Department of Artificial Intelligence and Data Science
Vivekanand Education Society's Institute of Technology
2021-2022

CONNECT 4

Submitted in partial fulfilment of the requirements
for the degree of

Bachelor of Engineering in
Artificial Intelligence and Data Science

by

Heramb Pawar-67

Manvi Gour-68

Shreyas Satre-69

under the guidance of
Supervisor:
Mrs. Sangeeta Oswal

VIVEKANAND EDUCATION SOCIETY'S
Institute of Technology

Department of Artificial Intelligence and Data Science

CERTIFICATE

This is to certify that **Heramb Pawar, Manvi Gour & Shreyas Satre** of Second Year of Artificial Intelligence and Data Science studying under the University of Mumbai have satisfactorily presented the Mini Project titled **Connect 4** as a part of MINI-PROJECT assessment for Semester-III under the guidance of **Mrs. Sangeeta Oswal** in the academic year 2021-2022.

Date: 26/01/2022

(Name and sign)
Head of Department

(Name and sign)
Supervisor/Guide

VIVEKANAND EDUCATION SOCIETY'S
Institute of Technology

Department of Artificial Intelligence and Data Science

DECLARATION

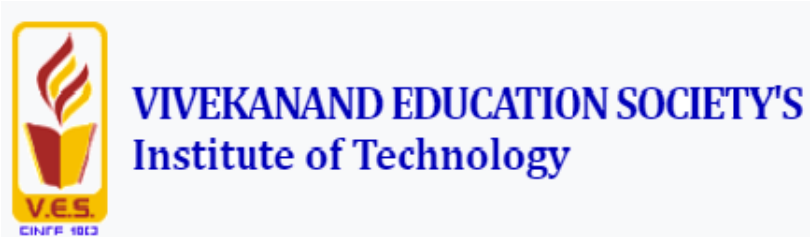
We, **Heramb Pawar, Manvi Gour, Shreyas Satre from class D6AD,** declare that this project represents our ideas in our own words without plagiarism and wherever others' ideas or words have been included, we have adequately cited and referenced the original sources.

We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our project work.

We declare that we have maintained a minimum 75% attendance, as per the University of Mumbai norms. We understand that any violation of the above will be cause for disciplinary action by the Institute.

Yours Faithfully,

1. **Heramb Pawar**
2. **Manvi Gour**
3. **Shreyas Satre**



ACKNOWLEDGEMENT

We are sincerely grateful to our college, VESIT, our respected principal **Dr. J. M. Nair** ma'am and our HOD **Dr. M. Vijayalakshmi** ma'am for giving us the opportunity to work on our Mini project i.e., an AI based game development. Completing this project would not have been possible without their guidance and support. We extend our gratitude to our mentor **Mrs. Sangeeta Oswal** ma'am who guided us in all the meetings with her. She provided us some valuable information and resources to help increase the efficiency of our project.

We would also like to thank our peers who helped us whenever we faced any difficulty while developing the game and our parents who always supported us in their own ways. Finally, we thank all the mentioned and not mentioned who were a part of our project.

VIVEKANAND EDUCATION SOCIETY'S
Institute of Technology

ABSTRACT

Connect Four is a two players game which takes place on a 7x6 rectangular board placed vertically between them. One player has 21 yellow coins and the other 21 red coins. Each player can drop a coin from the top of the board in one of the seven columns; the coin falls down and fills the lower unoccupied square. Of course, a player cannot drop a coin in a certain column if it's already full (i.e., it already contains six coins).

The objective of the game is to connect four coins vertically, horizontally or diagonally. If the board is filled and no one has aligned four coins then the game is drawn (i.e., after 42 moves if no one wins).

As students of the Artificial Intelligence and Data Science branch,implementation of AI into the game is done for the mini project. The aim of the project will be primarily to build a bot that can compete with a real player using AI.

An array of AI techniques can be applied to make the connect 4 bot. Foundational AI concepts, such as Minmax algorithm with alpha-beta purring, Decision tree can be learnt through this project. These concepts underlie real-world application areas such as natural language processing, computer vision, and robotics.

By working on this mini project, various requirements and topics are understood, which are duly mentioned in the report below.

