A PRELIMENERY REPORT ON

TIC - TAC - TOE USING TKINTER

SUBMITTED TO THE MRND LAB PRIVATE LIMITED, PUNE
IN THE FULFILLMENT OF THE REQUIREMENTS
FOR THE COMPLETION

OF

PYTHON INTERNSHIP

SUBMITTED BY

MAYUR KHARMATE
OMKAR KSHIRSAGAR

ACKNOWLEDGEMENT

With immense pleasure, we are presenting this Project Report as a part of the curriculum of Python Internship.

We are heartily thankful to **Prof. Geeta** whose encouragement, guidance and support from the initial to the final level enabled us to develop an understanding of the subject. We would also like to thank **Prof. Amol S Patil**, for giving us the opportunity to make a project on this interesting topic.

Lastly, we offer our regards to all those who supported us in any respect during the completion of the internship as well as our project.

NAME OF THE STUDENTS

Mayur Kharmate

Omkar Kshirsagar

ABSTRACT

The major objective of this internship was to learn the Python programming and its different libraries as it can provide a great enhancement to our project. This report is the detailed account of the process that we followed to build the project Tic Tac Toe using Tkinter. The project comprises two players, O and X, and has the options of win, loss, tie between the players, resetting the game and exiting from the game.

The process began with learning Python programming and its related libraries for creating GUI applications. We could understand and learn about Tkinter library which is used for creating GUI applications in python. Then we discussed about the features of Tkinter like canvas, buttons and labels that are used for developing a game like Tic Tac Toe. After that, we came up with a plan of how to implement it on python. We discussed the algorithms that need to be coded into the program for developing a logic for our game. We designed each feature of our game like player selection, win checker logic, etc.

We then divided among us tasks like coding individual functions and combining them together as a single program. We worked on basic logic first and then started adding more complex logics as per our plan and finally completed our project .

INDEX

LIST OF ABBREVATIONS

ABBREVIATION	**ILLUSTRATION**
GUI	Graphical user interface
RAM	Random Access Memory
GB	Gigabytes
GHz	Gigahertz
Tkinter	Tk interface
IDE	Integrated Development Environment
IDLE	Integrated Development and Learning Environment
SDLC	Software Development Life Cycle
AI	Artificial Intelligence

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

One of the most universally played childhood games is TIC TAC TOE. An interactive TIC TAC TOE game is developed where two players will be able to play against each other in a suitable GUI by using proper mouse movements. This game will start by showing a simple display, prompt the user for a move and then prints the new board.

The board looks like a large hash symbol (#) with nine slots that can each contain an X, an O, or a blank. There are two players, the X player and the O player. By default, player1 (the O player) takes the initiative. The game will end in two situations: a win, when one player gets three in a row, horizontally, vertically or diagonally. A draw, when there are no remaining places to choose and neither of them has won.

Here are some rules for the game:

- The player with symbol O goes first .
- Players alternate placing X s and O s on the board until either
- one player has three in a row, horizontally, vertically or diagonally; or all nine squares are filled.
- The player who can draw three X s or three O s in a row wins.
- If all nine squares are filled and none of the players have three in a row, the game is a draw.

2 SURVEY

2.1 EXISTING SYSTEM :

The Existing system is a simple game to play with paper and pencil between two people. Here the whole process will be carried out in the hand-written format making nine square grids, placing X’s and O’s and checking for the winner. This process will repeat every time.

So, it will be a tedious job to draw a nine square grid every time with paper and pencil. The human effort is more here. Along with that the retrieval of the information is not easy as the records are maintained in the hand-written papers. This application requires correct feed on input into the respective field. Suppose the wrong inputs are entered, then the whole process is to be done again. So, the users find it difficult to use.

2.2 PROPOSED SYSTEM:

To overcome the drawbacks of the existing system, the proposed system has been evolved. This project aims to reduce the paper work and saving time to generate accurate results from the player’s perspective. The system provides with the best Graphical User Interface. The efficient reports can be generated by using this proposed system.

2.2.1 ADVANTAGES OF PROPOSED SYSTEM :

- The game has been made user friendly with proper use of graphical interface.
- The user can play as many games without any interpretation.
- It is highly reliable, approximate result from user.
- The game has been made as a thorough expert system.
- The players can win the game, draw the game or will lose the game.
- It is a good brain exercise for all age group people.

2.3 PROBLEM DEFINATION

The project consists of developing and implementing a computer program that plays TIC TAC TOE against another player. This developed system will reduce the manual work that is for playing a game you’ve not to carry a pen or a pencil and a few sheets of papers on your pocket all the time. The system can able to provide hundreds or thousands of TIC TAC TOE game without any interruption.

