**PROJECT REPORT**

**on**

**Tic Tac Toe Computer vs Human**

**(CSE IV Semester Mini project )**

**2020-2021**

****

**Submitted to: Submitted by:**

Ms. Manika Manwal Mr. Priyanshu Padeliya

(CC-CSE-I-IV-Sem) Roll. No.: 1918568

**Guided by:** CSE-I-IV-Sem

Mr. Sandeep Kumain Session: 2020-2021

(Resource Person)

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY**

**GRAPHIC ERA HILL UNVERSITY, DEHRADUN**

**CERTIFICATE**

### Certified that Mr. Priyanshu Padeliya (Roll No.- 1918568) has developed mini project on “Tic Tac Toe Computer vs Human” for the CS IV Semester Mini Project Lab in Graphic Era Hill University, Dehradun. The project carried out by Students is their own work as best of my knowledge.

Date:14-05-2021

(Ms. Manika Manwal)

**Class Co-ordinator**

**CSE-I-IV-Sem**

(CSE Department)

GEHU Dehradun

(Mr. Sandeep Kumain)

**Project Guide**

Resource Person

(CSE Department)

GEHU Deh

**ACKNOWLEDGMENT**

We wish to thank our parents for their continuing support and encouragement. We also wish to thank them for providing us with the opportunity to reach this far in our studies.

We would like to thank particularly our project Co-ordinator Mr. B. P. Dubey and our Project Guide Mr. Sandeep Kumain for his patience, support and encouragement throughout the completion of this project and having faith in us.

At last but not the least We greatly indebted to all other persons who directly or indirectly helped us during this work.

**Mr. Priyanshu Padeliya**

**Roll No.- 1918568**

**CSE-I-IV-Sem**

**Session: 2020-2021**

**GEHU, Dehradun**

**TABLE OF CONTENTS**

**CHAPTER NO. / TITLE PAGE NO.**

**LIST OF SYMBOLS, ABBREVIATIONS viii**

**1. INTRODUCTION**

1.1 About Project 1

1.2 Python 2

1.3 GUI 3

1.4 Tkinter 3

**2. PROJECT**

2.1 Requirement Analysis 4

2.2 Software Specification 4

2.3 Module 5

2.4 Application 7

2.5 E-R Diagram 7

**3. SNAPSHOT OF PROJECT**

3.1 Game 8

3.2 1st players turn 8

3.3 Computer’s turm 9

3.4 Winning situation 9

3.5Draw situation 10

3.6 Reset 11

**4. CONCLUSION**  12

**REFERENCE** 13

**Appendix :code 14**

**LIST OF SYMBOLS, ABBREVIATIONS**

= EQUALS TO

!= NOT EQUALS TO

== EQUALITY

+ ADDITION

\n NEW LINE

-------------------------As per your Project------------------------------

**LIST OF ABBREVIATIONS or ACRONYMS**

GUI GRAPHICAL USER INTERFACE

-------------------------As per your Project------------------------------

**CHAPTER** **1**

1. **INTRODUCTION**
   1. **ABOUT PROJECT-:**

This project is based on “TIC-TAC-TOE”, which can play whenever you feel bored and can be played anywhere. You just need 2 peoples to play and enjoy. This game has very simple rules. It is also known as X’s and O’s. The player who wants to win, one has to get three same either X’s or O’s in a row- horizontally, vertically, diagonally, otherwise gets a tie.in which I have added an element so that u can play it with just one player and the other will be program itself.

The game can be played in a 3x3 grid or can be modified according to the mood and interest of the players but in this project, we are only going to play this in 3x3 grid as this is the easiest way to win and get rid of boredom.

