# 

# Check if the square is already filled

while (board[row][column] == symbol\_1)or (board[row][column] == symbol\_2):

filled = illegal(board, symbol\_1, symbol\_2, row, column)

row = int(input("Pick a row[upper row:"

"[enter 0, middle row: enter 1, bottom row: enter 2]:"))

column = int(input("Pick a column:"

"[left column: enter 0, middle column: enter 1, right column enter 2]"))

# Locates player's symbol on the board

if player == symbol\_1:

board[row][column] = symbol\_1

else:

board[row][column] = symbol\_2

return (board)

def isFull(board, symbol\_1, symbol\_2):

count = 1

winner = True

# This function check if the board is full

while count < 10 and winner == True:

gaming = startGamming(board, symbol\_1, symbol\_2, count)

pretty = printPretty(board)

if count == 9:

print("The board is full. Game over.")

if winner == True:

print("There is a tie. ")

# Check if here is a winner

winner = isWinner(board, symbol\_1, symbol\_2, count)

count += 1

if winner == False:

print("Game over.")

# This is function gives a report

report(count, winner, symbol\_1, symbol\_2)

def outOfBoard(row, column):

# This function tells the players that their selection is out of range

print("Out of boarder. Pick another one. ")

def printPretty(board):

# This function prints the board nice!

rows = len(board)

cols = len(board)

print("---+---+---")

for r in range(rows):

print(board[r][0], " |", board[r][1], "|", board[r][2])

print("---+---+---")

return board

def isWinner(board, symbol\_1, symbol\_2, count):

# This function checks if any winner is winning

winner = True

# Check the rows

for row in range (0, 3):

if (board[row][0] == board[row][1] == board[row][2] == symbol\_1):

winner = False

print("Player " + symbol\_1 + ", you won!")

elif (board[row][0] == board[row][1] == board[row][2] == symbol\_2):

winner = False

print("Player " + symbol\_2 + ", you won!")

# Check the columns

for col in range (0, 3):

if (board[0][col] == board[1][col] == board[2][col] == symbol\_1):

winner = False

print("Player " + symbol\_1 + ", you won!")

elif (board[0][col] == board[1][col] == board[2][col] == symbol\_2):

winner = False

print("Player " + symbol\_2 + ", you won!")

# Check the diagnoals

if board[0][0] == board[1][1] == board[2][2] == symbol\_1:

winner = False

print("Player " + symbol\_1 + ", you won!")

elif board[0][0] == board[1][1] == board[2][2] == symbol\_2:

winner = False

print("Player " + symbol\_2 + ", you won!")

elif board[0][2] == board[1][1] == board[2][0] == symbol\_1:

winner = False

print("Player " + symbol\_1 + ", you won!")

elif board[0][2] == board[1][1] == board[2][0] == symbol\_2:

winner = False

print("Player " + symbol\_2 + ", you won!")

return winner

def illegal(board, symbol\_1, symbol\_2, row, column):

print("The square you picked is already filled. Pick another one.")

def report(count, winner, symbol\_1, symbol\_2):

print("\n")

input("Press enter to see the game summary. ")

if (winner == False) and (count % 2 == 1 ):

print("Winner : Player " + symbol\_1 + ".")

elif (winner == False) and (count % 2 == 0 ):

print("Winner : Player " + symbol\_2 + ".")

else:

print("There is a tie. ")

# Call Main

main()

Def main():

# The main function

Introduction = intro()

Board = create\_grid()

Pretty = printPretty(board)

Symbol\_1, symbol\_2 = sym()

Full = isFull(board, symbol\_1, symbol\_2) # The function that starts the game is also in here.

Def intro():

# This function introduces the rules of the game Tic Tac Toe

Print(“Hello! Welcome to Pam’s Tic Tac Toe game!”)

Print(“\n”)

Print(“Rules: Player 1 and player 2, represented by X and O, take turns “

“marking the spaces in a 3\*3 grid. The player who succeeds in placing “

“three of their marks in a horizontal, vertical, or diagonal row wins.”)

Print(“\n”)

Input(“Press enter to continue.”)

Print(“\n”)

Def create\_grid():

# This function creates a blank playboard

Print(“Here is the playboard: “)

Board = [[“ “, “ “, “ “],

[“ “, “ “, “ “],

[“ “, “ “, “ “]]

Return board

Def sym():

# This function decides the players’ symbols

Symbol\_1 = input(“Player 1, do you want to be X or O? “)

If symbol\_1 == “X”:

Symbol\_2 = “O”

Print(“Player 2, you are O. “)

Else:

Symbol\_2 = “X”

Print(“Player 2, you are X. “)

Input(“Press enter to continue.”)