3 SOFTWARE REQUIREMENT SPECIFICATION

3.1 SYSTEM SPECIFICATION :

3.1.1 HARDWARE REQUIREMENT [minimum requirement]:

- Minimum RAM: - 1GB
- Minimum Hard Disk: - 128GB
- Processor: - Intel Pentium 4(1.50 GHz) or above
- Network Interface: Not Required

3.1.2 SOFTWARE REQUIREMENT [minimum requirement]:

- Operating System: - Support for both LINUX and WINDOWS users
- Back End : - Python 3.6.0 Interpreter
- Front End Language : - Python3
- Front Design : - Tkinter
- IDE : - Python IDLE

3.1.3 Performance Requirements

- High Speed : Our system processes the requested task simultaneously with other actions to provide a prompt answer. The system then has to wait for the process to finish.
- Accuracy : our System conducts processes correctly and report results precisely. The system output follows the user-specified format.

4 Software Development Life Cycle

4.1 ANALYSIS MODELS: SDLC MODEL SHOULD BE APPLIED

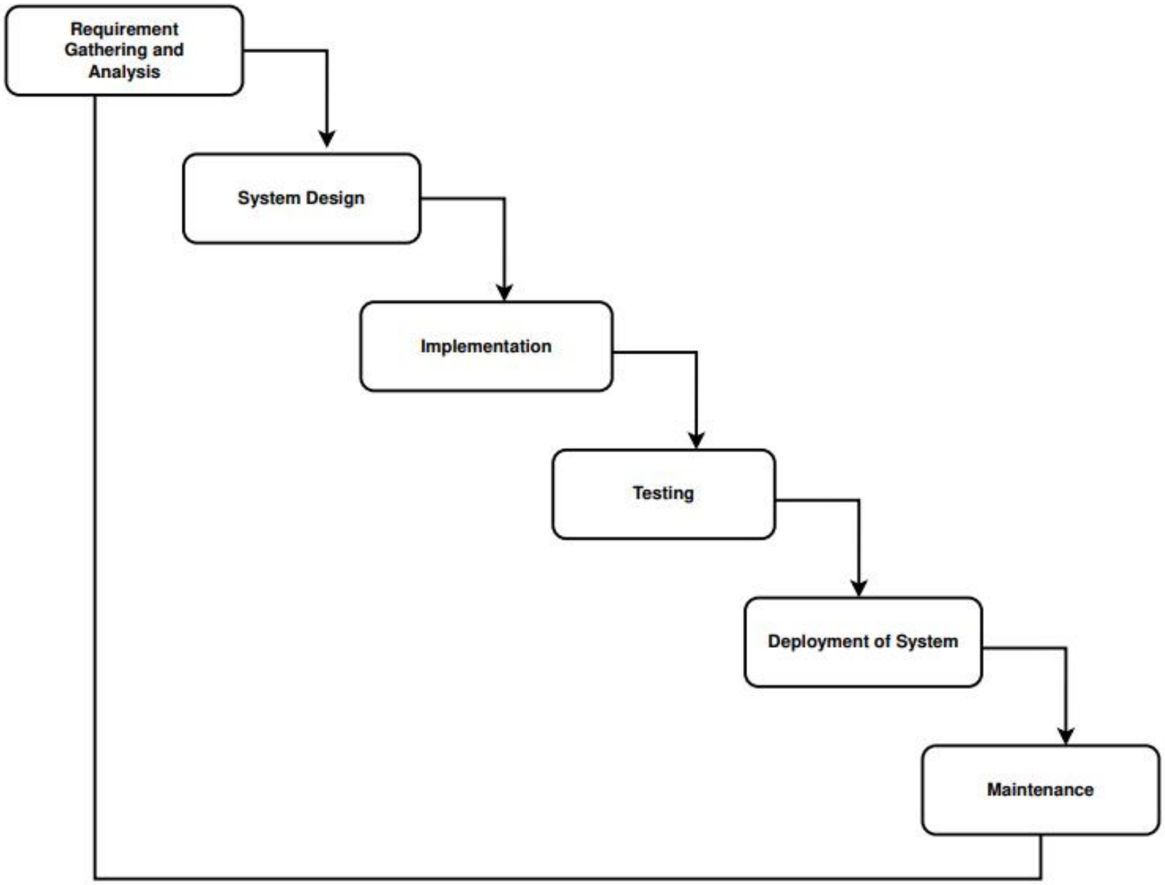


Fig 4.1 SDLC Model

4.2 SYSTEM IMPLEMENTATION PLAN

4.2**.1. Gathering and analyzing needs:**

Gathering and analyzing all the libraries and modules and functionalities required by the project. We identify the needs for our project in this waterfall step, including those for software and hardware, and interfaces.

4.2.2. System Design:

System design is crucial for project implementation since it indicates how the project will proceed. Which modules and function carry out which tasks and how they relate to one another. We develop a system that is user-friendly and easy to comprehend for end users throughout this stage of system design.

4.2.3. Implementation:

First, we divided the window in the 9 tiles where the game will be played. There are two Players in our Project: first ‘X’ and second ‘O’. The Game starts from the player ‘X’. The first player places its ‘X’ on any of the 9-position displayed. Which is then continued by the second player by placing it ‘O’ on any of the remaining positions. This way game continues until one of the players wins or it is a tie.

During the project's implementation phase, we successfully implemented the various modules needed to achieve the desired results at the various module levels.

The system is initially built as small programmes known as units with input from the system design and is then combined in the following phase.

4.2.4. Testing:

Various test cases are run to see if the project module is producing the desired results within the anticipated time frame.

Following the testing of each unit created during the implementation phase, the entire system is merged. The entire system is tested for errors and failures after integration. In order to comprehend the system flow, system modules, and order of execution, etc.

4.2.5. System Deployment:

Front end and back end of a website are deployed in the same file. The front end deploys the dedication file.

The product is either released to customers or deployed in their environments following functional and non-functional testing.

4.2.6. Maintenance:

If the client environment encounters any problems. Patches are published to address certain problems. Additionally, improved versions of the product are issued. To bring about these changes in the surroundings of the consumer, maintenance is performed. The progression is viewed as falling smoothly through the phases like a waterfall as they are all connected to one another.

Task	start date	Days to complete
Selection of Topic	**01-07-2022**	**3**
Submission of topics	**04-07-2022**	**1**
Data gathering on the topic	**05-07-2022**	**5**
Planning the prototype of project	**10-07-2022**	**4**
Researching about the technologies involved	**14-07-2022**	**10**
Designing of the Prototype	**24-07-2022**	**6**
Implementation of the prototype	**30-07-2022**	**11**
Debugging of the prototype	**10-08-2022**	**4**
Testing of the Prototype	**14-08-2022**	**2**
Fixing of all Defects in the prototype	**16-08-2022**	**7**
Final Implementation of model	**23-08-2022**	**10**
Deployment of model	**02-09-2022**	**6**

Table 4.1 Project Plan

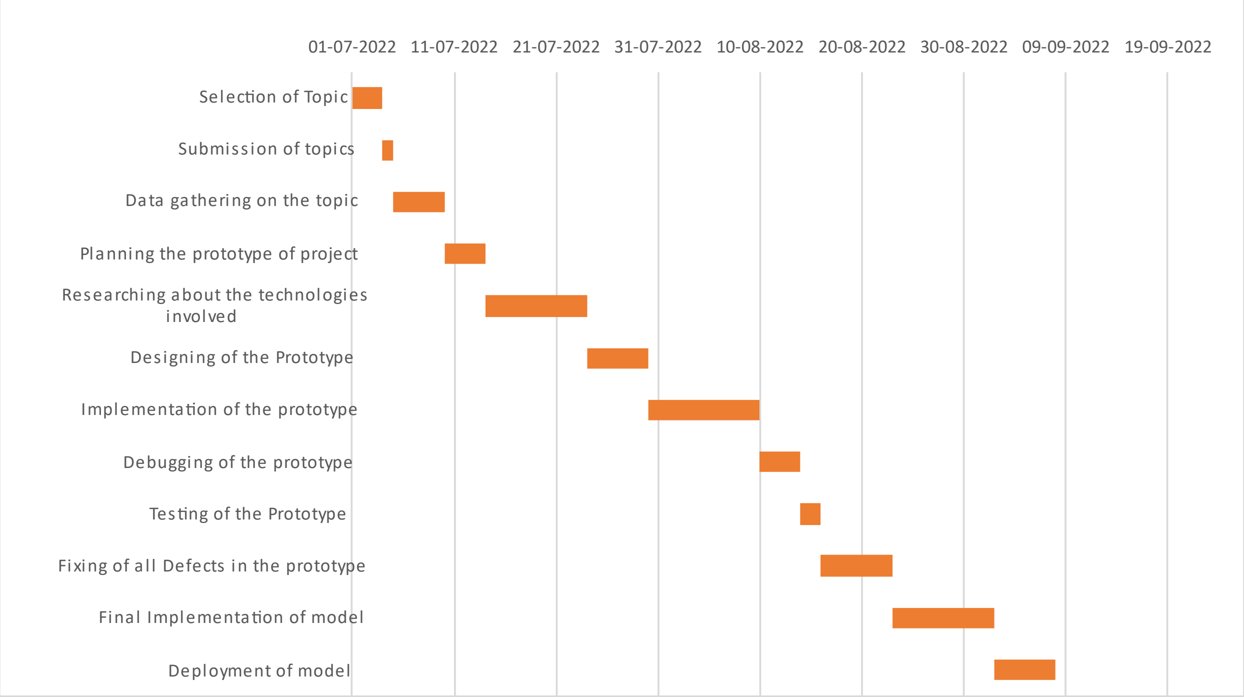


Fig 4.2 Gantt Chart

5 SYSTEM DESIGN

5.1 GUI (GRAPHICAL USER INTERFACE):

Our goal in this article is to provide you with an introductory of GUI programming. In order to learn the GUI programming, you must first understand a few core aspects of GUI. So, let’s begin the discussion. The primary GUI toolkit will be using Tk, Python’s default GUI. We’ll access Tk from its python interface called Tkinter. It is not the latest and greatest nor does it have the most robust set of GUI building blocks, but it is fairly simple to use and using it, you can build GUI’s that run on most platforms.

Setting up GUI application is similar to how an artist produces a painting. Conventionally there is a single canvas onto which the artist must put all the work. In GUI programming a top-level root windowing object contains all of the little windowing objects that will be a part of your GUI application. These can be text labels, buttons, list boxes etc. These individual little GUI components are known as widgets. Top level windows are those that show up stand alone as part of your application. Interestingly, you can have more than one top level window for your GUI, but only one of them should be your root window.

The top-level window can be created using this:

```
from tkinter import *
import tkinter.messagebox
from tkinter.font import Font

tk = Tk()
tk.resizable(False, False)
tk.title("Tic Tac Toe")
```

The object returned by tk = Tk() is usually referred to as the root window. Within this window you can place multiple component pieces together to form your GUI. Tk has three geometry managers that help with positioning your widget set.

- Placer: You provide the size of the widgets and locations to place them, this manager then places them for you.
- Packer: It packs widgets into the correct places.
- Grid: It is used to specify GUI widget placement based on grid coordinates.

Now once the packer has determined the sizes and alignments of your widgets, it will then place them on the screen for you. When all the widgets are in place we instruct the application to infinite main loop. In Tkinter the code that does it is:

tk.mainloop()

This is normally the last piece of sequential code the program runs.

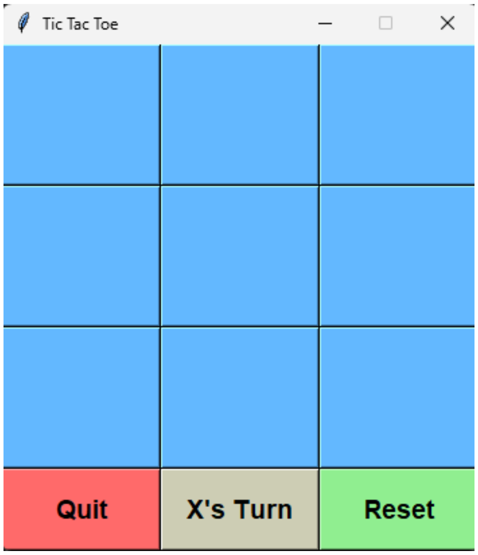


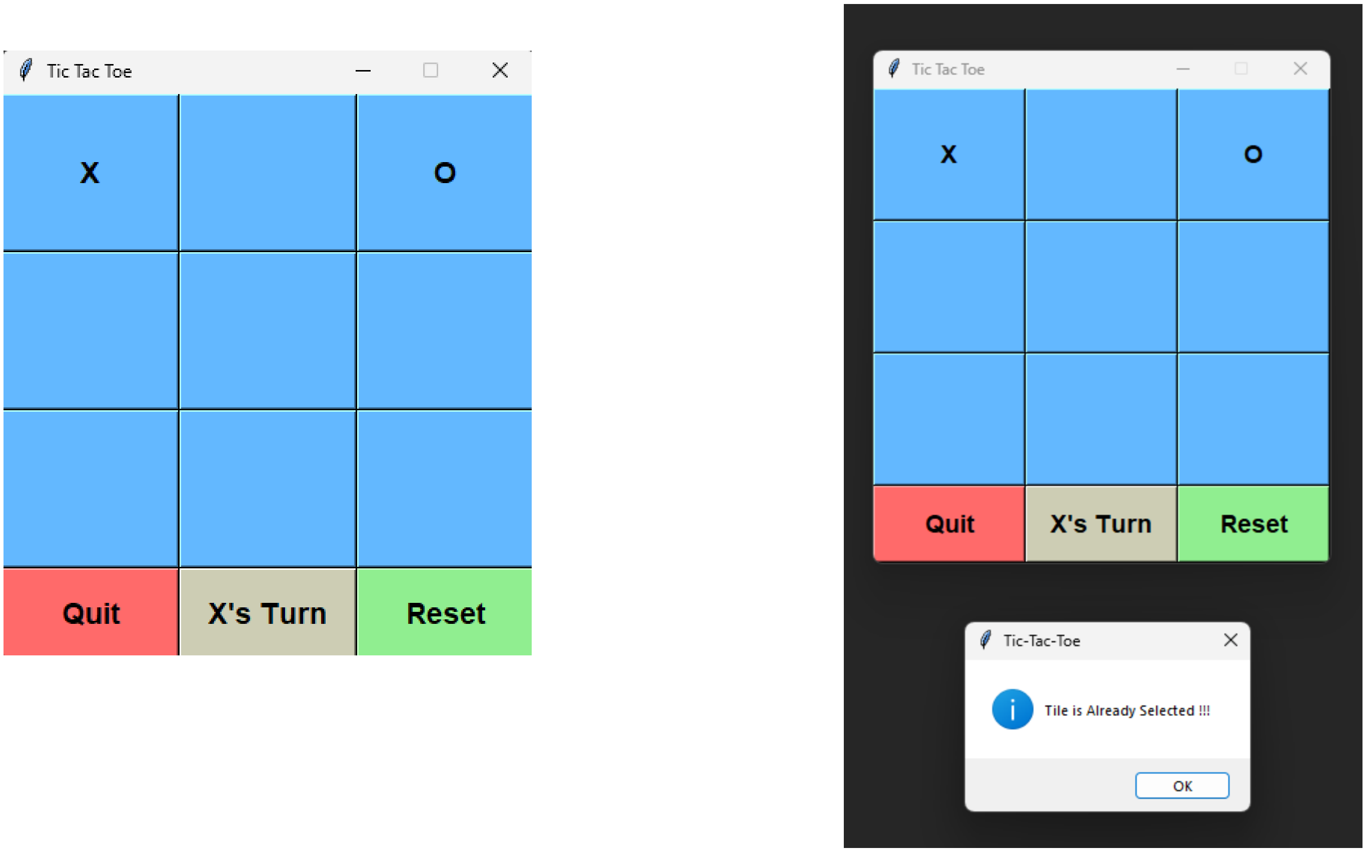
Fig 5.1 Tic Tac Toe – Empty Window

```
button = Button(tk, text=" ", font=my_font, bg='SteelBlue1', fg='black', height=4,
width=9,command=lambda: btnClick(button1)) button1.grid(row=1, column=1)
```

This is the statement which is used to divide the window into 12 grids of height 4 and width 9 and treating 9 of them as a button which the players can click in order to place their X or O. and remaining 3 which are at the bottom are for resetting game , Quit game and one is used to display the result .

- **def btnClick(buttons):**

  This function is used to mark button as X is the flag is true and O is the flag is false and if the tile is already selected then the prompt box will be displayed containing message "Tile is Already Selected !!!" .



Tile is Already Selected !!!

Fig 5.2 Tic Tac Toe –Tile Already selected Prompt

- **def checkForWin():**

  This function is used to check if any player is winning or not . Every time the player places his or her X or O this function is called to check if any of the win condition are met . If none of win condition are met the game continues until the all the tiles are filled which results in the Draw . The result of win or draw is displayed on result tile.

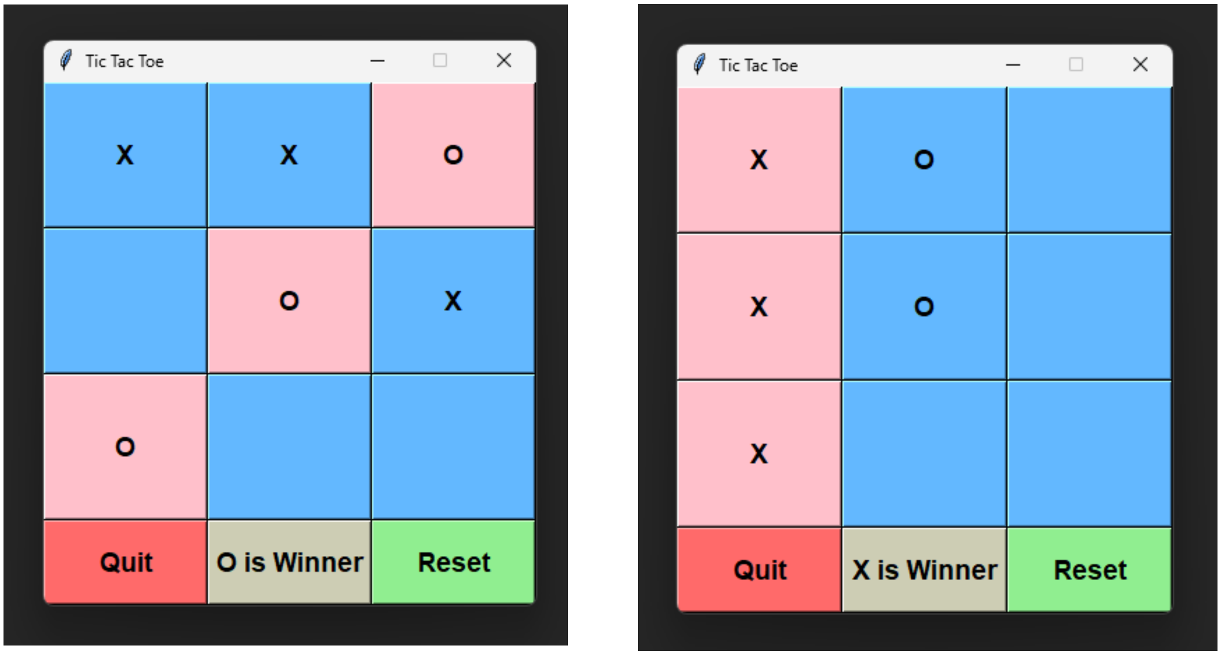


Fig 5.3 Tic Tac Toe – Winner Check

There are total of 8 winning conditions for each player which are shown as follows,

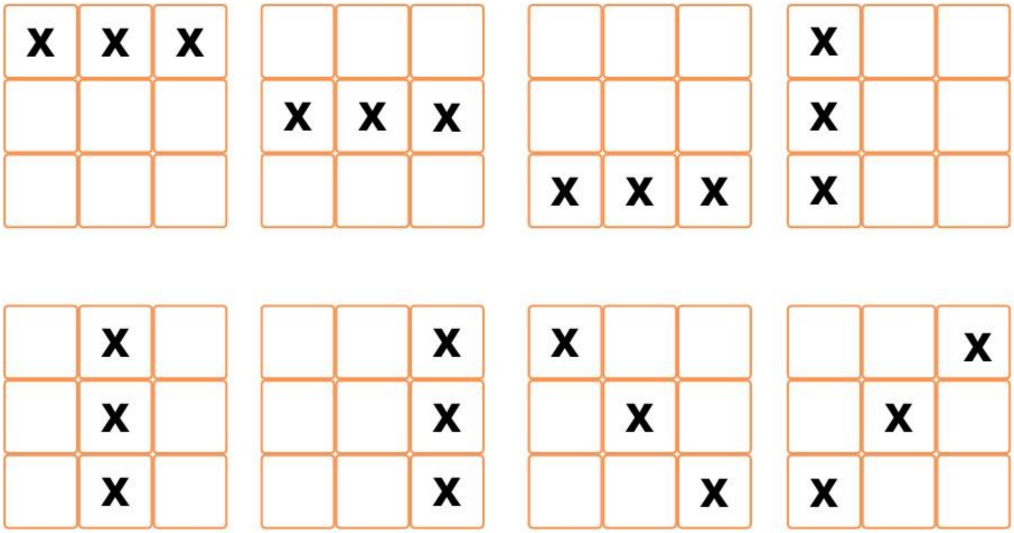


Fig 5.4 Tic Tac Toe – winning Arrangements

- **def color_result():**

  This function is used to color the tiles which has met the winning conditions. if the game results in the draw all the tiles will be colored in pink color. This helps in finding which winning condition has been met.

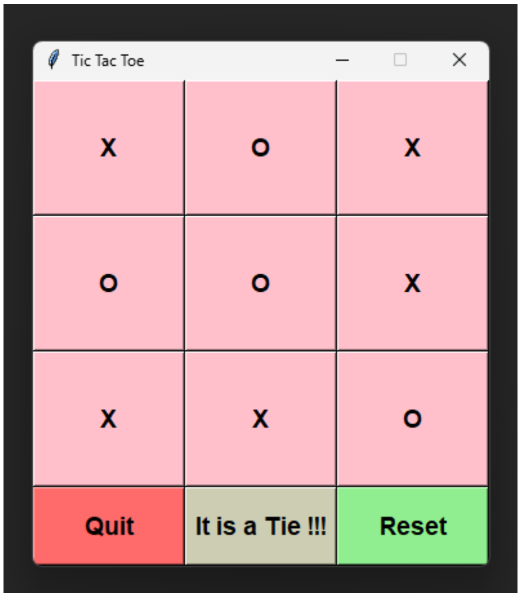


Fig 5.5 Tic Tac Toe – Game Draw (color all tiles)

- **def reset_game():**

  This function is used to reset the all the game tiles to initial condition. The turn is also reverted back to the player with ‘X’. Also, the color of tile which was changed due to win or draw is also reverted back to its original color.

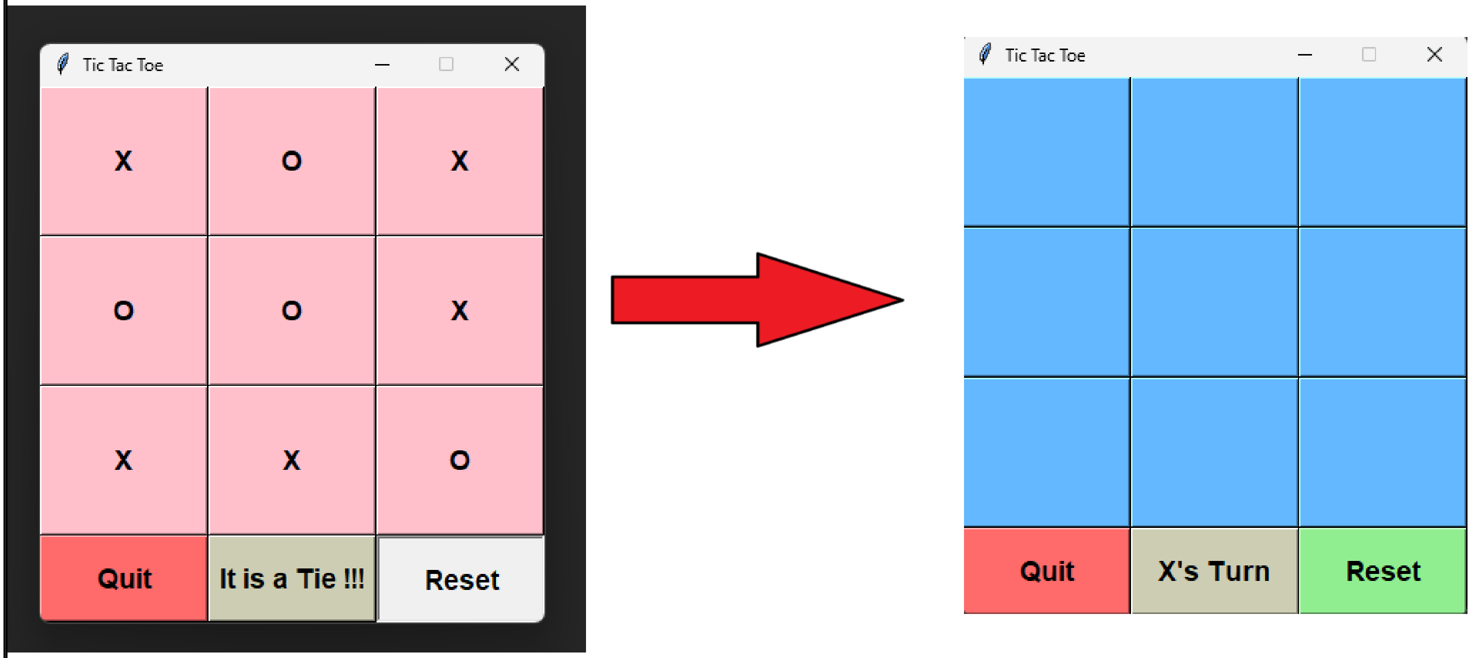


Fig 5.6 Tic Tac Toe – Reset game

6 OTHER SPECIFICATION

6.1 FUTURE SCOPE

1. **ALLOW THE PLAYERS TO CHOOSE THE MODE OF THE GAME:**

   An option to select the mode of the game that is whether the player wants to play the game with his/her friends or with the machine.

2. **ALLOW THE PLAYERS TO DEFINE THE METRICS:**

   An option to increase the board to size to n x n, where n can be any positive number (>0). This will increase the complexity of the code and when added with artificial intelligence will greatly make the game robust. Also, we can add a feature where k (< n) consecutive marks/dots (filled by the user) are a point/score. Or we can go even further and use a range of number n.

3. **INTRODUCTION OF ARTIFICIAL INTELLIGENCE:**

   An option of playing against the computer (machine) can be added too. An algorithm can be implemented for the machine based on actual rules a player uses to fill the Tic-Tac-Toe board. This can pose a challenge to the user when playing.

4. **INTRODUCTION OF DIFFICULTY LEVELS:**

   One option to select the difficulty levels of the game can be added. We would also like to create a computer AI that offered increasing resistance as the difficulty level increased.

5. **USER PROFILES AND SCORES:**

   One of the other enhancements could be that users can create their profiles and save their scores.

6. IN-GAME ASSISTANT:

An option to include an Assistant which will provide hints and suggestions to the user when playing against a bot.

6.2 FEASIBILITY

6.2.1 TECHNICAL FEASIBILITY:

The technical requirement for the system is economic and it doesn't use any other hardware or software. Technical evaluation must also assess whether the Existing System can be upgraded to use the new technology and whether the organization has the expertise to use it. Install all the upgrades frameworks into the Python IDLE package supported Linux based application. This application mostly depends two packages namely Tkinter package and pygame package provided by Python 3.6.0.

6.2.2 OPERATIONAL FEASIBILITY:

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no prerequisites for operating the product. Technical performance includes issues such as determining whether the system can sense the proper click of the mouse or after sensing the click it places the symbol in the desired cell and whether the system is organized in such a way that it always displays the correct result according to the moves made by the players.

7 CONCLUSION

In conclusion, I can say that this internship was a great experience. Thanks to this project, I acquired deeper knowledge concerning my technical skills, but I also personally benefited. Currently Python is a common part of most of the software applications, and it is one of the most popular languages for designing and developing of application used by professionals worldwide.

Keeping this into mind and using all the knowledge and expertise that we gained throughout the span of our training in python we successfully developed out project Tic Tac Toe game in python using Tkinter Library.

8 REFERENCES

[1] https://www.python.org

[2] https://www.google.com/

[3] https://docs.python.org/3/library/tkinter.html

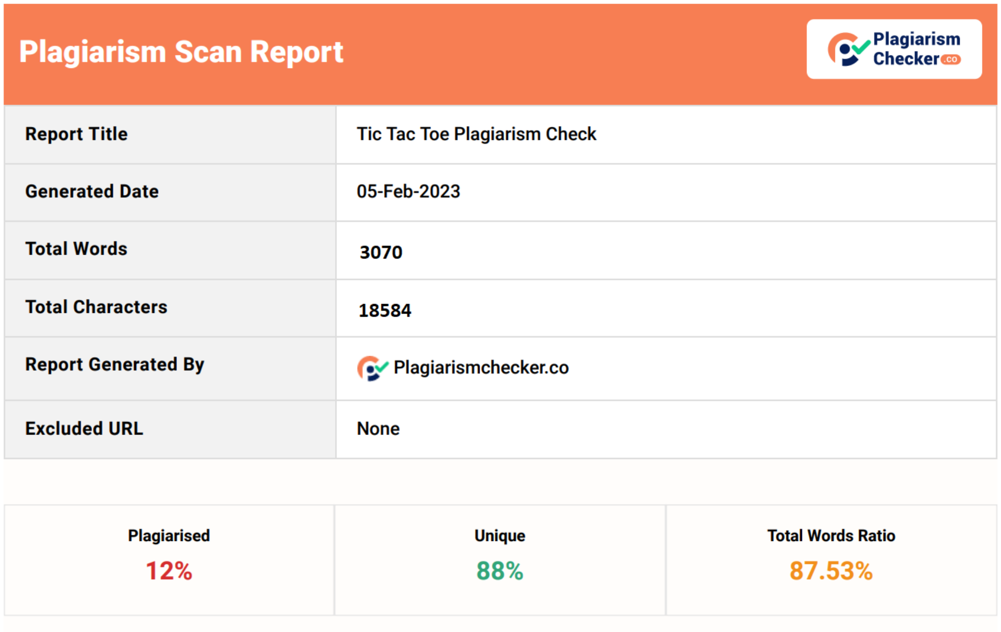
[4] https://www.geeksforgeeks.org/

[5] https://pypi.org/project/pip/

[6] https://www.javatpoint.com/

[7] https://www.w3schools.com/python/

8 PLAGIARISM CHECK REPORT



Plagiarism Scan Report

Report Title	Tic Tac Toe Plagiarism Check
Generated Date	05-Feb-2023
Total Words	**3070**
Total Characters	**18584**
Report Generated By	Plagiarismchecker.co
Excluded URL	None

Plagiarised	Unique	Total Words Ratio
12%	88%	87.53%