iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** II **Month of publication:** February 2022

DOI: https://doi.org/10.22214/ijraset.2022.40115

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue II Feb 2022- Available at www.ijraset.com

Tic Tac Toe by Minimax Alpha-Beta Pruning Using Arduino

Mallikarjun Mudda[1], Sandeep Hemdribhotla[2], T. Krishna Chaitanya[3], T. Rahul[4]
[1]*Associate Professor,* [2, 3, 4]*Students, Department of Electronics and Communication Engineering, Sreenidhi Institute of Science and Technology, India*

Abstract: ***Heuristic algorithms are put to the test in games. Because they incorporate an unexpected opponent, two-person games are more challenging than a basic puzzle. In a two-player game, you must presume that your opponent has the same information as you and can use it just as well. As a result, you must presume that your opponent will make the best available move at each level of the game. The minimax method is built on this foundation. The players in minimax are referred to as MAX (the player) and MIN (the player) (the opponent).***

Both aim to get the most out of their actions. The player, MAX, is attempting to maximise her score. And MIN is the opponent attempting to reduce MAX's score to the bare minimum. It is impossible to stretch the graph to its leaf nodes in most games.The number of layers in the state space is increased to n. The heuristic evaluation function assigns a value to each leaf node in this subgraph. The values are then passed back to the root node. The heuristic value of the best state that can be attained from that node is represented by the value propagated back. The process of alpha-beta pruning is used to reduce the amount of computation and searching required during minimax. Minimax is a two-pass search, with the first pass assigning heuristic values to nodes at the lowest depth and the second pass propagating those values up the tree. The depth-first approach is used in alpha-beta search. A MAX node's initial or temporary value is called an alpha value.Because MAX nodes receive the highest value from their descendants, an alpha value can never fall; it can only rise. A beta value is a value linked with a MIN node that is initial or transitory. A beta value can never increase; it can only go down because MIN nodes are assigned the lowest value among their progeny. Assume alpha = 6 for a MAX node. Then any branches originating from a MIN descendant with a beta value less than or equal to 6 won't be included in the search. So, if you know that the MAX node has an alpha of 6 and that one of its MIN children has a beta of less than or equal to 6, you don't need to look any farther down the MIN node. This is called Alpha Beta Pruning.

Keywords: component, formatting, style, styling, insert (key words)

I. INTRODUCTION

A. Motivation

The game industry is exploding, with several businesses of differing sizes, ethos, reach, and beyond. Developers put forth a lot of effort to make these video games successful. Every feature of each character, as well as the things in their environment, must be hand-coded. Repetitive work consumes a large portion of development time, resulting in an increase in errors and logical defects. Artificial intelligence has been utilised in software games to emulate human players, allowing game designers to create unique experiences and outcomes for each player. Computer chess players are well-known examples, wherein modern chess programs are trained to defeat best human players. AI based algorithms that can be implemented for games, but a need for optimal solutions is on a rise. We require a comparative analysis of multiple algorithms for understanding the most efficient and ideal one. In our work, through use of a game Tic-Tac-Toe various algorithms will be carried out with its prototype compared in terms of effective rate and optimality.

B. Problem Statement

Shortly after, difficulties of this nature emerged into a significant obstacle for the development of artificial intelligence, one of today's most popular topics in computer science. World champions in numerous strategic games, such as Chess, Checkers, Backgammon, and, most recently (2016), Go, have all been defeated by computers.

Although these algorithms are extremely successful, their decision-making process differs significantly from that of humans. Minimax algorithm has many branch divisions which are otherwise redundant in the implementation of Alpha Beta pruning.

C. Objectives

The objective of this project is to develop a java implementation of the Alpha Beta pruning algorithm and reduce the time complexities associated with the normal brute forces approaches.
The temporal complexity of the alpha-beta method is reduced due to the number(b) of times the function(M) must be run. 5. Final thoughts The tic tac toe game is created with the help of the alpha-beta algorithm and brute force. Based on an actual time setting, the outcome is evaluated with two players, a player and a machine. For game 3x 3, Randomized Alpha-Beta and Revamped Alpha-Beta were used.