Print(“\n”)

Return (symbol\_1, symbol\_2)

Def startGamming(board, symbol\_1, symbol\_2, count):

# This function starts the game.

# Decides the turn

If count % 2 == 0:

Player = symbol\_1

Elif count % 2 == 1:

Player = symbol\_2

Print(“Player “+ player + “, it is your turn. “)

Row = int(input(“Pick a row:”

“[upper row: enter 0, middle row: enter 1, bottom row: enter 2]:”))

Column = int(input(“Pick a column:”

“[left column: enter 0, middle column: enter 1, right column enter 2]”))

# Check if players’ selection is out of range

While (row > 2 or row < 0) or (column > 2 or column < 0):

outOfBoard(row, column)

row = int(input(“Pick a row[upper row:”

“[enter 0, middle row: enter 1, bottom row: enter 2]:”))

Column = int(input(“Pick a column:”

“[left column: enter 0, middle column: enter 1, right column enter 2]”))

# Check if the square is already filled

While (board[row][column] == symbol\_1)or (board[row][column] == symbol\_2):

Filled = illegal(board, symbol\_1, symbol\_2, row, column)

Row = int(input(“Pick a row[upper row:”

“[enter 0, middle row: enter 1, bottom row: enter 2]:”))

Column = int(input(“Pick a column:”

“[left column: enter 0, middle column: enter 1, right column enter 2]”))

# Locates player’s symbol on the board

If player == symbol\_1:

Board[row][column] = symbol\_1

Else:

Board[row][column] = symbol\_2

Return (board)

Def isFull(board, symbol\_1, symbol\_2):

Count = 1

Winner = True

# This function check if the board is full

While count < 10 and winner == True:

Gaming = startGamming(board, symbol\_1, symbol\_2, count)

Pretty = printPretty(board)

If count == 9:

Print(“The board is full. Game over.”)

If winner == True:

Print(“There is a tie. “)

# Check if here is a winner

Winner = isWinner(board, symbol\_1, symbol\_2, count)

Count += 1

If winner == False:

Print(“Game over.”)

# This is function gives a report

Report(count, winner, symbol\_1, symbol\_2)

Def outOfBoard(row, column):

# This function tells the players that their selection is out of range

Print(“Out of boarder. Pick another one. “)

Def printPretty(board):

# This function prints the board nice!

Rows = len(board)

Cols = len(board)

Print(“---+---+---“)

For r in range(rows):

Print(board[r][0], “ |”, board[r][1], “|”, board[r][2])

Print(“---+---+---“)

Return board

Def isWinner(board, symbol\_1, symbol\_2, count):

# This function checks if any winner is winning

Winner = True

# Check the rows

For row in range (0, 3):

If (board[row][0] == board[row][1] == board[row][2] == symbol\_1):

Winner = False

Print(“Player “ + symbol\_1 + “, you won!”)

Elif (board[row][0] == board[row][1] == board[row][2] == symbol\_2):

Winner = False

Print(“Player “ + symbol\_2 + “, you won!”)

# Check the columns

For col in range (0, 3):

If (board[0][col] == board[1][col] == board[2][col] == symbol\_1):

Winner = False

Print(“Player “ + symbol\_1 + “, you won!”)

Elif (board[0][col] == board[1][col] == board[2][col] == symbol\_2):

Winner = False

Print(“Player “ + symbol\_2 + “, you won!”)

# Check the diagnoals

If board[0][0] == board[1][1] == board[2][2] == symbol\_1:

Winner = False

Print(“Player “ + symbol\_1 + “, you won!”)

Elif board[0][0] == board[1][1] == board[2][2] == symbol\_2:

Winner = False

Print(“Player “ + symbol\_2 + “, you won!”)

Elif board[0][2] == board[1][1] == board[2][0] == symbol\_1:

Winner = False

Print(“Player “ + symbol\_1 + “, you won!”)

Elif board[0][2] == board[1][1] == board[2][0] == symbol\_2:

Winner = False

Print(“Player “ + symbol\_2 + “, you won!”)

Return winner

Def illegal(board, symbol\_1, symbol\_2, row, column):

Print(“The square you picked is already filled. Pick another one.”)

Def report(count, winner, symbol\_1, symbol\_2):

Print(“\n”)

Input(“Press enter to see the game summary. “)

If (winner == False) and (count % 2 == 1 ):

Print(“Winner : Player “ + symbol\_1 + “.”)

Elif (winner == False) and (count % 2 == 0 ):

Print(“Winner : Player “ + symbol\_2 + “.”)

Else:

Print(“There is a tie. “)

# Call Main

Main()