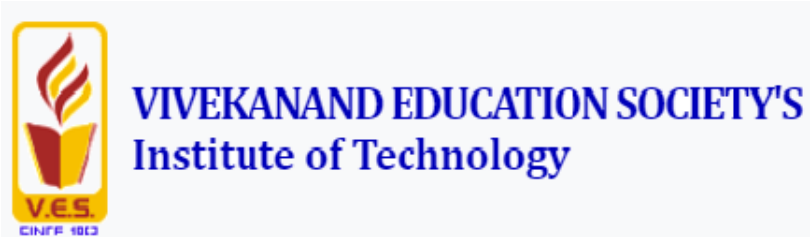


Table of Contents

Introduction

Connect-Four is a game for two persons. Both players have 21 identical coins. In the standard form of the game, one set of coins is yellow and the other set is red. The game is played on a vertical, rectangular board consisting of 7 vertical columns of 6 squares each. If a coin is put in one of the columns, it will fall down to the lowest unoccupied square in the column. As soon as a column contains 6 coins, no other token can be put in the column. Putting a coin in one of the columns is called: a move. The players make their moves in turns. There are no rules stating that the player with, for instance, the yellow coin should start first. It is assumed that the player with light colored coin starts the game. Both players will try to get four connected coins, either horizontally, vertically or diagonally. The first player who achieves one such group of four connected coin, wins the game. If all 42 coins are played and no player has achieved this goal, the game is drawn

1.1 Introduction

1.1.1 Other Elements of the Game:

- **Two player mode**.
- **Single player mode has adjustable difficulty levels**: Easy, Medium and Hard.
- **Game Review Option:**

  Once the game is over the player can look back at the sequence of moves made by them and can also see the best move calculated by AI against him and can compare the results.

1.1.2 Winning Strategies:

- **Place in the middle column!**
  If the player can play first, it is better to place it in the middle column. Since the board has seven columns, placing the discs in the middle allows connection to go up vertically, diagonally, and horizontally. In total, there are five possible ways. ·

- **Make a "7"!**
  A 7 trap is a name for a strategic move where one positions his coins in a configuration that resembles a 7. With three horizontal coins connected to two diagonal coins branching off from the rightmost horizontal coin. This coin formation is a good strategy because it gives players multiple directions to make a connect-four.

1.2 Problem Statement:

The game is categorized as a zero-sum game. Therefore, the minimax algorithm, which is a decision rule used in AI, can be applied. The project goal is to investigate how a decision tree is applied using the minimax algorithm in this game by Artificial Intelligence.

1.3 Objectives:

- We intend to make a game in which the computer is able to play intelligently the game of connect four. The game involves deep enough strategies that make the game interesting from the point of view of invoking artificial intelligence
- We intend to analyze the board before making the move. This would involve the successive analyzation of the strength of the computer as a consequence of effective moves.
- The graphics involved in the game would give us an insight into efficient handling of the graphics library in python (Pygame).
- If possible, we would like to make the game playable through networking as well.

1.4 Scope:

- Helps user to build problem solving skills and develop basic math's
- Provide Opportunity to detect patterns
- Boosts strategic thinking capabilities.
- Creates a quick-thinking mindset
- Enhances visual and perceptual-motor skills.

Literature/Techniques studied

2.1 Literature:

1. Zero-Sum Game:

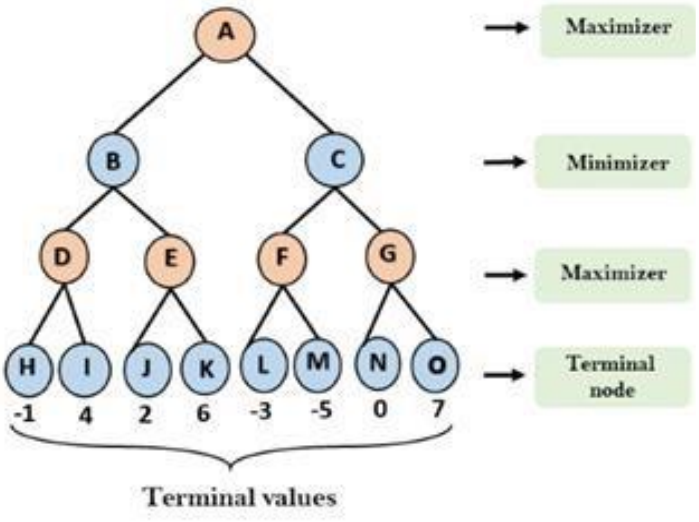
A zero-sum game describes a relationship, competition, or business deal where one person's gain is the other person's loss. The phrase "zero-sum game" comes from game theory and the notion that if one person wins and the other person loses, this produces a net gain of zero.

