Subject:**Data Sciences**

**Name:hidayatullah khan ID:14311**

**BS (,SE)** Instructor: M.Ayub Khan

Q1. Create anadvance Tic Tac Toe game in python (User interface must have

 reset and new game buttons use any python tool for coding ) and explain it in

detail including screen shots.

Ans:

We will use Pydroid3 for the development of tic tac toe game.

**How does the game work?**

It is numbered just like the keyboard no pad and the player can make his move

By entering number.

First we will use the dictionary to develop game . It is a permitive type of data type in python which stores the data in key: value format.the length of it will be 9 and each key will represennt a block in it .we will create a function printBoard() which we can use everytime we want to print the updated board in the game.

theBoard = {'a': ' ' , 'b': ' ' , 'c': ' ' ,

 'd': ' ' , 'e': ' ' , 'f': ' ' ,

 'g': ' ' , 'h': ' ' , 'i': ' ' }

def printBoard(board):

 print(board['a'] + '|' + board['b'] + '|' + board['c'])

 print('-+-+-')

 print(board['d'] + '|' + board['e'] + '|' + board['f'])

 print('-+-+-')

 print(board['g'] + '|' + board['h'] + '|' + board['i'])

So at the start our display will look like this

| |
-+-+-
| |
-+-+-
| |

The main function first of all it will take the input from the user and check if it is valid it will move otherwise wontif that path is full so it will ask the user to choose another one.

Def game():

 turn = 'X'

 count = 0

 for i in range(10):

 printBoard(theBoard)

 print("It's your turn," + turn + ".Move to which place?")

 move = input()

 if theBoard[move] == ' ':

 theBoard[move] = turn

 count += 1

 else:

 print("That place is already filled.\nMove to which place?")

 continue

For checking the winning condition there will be total of 8 conditions and the player that made the last move will be declare as the winner and if no player wins so it will be a tie.

 if count >= 5:

 if theBoard['a'] == theBoard['b'] == theBoard['c'] != ' ': # across the top

 printBoard(theBoard)

 print("\nGame Over.\n")

 print(" \*\*\*\* " +turn + " won. \*\*\*\*")

 break

 elif theBoard['d'] == theBoard['e'] == theBoard['f'] != ' ': # across the middle

 printBoard(theBoard)

 print("\nGame Over.\n")

 print(" \*\*\*\* " +turn + " won. \*\*\*\*")

 break

 elif theBoard['g'] == theBoard['h'] == theBoard['i'] != ' ': # across the bottom

 printBoard(theBoard)

 print("\nGame Over.\n")

 print(" \*\*\*\* " +turn + " won. \*\*\*\*")

 break

 elif theBoard['g'] == theBoard['d'] == theBoard['a'] != ' ': # down the left side

 printBoard(theBoard)

 print("\nGame Over.\n")

 print(" \*\*\*\* " +turn + " won. \*\*\*\*")

 break

 elif theBoard['h'] == theBoard['e'] == theBoard['b'] != ' ': # down the middle

 printBoard(theBoard)

 print("\nGame Over.\n")

 print(" \*\*\*\* " +turn + " won. \*\*\*\*")

 break

 elif theBoard['i'] == theBoard['f'] == theBoard['c'] != ' ': # down the right side

 printBoard(theBoard)

 print("\nGame Over.\n")

 print(" \*\*\*\* " +turn + " won. \*\*\*\*")

 break

 elif theBoard['a'] == theBoard['e'] == theBoard['i'] != ' ': # diagonal

 printBoard(theBoard)

 print("\nGame Over.\n")

 print(" \*\*\*\* " +turn + " won. \*\*\*\*")

 break

 elif theBoard['g'] == theBoard['e'] == theBoard['c'] != ' ': # diagonal

 printBoard(theBoard)

 print("\nGame Over.\n")

 print(" \*\*\*\* " +turn + " won. \*\*\*\*")

 break

# If neither X nor O wins and the board is full, we'll declare the result as 'tie'.

 if count == 9:

 print("\nGame Over.\n")

 print("It's a Tie!!")

 # we have to change the player after every move.

 if turn =='X':

 turn = 'O'

 else:

 turn = 'X'

**PlayAgain:**

We will ask the users to play again if they want.

board\_keys = []

for key in theBoard:

 board\_keys.append(key)

restart = input("Do want to play Again?(y/n)")

 if restart == "y" or restart == "Y":

 for key in board\_keys:

 theBoard[key] = " "

 game()

1 







