial intelligent heuristic approach (min-max algorithm and alpha-beta pruning algorithm) applied can really assist it in performing the best move in every different situation.

The rest of the paper is structured as follows: Section 2 introduces the related works. The methodology is presented in section 3. The development details are discussed in section 4. Then, the testing details are in section 5. Finally, the conclusion in the next section followed by the acknowledgment and the references are appended at the end.

2. Related works

In this section, we will introduce the related works before developing the tic tac toe game with artificial intelligence heuristics. We firstly study about the concept of Artificial Intelligence, then we also do research about heuristics in games and determine how we implement the concept in our game development.

Artificial Intelligence

Artificial Intelligence (AI) is the simulation of human intelligence processes by machines, especially computer systems. Artificial Intelligence exists when a machine can have human-based skills such as learning, reasoning, and solving problems [1].

AI can be categorized as either weak AI, general AI or strong AI. Weak AI is the most common and currently available AI. It is designed and trained for a specific task. Virtual personal assistants, such as Apple Siri is a form of weak AI. General AI is a type of intelligence that could perform any intellectual task with efficiency like a human. However, there is no such system exists as they are still under research. For strong AI, it is a machine that could surpass human intelligence and can perform any task better than humans. However, strong AI is still a hypothetical concept of Artificial Intelligence.

Nowadays, there is a lot of application of AI in different fields. In astronomy, AI can be very useful in solving complex universe problems. In healthcare, by applying AI a better and faster diagnosis than humans can be done. AI can also be used for gaming purposes. AI machines can be used to play strategy games like chess, where the machine can perform the best move in different situations. In our case, AI is used for gaming purposes too which the AI will be the opponent for players in this tic-tac-toe game.

Heuristics in games

Heuristic search is a technique to solve a problem faster than classic methods or to find an approximate solution when classic methods cannot. This is achieved by trading optimality, completeness, accuracy, or precision for speed. The heuristic can evaluate the available information and makes a decision on which branch to follow.

The heuristic is important because it is able to produce a solution that is good enough for the problem. Most problems are exponential. It reduces the number of alternatives from an exponential number to a polynomial number [2]. Nowadays, many effective heuristics methods or tools have been successfully applied in the industry because it does not require a lot of time, money and infrastructure.

The heuristic can be used in game development as they can help to inspire a creative player experience. In games, users want to have fun and enjoy playing the games. Solving problems or learning new things is part of the experience. Game players must use a heuristic to guide play along a path to a winning state. The minimax procedure in heuristics can predict the opponent's behavior and calculate the winning possibility. Thus, the players need to calculate all the possibilities to ensure success and at the same time, it increases the challenging level of the game, making players more attracted by this game.

3. Methodology

In this section, we will introduce our methodology on determining what kind of algorithm being used and the flow of game playing in the tic tac toe game. We firstly determine the rule-based strategy of the Tic-Tac-Toe game to make clear about the winning condition, study on Minimax Search Algorithm and Alpha-Beta Pruning Algorithm to determine how they implement in our game, and we also show the game flow chart and explain the flow of the game playing.

Rule-based Strategy

In the Tic-Tac-Toe game, the players can fill the nine spaces with any 2 different symbols to differentiate both

teams and the commonly used symbols are crosses ('X') and noughts ('O').

Here are the rules, in the order of importance [3], are:

Tic-Tac-Toe Rules

Rules	Description
Rule 1	If the player has a winning move, take it.
Rule 2	If the opponent has a winning move, block it.
Rule 3	If the player can create a fork(two winning ways) after this move, take it.
Rule 4	Do not let the opponent create a fork after the player's move.
Rule 5	Move in a way such as the player may win the most number of possible ways.

Figure 2: Rule in Tic -Tac -Toe

The winning condition for this game is placing three similar marks in horizontal, vertical or diagonal rows.

