**** A

Mini Project Report

on

**“TIC-TAC-TOE ”**

Submitted in Partial Fulfillment for the Award of Degree of

Bachelor of Technology (4th Sem)

**Guided By: -Ms Anuja Mishra Mam Submitted By:**

Payal Agarwal(18BCON653)

****

Department of Computer Science and Engineering

JECRC University, Jaipur (Raj.)

14 th April 2020

**JECRC University, Jaipur**

**Department of Computer Science and Engineering**

# CERTIFICATE OF APPROVAL

This is to certify that we have examined and approved the mini project for 4 TH semester in CS Engineering entitled **“TIC TAC TOE GAME”** submitted by

Payal Agarwal(18BCON653)

to JECRC UNIVERSITY,JAIPUR. We here by accord our approval of it as a mini project work carried out and presented in a manner required for its acceptance for the partial fulfillment for the Bachelor Degree of Technology in Computer Science Engineering for which it has been submitted. This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusions drawn as recorded in this mini project, it only signifies the acceptance of the mini project for the purpose it has been submitted.

**(Project Guide) (External) (HOD ETC)**

# Contents

1.CERTIFICATE

2.DECLARATION

3.ACKNOWLEDGEMENT

4.ABSTRACT

5.INTRODUCTION

#OVERVIEW

#BACKGROUND AND MOTIVATION

6.METHODOLOGY

7.RESULT/OUTCOME

8.CONCLUSION

# DECLARATION

This is to declare that the work in the mini project entitled **(“TIC TAC TOE”)** submitted by us in partial fulfillment of the requirement for the 4th semester for bachelor of technology with specialization in computer science is a bona fide work carried out by us under the supervision and guidance of Ms Anuja Mam. It does not contain materials copied from other published work ,and the work is not published anywhere.

Team member :

Payal Agarwal (18BCON653)

# ACKNOWLEDGEMENT

I would like to express my gratitude to my project guide Ms Anuja Mishra Mam for all the useful comment , remarks and encouragement throughout the session for this project. I would like to thanks my well wisher who have supported me throughout the session for this project.

I am also grateful to other members of the department and college who cooperated along for carrying out the work successfully.

Team member :

Payal Agarwal(18BCON653)

**ABSTRACT**

**“TIC TAC TOE”** report in Python**.** The purpose of this project is to develop a computerized tic-tac-toe game that can be used to revolutionize the traditional gaming system that currently implemented in majority of the infants. The traditional system that had most of the infants is the traditional manual gaming system which means all work is done through huge amount of paper and pen that is not effective and efficient. This causes the environment to encounter trouble due to the huge amount of paper wasting that operating. Thus, this computerized gaming system.

The chosen methodology for this project is throwaway prototyping methodology. This is because majority of the targeted user do not have the experience in using gaming system in tic-tac-toe game procedure as they implement traditional system previously. Therefore, this methodology enables developer to communicate with target user through using the prototyping, which can let target user to review, evaluate, visualize and learn about the system before the actual implementation of the final system. Furthermore, the system is a cross-platform system which involve all operating system.

**INTRODUCTION**

It is software that can be used by attendants to play game. It also keeps the record and tracks of all the scores. The software is very convenient to use. Python Programming language is used to develop this application. Functions, conditional statements and loops are mainly used to design this application.

This system makes the gaming easier and systematic are given.

The application has been successfully tested, has no errors and works absolutely fine.0

**METHODOLOGY**

# --------- Global Variables -----------

# Will hold our game board data

board = ["-", "-", "-",

"-", "-", "-",

"-", "-", "-"]

# Lets us know if the game is over yet

game\_still\_going = True

# Tells us who the winner is

winner = None

# Tells us who the current player is (X goes first)

current\_player = "X"

# ------------- Functions ---------------

# Play a game of tic tac toe

def play\_game():

# Show the initial game board

display\_board()

# Loop until the game stops (winner or tie)

while game\_still\_going:

# Handle a turn

handle\_turn(current\_player)

# Check if the game is over

check\_if\_game\_over()

# Flip to the other player

flip\_player()

# Since the game is over, print the winner or tie

if winner == "X" or winner == "O":

":

print(winner + " won.")

elif winner == None:

print("Tie.")

# Display the game board to the screen

def display\_board():

print("\n")

print(board[0] + " | " + board[1] + " | " + board[2] + " 1 | 2 | 3")

print(board[3] + " | " + board[4] + " | " + board[5] + " 4 | 5 | 6")

print(board[6] + " | " + board[7] + " | " + board[8] + " 7 | 8 | 9")

print("\n")

# Handle a turn for an arbitrary player

def handle\_turn(player):

# Get position from player

print(player + "'s turn.")

position = input("Choose a position from 1-9: ")

# Whatever the user inputs, make sure it is a valid input, and the spot is open

valid = False

while not valid:

# Make sure the input is valid

while position not in ["1", "2", "3", "4", "5", "6", "7", "8", "9"]:

position = input("Choose a position from 1-9: ")

# Get correct index in our board list

position = int(position) - 1

# Then also make sure the spot is available on the board

if board[position] == "-":

valid = True

else:

print("You can't go there. Go again.")