D. Main Contribution

Tic-tac-toe is a paper and pencil game played by two teams in which the players take turns numbering the gaps in a three-by-three grid. To win, one of the rows must have three X or O marks. Because tic-tac-toe is a simple game, it is used as a pedagogical technique to instil good sportsmanship. It's customary to write computer programmes to play tic-tac-toe properly, or to list the 765 basically different locations, or the 26,830 potential games on this space up to rotations and reflections.Brute Force Algorithms are a type of programming method that doesn't use any methods to boost performance, instead relying on brute computational strength to try out all possible solutions until the problem is solved.The Alpha-Beta Minimax algorithm is used to prune the branches of the tree that will not contribute favourably to the goal state that must be reached. Alpha – Beta Pruning allows you to make the most of your space while also making the most of your time. Tic-Tac-Toe Toe's is to win several games while also making efficient use of the system.

II. LITERATURE SURVEY

A. Tic-Tac-Toe Game

Tic-tac-toe is a game in which two players alternately place the marks X and O in one of the nine places on a three-by-three grid.

The first player (X) in the following example wins the game in seven steps:

There is no universally agreed-upon rule for who goes first, however in this article, the tradition is that X goes first.
Players quickly learn that the best play from both parties results in a tie. As a result, tic-tac-toe is frequently played by young children who may not yet have figured out the best approach.
Tic-tac-toe is frequently used as an educational tool for teaching the notions of good sportsmanship and the branch of artificial intelligence that deals with the search of game trees due to its simplicity. On this space, writing a computer programme to play faultless tic-tac-toe or enumerating the 765 basically different places or the 26,830 possible games up to rotations and reflections (game tree complexity) is simple.
The game can be simplified to a m,n,k-game, in which two players alternate laying stones of their own colour on an m-by-n board in order to get k of their own colour in a row. The 3,3,3-game is tic-tac-toe. [3] Harary's generalised tic-tac-toe is a more expansive version of the classic game. It's also known as a n power d game, with n equal 3 and d equal 2. [4] It can be made even more broad by using an arbitrary incidence structure with rows as lines and cells as points. The incidence structure of tic-tac-toe consists of nine points, three horizontal lines, three vertical lines, and two diagonal lines, each with at least three points.

B. Game Strategies

During the initial turn, the first player, who will be named "X," has three crucial unique positions to mark. On the surface, it appears that there are nine viable placements, one for each of the nine squares in the grid. However, by rotating the board, we can see that every corner mark is strategically identical to every other corner mark in the first turn. Every edge mark is the same way. There are only three conceivable first marks from a strategic standpoint: corner, edge, or centre. From all of these beginning positions, Player X can win or force a draw; but, playing the corner provides the opponent the lowest number of squares to play in order to avoid losing.

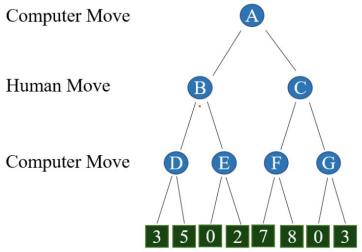


Figure 1

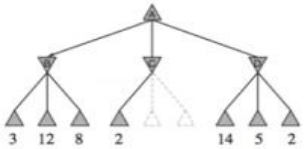
An example of a game tree is shown in Figure 1.
When it is the computer's move, the value of each node is decided by taking the maximum of the node's immediate children, and when it is the human's move, the value is determined by taking the minimum of the node's immediate children.
The game is then solved from the bottom up in this manner.The aim for the machine is for the game to conclude with the greatest possible value, whereas the human goal is for the game to end with the lowest possible value.Starting at the bottom of the tree, the minimax method asserts a value to each of the nodes.The value of a node when it is the computer's turn (nodes A, D, E, F, and G) is equal to the maximum of the children's values, but the value of a node when it is the human's turn (nodes A, D, E, F, and G) is equal to the lowest of the children's values.The value of node D in this tree is equal to the maximum of 3 and 5, which is 5.Similarly, nodes E, F, and G have values of 2, 8, and 3, respectively.Working their way up the tree, the human now has to pick between the lowest value of nodes D and E (5 and 2), which is 2, resulting in a value of 2 for node B.In the same way,nodeC'svalue is3.Finally,thecomputer at A should choose the maximum value out of nodes B and C, which is 3 for node C. So the best move for the computer at A is C.