Minimax Search Algorithm

Minimax algorithm is a kind of backtracking algorithm that is used in decision theory and game theory [4]. It uses game theory, decision theory, statistics and philosophy to find the optimal move for a player, assuming that the opponent also

plays optimally. It is commonly being used in two-player turn-based games such as Tic-Tac-Toe, Chess, etc. [11]

The player needs to fulfil two conditions in order to win a game. First, he needs to maximize his own chance of a win. There are two methods to maximize profit: a fork or win. A fork is an opportunity where he can win in two ways. A win is when there are two same 'X' or 'O' of him in a row, then he only needs to play the third one to get three same signs at a row. Second, he needs to minimize the opponent's winning chance. To minimize loss, he can block his opponent if two 'X' or 'O' of the opponent are in a row, or block opponent's fork.

The principle of the minimax search algorithm is to find the optimal path to minimize the maximum possible loss. As illustrated in Diagram 1, suppose that there are only one or two possible moves per player in each turn. $+\infty$ shows if the computer wins, and $-\infty$ shows if the computer loses. The AI or computer (or the maximizing player) is represented by circles while the opponent (or the minimizing player) is represented by squares. The look ahead is limited to four moves.

The algorithm evaluates the leaf nodes (terminating nodes or at a maximum depth of 4) using the heuristic evaluation

function. At level 3, the minimizing player will choose, for each node, the minimum of its children. In level 2, the maximizing player chooses the maximum of the children. The algorithm continues evaluating the maximum and minimum values of the child nodes alternately until it reaches the root node, where it chooses the move with the maximum value. This is the move that the player should make in order to minimize the maximum possible loss.

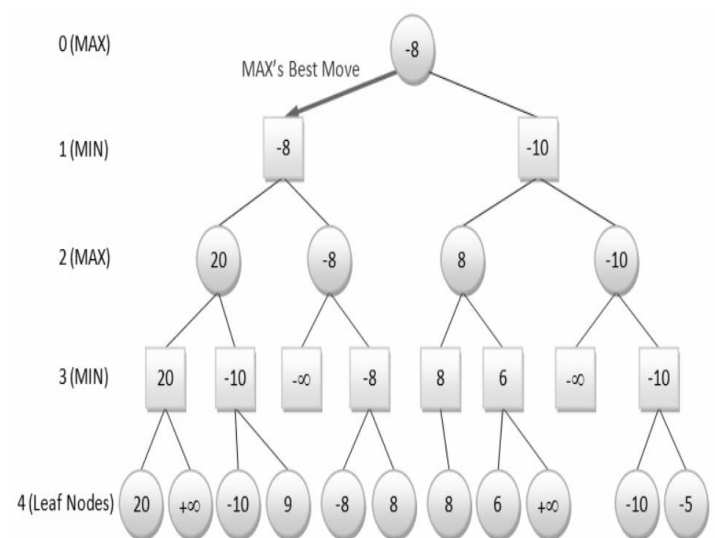


Figure 3: Illustration of the minimax algorithm

Alpha-Beta Pruning Algorithm

Alpha-Beta Pruning algorithm is an optimization algorithm for the minimax algorithm. It reduces the computational time by a huge factor. It allows faster search and even goes into deeper levels in the game tree. It will cut off branches of game trees that need not be searched when there is a better move that exists [5]. Alpha-beta pruning seeks to reduce the number of

nodes that needs to be evaluated in the search tree by the minimax algorithm. For the example illustrated in Figure 4, in the alpha cut-off, node C (MIN) cannot be more than 1 since node D returns 1. Since node B is 4 so there is no need to search the other children of node C, as node A will certainly pick node B over node C for the max node. As illustrated in Figure 4, the remaining children can be aborted if $\alpha \geq \beta$, for both the alpha cut-off and beta cut-off. [12]

