



Cracow University of Technology
Department of Computer Sciences



Python Programming – Erasmus
Ac. Year 2023/2024

Tic-Tac-Toe Game

Authors:

Karyna Ouahrani

Yuliia Boiko

1. Abstract

This project involves developing a classic Tic-Tac-Toe game using Python and the Tkinter library. The game allows a player to compete against an AI opponent with three difficulty levels: Random, Basic, and Unbeatable. The project includes several AI strategies to improve the gaming experience, a user-friendly graphical user interface, and gameplay instructions. The minimax algorithm ensures the AI plays optimally, simulating all possible moves to minimize potential losses and guarantee a win or draw. The main aim is to provide an engaging and challenging Tic-Tac-Toe game with a robust AI system, demonstrating the practical application of game theory and algorithmic strategies.

2. Introduction

2.1 Aim

The purpose of this project is to create a Tic-Tac-Toe game where users can play against an AI opponent with different levels of difficulty. The project aims to demonstrate the use of Python programming language and Tkinter for building graphical user interfaces (GUIs) and implementing AI strategies in a simple game setting.

2.2 Scope

The scope of this project encompasses the following aspects:

- Designing a user interface for the Tic-Tac-Toe game.
- Implementing game logic for player and AI moves.
- Creating multiple difficulty levels for the AI opponent.
- Ensuring the game handles win, loss, and draw conditions appropriately.
- Providing a seamless and enjoyable user experience.

2.3 Methodology

The methodology for this project includes:

- Using Tkinter to create the game's GUI.
- Implementing game logic in Python to manage the Tic-Tac-Toe board and determine the outcome of the game.
- Developing AI strategies ranging from random moves to an unbeatable algorithm using the minimax method.
- Testing the game to ensure reliability and smooth user interaction.

Theoretical Part

3. Game Theory in Tic-Tac-Toe

Tic-tat-toe is a simple children's game in which two players take turns drawing tokens (X's or O's) on a 3 x 3 grid. The game is an indirect descendent of games played by children in ancient Egypt and is related to several three-in-a-row children's games worldwide (Zaslavsky, 1982)."

The reason tic tac toe is interesting is because it's a beautiful illustration of a domain where a person's objectives could potentially collide. The competing objectives of this game, as in many others involving two players, are to win and not lose. [1]

Minimax Algorithm

The minimax algorithm is a recursive approach used in decision-making and game theory to minimize the possible loss for a worst-case scenario. When dealing with an AI opponent in Tic-Tac-Toe, the algorithm evaluates all possible moves by simulating each outcome and assigns a score to each possible board state. The AI then chooses the move with the highest score, ensuring it plays optimally. This algorithm ensures that the AI either wins or forces a draw, making it unbeatable when playing perfectly [2].

Having defined the main game concepts and its data structure, we are now ready to understand the essence of the minimax algorithm. The minimax algorithm is designed to determine the optimal strategy for one player, informing the best move for that player. The entire algorithm consists of four steps:

1. **Generate the entire game tree:** This involves creating all possible game states down to the terminal states (end of the game).[3]

2. **Evaluate the utility function:** Assign a value to each terminal state based on the outcome (win, lose, or draw). [3]
3. **Recursively determine utilities:** Use the evaluated values to determine the utility of higher-level nodes, alternating between minimizing and maximizing the utility values until reaching the root. [3]
4. **Choose the best move:** Select the move that leads to the highest utility value. [3]

In a programming context, imagine the computer playing as CIRCLE and the human playing as CROSS. The ‘MINIMAX DECISION’ function initiates the decision process and selects the best move for the computer from all legal moves. Each legal move is evaluated by the ‘MINIMAX VALUE’ recursive function, which continues to alternate turns between players until reaching a predefined maximum depth or a terminal game state. The utility function is then evaluated, and the process continues until the best move is determined. [3]

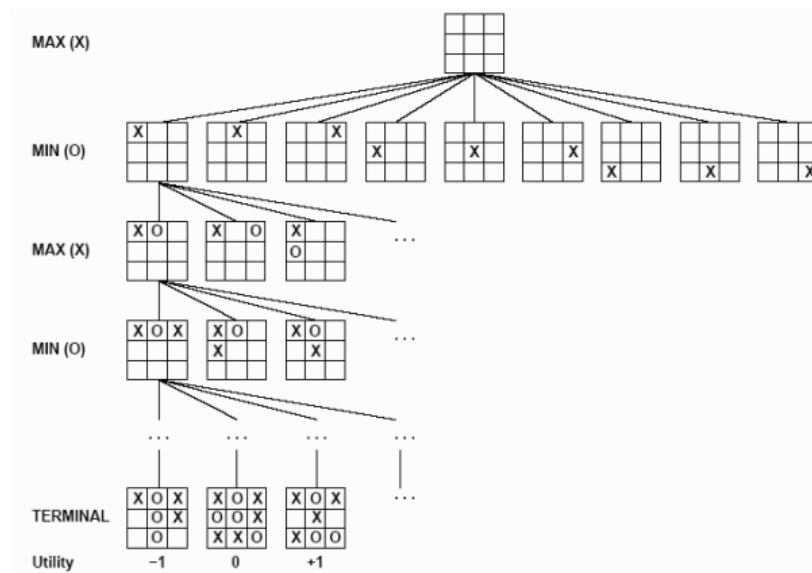


Fig1. The partial search tree for the Tic-Tac-Toe game.[3]