This game can be played by only 1 player, if player wants to win this game, has to fulfill some conditions:

* + The player has to get three same either X’s or O’s in a straight line/row – diagonally, vertically, horizontally.
  + And as this game is a 2-player game then the player should get the 2 different markers to play with their consecutive turns therefore one by one, which means the markers need to be change after every execution.
  + And at last, if the game gets tired, it will be reset, and you can play it once again.
  1. **Python -:**

**Python** is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant indentation](https://en.wikipedia.org/wiki/Off-side_rule). Its [language constructs](https://en.wikipedia.org/wiki/Language_construct) as well as its [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help [programmers](https://en.wikipedia.org/wiki/Programmers) write clear, logical code for small and large-scale projects.

Python is [dynamically-typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms), including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly, [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

[Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) began working on Python in the late 1980s, as a successor to the [ABC programming language](https://en.wikipedia.org/wiki/ABC_(programming_language)), and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension) and a garbage collection system using [reference counting](https://en.wikipedia.org/wiki/Reference_counting) and was discontinued with version 2.7.18 in 2020. Python 3.0 was released in 2008 and was a major revision of the language that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility) and much Python 2 code does not run unmodified on Python 3.

Python consistently ranks as one of the most popular programming languages.

So the version I am working on is python 3.9 and IDE is PyCharm which is a very easy to use and help full IDE for beginners in python coding.

* 1. **GUI (Graphical User Interface) -----:**

The **graphical user interface** (**GUI** )is a form of [user interface](https://en.wikipedia.org/wiki/User_interface) that allows [users](https://en.wikipedia.org/wiki/User_(computing)) to [interact with electronic devices](https://en.wikipedia.org/wiki/Human%E2%80%93computer_interaction) through graphical [icons](https://en.wikipedia.org/wiki/Icon_(computing)) and audio indicator such as primary notation, instead of [text-based user interfaces](https://en.wikipedia.org/wiki/Text-based_user_interface), typed command labels or text navigation. GUIs were introduced in reaction to the perceived steep [learning curve](https://en.wikipedia.org/wiki/Learning_curve) of [command-line interfaces](https://en.wikipedia.org/wiki/Command-line_interface) (CLIs), which require commands to be typed on a [computer keyboard](https://en.wikipedia.org/wiki/Computer_keyboard).

The actions in a GUI are usually performed through [direct manipulation](https://en.wikipedia.org/wiki/Direct_manipulation_interface) of the graphical elements. Beyond computers, GUIs are used in many handheld [mobile devices](https://en.wikipedia.org/wiki/Mobile_device) such as [MP3](https://en.wikipedia.org/wiki/MP3) players, portable media players, gaming devices, [smartphones](https://en.wikipedia.org/wiki/Smartphone) and smaller household, office and [industrial controls](https://en.wikipedia.org/wiki/Distributed_control_system). The term *GUI* tends not to be applied to other lower-[display resolution](https://en.wikipedia.org/wiki/Display_resolution) [types of interfaces](https://en.wikipedia.org/wiki/User_interface#Types), such as [video games](https://en.wikipedia.org/wiki/Video_game) (where *head-up display* ([HUD](https://en.wikipedia.org/wiki/Head-up_display_(video_gaming))) is preferred), or not including flat screens, like [volumetric displays](https://en.wikipedia.org/wiki/Volumetric_display)because the term is restricted to the scope of two-dimensional display screens able to describe generic information, in the tradition of the [computer science](https://en.wikipedia.org/wiki/Computer_science) research at the [Xerox Palo Alto Research Center](https://en.wikipedia.org/wiki/PARC_(company)).

In python there are many GUI libraries to work with for example Tkinter, PyQt,

PySide, Kivy, WXpython and many other but for this project I have used Tkinter for the game to be in a GUI form.

* 1. **Tkinter ----:**

**Tkinter** is a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) [binding](https://en.wikipedia.org/wiki/Language_binding) to the [Tk](https://en.wikipedia.org/wiki/Tk_(software)) [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) toolkit. It is the standard Python interface to the Tk GUI toolkit and is Python's [*de facto* standard](https://en.wikipedia.org/wiki/De_facto_standard) GUI. Tkinter is included with standard [Linux](https://en.wikipedia.org/wiki/Linux), [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) and [Mac OS X](https://en.wikipedia.org/wiki/Mac_OS_X) installs of Python.

The name *Tkinter* comes from *Tk interface*. Tkinter was written by Fredrik Lundh. Tkinter is [free software](https://en.wikipedia.org/wiki/Free_software) released under a [Python license](https://en.wikipedia.org/wiki/Python_license). As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete [Tcl](https://en.wikipedia.org/wiki/Tcl) [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) embedded in the Python [interpreter](https://en.wikipedia.org/wiki/Language_interpretation). Tkinter calls are translated into Tcl commands, which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

There are several popular GUI library alternatives available, such as [wxPython](https://en.wikipedia.org/wiki/WxPython), [PyQt](https://en.wikipedia.org/wiki/PyQt), [PySide](https://en.wikipedia.org/wiki/PySide), [Pygame](https://en.wikipedia.org/wiki/Pygame), and [PyGTK](https://en.wikipedia.org/wiki/PyGTK).

**CHAPTER** **2**

1. **PROJECT -----:**
   1. **Requirement Analysis -:**

My previous was on tic tac toe for 2 player with was developed in “ c++” and does not have any GUI and was a game for two players .So, to make it an advanced tic tac toe I decided to make a tic tac toe for single player and also add some GUI to make it look good and the most efficient way to use GUI was to use python as u know python come with its own GUI . so I used the Tkinter for this project .

* 1. **Software Specification -:**

The software which I made is a game tic tac toe in which a player plays the game with the computer to win. And the software in which this code is written is python 3.9 . and the IDE is Pycharm which is very basic and easy to use software.

* 1. **Module ----:**
* **Buttons** -: There are 12 buttons in this project 9 buttons are to mark the places of x and o and the rest three are the option menu buttons. These 9 are placed in such a manner that they form a grid of 3 X 3 and they perform only one command which is “ def\_sign “ to assign the button a text that marks the “X” on the button and Those three are as follows :
* Reset button -: Its basically clears the grid if the match is drawn or the player want to start the again. It performs a “Reset” command.
* Quit button -: It closed the game window
* Next button-: It decides the place to mark the “O” on the grid which is the mark of the computer.

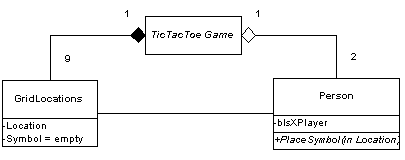
* **Functions -:** A function is a block of code which only runs when it is called. You can pass data, known as parameters, into a function. A function can return data as a result. The following are user defined function .
* **Compmove ()-:** this function for the 2nd player of the game it decides the place for the computer to put its marker on the button by the use of an algorithm called “MINIMAX “ which is used in board games . to predict the best move for the player .
* **Give\_number()-:** This function marks the marker “X” of the player 1 on the buttons , when they are clicked by the player. By calling an other method insert\_letter ();
* **insert\_letter ()-:** this function takes two arguments letter and position . to place the letter on the given position .
* **check\_win ()-:** It compares the values of the board array if the values of the board array is same in a row or column or in any diagonal then it return the value true .
* **spaceIsfree()-:**It basically checks the board array values if the value is null then it returns True .
* **check\_draw()-**:It also checks the value of the board if the board array is full then it means the game is draw so it returns True if there is no empty space in board array.
* **Minimax ()-**:it is basically the minimax algorithm which decides the best move for aur bot player to play .
* **Reset()-:**this function resets the text values of the buttons and the colour back to normal so that player can start over a new game .
  1. **APPLICATION --:**

The application of this project is to play a game when u r alone to get rid of boredom and as my previous project was to make a tic tac toe for two players so I chose this one to test my skills and make a single player tictactoe in GUI format.

This project is based on “TIC-TAC-TOE”, which can play whenever you feel bored and can be played anywhere. You just need 2 peoples to play and enjoy. This game has very simple rules. It is also known as X’s and O’s. The player who wants to win, one has to get three same either X’s or O’s in a row- horizontally, vertically, diagonally, otherwise gets a tie.in which I have added an element so that u can play it with just one player and the other will be program itself.

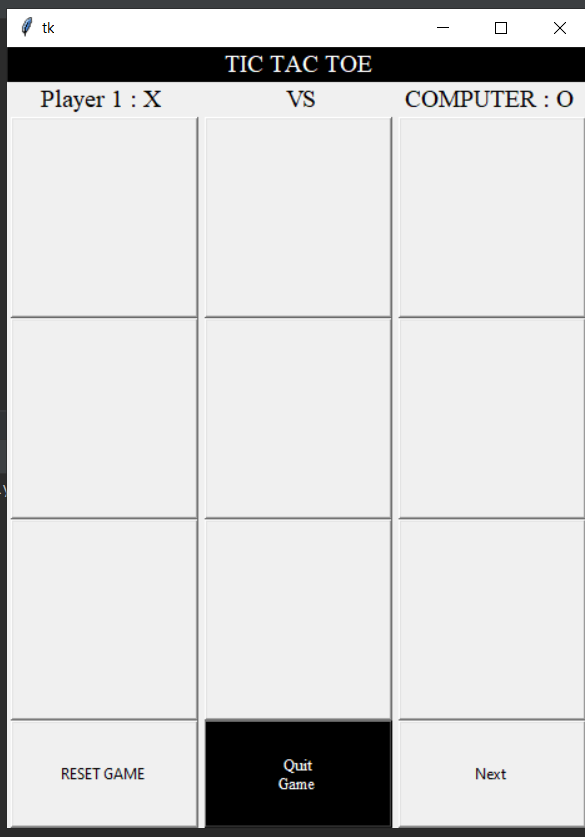
* 1. **ER DIAGRAM ----:**

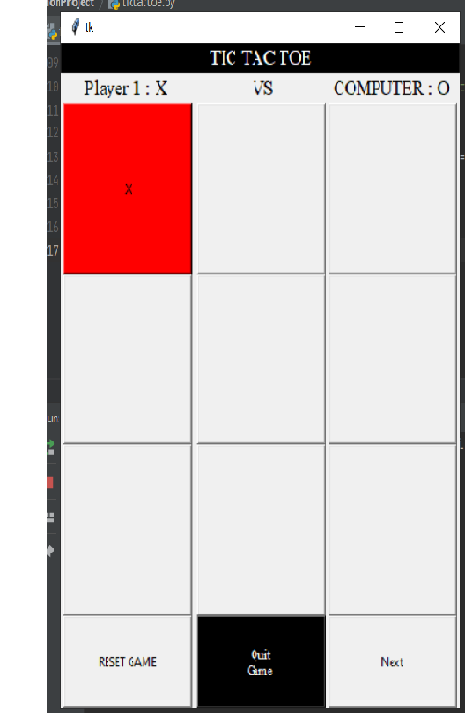
This is very basic er diagram for this project .

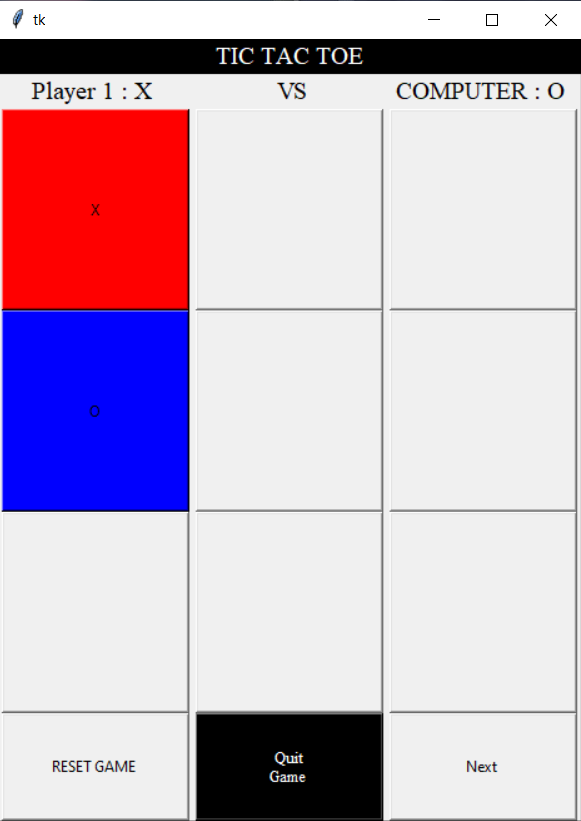
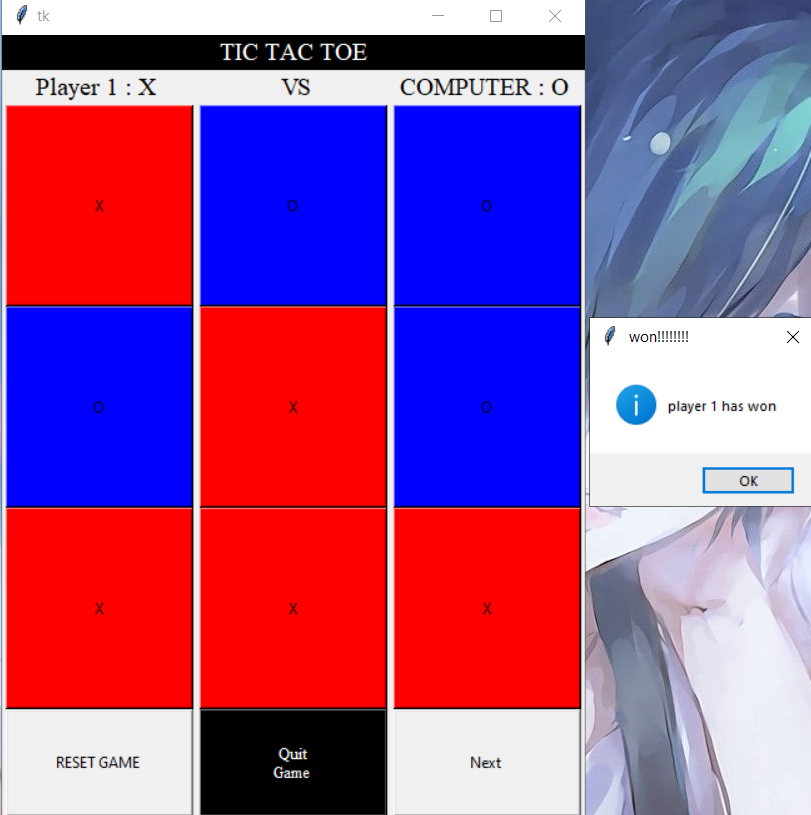


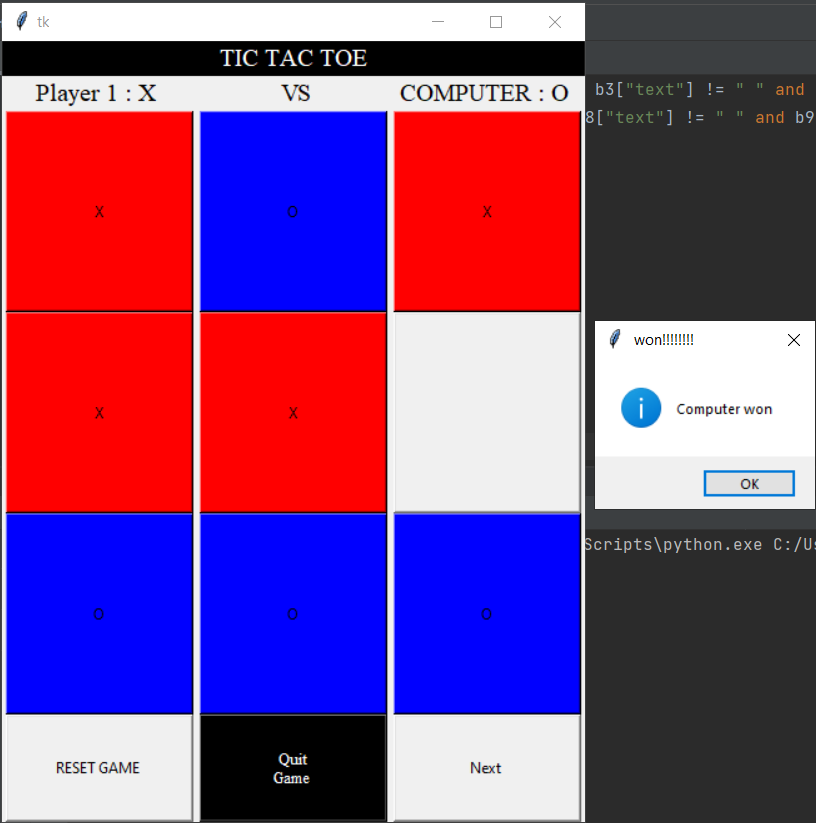
**CHAPTER** **3**

**SNAPSHOT OF PROJECT**

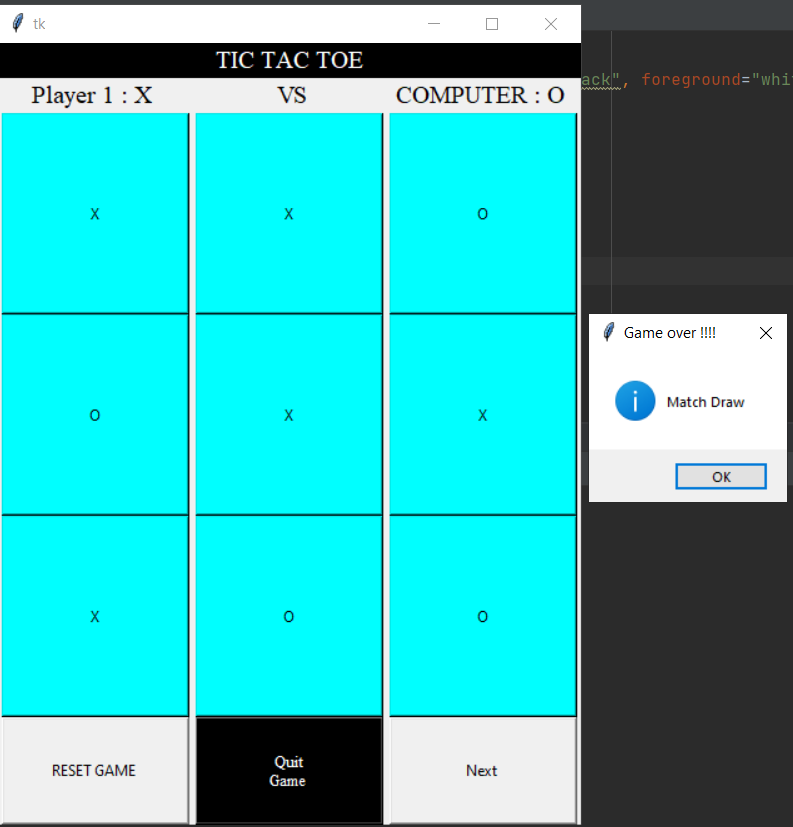
* **THE GAME -----:**
* **1st players turn ---:**

****

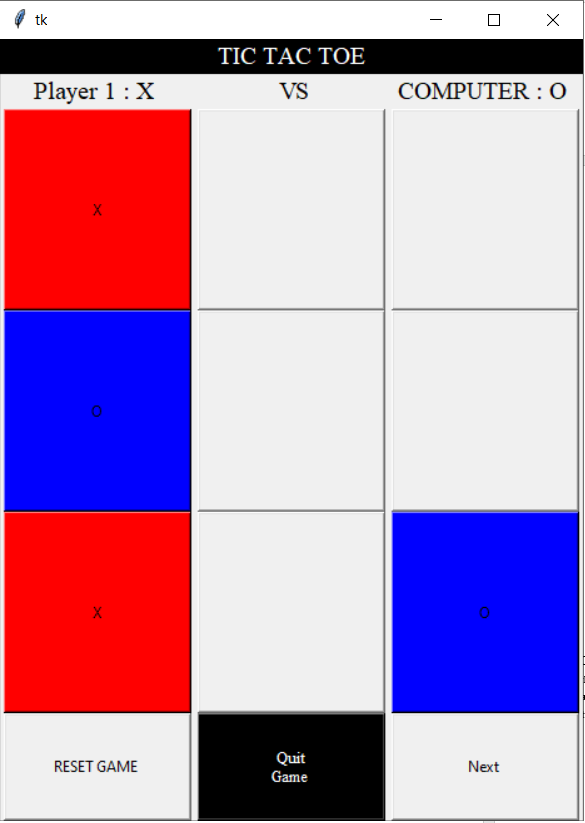
* **Computer’s turn ---:**
* **Wining situation ----:**

****

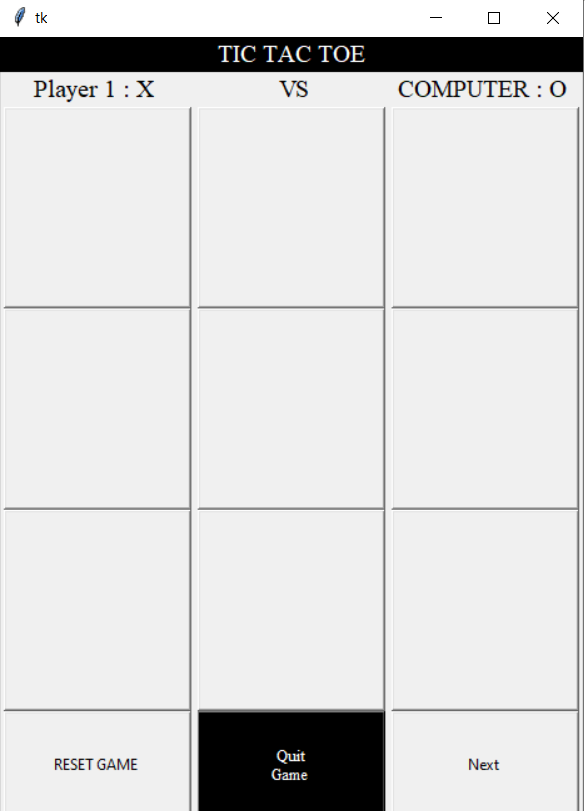
* **Draw situation ---:**

****

* **Reset ---:**

****

**Before reset ===**

****

**After reset =======**

**CHAPTER 4**

**CONCLUSION**

**4.1 CONCLUSION**

**From the game is project we have concluded that the tkinter in python is a good library to use and easy to work with if you know python and GUI is very inserting to use in our project and at last the game is fully functional and working**

**REFERENCE**

1. Head-First **Python** (**2nd** edition)
2. Stackoverflow website .
3. Geeks of geeks

**Appendix**

**Link ------:**

**https://drive.google.com/drive/folders/1nQIgQh0EfKXKjh8nfO4lB3iuaGScPtTy?usp=sharing**