Development of Tic-Tac-Toe Game Using Heuristic Search

Zain AM[1,2], Chai CW[1], Goh CC[1], Lim BJ[1], Low CJ[1], Tan SJ[1]

[1]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Skudai Johor, Malaysia
[2]UTM Big Data Centre, Ibnu Sina Institute for Scientific and Industrial Research, Universiti Teknologi Malaysia, Skudai Johor, Malaysia

E-mail: azlanmz@utm.my

Abstract: Tic-tac-toe is a fun game played by two players. Before starting the game, a 3x3 square grid is formed using two vertical and two horizontal lines. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner. In this study, we develop a computer program where a player can challenge artificial intelligent heuristic approach (opponent) to win the game. We want to prove that artificial intelligent heuristic approach enables in giving a great performance for development of tic-tac-toe game.

1. Introduction

The tic-tac-toe is a unique game as illustrated in Figure 1. A 3x3 grid is formed by using two vertical and two horizontal lines before the game starts. The players can fill the nine spaces with any two different sign normally crosses ('X') and noughts ('O') symbols [10]. In this study, we use Visual Studio 2019 software to develop the tic-tac-toe program based on artificial intelligent (AI) heuristic approach, min-max algorithm and alpha-beta pruning algorithm. When there are no player or less number of players be able to win the game, there is an evidence that the AI approach is effective for tic-tac-toe game development.

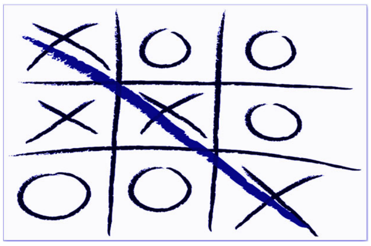


Figure 1. Tic-Tac-Toe game

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]. Nowadays, there is a lot of applications of AI in different fields. In astronomy, AI can be very useful in solving complex universe problems. In healthcare, AI could give a better and faster diagnosis than humans can be done. AI can also be used for gaming purposes, where AI machines can be used to play strategy games like chess, where the machine can perform the best move in different situations. In this study, AI is proposed for computer gaming where it acts as a challenger (opponent) in tic-tac-toe game.

2. Heuristics in Games

Heuristic search in AI 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 search technique can evaluate the available information and makes a decision on which branch to follow. The heuristic technique is capable to produce a solution that is good enough for the problem. [2].

Nowadays, many effective heuristics methods has been successfully applied in various problem domains. The heuristic methods are possible to be used in game development as they can help to inspire a creative player experience. In computer games, player want to enjoy the games, and programmer needs to have heuristic skills to guide the software for winning the game. Thus, the program needs to be intelligent in order to make user (player) more exciting with the game.

3. Methodology

We firstly create the rules needs for Tic-Tac-Toe game. Then we study the theory of Minimax Search and Alpha-Beta Pruning for tic-tac-toe game, logic for game development.

3.1 Rule-based Strategy

In the Tic-Tac-Toe game, the player is required to mark the nine spaces with any two different symbols normally crosses ('X') and noughts ('O'). The five rule-based strategy used in tac-tac-toe for player winning the games is given in Table 1 [3].

Table 1. Rule in Tic -Tac –Toe

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.

3.2 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. Firstly, the player needs to maximize the chance to win the game. Secondly, player needs to minimize the opponent's winning chance. The principle of the minimax search algorithm is to find the optimal path to minimize the maximum possible loss. Two possible results, $+\infty$ shows for computer wins, and $-\infty$ for computer loses. The steps of the Minimax search algorithm are summarized below.

i. Construct the complete game tree
ii. Evaluate scores for leaves using the evaluation function
iii. Back-up scores from leaves to root, considering the player type:
  - For max player, select the child with the maximum score

- For min player, select the child with the minimum score

iv. At the root node, choose the node with maximum value and perform the corresponding move

Figure 2 illustrates the minimax algorithm, where it evaluates the leaf nodes (terminating nodes or at a maximum depth of 4) using the heuristic evaluation function. The algorithm continues evaluating the maximum and minimum values of the child nodes alternately until it reaches the root node, where it chooses the maximum value.

