**Tic Tac Toe Game with AI Integrated and Front End Framework**

GUI based 5x5 tic tac toe game with 2 modes 1.player vs AI 2.player vs player. We have given certain conditions for the ai to perform its operations and the AI performs quite well and is difficult to beat. Player vs Player allows us to play it with 2 users.

**Base Paper:**

### <https://www.researchgate.net/publication/302973985_On_the_Development_of_Strategic_Games_based_on_a_Semiotic_Analysis_A_Case_Study_of_an_Optimized_Tic-Tac-Toe>

### Dataset Description:

No dataset used. It is all done on the basis of probability and permutation and combinations.

**Methodology:**

To build this game, I first created the board. This is a 5x5 game and hence a board with 25 boxes for entries.

Next, I created the winning conditions i.e. when the game should end, when either vertically or horizontally 5 elements of same character are repeated.

I wrote a function to get the input from user based on a mouse click, the entry is done only when that particular area/board location does not have any previous value on it.

For the AI to decide which moves to make next, I used basic if else conditions. The AI first covers the corners to make sure the probability of AI winning is high and the other moves are based on how the user gives his/her input. The program is designed in a very basic way but the AI is difficult to beat. It is easy to play a tie game though. We have used MinMax algorithm to build similar TicTacToe game on our website (ABHINAY LINK)

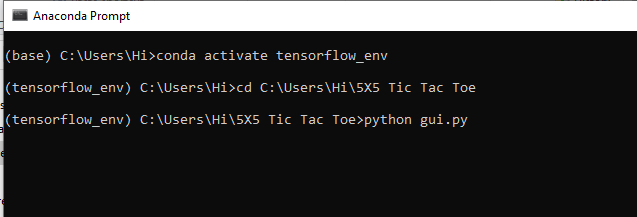
To make the game more interesting, I added a timeout condition wherein if the player does not make his/her move in 5 seconds (you can change this in code), the player loses the match.

**How to Execute?**

So, before execution we have some pre-requisites that we need to download or install i.e., anaconda environment, python and a code editor. **Anaconda**: Anaconda is like a package of libraries and offers a great deal of information which allows a data engineer to create multiple environments and install required libraries easy and neat.

To check on how to install anaconda environment, set up jupyter notebook refer to this article. You can skip the article if you have knowledge of installing anaconda, setting up environment and installing requirements.txt

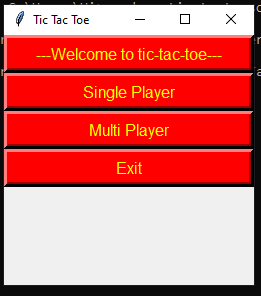
To run the code, open anaconda prompt. Go to virtual environment if created or operate from the base itself and go to your file location and type python gui.py to run the code.



1. **gui.py:** Game with timeout condition which makes it hard.
2. **gui2.py:** Game without timeout condition.

**Results:**

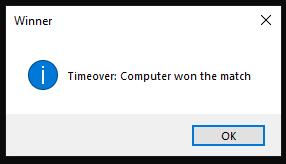
You can select any option based on your choice.

****

Demo game of user vs computer.



Result when I did not enter my move within the given time



**Issues faced/ Points to note:**

1. Ensure you have all libraries installed.
2. Give correct paths wherever necessary.

I built this code after referring to an article from geek for geeks. It was a 3x3 board, i just changed it to 5x5 board adding the win conditions and also gave time delay as an add on.

link: <https://www.geeksforgeeks.org/tic-tac-toe-game-with-gui-using-tkinter-in-python/>