Practical Part

4. Solution and Implementation

4.1 GUI Implementation:

The game's user interface is built using Tkinter. Key components include:

- **Main Menu:** Displays game instructions and options to start or exit the game.
- **Game Board:** A 3x3 grid where players place their marks.
- **Status Label:** Shows whose turn it is and displays the result of the game.
- **Restart Button:** Allows players to restart the game after it ends.
-

4.2 Game Logic and AI

The game logic manages the state of the board, the player turns, and determines the winner or if the game ends in a draw. The AI component includes three difficulty levels:

- **Random:** The AI selects a random empty cell for its move.
- **Basic:** The AI attempts to win or block the player's winning move; otherwise, it chooses randomly.
- **Unbeatable:** Uses the minimax algorithm to make optimal moves, ensuring it either wins or draws the game.

4.3 Results

The game was successfully implemented with a functional GUI and working AI. The Random and Basic difficulty levels provide a varied challenge, while the Unbeatable level ensures a competitive game. The game reliably detects wins, losses, and draws, providing appropriate feedback to the player.

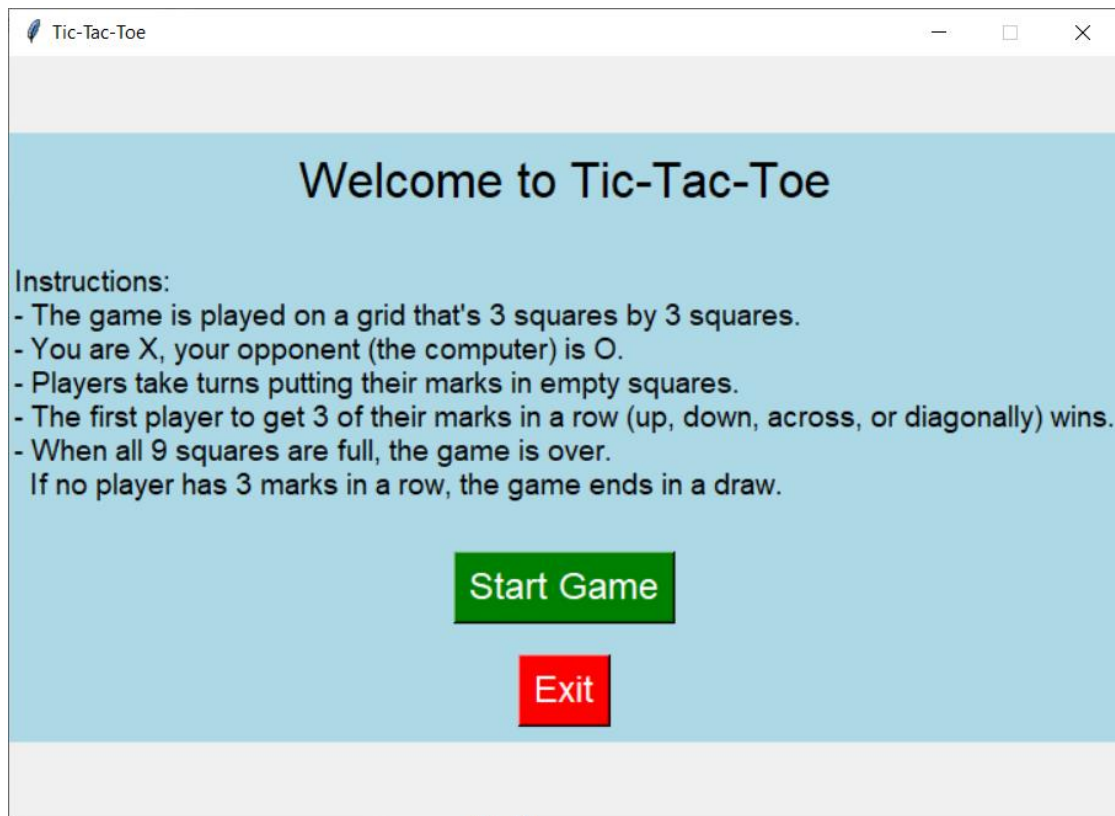


Fig 2. Main Menu of Tic-Tac-Toe

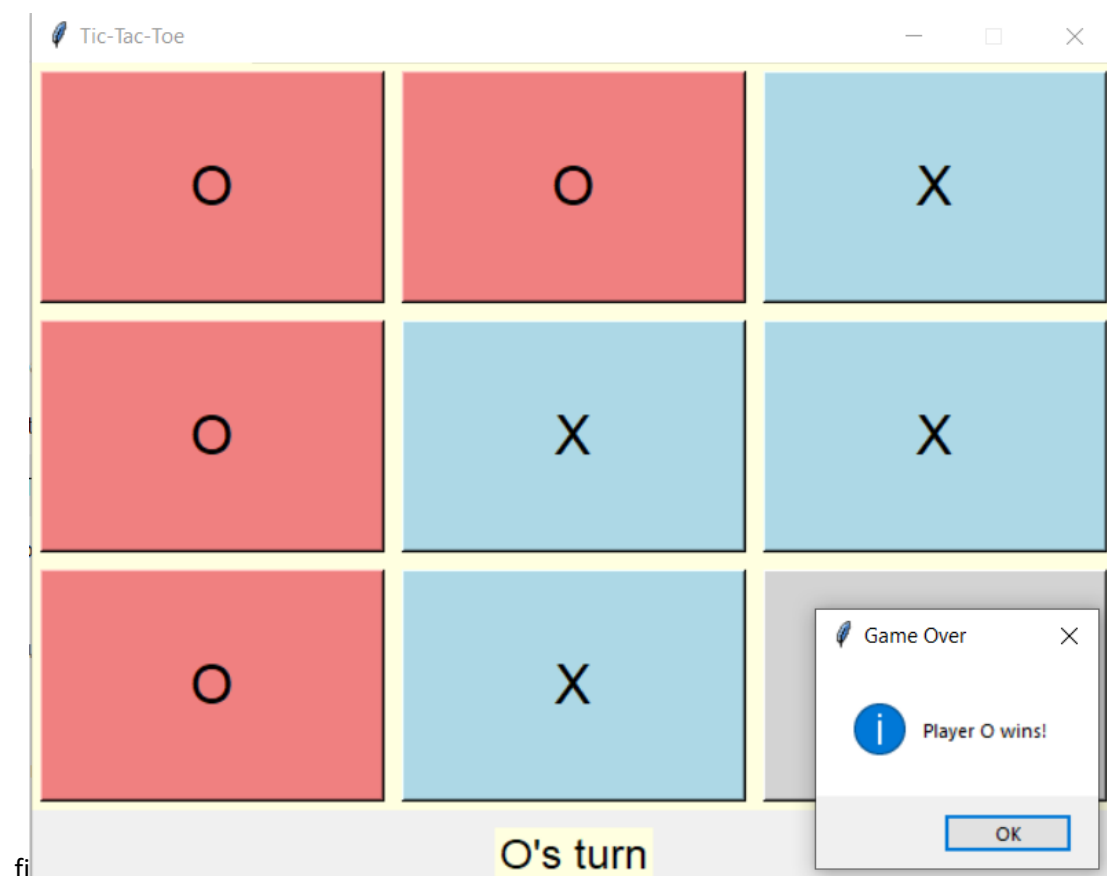


Fig 3. Game Board of Tic-Tac-Toe

Conclusion