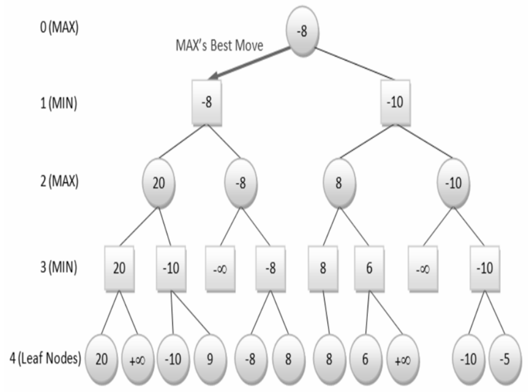


Figure 2. Illustration of the minimax algorithm

3.3 Alpha-Beta Pruning

Alpa-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 does not require to be searched when there is a better movement 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. The alpha cut-off process shown in Figure 3, node C (MIN) cannot be more than 1 since node D returns 1. Node B with value of 4 will not search the remaining children of node C, as node A will certainly pick node B over node C for the max node. The remaining children can be aborted if alpha ≥ beta, for both the alpha cut-off and beta cut-off [12]. The steps of the Alpha-Beta Pruning algorithm fundamentally is summarized as below.

i. Search down the tree to the given depth.
ii. Once reaching the bottom, calculate the evaluation for this node.(i.e. it's utility)
iii. Backtrack, propagating values and paths
iv. Attain the minimum score of the Alpha.

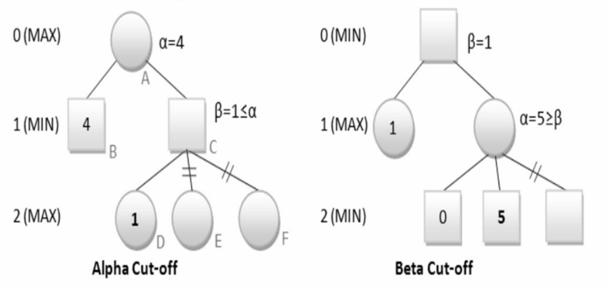


Figure 3. Illustration of the Alpha-Beta Pruning Algorithm

4. Development

This section discuss the development of the tic-tac-toe games using artificial intelligence heuristics as illustrated in Figure 4. Firstly, we perform pre-production process include rules-based study, game prototype and prototype testing. Secondly, we perform production process including programming and game design. Finally, we perform post-production process including maintenance and testing [6,7,9].

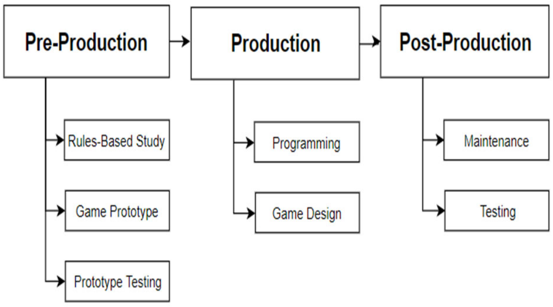
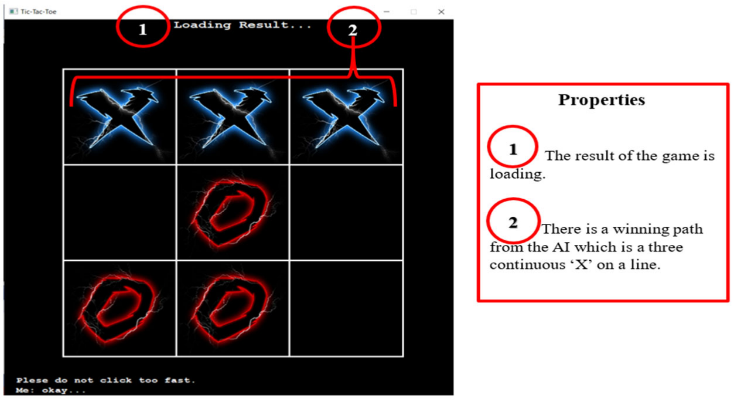

Pre-Production
Production
Post-Production
Rules-Based Study
Game Prototype
Prototype Testing
Programming
Game Design
Maintenance
Testing

Figure 4. Game development process

In the first phase, pre-production, we produce a proper 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. 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.

In the second phase, production, we focused on two parts: implementation of heuristic concept in the source code and provide more detailed design for the game. In the programming part, we provide a proper AI for perfect game [12,13]. Computer (opponent of game) needs to recognize game's difficulty level for all possible solution [14]. Then, we produce a design that enables the program fix with game rules. We develop a design for sound effects where it is important to give impression for the game's delivery [8]. Then, we design attractive theme for the game and find right elements to suit the game theme. We also apply background image for each of the scene. Example of game interface is illustrated in Figure 5.

