Version 1

Features:

1) 2 players will be to play the game, both humans
2) Board will be built using a dictionary:
    a) Board = {1: ‘ ‘, 2: ‘ ‘ …. 9: ‘ ‘}
3) Players can place piece on board by inputting board coordinate. E.g 5
    a) 1 - 9 the coordinates of the grid
4) Everytime piece is place the program will check if there is a winner
5) output draw if no one has won

Code:

```
def draw():
    #used to draw out the grid, in the format of a tictactoe game.
    for b in range(1,10):
        if b % 3 != 0:
            print(' '+positions[b]+' '+'|', end='')
        else:
            print(' '+positions[b]+' ', end='')
        if b == 3 or b == 6:
            print('\n '+'~ '*5)

def win(positions):
    #player1
    #checks if player 1 is the winner
    for i in range(1,10,3):
        #horizontal
        if positions[i] == 'x' and positions[i+1] == 'x' and positions[i+2] == 'x':
            return '\nPlayer1 Is the Winner'
    for i in range(1,4):
        #vertical
        if positions[i] == 'x' and positions[i+3] == 'x' and positions[i+6] == 'x':
            return '\nPlayer1 Is the Winner'
    #diagonal
    if positions[1] == 'x' and positions[5] == 'x' and positions[9] == 'x':
        return '\nplayer1 is the winner'
    elif positions[3] == 'x' and positions[5] == 'x' and positions[7] == 'x':
        return '\nplayer1 is the winner'

    #player2
    #checks if player 2 is the winner
    for i in range(1,10,3):
        #horizontal
        if positions[i] == 'o' and positions[i+1] == 'o' and positions[i+2] == 'o':
            return '\nPlayer1 Is the Winner'
    for i in range(1,4):
        #vertical
        if positions[i] == 'o' and positions[i+3] == 'o' and positions[i+6] == 'o':
            return '\nPlayer1 Is the Winner'
    #diagonal
    if positions[1] == 'o' and positions[5] == 'o' and positions[9] == 'o':
        return '\nplayer1 is the winner'
    elif positions[3] == 'o' and positions[5] == 'o' and positions[7] == 'o':
        return '\nplayer1 is the winner'

Player1 = 'x'
Player2 = 'o'

positions = {}

# creates a dictionary that is numbered 1-9 with each valued equal to ' '
for i in range(1, 10):
    positions[i] = ' '
```

```
# creates a dictionary that is numbered 1-9 with each valued equal to ' '
for i in range(1, 10):
    positions[i] = ' '

draw()
#repeats this code 5 times, this will fill the whole grid, if there is no winner
for i in range(5):
    #this checks if there is a winner, if there isn't a winner it will return 'none' which wont run this code below
    if win(positions) != None:
        print(win(positions))
        break

    #ask user to put there x on the grid, this will automatically put it in the dictionary which will then be printed out via draw()
    positions[int(input('\n(P1)Choose a number between 1-9(which is not taken), to put your "X" down: '))] = Player1
    draw()

    #this checks again, incase there is a winner after the previous input.
    if win(positions) != None:
        print(win(positions))
        break

    #an if statement which makes sure there isn't an extra value that is inputted into the grid since it repeats 5 times and theres 2 players so there would be an overall of 10 placements which is not needed.
    if i != 4:
        positions[int(input('\n(P2)Choose a number between 1-9(which is not taken), to put your "O" down: '))] = Player2
        draw()
```

Version 2

Features

1) Same as version one, however there will be 3 games
2) Print out who has won, out of 3 games and the score

Code:

```
def draw():
    #used to draw out the grid, in the format of a tictactoe game.
    for b in range(1,10):
        if b % 3 != 0:
            print(' '+positions[b]+' '+'|', end='')
        else:
            print(' '+positions[b]+' ', end='')
        if b == 3 or b == 6:
            print('\n '+'~ '*5)

def win(positions):
    #player1
    #checks if player 1 is the winner
    for i in range(1,10,3):
        #horizontal
        if positions[i] == 'x' and positions[i+1] == 'x' and positions[i+2] == 'x':
            return '\nPlayer1 Is the Winner'
    for i in range(1,4):
        #vertical
        if positions[i] == 'x' and positions[i+3] == 'x' and positions[i+6] == 'x':
            return '\nPlayer1 Is the Winner'
    #diagonal
    if positions[1] == 'x' and positions[5] == 'x' and positions[9] == 'x':
        return '\nplayer1 is the winner'
    elif positions[3] == 'x' and positions[5] == 'x' and positions[7] == 'x':
        return '\nplayer1 is the winner'

    #player2
    #checks if player 2 is the winner
    for i in range(1,10,3):
        #horizontal
        if positions[i] == 'o' and positions[i+1] == 'o' and positions[i+2] == 'o':
            return '\nPlayer1 Is the Winner'
    for i in range(1,4):
        #vertical
        if positions[i] == 'o' and positions[i+3] == 'o' and positions[i+6] == 'o':
            return '\nPlayer1 Is the Winner'
    #diagonal
    if positions[1] == 'o' and positions[5] == 'o' and positions[9] == 'o':
        return '\nplayer1 is the winner'
    elif positions[3] == 'o' and positions[5] == 'o' and positions[7] == 'o':
        return '\nplayer1 is the winner'

Player1 = 'x'
Player2 = 'o'
p1Score = 0
p2Score = 0

#makes it show that there is 3 games
for i in range(3):
    positions = {}
```

```
#makes it show that there is 3 games
for i in range(3):
    positions = {}

    # creates a dictionary that is numbered 1-9 with each valued equal to ' '
    for i in range(1, 10):
        positions[i] = ' '

    draw()
    #repeats this code 5 times, this will fill the whole grid, if there is no winner
    for i in range(5):
        #this checks if there is a winner, if there isn't a winner it will return 'none' which wont run this code below
        if win(positions) != None:
            print(win(positions))
            #if it returns player1 is ther winner, it will add a score to player1 else player2 gains a score
            if 'Player1' in win(positions):
                p1Score += 1
            else:
                p2Score += 1
            break
        #ask user to put there x on the grid, this will automatically put it in the dictionary which will then be printed out via draw()
        positions[int(input('\n(P1)Choose a number between 1-9(which is not taken), to put your "X" down: '))] = Player1
        draw()

        #this checks again, incase there is a winner after the previous input.
        if win(positions) != None:
            print(win(positions))
            # if it returns player1 is ther winner, it will add a score to player1 else player2 gains a score
            if 'Player1' in win(positions):
                p1Score += 1
            else:
                p2Score += 1
            break

        #an if statement which makes sure there isn't an extra value that is inputted into the grid since it repeats 5 times and theres 2 players so there would be an overall of 10 placements which is not needed.
        if i != 4:
            positions[int(input('\n(P2)Choose a number between 1-9(which is not taken), to put your "O" down: '))] = Player2
            draw()

#checks if player1 or player2 won the game, and prints the score out.
if p1Score > p2Score:
    print('player 1 is the winner with the score of p1:p2 - {}'.format((str(p1Score)+':'+str(p2Score))))
else:
    print('player 2 is the winner with the score of p1:p2 - {}'.format((str(p1Score) + ':' + str(p2Score))))
```

Version 3

Features

1) Not allow the user to put it in the same place as a position that is taken.

Code:

```
while True:
    pos = int(input('\n(P1)Choose a number between 1-9(which is not taken), to put your "X" down: '))
    if positions[pos] == ' ':
        positions[pos] = Player1
        break
    else:
        print('position is already taken please try again')
draw()

#this checks again, incase there is a winner after the previous input.
if win(positions) != None:
    print(win(positions))
    # if it returns player1 is ther winner, it will add a score to player1 else player2 gains a score
    if 'Player1' in win(positions):
        p1Score += 1
    else:
        p2Score += 1
    break

#an if statement which makes sure there isn't an extra value that is inputted into the grid since it repeats 5 times and ther
if i != 4:
    while True:
        pos = int(input('\n(P1)Choose a number between 1-9(which is not taken), to put your "o" down: '))
        if positions[pos] == ' ':
            positions[pos] = Player2
            break
        else:
            print('position is already taken please try again')
    draw()
```

Same code as previous one, however with the while loop and extra if statements. As screenshotted above.

Test:

```
(P1)Choose a number between 1-9(which is not taken), to put your "X" down: 1
position is already taken please try again
   |   |
 ~ ~ ~ ~ ~
   |   |
 ~ ~ ~ ~ ~
   |   |
(P1)Choose a number between 1-9(which is not taken), to put your "X" down: 2
position is already taken please try again
   |   |
 ~ ~ ~ ~ ~
   |   |
 ~ ~ ~ ~ ~
   |   |
(P1)Choose a number between 1-9(which is not taken), to put your "X" down:
```

The problem of this was that i had an if statement that is look for a space and if there is a space it will print error message, which is something i didn’t want.

Version 4

Features

1) Create a GUI with tkinter for tic tac toe
2) When clicked if x bg = blue and when o = pink
3) Be able to play as much as the user wants to
4) Keep the score of each users until they want to end.
5) Be able to draw with the user

Code:

```
47
48          # player2
49      elif button1['text'] == 'o' and button2['text'] == 'o' and button3['text'] == 'o':
50          return '\nPlayer2 Is the Winner'
51      elif button4['text'] == 'o' and button5['text'] == 'o' and button6['text'] == 'o':
52          return '\nPlayer2 Is the Winner'
53      elif button7['text'] == 'o' and button8['text'] == 'o' and button9['text'] == 'o':
54          return '\nPlayer2 Is the Winner'
55      #vertical
56      elif button1['text'] == 'o' and button4['text'] == 'o' and button7['text'] == 'o':
57          return '\nPlayer2 Is the Winner'
58      elif button2['text'] == 'o' and button5['text'] == 'o' and button8['text'] == 'o':
59          return '\nPlayer2 Is the Winner'
60      elif button3['text'] == 'o' and button6['text'] == 'o' and button9['text'] == 'o':
61          return '\nPlayer2 Is the Winner'
62      #diagonal
63      elif button1['text'] == 'o' and button5['text'] == 'o' and button9['text'] == 'o':
64          return '\nPlayer2 Is the Winner'
65      elif button3['text'] == 'o' and button5['text'] == 'o' and button7['text'] == 'o':
66          return '\nPlayer2 Is the Winner'
67
68  def change1():
69      check()
70      if check() == None:
71          if num1 == True:
72              if player == True:
73                  button1.configure(text=P1, bg="blue")
74                  global player, back
75                  player = False
76                  back = P1
77              elif player == False:
78                  button1.configure(text=P2, bg="deep pink")
79                  global player, back
80                  player = True
81                  back = P1
82              global num1
83              num1 = False
84          else:
85              print('Position Taken')
86      else:
87          tkinter.messagebox.showinfo("WInner Of TicTacToe", "{}".format(check()))
```

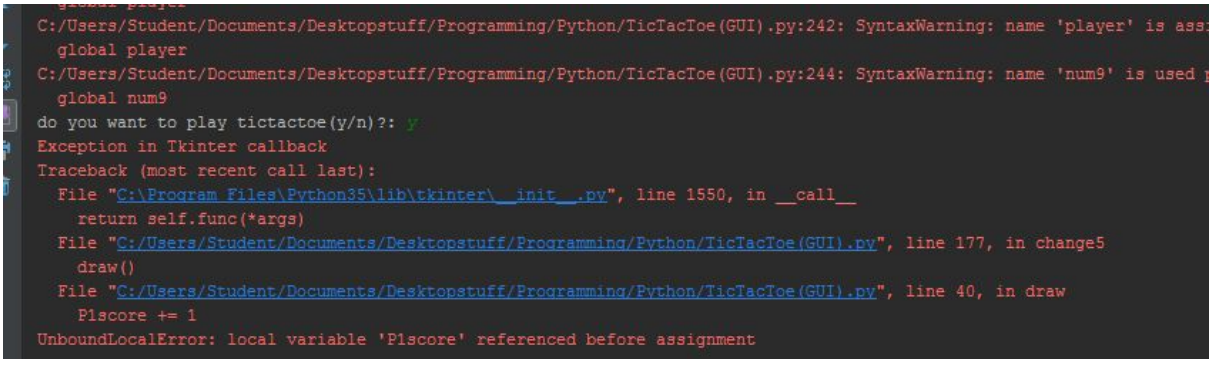
```
cont = 'y'

while cont != 'n':
    cont = input('do you want to play tictactoe(y/n)?: ').lower()

    if cont == 'y':
        tictac()
    else:
        print('goodBye')
```

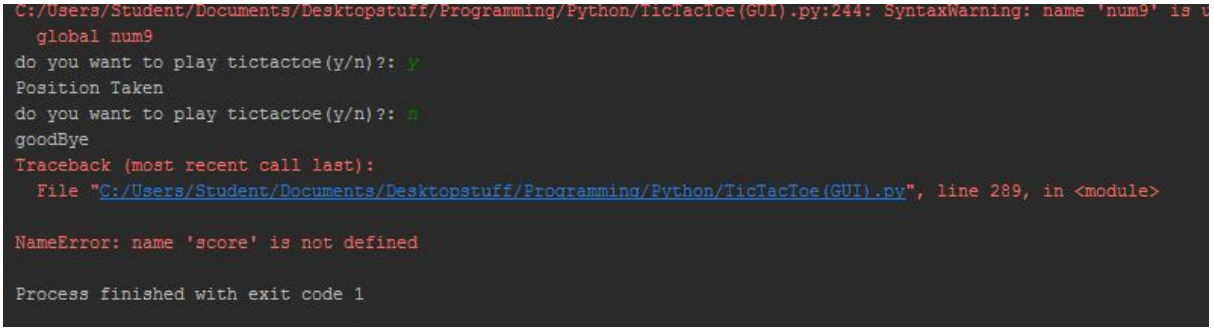
```
def tictac():
    master=Tk()

    master.title('TicTacToe')

    global player, P1, P2, empty, num1, num2, num3, num4, num5, num6, num7, num8, num9

    player = True
    P1 = 'x'
    P2 = 'o'
    empty = ' '
```

```
def draw():
    if num1 != True and num2 != True and num3 != True and num4 != True and num5 != True and num6 != True and num7 != True and num8 != True and num9 != True and check() == None:
        tkinter.messagebox.showinfo("Draw - TicTacToe", 'The game was a score')
        master.destroy()
    elif check() != None:
        tkinter.messagebox.showinfo("WInner Of TicTacToe", "{}".format(check()))
        master.destroy()
    else:
        print('Position Taken')
```

```
    def change1():
        check()
        if check() == None:
            if num1 == True:
                if player == True:
                    button1.configure(text=P1, bg="blue")
                    global player, back
                    player = False
                    back = P1
                elif player == False:
                    button1.configure(text=P2, bg="deep pink")
                    global player, back
                    player = True
                    back = P1
                global num1
                num1 = False
            else:
                draw()
```

Test


C:/Users/Student/Documents/Desktopstuff/Programming/Python/TicTacToe(GUI).py:242: SyntaxWarning: name 'player' is ass
global player
C:/Users/Student/Documents/Desktopstuff/Programming/Python/TicTacToe(GUI).py:244: SyntaxWarning: name 'num9' is used
global num9
do you want to play tictactoe(y/n)?: y
Exception in Tkinter callback
Traceback (most recent call last):
File "C:\Program Files\Python35\lib\tkinter__init__.py", line 1550, in __call__
return self.func(*args)
File "C:/Users/Student/Documents/Desktopstuff/Programming/Python/TicTacToe(GUI).py", line 177, in change5
draw()
File "C:/Users/Student/Documents/Desktopstuff/Programming/Python/TicTacToe(GUI).py", line 40, in draw
P1score += 1
UnboundLocalError: local variable 'P1score' referenced before assignment


C:/Users/Student/Documents/Desktopstuff/Programming/Python/TicTacToe(GUI).py:244: SyntaxWarning: name 'num9' is u
global num9
do you want to play tictactoe(y/n)?: y
Position Taken
do you want to play tictactoe(y/n)?: n
goodBye
Traceback (most recent call last):
File "C:/Users/Student/Documents/Desktopstuff/Programming/Python/TicTacToe(GUI).py", line 289, in <module>
NameError: name 'score' is not defined
Process finished with exit code 1