C. The Minimax Algorithm

In decision-making and game theory, the mini-max algorithm is a recursive or backtracking method.
It suggests the best move for the player, provided that the opponent is likewise playing well.The Mini-Max method searches the game tree using recursion.In AI, the Min-Max algorithm is mostly employed for game play.Chess, checkers, tic-tac-toe, and other two-player games are examples.This Algorithm calculates the current state's minimax choice.The game is played by two players, one named MAX and the other named MIN, in this algorithm.Both players are fighting it, since the opponent player receives the smallest advantage while they receive the most profit.Both players in the game are adversaries, with MAX selecting the maximum value and MIN selecting the min.For the investigation of the whole game tree, the minimax method uses a depth-first search strategy.The minimax method descends all the way to the tree's terminal node, then recursively backtracks the tree.

D. Alpha Beta Pruning

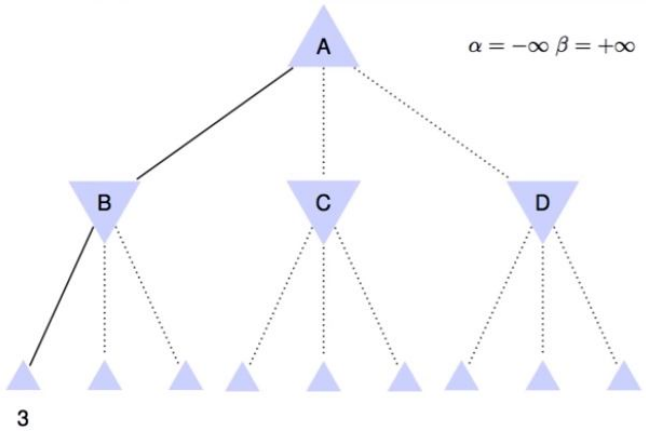
The minimax approach employs a depth-first search strategy to investigate the whole game tree. The minimax approach descends all the way to the tree's terminal node, then backtracks the tree recursively.The goal is to find the game tree's minimax. Pruning is a strategy for computing the proper minimax choice without having to inspect each node of the game tree. It's named alpha-beta pruning because it involves two threshold parameters, Alpha and Beta, for future expansion. At any point along the Maximizer path, Alpha is the best (highest-value) option we've uncovered so far. The value of alpha starts at -.
Beta: At any point along the Minimizer path, this is the best (lowest-value) option we've uncovered thus far. Beta has a value of + in the start. The Alpha-beta pruning process removes those nodes that aren't truly effecting the final result but are slowing down the algorithm.Hence by pruning these nodes, it makes the algorithm fast.



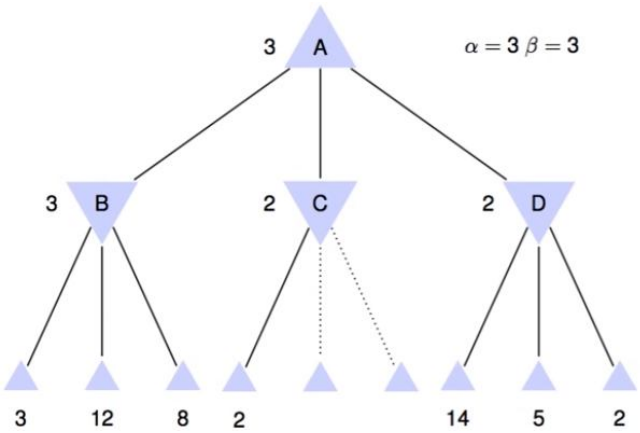
$$\begin{aligned} minimax(root) &= max(min(3,12,8), min(2,x,y), min(14,5,2)) \\ &= max(3, min(2,x,y), 2) \\ &= max(3,z,2) \\ &= 3 \end{aligned}$$

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue II Feb 2022- Available at www.ijraset.com

On the diagram above, we have an example. If we execute minimax at the root level A, the nodes B, C, and D will be the maximum of their values below the tree, and each of these nodes will have to be the minimum of their values. We don't require z because the result is 3. We don't require z, which is the min(2,x,y), because z = 2, and the max(3,z,2) equals 3, therefore we don't need to look at the other two C nodes.The value of the best choice so far for MAX (highest value) is Alpha, and the value of the best choice so far for MIN is Beta (lowest value). Each node maintains a record of its Alpha and Beta values.



We begin with Alpha = - and Beta = +, and we want to raise the Alpha value while also ensuring that Alpha is less than or equal to Beta.We have 3 as the first value in the previous example, and now we must determine whether 3 is less than Alpha (-). We can update Beta = 3 because 3 is larger than Alpha and less than Beta (+). This procedure is repeated for all of the values in the tree. The root node A's minimax at the end of the procedure is 3, and we didn't look at the two undefined final nodes of C.Furthermore, rearranging the final nodes of D by 2,5,14 reveals the answer for this node, which is 2.