In the third phase, we perform post-production development process including maintenance and testing. Maintenance is very important next to game production, where it is necessary for programmer to recheck the source code and fix all the bugs produced after executing the game design. Then, testing is a process to test the efficiency of the game discussed in next section.



Loading Result...

Plese do not click too fast.
Me: okay...

Properties

1 The result of the game is loading.

2 There is a winning path from the AI which is a three continuous 'X' on a line.

Figure 5. Game interface

5. Testing

We recruit ten players in order to test the efficiency of tic-tac-toe game that we developed. Each game consists of 3 rounds and the results are recorded in Table 2. Then, we generate a bar chart in Figure 6 to demonstrate the winner of the game. We observed that there is no player won the game against the computer (opponent). The results showed the results of 30 games (three trials given to each player), and it was found that the players win, lose, and draw the games against the computer are 0, 5, and 25 respectively.

Table 2. Result of the testing

Player #	Win	Lose	Draw	No. of games
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**

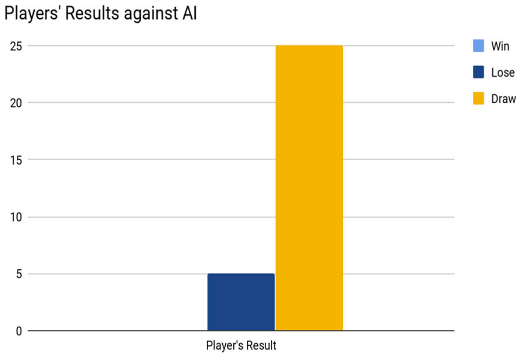


Figure 6. Bar chart for players' results against AI

6. Conclusion

This paper has discussed and developed the tic tac toe game using artificial intelligence heuristic approach. It was proved that min-max and alpha-beta pruning algorithms have extraordinary talent in giving a great performance for tic-tac-toe game.

Acknowledgement

Special appreciations to editor and all reviewers on the useful advices and comments provided. The authors greatly acknowledge the Research Management Centre, Universiti Teknologi Malaysia (UTM) (UTM-TDR - vot. No. Q.J130000.3551.05G41).

References

[1] Russel S, Norvig P. *Artificial intelligence: a modern approach.* Pearson Education Limited; 2013.

[2] Pearl J. *Intelligent search strategies for computer problem solving*. Addision Wesley. 1984.

[3] Stuart R, Peter N. *Artificial intelligence: a modern approach*. 2003

[4] Moret BM. *Decision trees and diagrams*. ACM Computing Surveys (CSUR). 1982 Dec 1;14(4):593-623.

[5] Fullerton T. *Game design workshop: a playcentric approach to creating innovative games*. CRC press; 2014 Mar 5.

[6] Brathwaite B, Schreiber I. *Challenges for game designers*. Nelson Education; 2009.

[7] Bethke E. *Game development and production*. Wordware Publishing, Inc.; 2003.

[8] Prasad B, Reddy MS, Reddy VS, Kishore RR. *Game Development-Bounty Rescuestep*. Empirical Research Press Ltd.. 2017.

[9] Siegel AN. *Combinatorial game theory*. American Mathematical Soc.; 2013 Aug 1.

[10] Motreanu D, Panagiotopoulos PD. *Minimax theorems and qualitative properties of the solutions of hemivariational inequalities*. Springer Science & Business Media; 2013 Dec 1.

[11] Knuth DE, Moore RW. *An analysis of alpha-beta pruning*. Artificial intelligence. 1975 Dec 1;6(4):293-326.

[12] Sarkheyli A, Zain AM, Sharif S. A multi-performance prediction model based on ANFIS and new modified-GA for machining processes. Journal of intelligent manufacturing. 2015 Aug 1;26(4):703-16.

[13] Zhou KQ, Zain AM, Mo LP. A decomposition algorithm of fuzzy Petri net using an index function and incidence matrix. Expert Systems with Applications. 2015 May 15;42(8):3980-90.

[14] Deris AM, Zain AM, Sallehuddin R. Overview of support vector machine in modeling machining performances. Procedia Engineering. 2011 Jan 1;24:308-12.