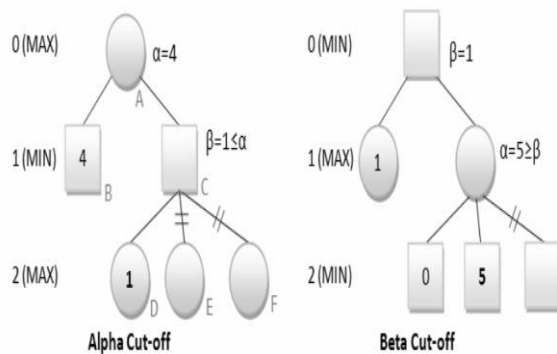


Figure 4: Illustration of the Alpha-Beta Pruning Algorithm

Game Flowchart

The game flowchart is shown in Figure 5. It will show how the game flow is for the tic-tac-toe game we have created.

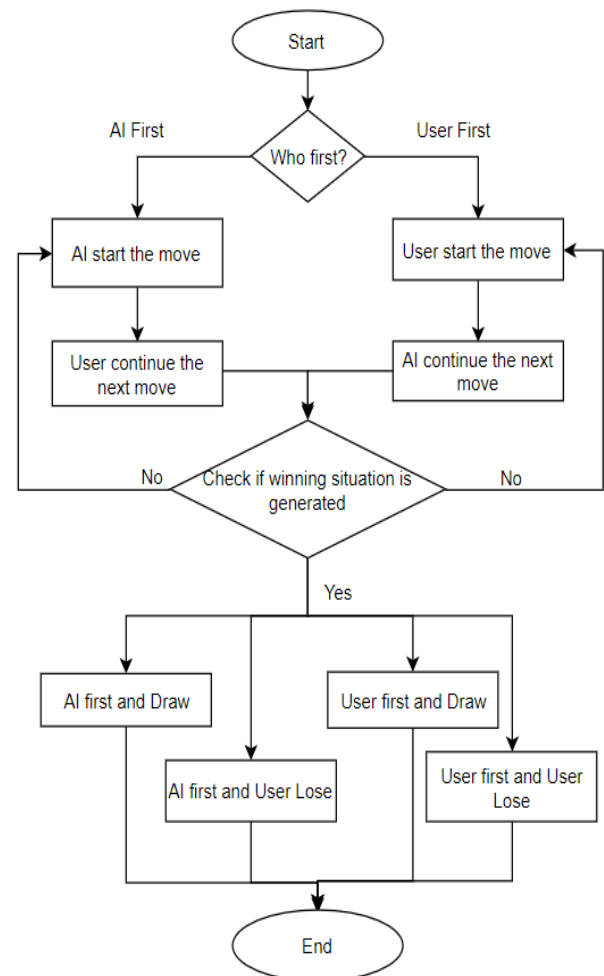


Figure 5: Game flowchart for the tic-tac-toe game

The flowchart above will briefly explain the game flow of this tic-tac-toe game we created using visual studio 2019. After starting the game, we will prompt out a menu and let the user pick whether he/she wants to be the first to make a move or let the AI to make a move first. Either one side makes a move first, the next turn will be for another side to make a move. Then, the winning situation is checked after both sides had made their move. If the winning condition is not generated, the game will continue with the same sequences as before. If the winning

condition is generated, the final result is shown to the user. Then, the user will be prompted again whether he/she wants to restart the game or exit the game.

4.Development

In this section, we will introduce our development process on how to produce the tic tac toe game with artificial intelligence heuristics. First, we perform pre-production process include rules-based study, game prototype and prototype testing. Second, we perform production process including programming and game design. Lastly, we perform post-production process including maintenance and testing.

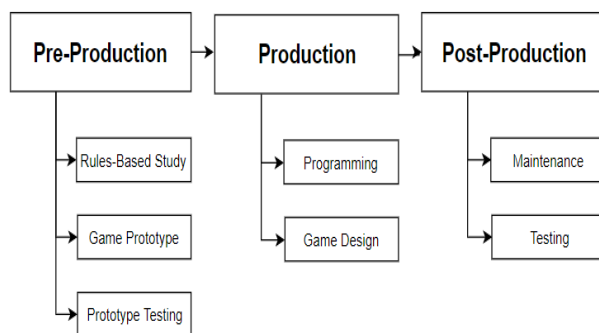


Figure 6: Game development process

This is a planning phase of the project. It aims to produce clear and easy to understand concept [6]. Concept is the game proposal or game plan. It includes the gameplay descriptions, features, target audience and also game interface design.