Importantly, when dealing with a big number of od nodes, this technique saves us a lot of time and allows us to get to the answer faster without compromising the end result. Meta reasoning is the term for this technique.

E. Arduino

Arduino is an open-source programmable circuit board that may be used in a number of basic and complicated makerspace projects. This board has a microcontroller that may be designed to detect and control items in the real world. The Arduino can interact with a wide range of outputs, including LEDs, motors, and displays, by responding to sensors and inputs. Arduino has been a popular choice for makers and makerspaces wishing to construct interactive hardware projects because to its adaptability and low cost Massimo Banzi launched Arduino in 2005 in Italy as a method for non-engineers to use a low-cost, easy-to-use tool for building hardware projects. Because the board is open source, it is distributed under a Creative Commons licence, allowing anybody to make their own. There are hundreds of Arduino compatible clones and modifications accessible on the internet, but the only official boards have Arduino in their name. We'll go through a couple of the Arduino boards that are available and how they vary in the following section.

Arduino is a fantastic platform for developing ideas and inventions, but choosing the proper board may be difficult. If you're new to this, you could have assumed there was only one "Arduino" board and that was it.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue II Feb 2022- Available at www.ijraset.com

In truth, there are several versions of the genuine Arduino boards, as well as hundreds of clones from rivals. But don't worry; later in this guide, we'll show you which one to start with. The following are some examples of the many types of Arduino boards available. The authentic Arduino boards are those with the Arduino logo on them, although there are many excellent clones on the market as well Below are a few examples of the different types of Arduino boards out there. The boards with the name Arduino on them are the official boards but there are also a lot of really great clones on the market as well. One of the best reasons to buy a clone is the fact they are generally less expensive than their official counterpart. Adafruit and Sparkfun for example, sell variations of the Arduino boards which cost less but still have the same quality of the originals. One word of caution, be careful when buying boards from companies you don't know.

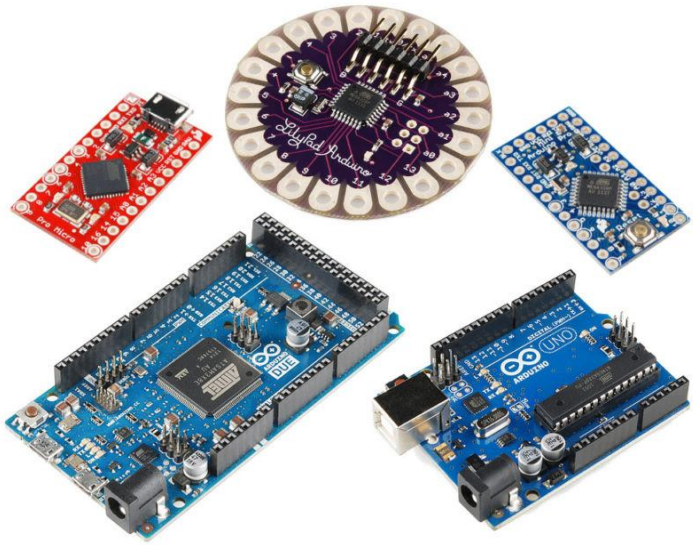
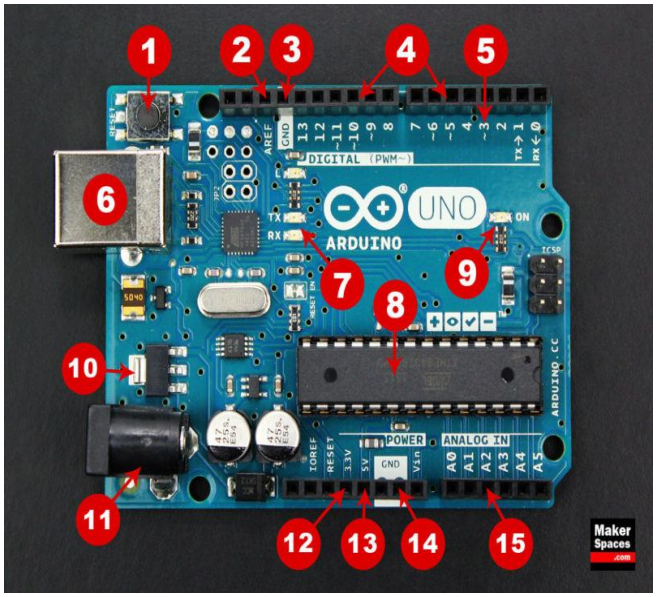


Image credit – Sparkfun.com

Next, we're going to focus on our favorite Arduino board which we recommend beginners start with.

