Faculty of Computers and Artificial Intelligence
Information Systems Department
Artificial intelligence
Fall semester 2023-2024

Artificial intelligence

Team Members

	Name	ID	Total grade(35)
1			
2			
3			
4			
5			
6			

:Tic Tac Toe Game Documentation

:Introduction and Overview

:Project Idea and Overview

The Tic Tac Toe game is a classic two-player board game where players take turns marking a cell on a 3x3 grid. The player who succeeds in placing three of their symbols (X or O) in a horizontal, vertical, or diagonal row wins the game. This project implements a simple GUI-based Tic Tac Toe game in Python using the Tkinter library. The primary objective is to create an interactive and engaging user experience, allowing players to compete against each other .or challenge a computer opponent

:Similar Applications

Similar applications include various online platforms, mobile apps, and desktop applications that offer a Tic Tac Toe gaming experience. These applications typically follow the same rules .and gameplay, allowing users to play against each other or against computer opponents

:Literature Review

The project drew inspiration from classic game development concepts and Python GUI frameworks. While there may not be specific academic papers for this type of project, general game development principles and Tkinter documentation were referred to. The focus was on creating a user-friendly interface and implementing a basic AI for single-player .mode

:Proposed Solution & Dataset

:Main Functionalities/Features

:User Interaction

.Players can click on cells to make moves .1

:Description

Users interact with the game by clicking on the individual cells of the 3x3 grid, selecting their .desired move

:Implementation

Tkinter buttons represent each cell, and the make_move function is triggered upon a user .click

.The game switches turns between the human player and the computer opponent .2

:Description

The game seamlessly alternates between the human player and the computer opponent .after each move

:Implementation

The make_move function updates the game state and switches the current player to ensure .turns are correctly managed

.A "New Game" button resets the game board .3

:Description

.Users have the option to start a new game at any point by clicking the "New Game" button

:Implementation

.The reset_game function resets the game board, allowing players to initiate a fresh round

:AI Opponent

.The computer opponent uses the algorithm to make strategic moves .1

:Description

The AI opponent employs the algorithm, a recursive decision-making algorithm, to .determine optimal moves

:Implementation

The function evaluates possible moves, assigning scores to each, and selects the move with .the highest score for the computer and the lowest for the human player

.The game provides a challenging single-player experience .2

:Description

The implementation of the algorithm ensures that the computer opponent presents a .challenging and competitive experience for the player

:Implementation

The AI's strategic decision-making enhances the difficulty level, creating an engaging single-.player mode

:Dataset

.The game does not require a dataset as it is a rule-based game with no training data

:Applied Algorithms

:Minimax Algorithm

:Functionality

.Determines the best move for the computer player .1

Recursively evaluates possible moves and selects the one with the highest score for the .2
.computer and the lowest score for the human player

:Alpha-Beta Pruning Algorithm

:Functionality

Enhances the minimax algorithm by pruning branches that cannot possibly influence the .1
.final decision

Utilizes alpha and beta values to keep track of the best-known option for the maximizer .2
.and minimizer, respectively

Prunes branches when it is determined that they won't affect the final decision, reducing .3
.the number of nodes evaluated

:Heuristic Reduction

:Functionality

Introduces heuristic evaluation to estimate the desirability of a particular game state .1
.without exploring all possible moves

Reduces the depth of search by evaluating positions using a heuristic function, providing .2
.faster decisions

.Balances accuracy with computational efficiency to enhance the AI's performance .3

:Heuristic Symmetry Reduction

:Functionality

1. Utilizes symmetrical positions to reduce the search space, improving computational efficiency.

2. Takes advantage of the fact that certain game states are equivalent due to symmetries, avoiding redundant evaluations.

3. Enhances the AI's performance by considering symmetrical positions and reducing the number of explored nodes.

:Design Choices

:Minimax Algorithm

The minimax algorithm ensures the computer makes optimal moves to either win or draw, .providing a strong foundation for strategic decision-making

:Alpha-Beta Pruning Algorithm

Alpha-Beta pruning enhances the minimax algorithm by intelligently pruning branches that cannot influence the final decision. This optimization reduces the number of nodes .evaluated, making the algorithm more efficient without sacrificing accuracy

:Heuristic Reduction

Incorporating heuristic reduction introduces a heuristic evaluation to estimate the desirability of a game state. This heuristic allows the AI to make faster decisions by evaluating positions without exploring all possible moves, balancing computational .efficiency with accuracy