# Put the game piece on the board

board[position] = player

# Show the game board

display\_board()

# Check if the game is over

def check\_if\_game\_over():

check\_for\_winner()

check\_for\_tie()

# Check to see if somebody has won

def check\_for\_winner():

# Set global variables

global winner

# Check if there was a winner anywhere

row\_winner = check\_rows()

column\_winner = check\_columns()

diagonal\_winner = check\_diagonals()

# Get the winner

if row\_winner:

winner = row\_winner

elif column\_winner:

winner = column\_winner

elif diagonal\_winner:

winner = diagonal\_winner

else:

winner = None

# Check the rows for a win

def check\_rows():

# Set global variables

global game\_still\_going

# Check if any of the rows have all the same value (and is not empty)

row\_1 = board[0] == board[1] == board[2] != "-"

row\_2 = board[3] == board[4] == board[5] != "-"

row\_3 = board[6] == board[7] == board[8] != "-"

# If any row does have a match, flag that there is a win

if row\_1 or row\_2 or row\_3:

game\_still\_going = False

# Return the winner

if row\_1:

return board[0]

elif row\_2:

return board[3]

elif row\_3:

return board[6]

# Or return None if there was no winner

else:

return None

# Check the columns for a win

def check\_columns():

# Set global variables

global game\_still\_going

# Check if any of the columns have all the same value (and is not empty)

column\_1 = board[0] == board[3] == board[6] != "-"

column\_2 = board[1] == board[4] == board[7] != "-"

column\_3 = board[2] == board[5] == board[8] != "-"

# If any row does have a match, flag that there is a win

if column\_1 or column\_2 or column\_3:

game\_still\_going = False

# Return the winner

if column\_1:

return board[0]

elif column\_2:

return board[1]

elif column\_3:

return board[2]

# Or return None if there was no winner

else:

return None

# Check the diagonals for a win

def check\_diagonals():

# Set global variables

global game\_still\_going

# Check if any of the columns have all the same value (and is not empty)

diagonal\_1 = board[0] == board[4] == board[8] != "-"

diagonal\_2 = board[2] == board[4] == board[6] != "-"

# If any row does have a match, flag that there is a win

if diagonal\_1 or diagonal\_2:

game\_still\_going = False

# Return the winner

if diagonal\_1:

return board[0]

elif diagonal\_2:

return board[2]

# Or return None if there was no winner

else:

return None

# Check if there is a tie

def check\_for\_tie():

# Set global variables

global game\_still\_going

# If board is full

if "-" not in board:

game\_still\_going = False

return True

# Else there is no tie

else:

return False

# Flip the current player from X to O, or O to X

def flip\_player():

# Global variables we need

global current\_player

if current\_player == "X":

current\_player = "O"

# Or if the current player was O, make it X

elif current\_player == "O":

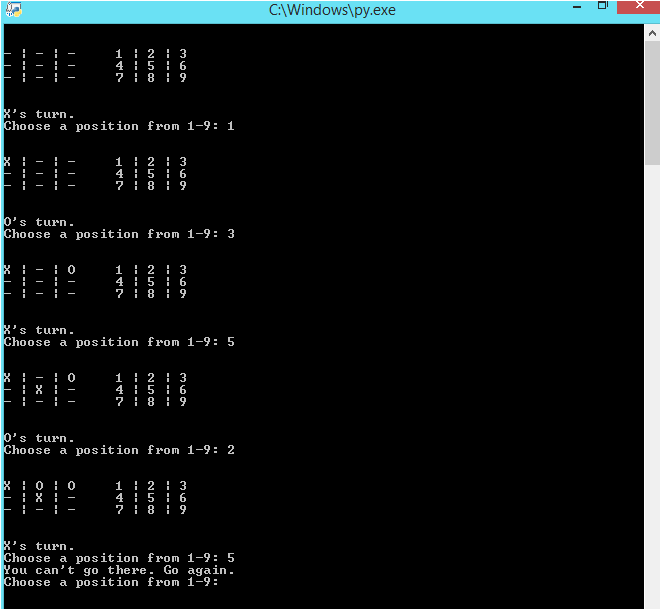
current\_player = "X"

# ------------ Start Execution -------------

# Play a game of tic tac toe

play\_game()

**OUTPUT**



CONCLUSION

Therefore, conclusion of the proposed system is based on user’s need and is user centered. The system is developed in considering all issues related to all user which are included in this system. All attendants can use this. Thus, implementation of Tic-Tac-Toe System is done to help people. Based on the result of this research, it can be concluded: it computes the winner of the game.

---------------------THANKU YOU--------------------------