1) *Arduino UNO:* The Arduino Uno is one of the most popular Arduino boards. Despite the fact that it was not the first board to be launched, it is still the most popular and well-documented on the market. Because of its widespread use, the Arduino Uno has a plethora of project guides and forums available on the internet to assist you in getting started or getting out of a bind. Because of its amazing features and ease of use, we're big admirers of the Uno.



F. Board Breakdown

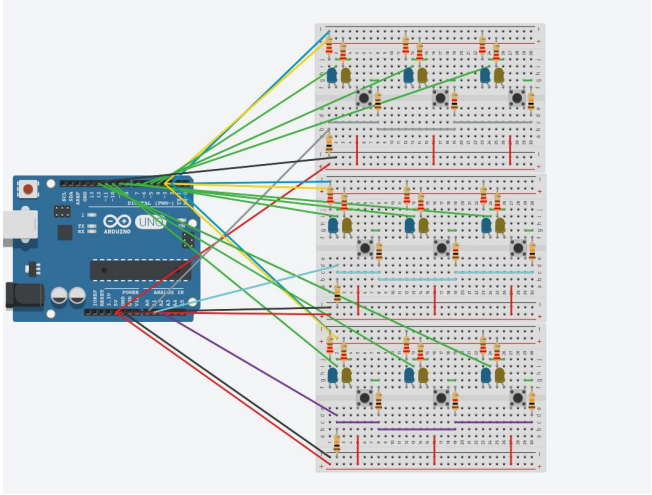
Here are the components that make up an Arduino board and what each of their functions are.

1) *Reset Button:* This will restart any code that is loaded to the Arduino board
2) *AREF:* Stands for "Analog Reference" and is used to set an external reference voltage
3) *Ground Pin:* There are a few ground pins on the Arduino and they all work the same
4) *Digital Input/Output:* Pins 0-13 can be used for digital input or output
5) *PWM:* The pins marked with the (~) symbol can simulate analog output
6) *USB Connection*: Used for powering up your Arduino and uploading sketches
7) *TX/RX:* Transmit and receive data indication LEDs
8) *ATmega Microcontroller:* This is the brains and is where the programs are stored
9) *Power LED Indicator:* This LED lights up anytime the board is plugged in a power source
10) *Voltage Regulator:* This controls the amount of voltage going into the Arduino board
11) *DC Power Barrel Jack:* This is used for powering your Arduino with a power supply
12) *3.3V Pin:* This pin supplies 3.3 volts of power to your projects
13) *5V Pin:* This pin supplies 5 volts of power to your projects
14) *Ground Pins:* There are a few ground pins on the Arduino and they all work the same
15) *Analog Pins:* These pins can read the signal from an analog sensor and convert it to digital

The Arduino Uno requires a power supply to function and may be charged in a variety of ways. You may connect the board directly to your computer using a USB cable, as most users do. Consider utilising a 9V battery pack to power your project if you want it to be portable. The final option is to utilise a 9V AC power supply.

III. WORKING OF THE PROJECT

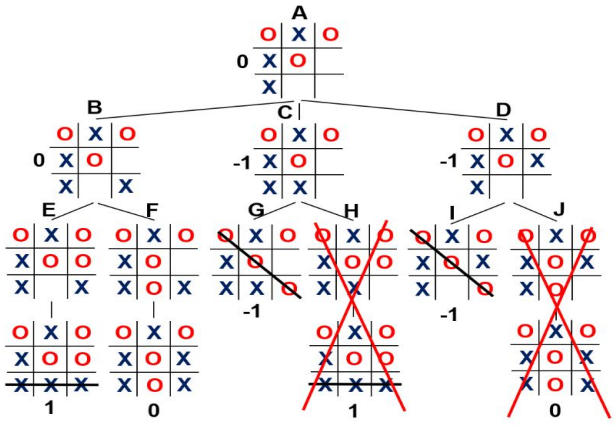
The heart of our project is minimax algorithm with alpha beta pruning and the implementation of tic tac toe game. The tic tac toe game is represented in the form of LED's. Representing the 3X3 matrix of the game, two leds are used to represent either o or x at each cell. Alpha beta pruning code is implemented using Arduino uno. For giving input push buttons are used. Each turn of the input, led is turned on alternatively(each player selects either o or x)



The traversal begins on node A, where the computer is in command, hence player is set to one. Then, on A's first child, node B, whose player is -1, Minimaxab is called.

1) Minimaxab is then invoked on E, node B's first child. We contact E's lone kid, who responds with a 1. At B, E returns 1 and val = 1.
2) The computer now determines min(val, best), resulting in best = min(, 1) = 1.
3) beta = min(beta, best) = min(, 1) = 1; beta = min(beta, best) = min(, 1) = 1; beta = min(beta, The for loop continues since alpha is still alpha, beta equals 0. Because B has no more children, minimaxab returns 0 to node B.
4) The computer compares the value of B's right child, which is zero, to the value of val, which is one. Because 0 1 equals 0, val equals 0 and beta

```
Minimaxab(board, player, alpha, beta) :
    if board is a leaf node :
        return the value of board
    if player = 1 :
        best = -∞
        for each child of board :
            val = minimaxab(child, -1, alpha, beta)
            best = max(val, best)
            alpha = max(best, alpha)
            if alpha ≥ beta :
                break
        return best
    else :
        best = ∞
        for each child of board :
            val = minimaxab(child, -1)
            best = min(val, best)
            beta = min(best, beta)
            if alpha ≥ beta :
                break
        return best
```



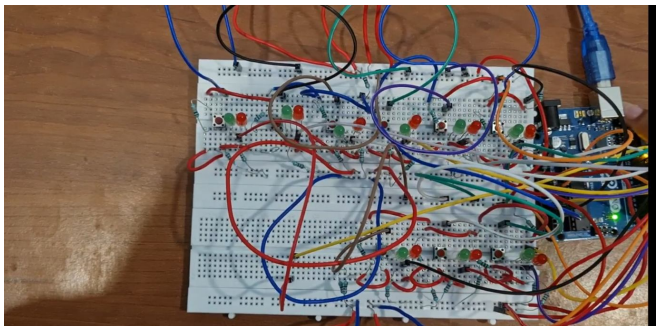
A simple for loop is used in the standard minimax method to examine child nodes in the same order in every area of the tree. Each child node in Tic-Tac-Toe is rated from 1 to 9, starting at the top left and going left to right, top to bottom. The sequence in which each of the nine children is assessed would be arbitrary if randomization was enabled. Ordinary minimax randomness, on the other hand, will not assist because the full tree must be produced anyhow. However, in the case of alpha-beta pruning, the order in which the children are assessed is critical since it may result in more or less pruning depending on whether the computer is "fortunate" enough to short-circuit sections of the tree.Both random and non-random algorithms will need to analyse the entire tree in the worst-case scenario. With randomization enabled, the computer has a greater chance of finding "fortunate" offspring by going through them in a random sequence at each node. This is because, with 1/-1 termination, the computer stops looking once it discovers something. Without randomization, the machine is more prone to repeat failed searches.

The number of nodes created in the initial move of a 3x3 Tic-Tac-Toe game was reduced to 1,893 when all four of these characteristics were enabled, compared to a total tree size of 549,945 when regular minimax was used. However, this was based on the assumption that screening for a "forced relocation" was a good idea (items 1 and 2).This is done for free and does not qualify as node generation. We may still deduce that the runtime was considerably decreased compared to the original minimax.

Other finite board games like Tic-Tac-Toe and Hex aren't too difficult to adapt the minimax algorithm and its modifications to. Only the functions that check for win conditions on the board must be written.

Aside from that, only small changes to the code are required to accommodate for the varying board sizes and regulations. Another game that might be investigated using our algorithms in the future is Connect Four. We might also improve our code by developing an evaluation function that can evaluate non-terminal node values.

IV. RESULTS AND CALCULATIONS



As shown in the above photograph, we have connected the circuit and implemented the tic tac toe game. Theleds(one for o, one for x) are in coordination and are completing the game.

A. Algorithmic Complexity

Randomization had no effect in Tic-Tac-Toe with forced steps switched on. Pruning doesn't help much in Tic-Tac-Toe since there are so many needless movements. Without pruning, randomness has no value since there is no risk of tree short-circuiting. Randomness reduced the number of nodes tested during 1000 trials by around 6-10% when the simulation was run without the forced movement characteristics. The Alpha-Beta Pruning has the following level of difficulty
O ((nM)^N).
For the tic tac toe game, devised an improvised AB algorithm.
Our Algorithm is built up in such a way that we take up the ends, knowing that successful methods begin with taking the corners. The method gradually reduces the state space by pruning and time with few recursive calls while the player plays. The complexity that has been estimated is: O (bM)

V. CONCLUSIONAND FUTURE SCOPE

Thus we have implemented the tic tac toe game on Arduino using the Alpha Beta pruning algorithm. We have demonstrated the working of the LEDs and the zero sum game feature of the game(assuming both are efficient players). We have increased the efficiency of the process by elimating the need for checking the redundant nodes. By adding more characteristics to the Tic-Tac-Toe algorithm, we can make it even better.
These are some of them:

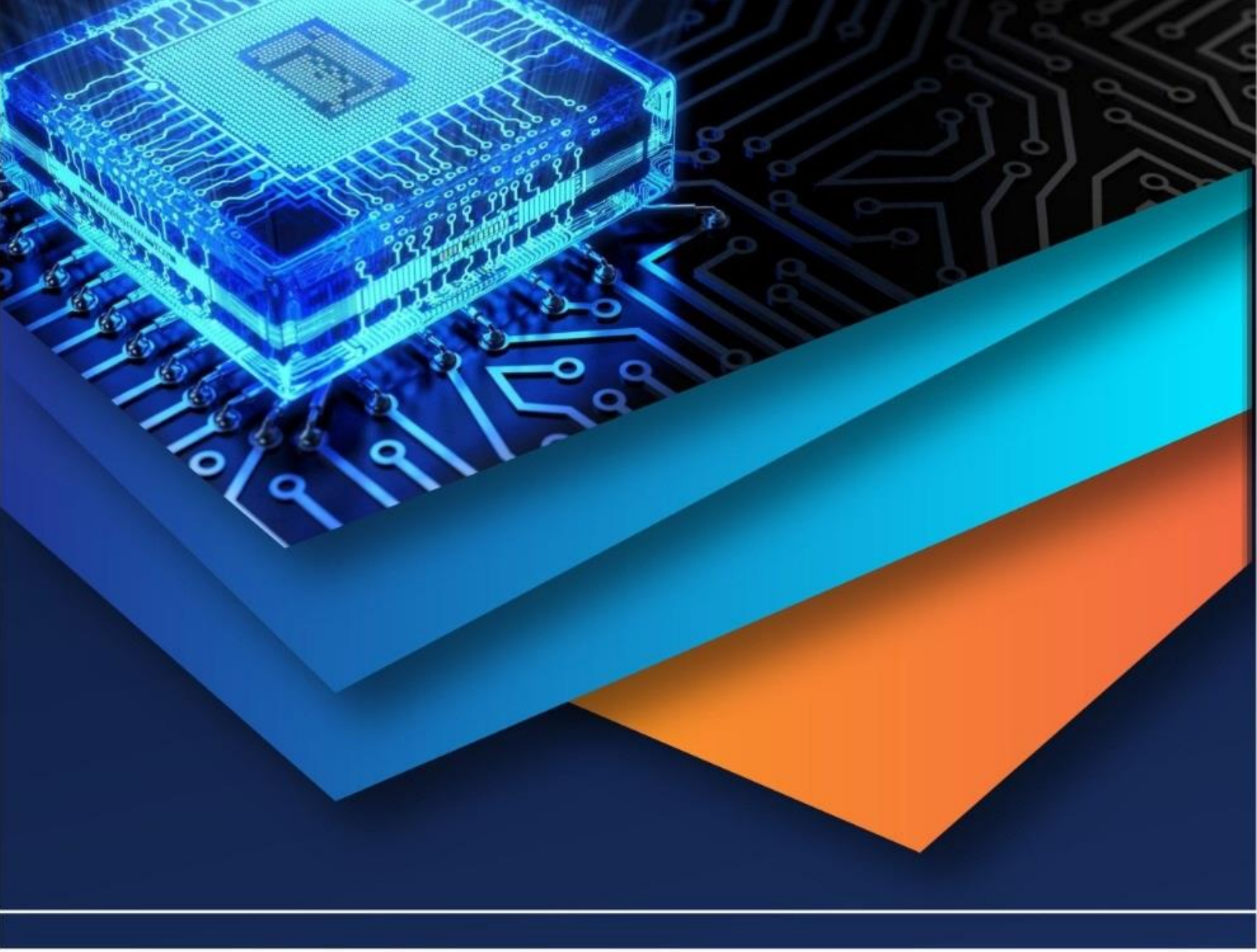
1) Check to see if there is a spot where the computer can win the game right away and play there.
2) If the game cannot be won right away, look for a spot where the human will win the following turn and play there to prevent the human from doing so.