We have done a lot of rules-based study of tic tac toe game in order to produce clear game description to let user to understand. Also, we produce a paper

prototype for game interface including gameplay ideas and features. It acts as a proof of concept and to test ideas, by modifying some of the game features [7]. Later, we take the prototype testing result to produce the real game interface. This process allows us to gain more understanding about the final interface and the features to let users be more attracted by our game.

Production

This is the main stage of the game development process which cost us most of the time. We are focusing on two parts: implement heuristic concept in the source code and make more detailed design for the game.

In the programming part, we are writing a proper AI that plays a perfect game. An AI player needs to know the following: its intelligence level (which is the game's difficulty level), and the game it plays. This AI will consider all possible situations and makes the most optimal move. The AI is using both Minimax Search Algorithm and Alpha-Beta Pruning Algorithm. We also produce new functionality for to fulfill the game design and fix any bugs introduced during the game development process.

Game design is very important in designing the game content and game playing rules. We develop sound effects of

each action in the game playing process. Sound effects are important and can bring impact for the game's delivery [8]. Also, we design an attractive theme for the game and find suitable elements to suit the game theme. We also apply background image for each of the scene.

Post-Production

After all of the production process, we perform post-production development process including maintenance and testing process.

Maintenance is very important after game production [9]. We need to recheck the source code and fix bugs produced after we implement all of the game design. We also perform testing process to test the efficiency of AI in the game.

5. Testing

For the testing of the efficiency of artificial intelligent heuristic approach (min-max algorithm) applied in artificial intelligence in assisting it in performing the best move in every different situation, we had recruited 10 players who have played tic-tac-toe game before to complete with the artificial intelligence in a tic-tac-toe game. Each game consists of 3 rounds and the results are recorded in Figure 7.

<u>Players' Results against AI</u>				
Player #	Win	Lose	Draw	Total
1	0	1	2	3
2	0	0	3	3
3	0	2	1	3
4	0	0	3	3
5	0	0	3	3
6	0	1	2	3
7	0	0	3	3
8	0	0	3	3
9	0	0	3	3
10	0	1	2	3
Total	0	5	25	30

Figure 7: Result of the testing

A bar chart is tabulated and shown in Figure 8 for the analysis of data. As we can see in the bar chart, there is no win by the players when they are playing against artificial intelligence in the tic-tac-toe game. In 30 rounds, there are 25 draws and even 5 loses from the player.

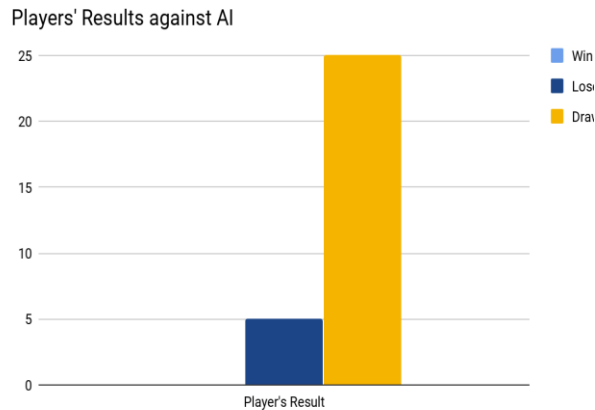


Figure 8: Bar chart for players' results against AI

A pie chart is also tabulated and shown in Figure 9 for the analysis of data. As we can see in the pie chart, there are about 83.3% of the draw and 16.7% of loss for the players' results when they are playing against AI in the tic-tac-toe game.

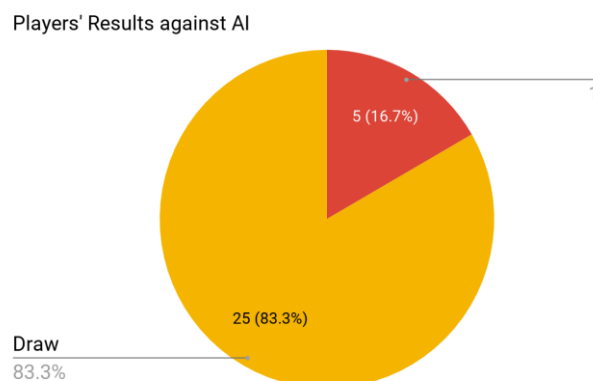


Figure 9: Pie chart for players' results against AI

From the data we have collected, it is proved that the artificial intelligence heuristic approach (min-max algorithm) applied in artificial intelligence is really efficient in assisting it in performing the best move in every different situation.

6. Conclusion

In this paper, we have introduced some introduction about the game, the related works regarding our paper such as the artificial intelligence heuristics and some algorithms. We also developed a tic-tac-toe game using the artificial intelligent heuristic approach. According to the testing we have done, it is proved that the artificial intelligence heuristic approach (min-max algorithm) applied in artificial intelligence is really efficient in assisting it in performing the best move in every different situation.

Acknowledgment

We thank Assoc.Prof Dr Azlan (University of Technology Malaysia), for giving guidelines and assistance in methodology and we thank Ong Le Foo, Tan Shi Xuan and Soh Jia Jun for their pearl of wisdom with us during the discussion. We also thank Mohd Azma, Looi Yao Wei, Fatin Nadiah, Steven Yong Wei, Mohd Luqman, Ngieng Chai Wen, Dinesh Aramugam, Ng Jia Qi, Hani Haziqah, and Johnson Tan for participating in our testing session.

References

1. Russell, Stuart J.; Norvig, Peter (2009). *Artificial Intelligence: A Modern Approach* (3rd ed.). Upper Saddle River, New Jersey: Prentice Hall. ISBN 978-0-13-604259-4.
2. Pearl, Judea (1984). *Heuristics: intelligent search strategies for computer problem solving*. United States: Addison-Wesley Pub. Co., Inc., Reading, MA. P. 3. OSTI 5127296
3. Java Graphics Tutorial -Case Study on Tic-Tac-Toe Part 2: With AI, Retrieved from https://www.ntu.edu.sg/home/ehchua/programming/java/JavaGame_TicTacToe_AI.html
4. Russell, Stuart J.; Norvig, Peter (2003), *Artificial Intelligence: A Modern Approach* (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, pp. 163–171, ISBN 0-13-790395-2
5. Saks, M.; Wigderson, A. (1986). "Probabilistic Boolean Decision Trees and the Complexity of Evaluating Game Trees". 27th Annual Symposium on Foundations of Computer Science. pp.29–38.doi:10.1109/SFCS.1986.44. ISBN 0-8186-0740-8.
6. Chandler, Heather Maxwell (2009). *The Game Production Handbook* (2nd ed.). Hingham, Massachusetts: Infinity Science Press. ISBN 978-1-934015-40-7.
7. Brathwaite, Brenda; Schreiber, Ian (2009). *Challenges for Game Designers*. Charles River Media. ISBN 1-58450-580-X.
8. Bethke, Erik (2003). *Game development and production*. Texas: Wordware Publishing, Inc. ISBN 1-55622-951-8.
9. Moore, Michael E. ; Novak, Jeannie (2010). *Game Industry Career Guide*. Delmar: Cengage Learning. ISBN 978-1-4283-7647-2.
10. Tic-Tac-Toe-Wikipedia Retrieved from <https://en.wikipedia.org/wiki/Tic-tac-toe>
11. Minimax-Wikipedia Retrieved from <https://en.wikipedia.org/wiki/Minimax>
12. Alpha-beta Pruning-Wikipedia Retrieved from https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning