A MINI PROJECT REPORT

On

TIC TAC TOE USING TKINTER.

Submitted in partial fulfillment of the requirement
of the University of Mumbai for the Course

In

Computer Engineering (IV SEM)

Submitted By
Rakshit Shah
Tejas Sheth
Vikas Kumar Sethiya

Subject Incharge
Merlin Priya Jacob

CERTIFICATE

This is to certify that the requirements for the project report entitled **'Tic Tac Toe using Tkinter'** have been successfully completed by the following students:

Name	Moodle Id
Rakshit Shah	19102008
Tejas Sheth	19102026
Vikas Kumar Sethiya	19102028

In partial fulfillment of the course Python Programming (CSL 405) in Sem: IV of Mumbai University in the Department of Computer Engineering during the academic year 2020-2021.

Sub-in-Charge

PROJECT APPROVAL

The project entitled '**Tic Tac Toe using Tkinter**' by **Rakshit Shah, Tejas Sheth, and Vikas Kumar Sethiya** is approved for the course of Python Programming (CSL 405) in Sem: IV of Mumbai University in the Department of Computer Engineering.

Subject-in-Charge
Merlin Priya Jacob

Date:

Place: Thane

Table of Contents

ABSTRACT

Tic-tac-toe is originally a pen-and-paper game, relying on the players' intellectual strength and cunning strategy. Generating a game of tic-tac-toe using Python language helps us play around with probabilities, which is one of the pillars for Artificial Intelligence and Machine Learning domains. We have used the Tkinter package in our project, which is the standard Python interface to the Tk GUI toolkit. The project is executed on Visual Studio, using the Tkinter module.

Our project can be advanced in the future by including some sort of gaming interface. It can also be played against the computer in the future.

PROBLEM DEFINITION

Q.To Create a Tic Tac Toe Game using the Tkinter module for GUI.

The game is played on a three by three board. Each player has a marker. One player has an 'X', the other an 'O'. Players get alternate turns to place their markers on the board. The first player to get three in a row either diagonally, horizontally, or vertically, wins the games. In the event all squares are taken on the board without a winner then it is a tie. After the game ends Score is provided in the scoreboard window.

INTRODUCTION

The "Tic Tac Toe game using Tkinter" is a module-based program, in which different types of modules are used in order to create a smooth functioning of the game. The Code consists of 6 modules named tic_tac_toe which is the main module where the code starts to execute itself and other modules named key_press, score_file, draw_grid, chance, win_con are used in order to keep track of the moves, count the score and declare a winner respectively.

The Tkinter is used to create the frame of the main page, scoreboard and playing frame and respective buttons have been added for smooth flow of the game.

Description of Modules Used

TKINTER :
1. The Tkinter package ("Tk interface") is the standard Python interface to the Tk GUI toolkit. Both Tk and Tkinter are available on most Unix platforms, and on Windows systems.
2. Most of the time, Tkinter is all you need, but many additional modules are available as well. The Tk interface is located in a binary module named _tkinter. This module contains the low-level interface to Tk, and should never be used directly by application programmers. It is usually a shared library (or DLL), but might in some cases be statically linked with the Python interpreter.
3. The message box module is used to display the message boxes in the python applications. Various functions are used to display the relevant messages depending upon the application requirements.

TIC TAC TOE :
1. The main module in which main_body() is defined consists of the Tkinter window, title, and dimensions.
2. It contains the mainframe, score frame, and player frame.
3. It also contains two labels for the "Player1" and "Player2" name entry which then is displayed on top of the next frame to indicate Player 1 or Player 2 chance.
4. We also have a Back button, ScoreBoard, and Submit button to move ahead, check the score, and go back respectively.
5. Another module draw_grid is called which creates a 3x3 grid for the User to play.

DRAW_GRID:
1. 9 variables are declared, one for every block, to set the Button Value according to User Click.
2. Every variable is assigned to a grid box, which will be a part of the frame_player.
3. Lambda function is used to take the Players' input and is then used as a parameter in the key_press module.
4. The key_press module is called with the frame_player, box number, place value, and buttons as the parameters.

KEY_PRESS
1. In this module, the message box is imported from the Tkinter package and few global variables are declared.
2. We also import chance,win_con,draw_grid from the TicTacToe module.
3. A dictionary keypress_count is used to keep a record of whether the block is previously clicked or not.

4. A function limit_keypress is created to avoid the user from manipulating the block value once clicked.
5. The win_con module is called and it checks the win condition after every move.
6. the chance module is used to keep a track of players chance

CHANCE
1. It provides the value of X and O to the grid.
2. for loop is used to increment the value of counter after every player move
3. For even number Player 1 will play and for odd number Player, 2 will play and return X and O respectively.
4. After every move, The Button of Player 1 and 2 either gets Active or Disabled depending on the chance of the player.

WIN_CON
1. In this, we have imported permutation from the itertools module and the score_file module.
2. We have defined a win_con program in which the Permutation function is used which will generate 8 winning possibility
3. We then use a for loop to check whether any of the winning possibilities is being satisfied or not.
4. Once it does, a message box saying "Player _ Wins" is displayed.

SCORE_FILE
1. In this module, we have imported fileinput and have defined two programs score_save() and get_scores().
2. In score_save() we save the score of the player from win_con and save it to a file by using the concept of file handling.
3. The score keeps updating every time the player wins.
4. The get_score() program converts the file values in a list and then the loop is used to create a label for each and every line of the file and display it in the frame_score window.
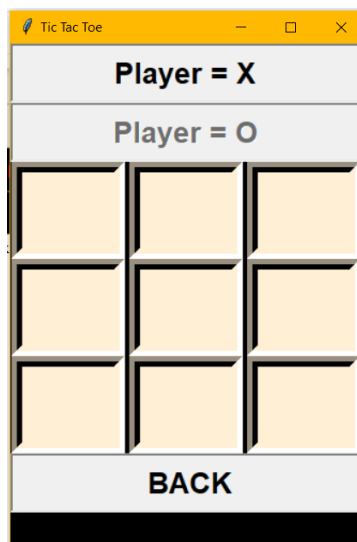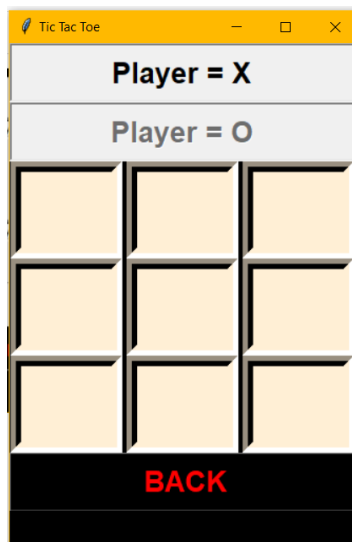
Implementation Details with ScreenShot.

1. We First Run the main file Tic_Tac_Toe and initially, the Tkinter mainframe is displayed

2. We then enter Player 1 and Player 2 Names respectively and Press the start button to move ahead to the Player_Frame.

3. We can even go back by pressing the "Back" button provided in the next frame.

4. The boxes are created using the draw_grid module and each button is assigned with a string variable to take the input.

5. The "Back" button is disabled after the Player plays the first move.

6. The game is executed by alternate chances of each player and the chance() module takes care of switching between both the players by displaying whose chance it is.

7. The limit_keypress function in the key_press module helps in avoiding manipulation of the move.
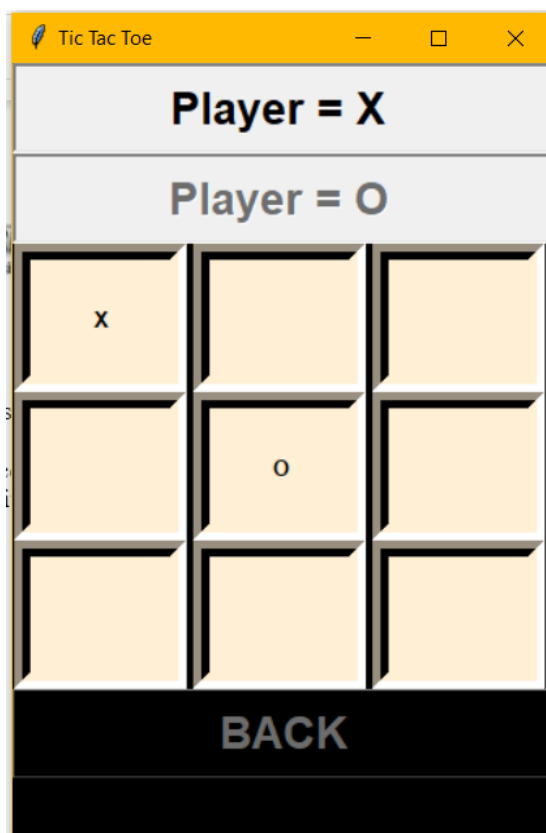
[Player 2 = "O" chance]   [Player 1 = "X" chance]

8. The game has 9 moves and there are three possibilities in-game :
    a. First Player Wins
   b. Second Player Wins
    c. Draw

a. First Player Wins: When the game encounters 3X in a row straight or diagonally Player 1 wins.
   i. This is possible due to the win_con module and the permutation tool imported from python itertools.
   ii. After Player 1 wins a dialogue box appears saying "Player1 WINS"
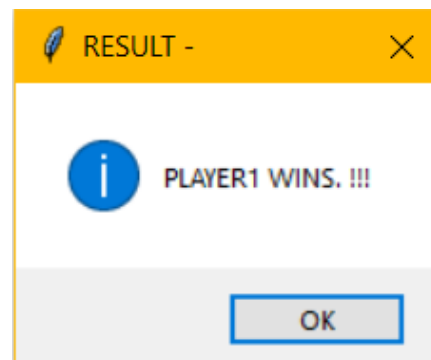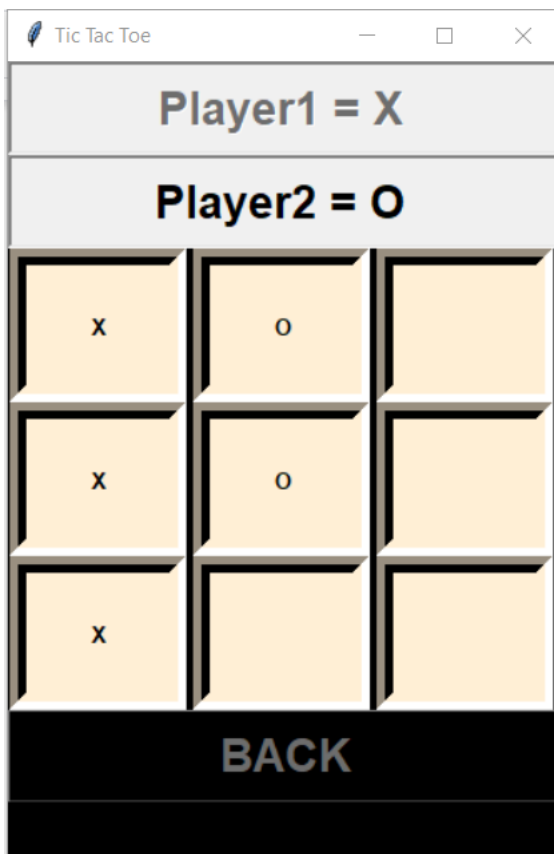
b. Second Player Wins: When the game encounters 3O in a row straight or diagonally Player 2 wins.

    i.   This is possible due to the win_con module and the permutation tool imported from python itertools.

    ii.   After Player 2 wins a dialogue box appears saying "Player2 WINS"
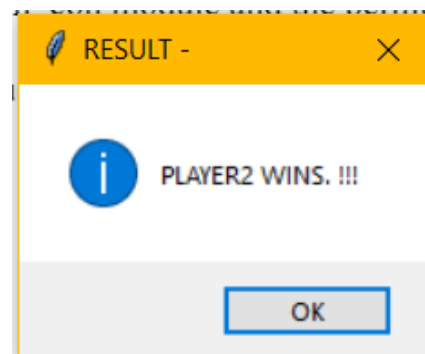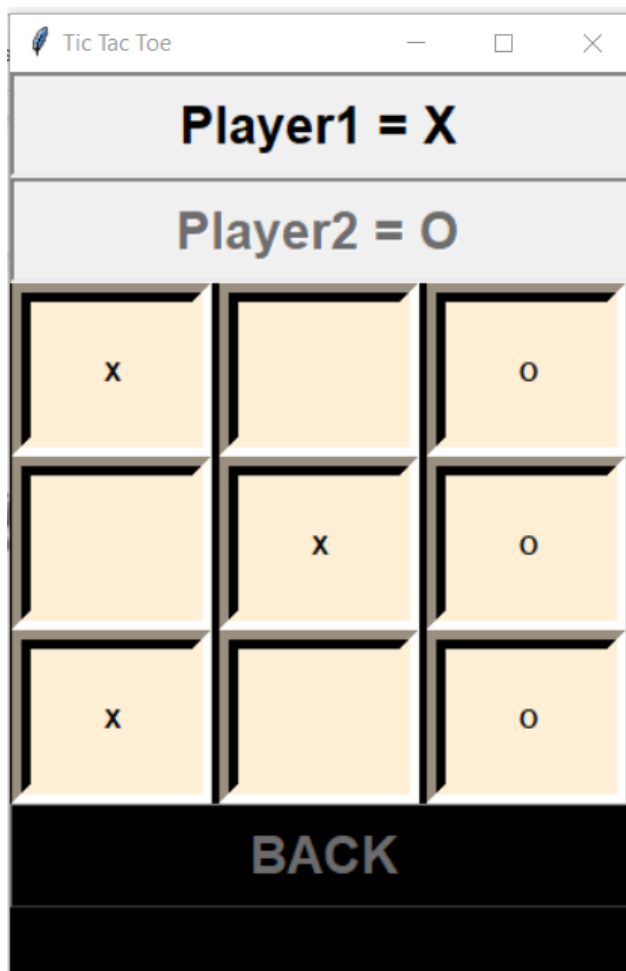
c. Draw:
   i. If all the 9 blocks get filled by X and O and still we have no winner then the game gets Draw.
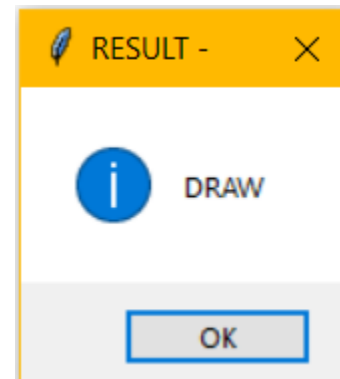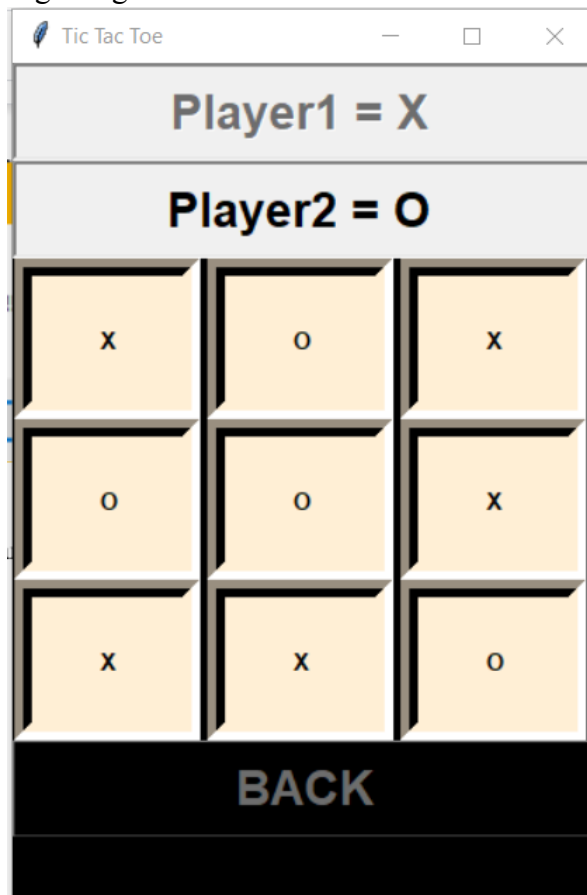
9. After the Result is shown the user is driven back to the mainframe and he gets three options:
   a. Start - To Play Again
   b. ScoreBoard - To check the ScoreBoard
   c. Exit - To Terminate the game.

10. When the ScoreBoard Button is clicked, the frame_score is raised and all the scores that have been stored in the file using the file handling concept are displayed in order.
   a. The score is calculated on the basis of moves left for a player and for every move left 5 points are awarded in the win_con module.

b. The score is then sent to the score_file module and there it is appended to the score.txt file.
c. If a player with the same name plays multiple times, Its score is updated after every game he/she wins.

10. We can go back to the mainframe by clicking on the "Back" button in the scoreboard fraq

[Main frame after pressing "Back" button]

11. Finally we can click on the "Exit" button to terminate the game.

Conclusion and Future Scope

Conclusion: Thus we conclude that we have studied and practiced Python GUI using Tkinter, and usage of modules for better understanding.

Future Scope :

With respect to this game our future scope would include :

1. User v/s Computer mode.
2. A timer for every move

REFERENCE

1. Permutation from itertools : https://www.youtube.com/watch?v=YX6-GnEUXLo
2. Python Tkinter : https://docs.python.org/3/library/tkinter.html
3. Python FileInput : https://docs.python.org/3/library/fileinput.html

ACKNOWLEDGEMENT

We have great pleasure in presenting the mini project report on "Tic-Tac-Toe using Tkinter". We take this opportunity to express our sincere thanks towards our guide Prof.Merlin Jacob, Department of Computer Engineering, APSIT, Thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude for the constant encouragement, support and guidance throughout the development of the project.