2. Minimax Algorithm:

Minimax is a kind of backtracking algorithm that is used in decision making and game theory to find the optimal move for a player, assuming that your opponent also plays optimally. It is widely used in two player turn-based games such as Tic-Tac-Toe, Backgammon, Mancala, Chess, etc.

In Minimax the two players are called maximizer and minimizer. The maximizer tries to get the highest score possible while the minimizer tries to do the opposite and get the lowest score possible.

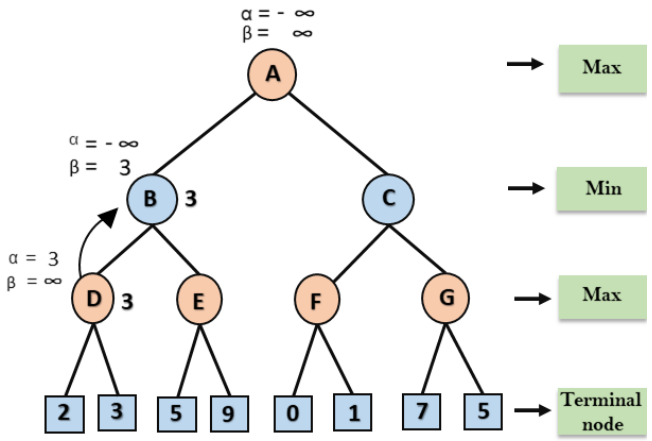
Every board state has a value associated with it. In a given state if the maximizer has upper hand, then, the score of the board will tend to be some positive value. If the minimizer has the upper hand in that board state, then it will tend to be some negative value. The values of the board are calculated by some heuristics which are unique for every type of game.



3. Alpha Beta Purring:

Alpha-beta pruning is a modified version of the minimax algorithm. It is an optimization technique for the minimax algorithm.

As we have seen in the minimax search algorithm that the number of game states it has to examine are exponential in depth of the tree. Since we cannot eliminate the exponent, but we can cut it to half.



Hence there is a technique by which without checking each node of the game tree we can compute the correct minimax decision, and this technique is called pruning. This involves two threshold parameter Alpha and beta for future expansion, so it is called alpha-beta pruning. It is also called as Alpha-Beta Algorithm.

Alpha-beta pruning can be applied at any depth of a tree, and sometimes it not only prunes the tree leaves but also entire sub-tree. The two-parameter can be defined as:

- **Alpha:** The best (highest-value) choice we have found so far at any point along the path of Maximizer. The initial value of alpha is -∞.
- **Beta:** The best (lowest-value) choice we have found so far at any point along the path of Minimizer. The initial value of beta is +∞.

The Alpha-beta pruning to a standard minimax algorithm returns the same move as the standard algorithm does, but it removes all the nodes which are not really affecting the final decision but making algorithm slow. Hence by pruning these nodes, it makes the algorithm fast

4. Decision Tree:

Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

The decisions or the test are performed on the basis of features of the given dataset.

It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.

5. Pygame Module:

- Pygame is a cross-platform set of Python modules which is used to create video games.
- It consists of computer graphics and sound libraries designed to be used with the Python programming language.
- Pygame was officially written by Pete Shinners to replace PySDL.
- Pygame is suitable to create client-side applications that can be potentially wrapped in a standalone executable.

2.2 Papers/Links/Findings:

After going through the below-mentioned websites, courses, source codes and some videos a few observations were made that would need to be kept in mind while implementing the game and building a bot.

- https://connect4.gamesolver.org/en/
- https://www.youtube.com/watch?v=STjW3eH0Cik
- https://www.youtube.com/watch?v=XpYz-q1lxu8
  https://www.youtube.com/watch?v=8392NJjj8s0
- https://roadtolarissa.com/connect-4-ai-how-itworks/#:~:text=The%20connect%204%20playing%20program,considered%20has%20actually%20taken%20place.
- https://www.python.org/

Analysis of the System

3.1 Initial Game Proposal:

- The program will consist of 2 mode Single player & multi-player mode.
- The user needs to select any one of these. Then the program will ask the player to select the difficulty level.
- Later an interface with a 7*6 grid will be displayed to the user.
- The user will be given the option to select a coin of certain color i.e., either red or yellow.
- He'll then be asked to drop the coin in any one of the columns.
- After the user plays his turn, the bot will make his move.
- The one who is able to make a pair of 4 coins will be considered as the winner of game
- Once the game is over the player can look back at the sequence of moves made by him and can also see the best move calculated by the bot against him and can compare the results.

3.2 Things to be considered while making the Bot

The bot needs to be better at positioning the coin. In order to compete with a real player, the bot must do at least the following:

1. It must select a winning move if it exists.

2. It must select a blocking move (i.e., prevent opponent from winning) if it exists.

But when there is neither a winning move nor a blocking move than in such situation the creativity of bot's intelligence comes in picture.

1. One possible strategy is to simulate moves and see which moves gives you the most two-in-a-rows, three-in-a rows, etc. i.e. simulate placing a coin in a column and see what the possible outcomes are. You will then need to score and rank each move.

2. The rigorous way to solve Connect 4 is to implement the Minimax algorithm with alpha beta pruning which would guarantee a win or a tie for your bot

3.3 Proposed Solutions:

Implementing Mini-Max Algorithm in Connect 4:

The connect 4 playing programs uses an algorithm. Every time the computer decides what move to make next, it considers all of its possible moves. The computer then pretends that each of the moves it has considered has actually taken place. For every possible move, it looks at all the moves the other player could make in response. For example, while considering what would happen if it played a red piece down the 7th column, the computer would examine the following yellow plays

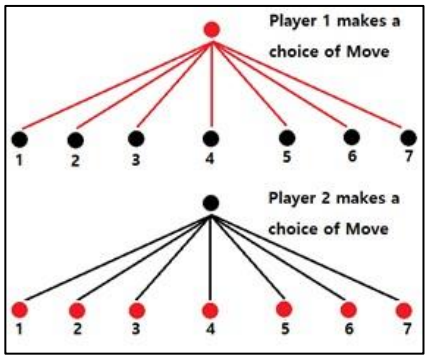
Each of these yellow moves is analyzed in the same way. 8 initial moves each have 8 responses for a total 64 distinct board positions after two moves. After three moves, there are 64*8 = 512 possibilities. The number of times this process repeats is determined by the search depth of the algorithm. When the program is called the first time, it is passed a depth parameter. Each time another layer of moves is considered, the depth is decremented by one.

Once the depth is equal to 0, all of the moves on the current layer are evaluated. Winning moves for red are given a score of 1000, while winning scores for yellow are given a score of -1000. Every time a layer of 8 possible moves is considered, if the red player was placing a piece, the move with the highest score is picked, while the yellow player does the opposite. After a move has been picked, it is sent back up the chain and used to evaluate the move before it along with the 7 other evaluations which have also been sent back up. The algorithm is called minmax because the computer is looking to minimize its maximum losses.

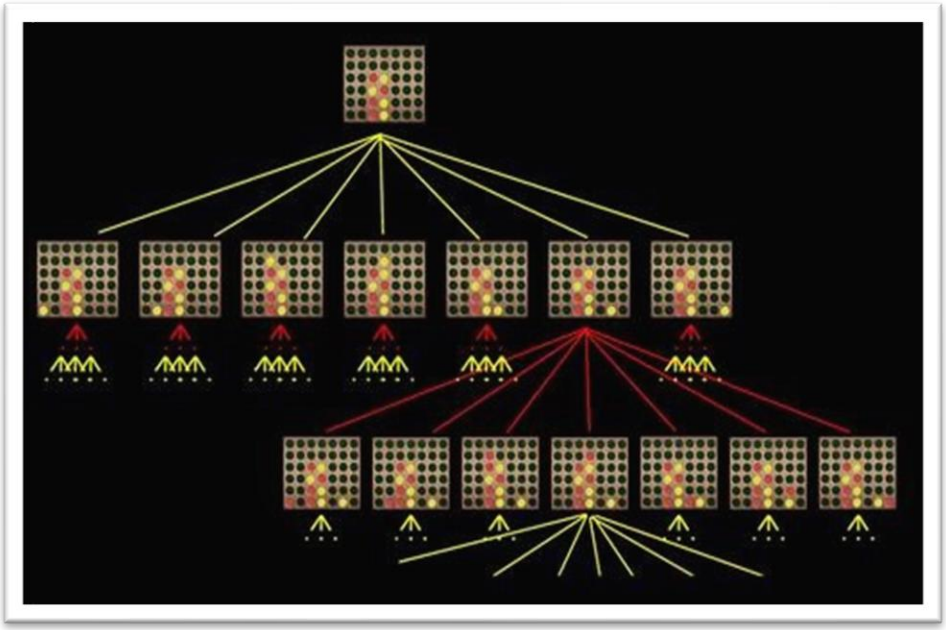
Decision tree in Connect Four:

When the game begins, the first player gets to choose one column among seven to place the colored coin. There are 7 columns in total, so there are 7 branches of a decision tree each time. After the first player makes a move, the second player could choose one column out of seven, continuing from the first player's choice of the decision tree.

Notice that the decision tree continues with some special cases. First, if both players choose the same column 6 times in total, that column is no longer available for either player. It means that their branches of choice are reduced by one. Second, when both players make all choices (42 in this case) and there are still no 4 discs in a row, the game ends as a draw, and the decision tree stops. Finally, if any player makes 4 in a row, the decision tree stops, and the game ends.



Possible moves for each iteration of the Connect Four game shown in the decision tree



Decision tree of Connect Four possible moves

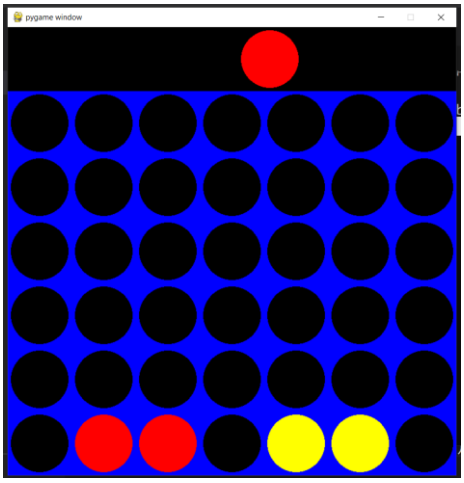
3.4 . Design of the proposed system:

- The system will be divided into three scripts:
- The first script will generate and handle the adjacency matrix
- The second script will receive the adjacency matrix and will implement Mini-Max algorithm logic along with alpha-beta pruning on the generated matrix.
- Third script will import the second script and give a GUI to our project.

Results and Discussion

Various ways and algorithms were discussed in which the AI part can be implemented in the game and which would be better and more efficient. Solution for complexity issues regarding Mini-max algorithm was discussed and found. We also discussed about the GUI of the game and developed it using Pygame module. Finally, we tried to create a basic interface for Connect 4 game.

Basic Interface Developed:



Conclusion

These observations and learnings mentioned in this paper will help us to make better decisions in making the Connect 4 game. And also help to implement an AI Bot that tries to win the game against real players.

Future Work

If given more time and resources, there are a few additional features that could be implemented. For example, building a mechanical device that will allow the AI to put its piece into the appropriate column on the physical board. In addition, certain improvements could be made to the actual AI itself, to make it smarter. For example, enabling an offensive vs defensive mode, where the AI will make offensive or defensive moves depending on whether it is winning or losing. In addition, implementing more user Interface features and quality of life improvements, such as building a GUI to prompt the user to input their move, or printing a message once the game has finished