The project's objectives have all been successfully executed. It was effective in creating a Tic-Tac-Toe game that is convenient to play, using Python for game logic and Tkinter for the graphical user interface (GUI). The resulting program offers a fun and engaging gameplay experience with various AI difficulty levels, including an unbeatable AI using the minimax algorithm. The project's requirements and expectations have all been met by the user-friendly Tic-Tac-Toe game that has been produced via the integration of these technologies.

This application serves as a valuable tool for both entertainment and educational purposes. For people in various fields, both professionals can benefit greatly from this application. It can be used by students and educators to understand game theory and algorithmic strategies like minimax in a practical setting. The tool can also serve as a leisure activity for anyone looking to enjoy a classic game with a challenging AI opponent. All things considered, the game provides a practical and enjoyable option for anyone interested in gaming, AI, or programming.

Bibliography

1. Pilgrim R. A. TIC-TAC-TOE: Introducing expert systems to middle school students //Acm Sigcse Bulletin. – 1995. – T. 27. – №. 1. – C. 340-344.
2. Minimax algorithm: <https://en.wikipedia.org/wiki/Minimax>
3. Thamrin H., Pamungkas R. W. A. Algoritma Minimax untuk Game Tic Tac Toe yang Menantang //Indonesian Journal of Computer Science. – 2023. – T. 12. – №. 3.