:Heuristic Symmetry Reduction

The addition of heuristic symmetry reduction takes advantage of symmetrical positions to further reduce the search space. By considering symmetrical positions and avoiding redundant evaluations, the algorithm becomes more efficient, leveraging the inherent .symmetries in certain game states

:GUI Implementation

:Tkinter Library

.Used for creating the graphical user interface

.Buttons represent cells on the game board

Labels display the current player and game messages

:Experiments & Results

:Testing

Manual testing was performed to ensure the game functions as expected. The game .underwent rigorous testing for player moves, computer moves, win scenarios, and ties

:Results

The Tic Tac Toe game provides an interactive and challenging experience. The combination of minimax, Alpha-Beta Pruning, Heuristic Reduction, and Heuristic Symmetry Reduction .algorithms ensures a competitive and intelligent computer opponent

:Analysis, Discussion, and Future Work

:Analysis of Results

The game successfully implements the core Tic Tac Toe functionalities. The minimax algorithm, complemented by Alpha-Beta Pruning, Heuristic Reduction, and Heuristic .Symmetry Reduction, provides a competent and efficient computer opponent

:Advantages / Disadvantages

:Advantages

.Simple and intuitive user interface .1

Competitive single-player mode with advanced AI strategies, including minimax with .2
Alpha-Beta Pruning, Heuristic Reduction, and Heuristic Symmetry Reduction.

:Disadvantages

While advanced algorithms are incorporated, the current version maintains simplicity and .1
.lacks certain advanced features

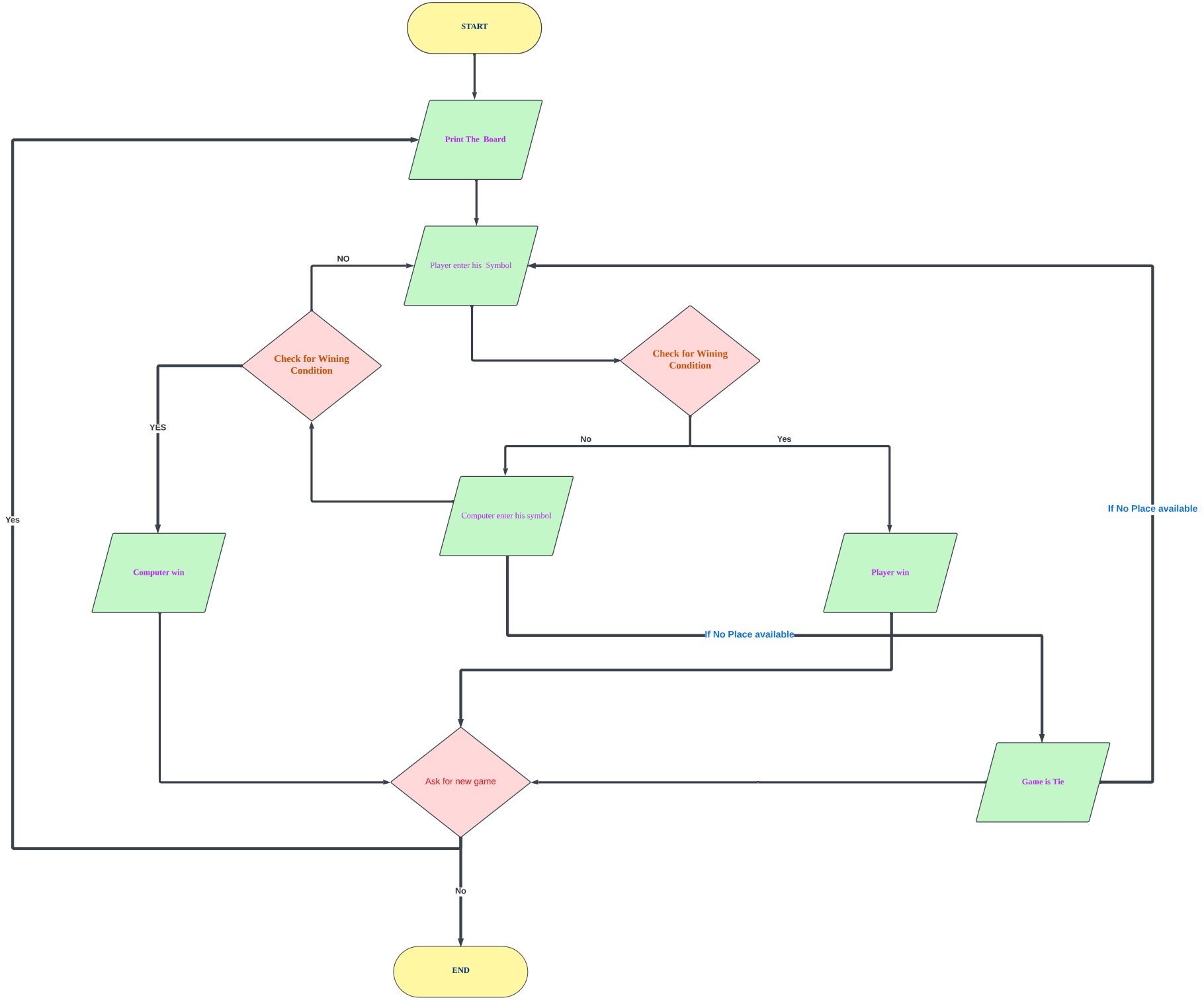
:Insights and Future Work

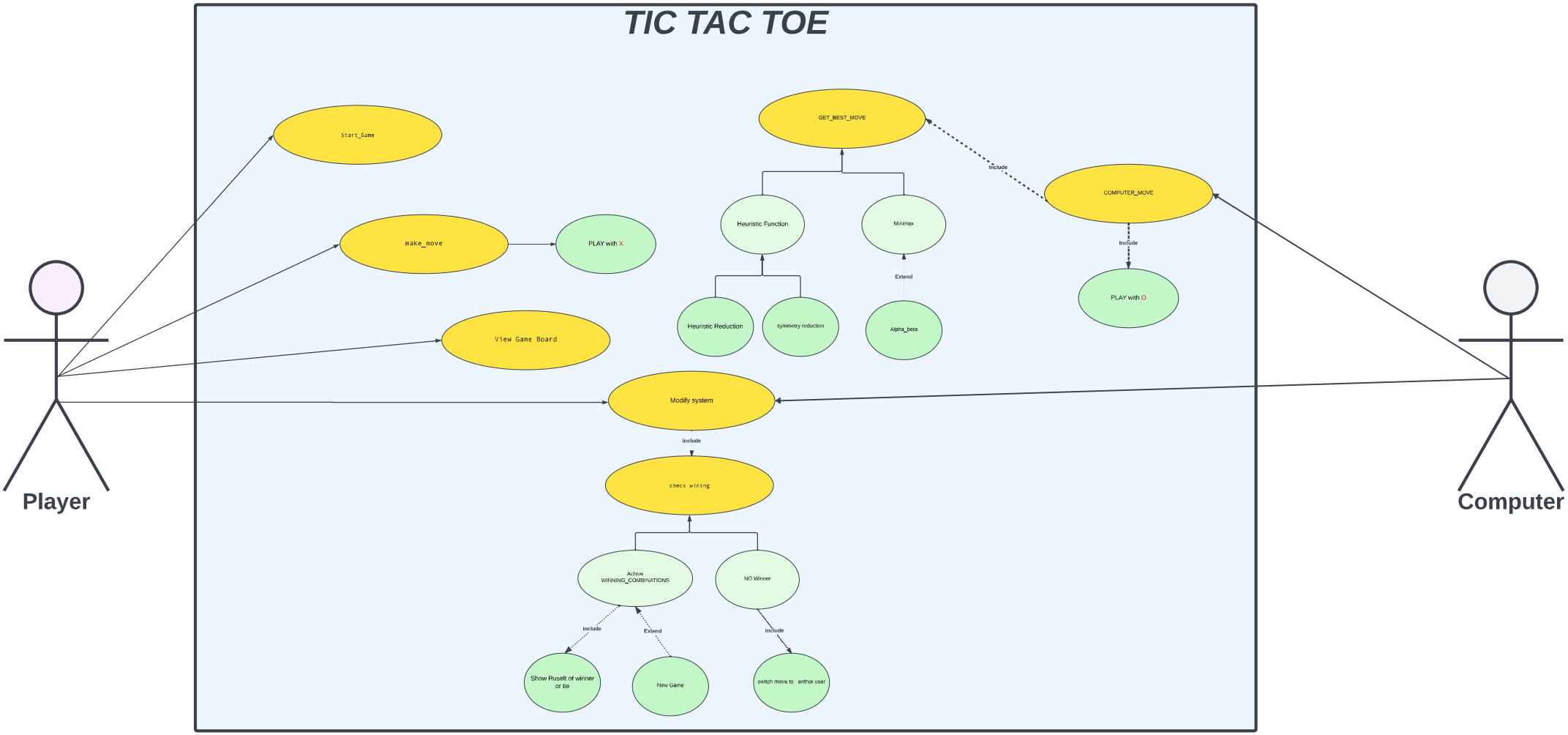
:Future modifications may include

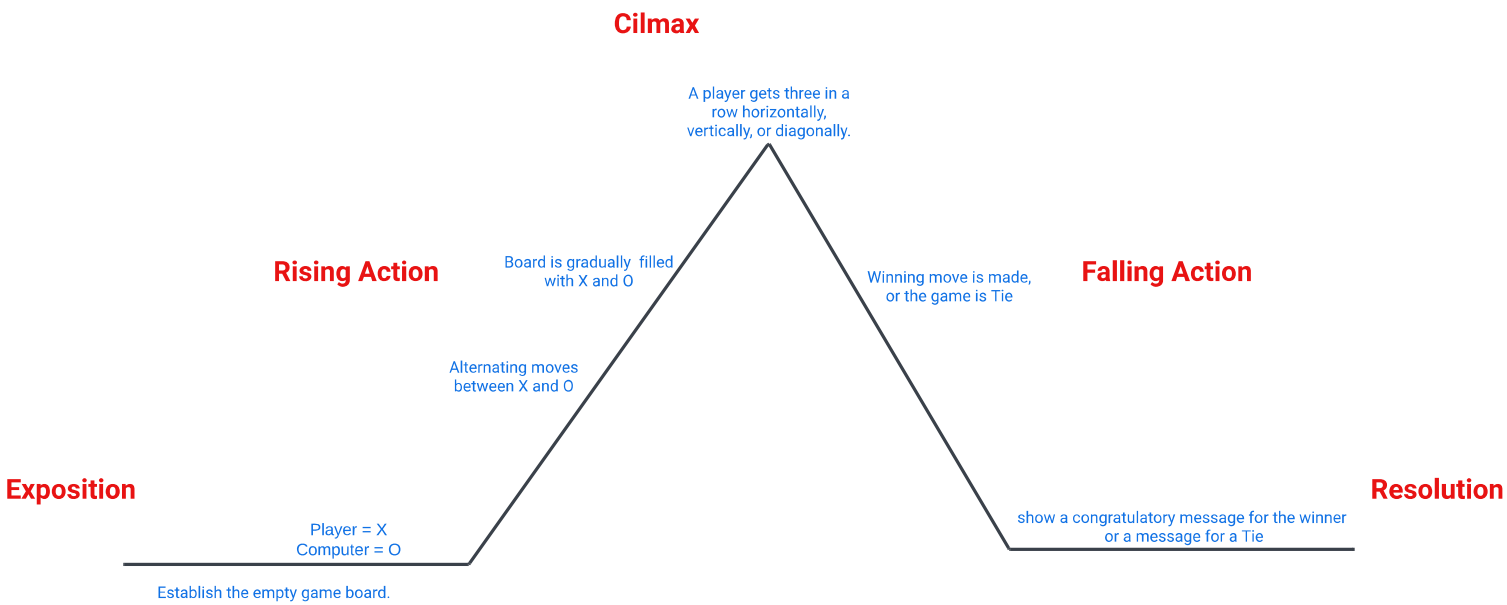
.Implementing additional AI strategies to diversify gameplay further .1

Enhancing the GUI with more advanced features and animations to improve user .2
.engagement

Exploring possibilities to expand the game to support network multiplayer, creating a .3
.more collaborative or competitive gaming experience


START
Print The Board
Player enter his Symbol
NO
Check for Wining Condition
Check for Wining Condition
YES
No
Yes
Computer enter his symbol
Yes
Computer win
Player win
If No Place available
If No Place available
Ask for new game
Game is Tie
No
END


TIC TAC TOE
Start_Game
make_move
PLAY with X
View Game Board
GET_BEST_MOVE
Heuristic Function
Minimax
Heuristic Reduction
symmetry reduction
Alpha_beta
Extend
Include
COMPUTER_MOVE
Include
PLAY with O
Modify system
include
check wining
Achive WINNING_COMBINATIONS
NO Winner
include
Extend
include
Show Ruselt of winner or tie
New Game
switch move to anthor user
Player
Computer


Cilmax
A player gets three in a row horizontally, vertically, or diagonally.
Rising Action
Board is gradually filled with X and O
Falling Action
Winning move is made, or the game is Tie
Alternating moves between X and O
Exposition
Resolution
Player = X
Computer = O
show a congratulatory message for the winner or a message for a Tie
Establish the empty game board.

FUNCTIONAL Block Diagram