Other finite board games like Tic-Tac-Toe and Hex aren't too difficult to adapt the minimax algorithm and its modifications to. Only the functions that check for win conditions on the board must be written.
Aside from that, only small changes to the code are required to accommodate for the varying board sizes and regulations. Another game that might be investigated using our algorithms in the future is Connect Four. We might also improve our code by developing an evaluation function that can evaluate non-terminal node values.Then, if a node is in a non-terminal state at a particular depth, the evaluation function is applied to it, and the node is considered as a leaf with that value. A more advanced software would try to calculate the "volatility" of different board situations and go deeper into regions of the game tree where movements can have a greater impact on the game's status.

REFERENCES

[1] Nidal M. Jodeh, B., MAS. "Development of Autonomous Unmanned Aerial Vehicle Research Platform: Modeling, Simulating, and Flight Testing." Air Force Institute of Technology, March 2006.
[2] Donohue, C. "NASA Study To Use a Predator B-class Unmanned Aerial System (UAS) In Support Of Arctic/Antarctic Polar Missions." Anarctic Meteorological Observation, Modeling and Forecasting Workshop, 13-15 June 2006.
[3] Kargin, V. "Design of an Autonomous Landing Control Algorithm for a Fixed Wing Uav." Middle East Technical University, October 2007.
[4] Mueller, E. R. "Hardware-in-the-loop Simulation Design for Evaluation of Unmanned Aerial Vehicle Control Systems," AIAA Modeling and Simulation Technologies Conference and Exhibit. Hilton Head, South Carolina,2007.
[5] Leong, H. I. "Development of a 6DoF Nonlinear Simulation Model Enhanced with Fine Tuning Procedures,"2008.

[6] Jager, R. "Test and Evaluation of the Piccolo II Autopilot System on a One-Third Scale Yak-54." Vol. Msc, University of Kansas, April, 2008.
[7] Paw, Y. C. "Synthesis and Validation of Flight Control for UAV." University of Minnesota, 2009.
[8] Al-Radaideh, A., Al-Jarrah, M., Jhemi, A., and Dhaouadi, R. "Arf60 Aus-UavModeling,SystemIdentification,Guidance and Control: Validation Through Hardware In The Loop Simulation," Mechatronics and its Applications, 2009. ISMA'09.6th International Symposium on. IEEE, 2009, pp. 1-11.
[9] Bayezit, I. "The Autonomy of Fixed-Wing Aerial Vehicles And Experimental Design Steps On Implementing Autonomous Navigation, Landing And Take Off Operations For The Trainer60 Model Aircraft," 5. Ankara International Aerospace Conference. METU, Ankara, August, 2009.
[10] Edwards, C. D. "Nonlinear Six Degree-of-Freedom Simulator for a Small Unmanned Aerial Vehicle." Mississippi State University, May 2010.
[11] Mallikarjun Mudda[1], .Manjunath R[2], Krishnamurthy N[3] "Enhanced Image Registration Technique for Medical Image Segmentation" in International Medical Sciences Academy, Vol 30; No. 3
[12] Mallikarjun Mudda, Syed Jahangir "Target Tracking System: PAN TILT", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-8 Issue-5, June 2019.
[13] Mallikarjun Mudda , R. Manjunath & N. Krishnamurthy (2020): Brain Tumor Classification Using Enhanced Statistical Texture Features, IETE Journal of Research, DOI: 10.1080/03772063.2020.1775501
[14] Mallikarjun Mudda, Farheen: Artificial neural network based weather pridiction system", international journal of scientific & technology research. volume 9, issue 02, ISSN 2277-8616, PP-5587-5594, FEBRUARY 2020.
[15] Shivleela Mudda, K.M. Gayathri, Mallikarjun Mudda: Compact High Gain Microstrip Patch Multi-Band Antenna for Future Generation Portable Devices Communication, 2021 International Conference on Emerging Smart Computing and Informatics (ESCI) | 978-1-7281-8519-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/ESCI50559.2021.9396776.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY