**DE LA SALLE UNIVERSITY**

2401 Taft Avenue, Manila

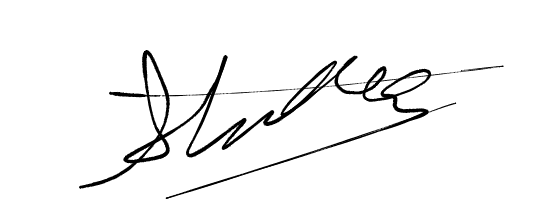
Final Project in

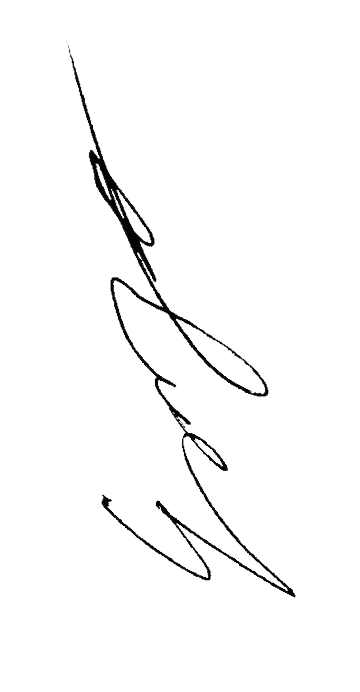
**LBYCPA1**

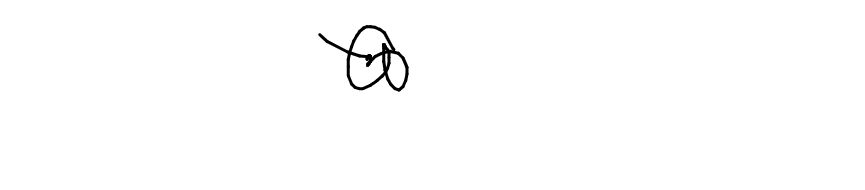
CPE, EQ1

**Alvin and the Tic Tacs**

Submitted by:

Ng, Stanley - 

Ong, Enzo- 

Roldan, Anton -

Ruiz, Ramon

April 14, 2023

**Table of Contents**

**I. Introduction**

**A. Background of the Study…………………………………………………………..3**

**B. Problem Statement……………………………………………………………….…3**

**C. Objectives………………………………………………………………….………….3**

**C.1 General Objective…………………………………………………………………..3**

**C.2 Specific Objectives…………………………………………………………..……..3**

**D. Significance of the Project………………………………………………………….3**

**II. Review of Related Literature………………………………………………………3-4**

**III. Methodology…………………………………………………………………………..5**

**A. Conceptual Framework – IPO Chart (Input-Process-Output-Chart)...............5**

**B. Hierarchy Chart (Refer to the lecture on modules)..........................................6**

**C. Flowchart………………………………………………………………………………7-9**

**D. Pseudocode………………………………………………………………………….10-11**

**IV. Results……………………………………………………………………………12-16**

**V. Discussion of Results………………………………………………………………17**

**VI. Analysis, Conclusion and Future Directives……………………………………17**

**References………………………………………………………………………………….18**

**Appendices**

**A. User’s Manual…………………………………………………………………………19**

**B. Source Code – include comments in your source codes……………………..20-55**

**C. Work breakdown – itemize the work done by each member of the group… 56**

**D. Personal Data Sheet……………………………………………………………….. 57 - 58**

**List of Figures**

**Hiearchy Chart………………………………………………………………………….5**

**Flowcharts…………………………………………………………………………….6-8**

**List Tables**

**IPO Chart………………………………………………………………………………5**

**Project Description**

This program will allow two players to play tic-tac-toe without the need for pen and paper. The first player will select a character from the list, then the second player will select a character different from player one’s. The character chosen will have different markings on the grid. For example, character A will have Xs on the grid while character C will have Os. One person will select where they would place their mark on the grid. Then the second person would place their mark on their selected place on the grid. This would continue until all the spaces on the grid are filled. If there are 3 matching marks lined up together (vertically, diagonally, or horizontally), the code will output that mark as the winner. If the grid is filled without 3 matching marks, the code will output a draw. If the user tries to input their mark on an already filled spot on the grid, the code will reject it and the user has to try again.

1. **Introduction**
   1. **Background of the study**

The students were instructed to create a working Python code based on everything they have learned in class, in which they chose to create an interactable and functioning tic-tac-toe board. In order to make the board game unique, the students implemented the hit film Alvin and the Chipmunks into the board game.

* 1. **Problem Statement**

Tic-tac-toe on its own is a fun game, but the students wanted to add a unique twist to it. The unique twist allows the players to be able to select their own Alvin and the Chipmunks character to play as.

* 1. **Objectives**

1. To promote strategic thinking and planning among the players of the game
2. To give an enjoyable and engaging activity for players of all ages
3. To improve spatial reasoning and social skills
4. To give people a chance to play as their favourite chipmunk
   1. **Significance of the Project**

This project can help develop thinking and social skills for the players, develop cognitive abilities by learning to exercise logic and strategy, and it gives the well-known classic board game, tic-tac-toe, a new light with the unique character selection of the Alvin and the Chipmunks characters.

1. **Review of Related Literature**

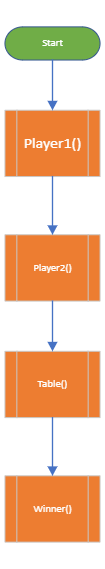
Tic-tac-toe, or “Naughts and Crosses” in other countries, is a board game where there are two players. One player plays as the “O” and the other plays as the “X,” and they each take turns placing their designated symbol on a 3x3 grid, where they must get three of their marks in a row, be it in a straight line, diagonally, or across, with the player with the “X” symbol going first (The Exploratorium, 2000). There are conflicting stories when it comes to the origins of tic-tac-toe. There are some sources that suggest that this game dates as far back as the Egyptians (Franklin, 2022), but it is said that the earliest variation of this game was played in the Roman Empire way back in the First Century BCE, which went by the name “Terni Lapilli,” which translates to “three pebbles at a time” (Gamesver, 2022). The name tic-tac-toe is said to have come from a completely different game that is no longer played, where players who have their eyes closed, throw a pencil down onto a slate with numbers written on it, wherein the score you earn is where the pencil lands on (Veto, 2017).

Alvin and the Chipmunks is a widely popular kids franchise, with three chipmunks at the center of it all, named Alvin, Simon, and Theodore. The series was first a children’s music act, which was created by Ross Bagdasarian Sr. in 1958, which was then created into a television series named “The Alvin Show,” which aired in 1961 (Sherman, 2021). Other versions of the franchise followed, with the film from 2007 leading this franchise to mainstream popularity. The film is about the three previously named chipmunks who are adopted by an aspiring songwriter named David Seville, who discovers their musical abilities (IMDb, 2007). After the release of this live action movie, it inspired people to create “Chipmunk versions” of songs which became a trend during that time, wherein they take a song and speed it up, making it sound like the squeaky voice of the chipmunks were singing it (Gawaran, 2022).

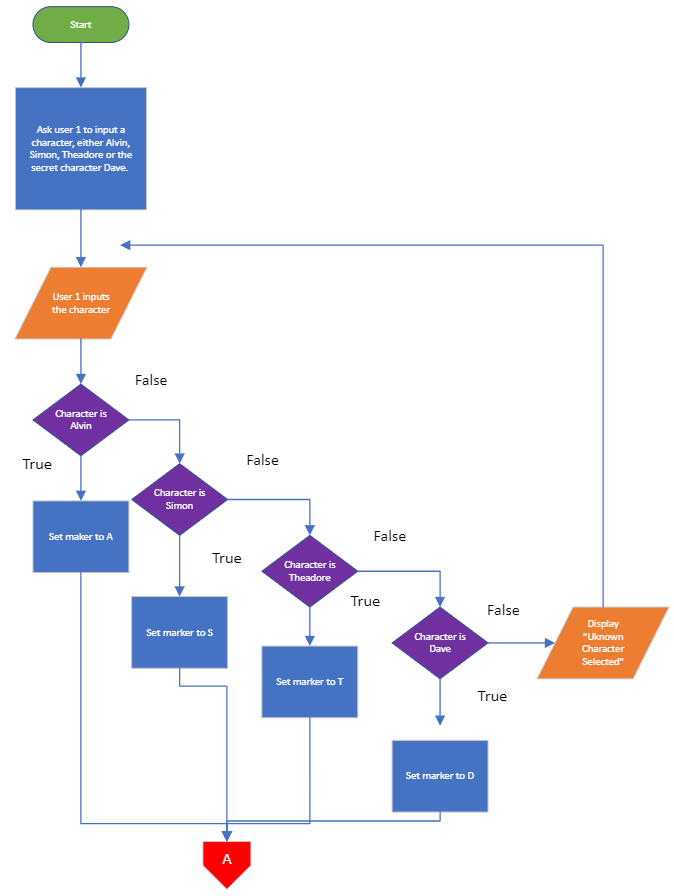
1. **Methodology**
   1. **Conceptual Framework - IPO Chart**

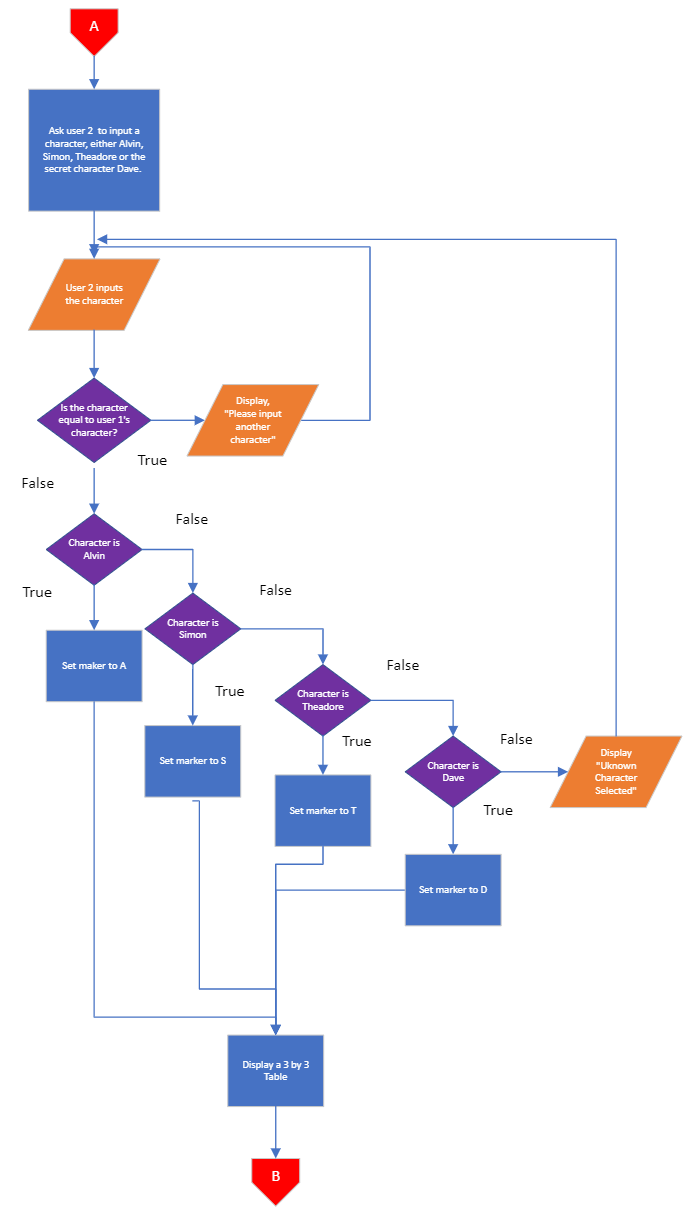
| **Input** | **Process** | **Output** |
| --- | --- | --- |
| The user will input the character of choice as a string (“Alvin”, “Simon” or “Theadore”). A secret character (“Dave”) will not be displayed to the user, but if the user is to input “Dave”, the character will be used. Once the first user chooses, the second user will then be able to choose, but not the same character as the first user. | Player number 1 will be prompted to click on one of the boxes in a 3 by 3 table. The box they click will then show a marker, an A if they chose Alvin, a S if they chose Simon, and a T if they chose Theadore. However, if they chose Dave, the marker will be a D, and they will be allowed to place 2 markers in a row for each turn. Once done, the next player will click on a box and their respective marker will be shown. Both players will then take turns clicking on boxes in the table. A player will win if 3 of their markers connect in either a horizontal, vertical or slanted line. The game will stop once a player wins, or if all the boxes are filled out with no winner. | The code will display, character “is the winner!”. However, if there is a tie, the code will display, “Tie.” |

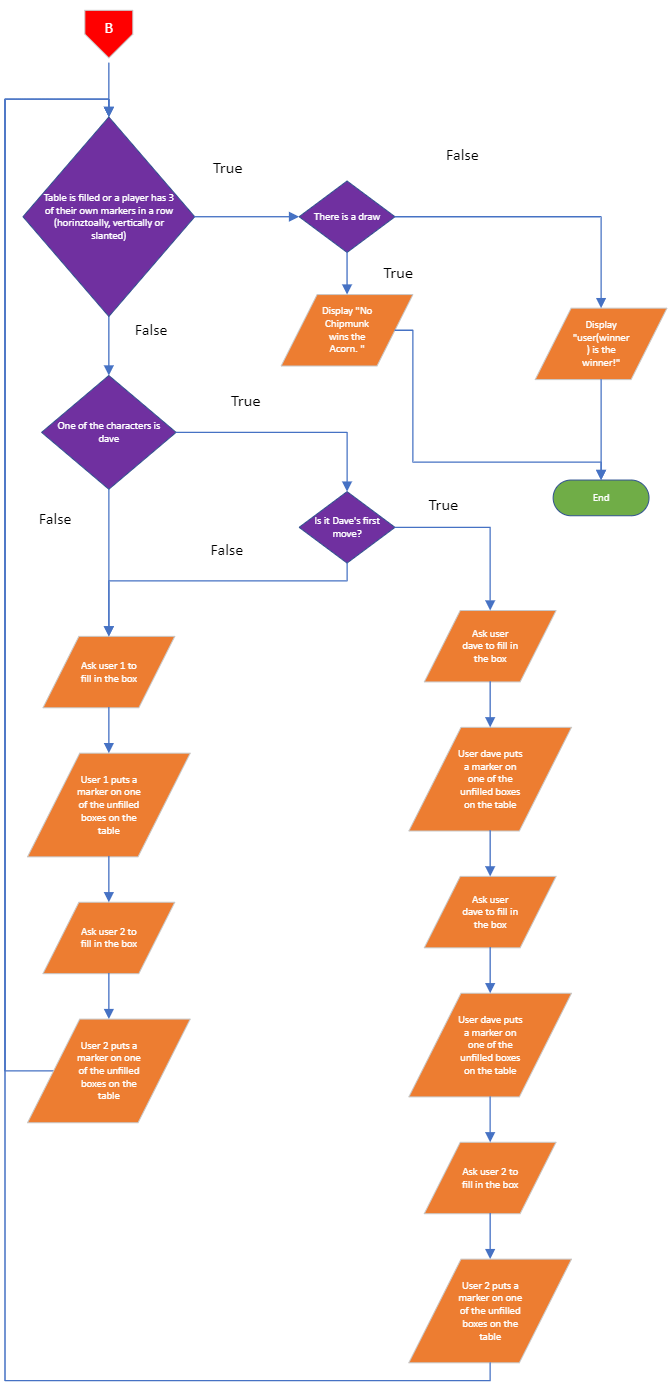
* 1. **Hierarchy Chart**

****

* 1. **Flowchart**







* 1. **Pseudocode**

**Pseudocode:**

Declare character variable user1, user2

Get user1

If user1 = “Alvin” do

Set marker to A

Else If user1 = “Simon” do

Set marker to S

Else If user1 = “Theadore” do

Set marker to T

Else If user1 = “Dave” do

Set marker to D

Else, ask user to input another character again.

End if

Get user2

If user2 == user1, do

Display “Please input another character”

Else, continue

End if.

If user1 = “Alvin” do

Set marker to A

Else If user1 = “Simon” do

Set marker to S

Else If user1 = “Theadore” do

Set marker to T

Else If user1 = “Dave” do

Set marker to D

Else, ask user to input another character again.

End if

Display 3 by 3 table

Declare integer move = 1

Until Table is filled or there are 3 markers in a row, do

If user1 or user2 == “Dave”, do

If move = 1, do

Get user “Dave” input

Store input in table

Get user “Dave” input

Store input in the table

Get user2 input

Store input in table

move = move + 1

Else, continue

Else, do

Get user1 input

Store input in table

Get user2 input

Store input in table

move = move + 1

End if.

End until.

If table shows a draw, do

Display “There is a draw.”

Else, do

Display User(winner) “is the winner.”

End if.

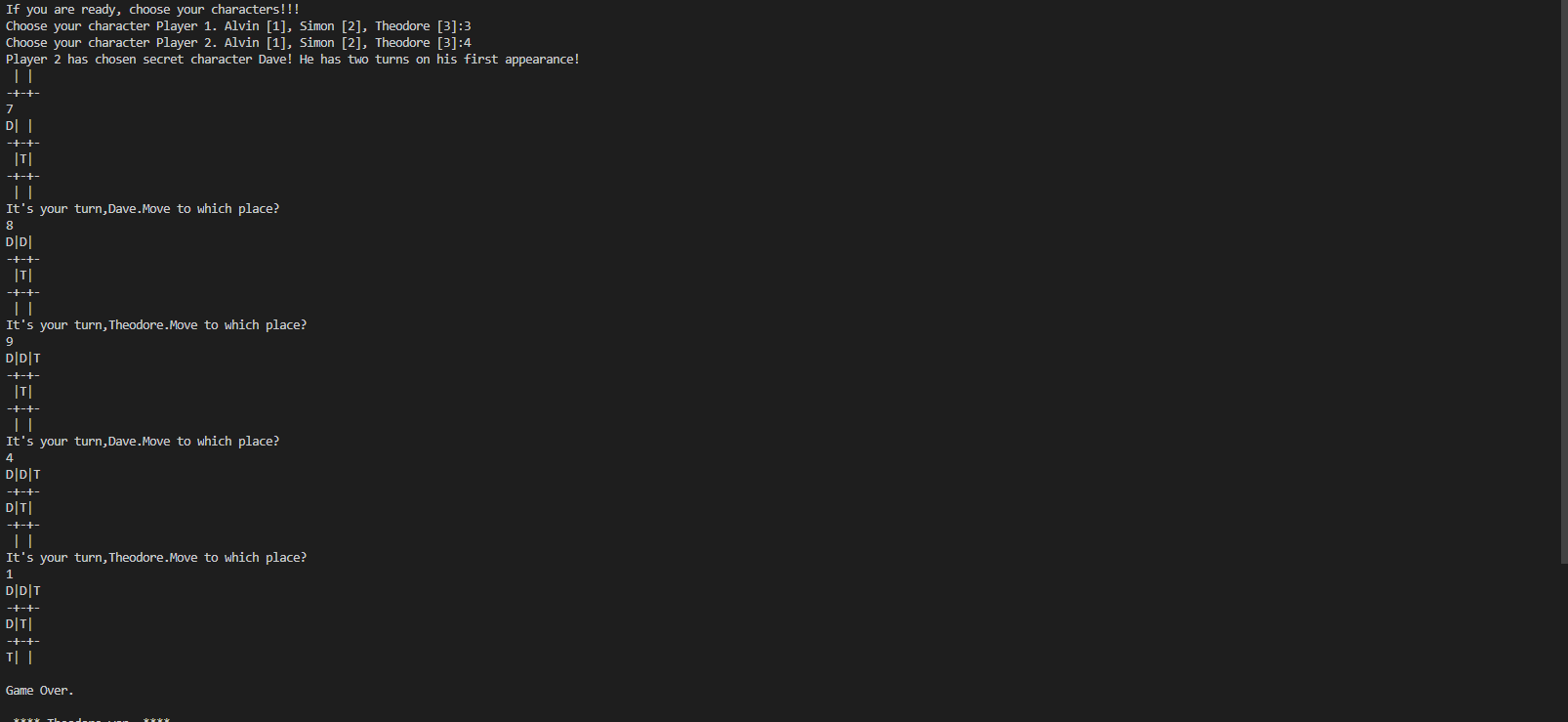
1. **Results**

**Sample Input/Output Dialogue:**

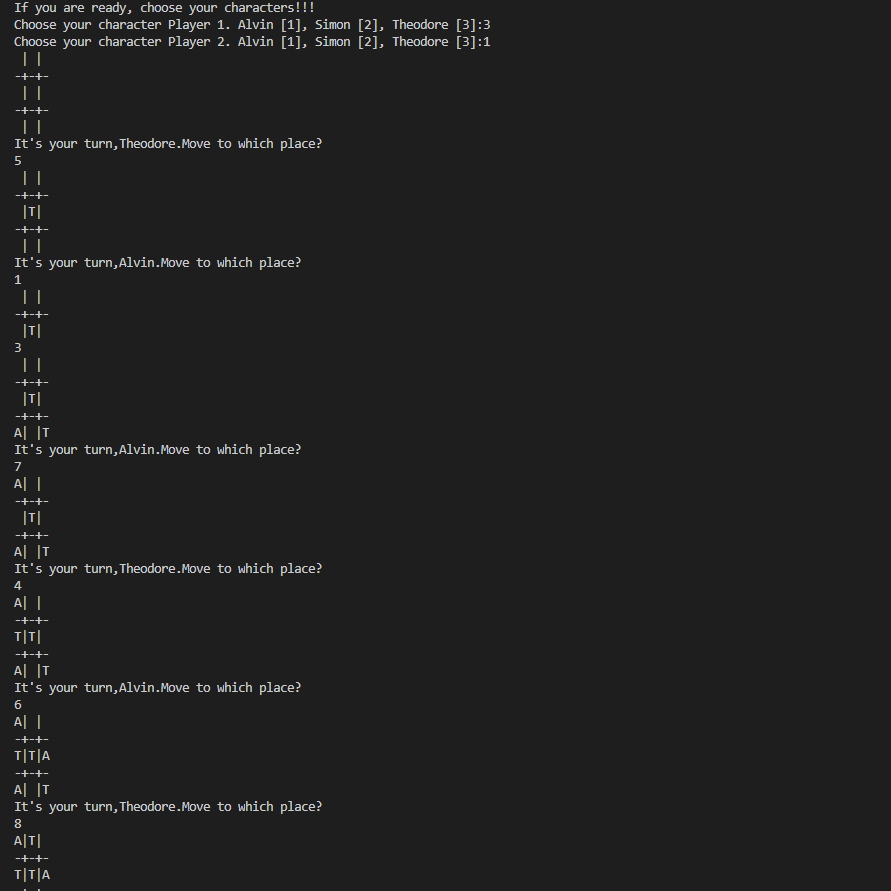
| If you are ready, choose your characters!!!  Choose your character Player 1. Alvin [1], Simon [2], Theodore [3]:3  Choose your character Player 2. Alvin [1], Simon [2], Theodore [3]:4  Player 2 has chosen secret character Dave! He has two turns on his first appearance!  | |  -+-+-  | |  -+-+-  | |  It's your turn,Theodore.Move to which place?  5  | |  -+-+-  |T|  -+-+-  | |  It's your turn,Dave.Move to which place?  7  D| |  -+-+-  |T|  -+-+-  | |  It's your turn,Dave.Move to which place?  8  D|D|  -+-+-  |T|  -+-+-  | |  It's your turn,Theodore.Move to which place?  9  D|D|T  -+-+-  |T|  -+-+-  | |  It's your turn,Dave.Move to which place?  4  D|D|T  -+-+-  D|T|  -+-+-  | |  It's your turn,Theodore.Move to which place?  1  D|D|T  -+-+-  D|T|  -+-+-  T| |  Game Over.  \*\*\*\* Theodore won. \*\*\*\*  Do want to play Again?(y/n)n |
| --- |

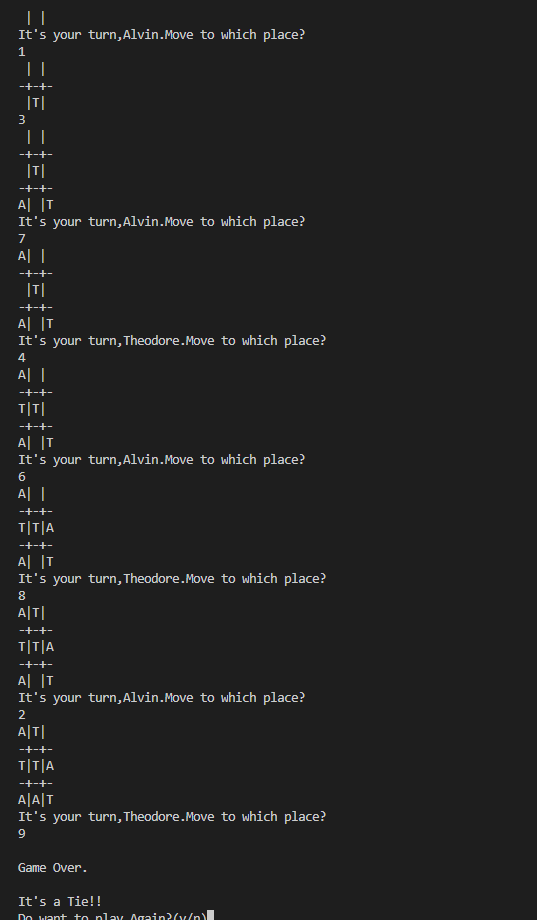
**Screenshots of outputs (At least three sets of data should be tested)**

**Sample 1:**

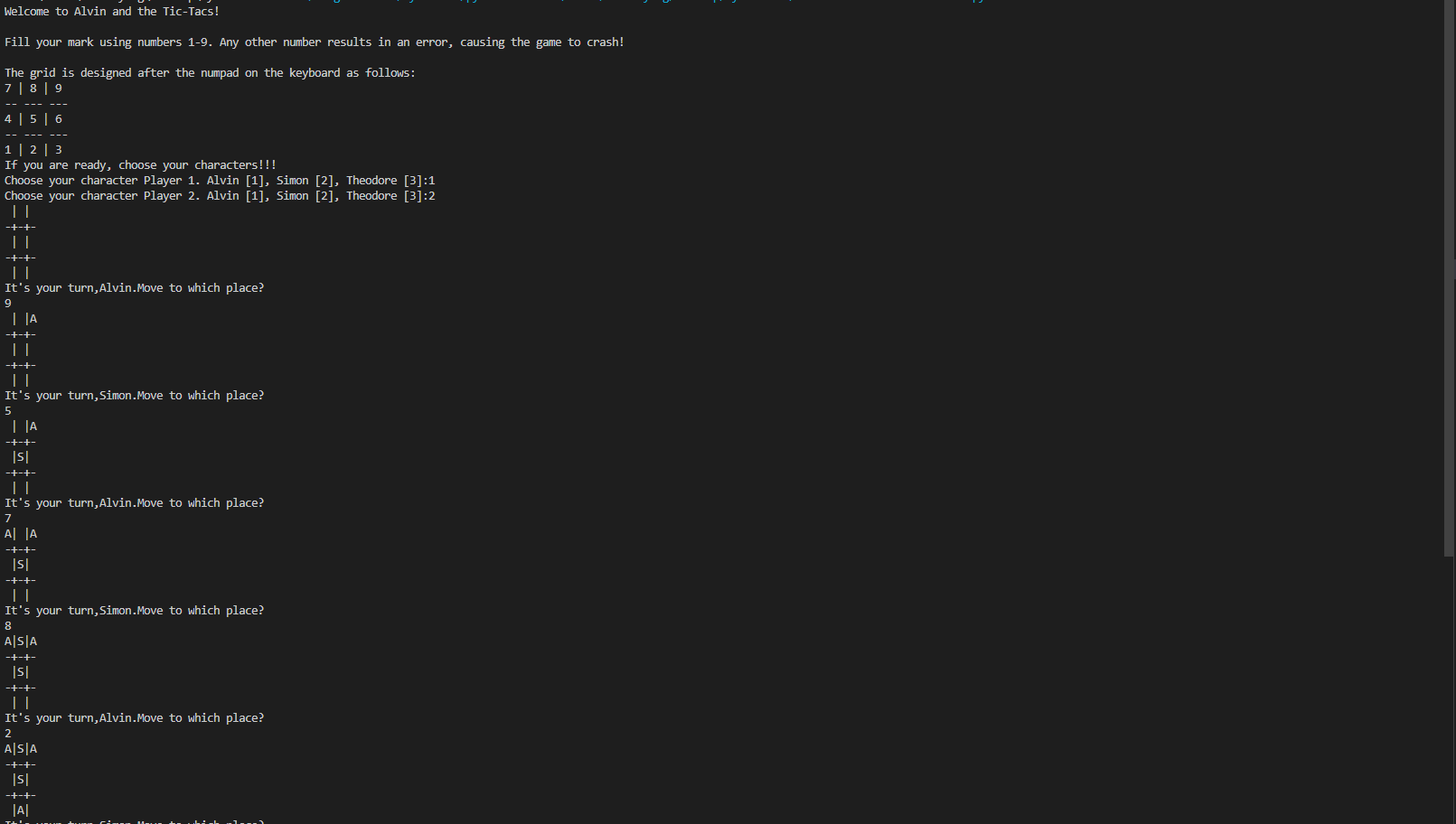
****

**Sample 2:**

****

****

**Sample 3:**

****

****

1. **Discussion of Results**

The Python program made allows 2 users to play tic tac toe with a chipmunk twist. First, both users are to select the character they wish to play. Then, both players play a game of tic tac toe, where their markers represent which character they are playing. However, unlike a normal tic tac toe game, players have the option to pick a secret character, in this case, Dave, the father figure of the chipmunks. Dave, unlike the other characters, has the ability to place two of his own markers at the very beginning of the round, giving him an advantage heading into the game. This tic tac toe game follows the normal rules of tic tac toe, where there is a 3 by 3 table and a winner is selected if a player gets three of his markers in a row. Both players fill in the tic tac toe board using the num pad on their keyboard, with the numbers 1 through 9 representing the 3 by 3 tic tac toe board. Our game allows players to play tic tac toe together, although both players must be present in the same room using the same computer to play, as this tic tac toe program does not have any online capabilities yet.

1. **Analysis, Conclusion, and Further Directives**

The students have successfully made a functioning tic-tac-toe board, complete with a character select function of the Alvin and the Chipmunks franchise. The students have thought of further improvements that could be done to this project, namely allowing online play to allow users to play the game without having to be in the same room as each other. Including a competitive ladder for the future in order for people to have a reason to keep playing, and the possibility of players choosing which characters they want to ban in the competitive scene.

**References:**

Franklin, J. (2022). The History of Tic-Tac-Toe. *PlayorDown*.

<https://playordown.com/blog/the-history-of-tic-tac-toe/>

Gamesver. (2022). Tic-Tac-Toe (Noughts & Crosses): Origins, History, Evolution,…

<https://www.gamesver.com/tic-tac-toe-noughts-crosses-origins-history-evolution/>

Gawaran, A. (2022). Alvin & The Chipmunks: Every Movie in the Franchise, Ranked.

*MovieWeb*. <https://movieweb.com/alvin-and-the-chipmunks-movies/>

IMDb. (2007). Alvin and the Chipmunks (2007). <https://www.imdb.com/title/tt0952640/>

Sherman, A. (2021). Alvin and the Chipmunks owner is looking to sell for about $300 million,

sources say. *CNBC*. [https:// www.cnbc.com/2021/11/02/alvin-and-the-chipmunks-owner-seeks-sale-for-300-million-sources.html](https://www.cnbc.com/2021/11/02/alvin-and-the-chipmunks-owner-seeks-sale-for-300-million-sources.html)

The Exploratorium. (2000). Brain Explorer: Tic Tac Toe.

<https://www.exploratorium.edu/brain_explorer/tictactoe.html>

Veto, M. (2017). What Is the Origin of the Name Tic Tac Toe? *Our Pastimes*.

<https://ourpastimes.com/origin-name-tic-tac-toe-4682851.html>

**Appendices:**

1. **User’s Manual**

Welcome to Alvin and the Tic-Tacs User Manual!

This Tic-Tac-Toe game is a computer program that allows you to play the classic game of Tic-Tac-Toe on your computer. This program allows you to choose from three playable characters: Alvin, Simon or Theodore, each with their own unique marker! Player one may choose anyone from the three chipmunks, but player two cannot be the same character as player one (e.g if player one chooses Alvin, player two cannot be Alvin).

Getting Started:

Download the Tic-Tac-Toe code on your computer.

Run the code in any python integrated environment (Recommended Jupyter Notebook).

Gameplay:

The game is played on a 3x3 grid which resembles the numpad on the keyboard.

Player one chooses their character from Alvin, SImon, or Theodore.

Player one chooses their character from Alvin, SImon, or Theodore, excluding player one’s choice.

Player one will go first and place their character on any square of the grid.

Player two will go next and place their character on any square of the grid.

The game continues until one player wins by getting three of their characters in a row (horizontally, vertically, or diagonally) or the game ends in a tie.

The player can choose to play again or exit the program after the game ends.

Secret Character: Dave

If the player inputs 4, they will choose a secret playable character known as Dave, the father figure of the Three Chipmunks.

Unlike the chipmunks, Dave has two turns on his first appearance, allowing the player using him to gain a headstart on the game.

Only one player can play as Dave

1. **Source Code**

#Implementation of Two Player Tic-Tac-Toe game in Python.

from IPython.display import Image

Image(filename='Chipmunks.png')

def Alvin\_vs\_Simon():

''' We will make the board using dictionary

in which keys will be the location(i.e : top-left,mid-right,etc.)

and initialliy it's values will be empty space and then after every move

we will change the value according to player's choice of move. '''

theBoard = {'7': ' ' , '8': ' ' , '9': ' ' ,

'4': ' ' , '5': ' ' , '6': ' ' ,

'1': ' ' , '2': ' ' , '3': ' ' }

board\_keys = []

for key in theBoard:

board\_keys.append(key)

''' We will have to print the updated board after every move in the game and

thus we will make a function in which we'll define the printBoard function

so that we can easily print the board everytime by calling this function. '''

def printBoard(board):

print(board['7'] + '|' + board['8'] + '|' + board['9'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['1'] + '|' + board['2'] + '|' + board['3'])

# Now we'll write the main function which has all the gameplay functionality.

def game():

turn = 'Alvin'

mark\_alvin = 'A'

mark\_simon = 'S'

count = 0

for i in range(10):

printBoard(theBoard)

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

move = input()

if theBoard[move] == ' ':

if turn == 'Alvin':

theBoard[move] = mark\_alvin

else:

theBoard[move] = mark\_simon

count += 1

else:

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

continue

# Now we will check if player X or O has won,for every move after 5 moves.

if count >= 5:

if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ': # across the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ': # across the bottom

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ': # down the left side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # 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!!")

break

# Now we have to change the player after every move.

if turn =='Alvin':

turn = 'Simon'

else:

turn = 'Alvin'

# Now we will ask if player wants to restart the game or not.

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

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

for key in board\_keys:

theBoard[key] = " "

game()

if \_\_name\_\_ == "\_\_main\_\_":

game()

def Alvin\_vs\_Theodore():

''' We will make the board using dictionary

in which keys will be the location(i.e : top-left,mid-right,etc.)

and initialliy it's values will be empty space and then after every move

we will change the value according to player's choice of move. '''

theBoard = {'7': ' ' , '8': ' ' , '9': ' ' ,

'4': ' ' , '5': ' ' , '6': ' ' ,

'1': ' ' , '2': ' ' , '3': ' ' }

board\_keys = []

for key in theBoard:

board\_keys.append(key)

''' We will have to print the updated board after every move in the game and

thus we will make a function in which we'll define the printBoard function

so that we can easily print the board everytime by calling this function. '''

def printBoard(board):

print(board['7'] + '|' + board['8'] + '|' + board['9'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['1'] + '|' + board['2'] + '|' + board['3'])

# Now we'll write the main function which has all the gameplay functionality.

def game():

turn = 'Alvin'

mark\_alvin = 'A'

mark\_theodore = 'T'

count = 0

for i in range(10):

printBoard(theBoard)

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

move = input()

if theBoard[move] == ' ':

if turn == 'Alvin':

theBoard[move] = mark\_alvin

else:

theBoard[move] = mark\_theodore

count += 1

else:

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

continue

# Now we will check if player X or O has won,for every move after 5 moves.

if count >= 5:

if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ': # across the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ': # across the bottom

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ': # down the left side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # 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!!")

break

# Now we have to change the player after every move.

if turn =='Alvin':

turn = 'Theodore'

else:

turn = 'Alvin'

# Now we will ask if player wants to restart the game or not.

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

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

for key in board\_keys:

theBoard[key] = " "

game()

if \_\_name\_\_ == "\_\_main\_\_":

game()

def Alvin\_vs\_Dave():

''' We will make the board using dictionary

in which keys will be the location(i.e : top-left,mid-right,etc.)

and initialliy it's values will be empty space and then after every move

we will change the value according to player's choice of move. '''

theBoard = {'7': ' ' , '8': ' ' , '9': ' ' ,

'4': ' ' , '5': ' ' , '6': ' ' ,

'1': ' ' , '2': ' ' , '3': ' ' }

board\_keys = []

for key in theBoard:

board\_keys.append(key)

''' We will have to print the updated board after every move in the game and

thus we will make a function in which we'll define the printBoard function

so that we can easily print the board everytime by calling this function. '''

def printBoard(board):

print(board['7'] + '|' + board['8'] + '|' + board['9'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['1'] + '|' + board['2'] + '|' + board['3'])

# Now we'll write the main function which has all the gameplay functionality.

def game():

turn = 'Alvin'

mark\_dave = 'D'

mark\_alvin = 'A'

count = 0

dave\_count = 0

for i in range(10):

printBoard(theBoard)

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

move = input()

if theBoard[move] == ' ':

if turn == 'Dave':

theBoard[move] = mark\_dave

else:

theBoard[move] = mark\_alvin

count += 1

else:

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

continue

# Now we will check if player X or O has won,for every move after 5 moves.

if count >= 5:

if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ': # across the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ': # across the bottom

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ': # down the left side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # 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!!")

break

# Now we have to change the player after every move.

if turn =='Dave':

turn = 'Dave'

if (dave\_count % 2) == 0:

turn = 'Dave'

dave\_count += 1

else:

turn = "Alvin"

else:

turn = 'Dave'

# Now we will ask if player wants to restart the game or not.

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

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

for key in board\_keys:

theBoard[key] = " "

game()

if \_\_name\_\_ == "\_\_main\_\_":

game()

def Simon\_vs\_Alvin():

''' We will make the board using dictionary

in which keys will be the location(i.e : top-left,mid-right,etc.)

and initialliy it's values will be empty space and then after every move

we will change the value according to player's choice of move. '''

theBoard = {'7': ' ' , '8': ' ' , '9': ' ' ,

'4': ' ' , '5': ' ' , '6': ' ' ,

'1': ' ' , '2': ' ' , '3': ' ' }

board\_keys = []

for key in theBoard:

board\_keys.append(key)

''' We will have to print the updated board after every move in the game and

thus we will make a function in which we'll define the printBoard function

so that we can easily print the board everytime by calling this function. '''

def printBoard(board):

print(board['7'] + '|' + board['8'] + '|' + board['9'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['1'] + '|' + board['2'] + '|' + board['3'])

# Now we'll write the main function which has all the gameplay functionality.

def game():

turn = 'Simon'

mark\_alvin = 'A'

mark\_simon = 'S'

count = 0

for i in range(10):

printBoard(theBoard)

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

move = input()

if theBoard[move] == ' ':

if turn == 'Simon':

theBoard[move] = mark\_simon

else:

theBoard[move] = mark\_alvin

count += 1

else:

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

continue

# Now we will check if player X or O has won,for every move after 5 moves.

if count >= 5:

if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ': # across the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ': # across the bottom

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ': # down the left side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # 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!!")

break

# Now we have to change the player after every move.

if turn =='Simon':

turn = 'Alvin'

else:

turn = 'Simon'

# Now we will ask if player wants to restart the game or not.

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

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

for key in board\_keys:

theBoard[key] = " "

game()

if \_\_name\_\_ == "\_\_main\_\_":

game()

def Simon\_vs\_Theodore():

''' We will make the board using dictionary

in which keys will be the location(i.e : top-left,mid-right,etc.)

and initialliy it's values will be empty space and then after every move

we will change the value according to player's choice of move. '''

theBoard = {'7': ' ' , '8': ' ' , '9': ' ' ,

'4': ' ' , '5': ' ' , '6': ' ' ,

'1': ' ' , '2': ' ' , '3': ' ' }

board\_keys = []

for key in theBoard:

board\_keys.append(key)

''' We will have to print the updated board after every move in the game and

thus we will make a function in which we'll define the printBoard function

so that we can easily print the board everytime by calling this function. '''

def printBoard(board):

print(board['7'] + '|' + board['8'] + '|' + board['9'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['1'] + '|' + board['2'] + '|' + board['3'])

# Now we'll write the main function which has all the gameplay functionality.

def game():

turn = 'Simon'

mark\_theodore = 'T'

mark\_simon = 'S'

count = 0

for i in range(10):

printBoard(theBoard)

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

move = input()

if theBoard[move] == ' ':

if turn == 'Simon':

theBoard[move] = mark\_simon

else:

theBoard[move] = mark\_theodore

count += 1

else:

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

continue

# Now we will check if player X or O has won,for every move after 5 moves.

if count >= 5:

if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ': # across the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ': # across the bottom

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ': # down the left side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # 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!!")

break

# Now we have to change the player after every move.

if turn =='Simon':

turn = 'Theodore'

else:

turn = 'Simon'

# Now we will ask if player wants to restart the game or not.

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

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

for key in board\_keys:

theBoard[key] = " "

game()

if \_\_name\_\_ == "\_\_main\_\_":

game()

def Simon\_vs\_Dave():

''' We will make the board using dictionary

in which keys will be the location(i.e : top-left,mid-right,etc.)

and initialliy it's values will be empty space and then after every move

we will change the value according to player's choice of move. '''

theBoard = {'7': ' ' , '8': ' ' , '9': ' ' ,

'4': ' ' , '5': ' ' , '6': ' ' ,

'1': ' ' , '2': ' ' , '3': ' ' }

board\_keys = []

for key in theBoard:

board\_keys.append(key)

''' We will have to print the updated board after every move in the game and

thus we will make a function in which we'll define the printBoard function

so that we can easily print the board everytime by calling this function. '''

def printBoard(board):

print(board['7'] + '|' + board['8'] + '|' + board['9'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['1'] + '|' + board['2'] + '|' + board['3'])

# Now we'll write the main function which has all the gameplay functionality.

def game():

turn = 'Simon'

mark\_dave = 'D'

mark\_simon = 'S'

count = 0

dave\_count = 0

for i in range(10):

printBoard(theBoard)

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

move = input()

if theBoard[move] == ' ':

if turn == 'Dave':

theBoard[move] = mark\_dave

else:

theBoard[move] = mark\_simon

count += 1

else:

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

continue

# Now we will check if player X or O has won,for every move after 5 moves.

if count >= 5:

if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ': # across the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ': # across the bottom

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ': # down the left side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # 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!!")

break

# Now we have to change the player after every move.

if turn =='Dave':

turn = 'Dave'

if (dave\_count % 2) == 0:

turn = 'Dave'

dave\_count += 1

else:

turn = "Simon"

else:

turn = 'Dave'

# Now we will ask if player wants to restart the game or not.

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

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

for key in board\_keys:

theBoard[key] = " "

game()

if \_\_name\_\_ == "\_\_main\_\_":

game()

def Theodore\_vs\_Alvin():

''' We will make the board using dictionary

in which keys will be the location(i.e : top-left,mid-right,etc.)

and initialliy it's values will be empty space and then after every move

we will change the value according to player's choice of move. '''

theBoard = {'7': ' ' , '8': ' ' , '9': ' ' ,

'4': ' ' , '5': ' ' , '6': ' ' ,

'1': ' ' , '2': ' ' , '3': ' ' }

board\_keys = []

for key in theBoard:

board\_keys.append(key)

''' We will have to print the updated board after every move in the game and

thus we will make a function in which we'll define the printBoard function

so that we can easily print the board everytime by calling this function. '''

def printBoard(board):

print(board['7'] + '|' + board['8'] + '|' + board['9'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['1'] + '|' + board['2'] + '|' + board['3'])

# Now we'll write the main function which has all the gameplay functionality.

def game():

turn = 'Theodore'

mark\_theodore = 'T'

mark\_alvin = 'A'

count = 0

for i in range(10):

printBoard(theBoard)

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

move = input()

if theBoard[move] == ' ':

if turn == 'Theodore':

theBoard[move] = mark\_theodore

else:

theBoard[move] = mark\_alvin

count += 1

else:

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

continue

# Now we will check if player X or O has won,for every move after 5 moves.

if count >= 5:

if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ': # across the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ': # across the bottom

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ': # down the left side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # 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!!")

break

# Now we have to change the player after every move.

if turn =='Theodore':

turn = 'Alvin'

else:

turn = 'Theodore'

# Now we will ask if player wants to restart the game or not.

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

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

for key in board\_keys:

theBoard[key] = " "

game()

if \_\_name\_\_ == "\_\_main\_\_":

game()

def Theodore\_vs\_Simon():

''' We will make the board using dictionary

in which keys will be the location(i.e : top-left,mid-right,etc.)

and initialliy it's values will be empty space and then after every move

we will change the value according to player's choice of move. '''

theBoard = {'7': ' ' , '8': ' ' , '9': ' ' ,

'4': ' ' , '5': ' ' , '6': ' ' ,

'1': ' ' , '2': ' ' , '3': ' ' }

board\_keys = []

for key in theBoard:

board\_keys.append(key)

''' We will have to print the updated board after every move in the game and

thus we will make a function in which we'll define the printBoard function

so that we can easily print the board everytime by calling this function. '''

def printBoard(board):

print(board['7'] + '|' + board['8'] + '|' + board['9'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['1'] + '|' + board['2'] + '|' + board['3'])

# Now we'll write the main function which has all the gameplay functionality.

def game():

turn = 'Theodore'

mark\_theodore = 'T'

mark\_simon = 'S'

count = 0

for i in range(10):

printBoard(theBoard)

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

move = input()

if theBoard[move] == ' ':

if turn == 'Theodore':

theBoard[move] = mark\_theodore

else:

theBoard[move] = mark\_simon

count += 1

else:

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

continue

# Now we will check if player X or O has won,for every move after 5 moves.

if count >= 5:

if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ': # across the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ': # across the bottom

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ': # down the left side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # 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!!")

break

# Now we have to change the player after every move.

if turn =='Theodore':

turn = 'Simon'

else:

turn = 'Theodore'

# Now we will ask if player wants to restart the game or not.

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

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

for key in board\_keys:

theBoard[key] = " "

game()

if \_\_name\_\_ == "\_\_main\_\_":

game()

def Theodore\_vs\_Dave():

''' We will make the board using dictionary

in which keys will be the location(i.e : top-left,mid-right,etc.)

and initialliy it's values will be empty space and then after every move

we will change the value according to player's choice of move. '''

theBoard = {'7': ' ' , '8': ' ' , '9': ' ' ,

'4': ' ' , '5': ' ' , '6': ' ' ,

'1': ' ' , '2': ' ' , '3': ' ' }

board\_keys = []

for key in theBoard:

board\_keys.append(key)

''' We will have to print the updated board after every move in the game and

thus we will make a function in which we'll define the printBoard function

so that we can easily print the board everytime by calling this function. '''

def printBoard(board):

print(board['7'] + '|' + board['8'] + '|' + board['9'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['1'] + '|' + board['2'] + '|' + board['3'])

# Now we'll write the main function which has all the gameplay functionality.

def game():

turn = 'Theodore'

mark\_dave = 'D'

mark\_theodore = 'T'

count = 0

dave\_count = 0

for i in range(10):

printBoard(theBoard)

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

move = input()

if theBoard[move] == ' ':

if turn == 'Dave':

theBoard[move] = mark\_dave

else:

theBoard[move] = mark\_theodore

count += 1

else:

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

continue

# Now we will check if player X or O has won,for every move after 5 moves.

if count >= 5:

if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ': # across the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ': # across the bottom

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ': # down the left side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # 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!!")

break

# Now we have to change the player after every move.

if turn =='Dave':

turn = 'Dave'

if (dave\_count % 2) == 0:

turn = 'Dave'

dave\_count += 1

else:

turn = "Theodore"

else:

turn = 'Dave'

# Now we will ask if player wants to restart the game or not.

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

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

for key in board\_keys:

theBoard[key] = " "

game()

if \_\_name\_\_ == "\_\_main\_\_":

game()

def Dave\_vs\_Alvin():

''' We will make the board using dictionary

in which keys will be the location(i.e : top-left,mid-right,etc.)

and initialliy it's values will be empty space and then after every move

we will change the value according to player's choice of move. '''

theBoard = {'7': ' ' , '8': ' ' , '9': ' ' ,

'4': ' ' , '5': ' ' , '6': ' ' ,

'1': ' ' , '2': ' ' , '3': ' ' }

board\_keys = []

for key in theBoard:

board\_keys.append(key)

''' We will have to print the updated board after every move in the game and

thus we will make a function in which we'll define the printBoard function

so that we can easily print the board everytime by calling this function. '''

def printBoard(board):

print(board['7'] + '|' + board['8'] + '|' + board['9'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['1'] + '|' + board['2'] + '|' + board['3'])

# Now we'll write the main function which has all the gameplay functionality.

def game():

turn = 'Dave'

mark\_dave = 'D'

mark\_alvin = 'A'

count = 0

dave\_count = 0

for i in range(10):

printBoard(theBoard)

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

move = input()

if theBoard[move] == ' ':

if turn == 'Dave':

theBoard[move] = mark\_dave

else:

theBoard[move] = mark\_alvin

count += 1

else:

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

continue

# Now we will check if player X or O has won,for every move after 5 moves.

if count >= 5:

if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ': # across the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ': # across the bottom

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ': # down the left side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # 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!!")

break

# Now we have to change the player after every move.

if turn =='Dave':

turn = 'Dave'

if (dave\_count % 2) == 0:

turn = 'Dave'

dave\_count += 1

else:

turn = "Alvin"

else:

turn = 'Dave'

# Now we will ask if player wants to restart the game or not.

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

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

for key in board\_keys:

theBoard[key] = " "

game()

if \_\_name\_\_ == "\_\_main\_\_":

game()

def Dave\_vs\_Simon():

''' We will make the board using dictionary

in which keys will be the location(i.e : top-left,mid-right,etc.)

and initialliy it's values will be empty space and then after every move

we will change the value according to player's choice of move. '''

theBoard = {'7': ' ' , '8': ' ' , '9': ' ' ,

'4': ' ' , '5': ' ' , '6': ' ' ,

'1': ' ' , '2': ' ' , '3': ' ' }

board\_keys = []

for key in theBoard:

board\_keys.append(key)

''' We will have to print the updated board after every move in the game and

thus we will make a function in which we'll define the printBoard function

so that we can easily print the board everytime by calling this function. '''

def printBoard(board):

print(board['7'] + '|' + board['8'] + '|' + board['9'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['1'] + '|' + board['2'] + '|' + board['3'])

# Now we'll write the main function which has all the gameplay functionality.

def game():

turn = 'Dave'

mark\_dave = 'D'

mark\_simon = 'S'

count = 0

dave\_count = 0

for i in range(10):

printBoard(theBoard)

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

move = input()

if theBoard[move] == ' ':

if turn == 'Dave':

theBoard[move] = mark\_dave

else:

theBoard[move] = mark\_simon

count += 1

else:

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

continue

# Now we will check if player X or O has won,for every move after 5 moves.

if count >= 5:

if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ': # across the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ': # across the bottom

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ': # down the left side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # 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!!")

break

# Now we have to change the player after every move.

if turn =='Dave':

turn = 'Dave'

if (dave\_count % 2) == 0:

turn = 'Dave'

dave\_count += 1

else:

turn = "Simon"

else:

turn = 'Dave'

# Now we will ask if player wants to restart the game or not.

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

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

for key in board\_keys:

theBoard[key] = " "

game()

if \_\_name\_\_ == "\_\_main\_\_":

game()

def Dave\_vs\_Theodore():

''' We will make the board using dictionary

in which keys will be the location(i.e : top-left,mid-right,etc.)

and initialliy it's values will be empty space and then after every move

we will change the value according to player's choice of move. '''

theBoard = {'7': ' ' , '8': ' ' , '9': ' ' ,

'4': ' ' , '5': ' ' , '6': ' ' ,

'1': ' ' , '2': ' ' , '3': ' ' }

board\_keys = []

for key in theBoard:

board\_keys.append(key)

''' We will have to print the updated board after every move in the game and

thus we will make a function in which we'll define the printBoard function

so that we can easily print the board everytime by calling this function. '''

def printBoard(board):

print(board['7'] + '|' + board['8'] + '|' + board['9'])

print('-+-+-')

print(board['4'] + '|' + board['5'] + '|' + board['6'])

print('-+-+-')

print(board['1'] + '|' + board['2'] + '|' + board['3'])

# Now we'll write the main function which has all the gameplay functionality.

def game():

turn = 'Dave'

mark\_dave = 'D'

mark\_theodore = 'T'

count = 0

dave\_count = 0

for i in range(10):

printBoard(theBoard)

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

move = input()

if theBoard[move] == ' ':

if turn == 'Dave':

theBoard[move] = mark\_dave

else:

theBoard[move] = mark\_theodore

count += 1

else:

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

continue

# Now we will check if player X or O has won,for every move after 5 moves.

if count >= 5:

if theBoard['7'] == theBoard['8'] == theBoard['9'] != ' ': # across the top

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['4'] == theBoard['5'] == theBoard['6'] != ' ': # across the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['2'] == theBoard['3'] != ' ': # across the bottom

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['4'] == theBoard['7'] != ' ': # down the left side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['2'] == theBoard['5'] == theBoard['8'] != ' ': # down the middle

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['3'] == theBoard['6'] == theBoard['9'] != ' ': # down the right side

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['7'] == theBoard['5'] == theBoard['3'] != ' ': # diagonal

printBoard(theBoard)

print("\nGame Over.\n")

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

break

elif theBoard['1'] == theBoard['5'] == theBoard['9'] != ' ': # 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!!")

break

break

# Now we have to change the player after every move.

if turn =='Dave':

turn = 'Dave'

if (dave\_count % 2) == 0:

turn = 'Dave'

dave\_count += 1

else:

turn = "Theodore"

else:

turn = 'Dave'

# Now we will ask if player wants to restart the game or not.

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

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

for key in board\_keys:

theBoard[key] = " "

game()

if \_\_name\_\_ == "\_\_main\_\_":

game()

print("Welcome to Alvin and the Tic-Tacs!\n")

print("Fill your mark using numbers 1-9. Any other number results in an error, causing the game to crash!\n")

print("The grid is designed after the numpad on the keyboard as follows:")

print("7 | 8 | 9")

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

print("4 | 5 | 6")

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

print("1 | 2 | 3")

print("If you are ready, choose your characters!!!")

player1 = int(input("Choose your character Player 1. Alvin [1], Simon [2], Theodore [3]:"))

player2 = int(input("Choose your character Player 2. Alvin [1], Simon [2], Theodore [3]:"))

while True:

if player1 == 1 and player2 == 2:

Alvin\_vs\_Simon()

break

elif player1 == 1 and player2 == 3:

Alvin\_vs\_Theodore()

break

elif player1 == 1 and player2 == 4:

print("Player 2 has chosen secret character Dave! He has two turns on his first appearance!")

Alvin\_vs\_Dave()

break

elif player1 == 1 and player2 == 1:

print("This character is taken. Please choose another character.")

player2 = int(input("Choose your character Player 2. Alvin [1], Simon [2], Theodore [3]:"))

continue

elif player1 == 2 and player2 == 1:

Simon\_vs\_Alvin()

break

elif player1 == 2 and player2 == 3:

Simon\_vs\_Theodore()

break

elif player1 == 2 and player2 == 4:

print("Player 2 has chosen secret character Dave! He has two turns on his first appearance!")

Simon\_vs\_Dave()

break

elif player1 == 2 and player2 == 2:

print("This character is taken. Please choose another character.")

player2 = int(input("Choose your character Player 2. Alvin [1], Simon [2], Theodore [3]:"))

continue

elif player1 == 3 and player2 == 1:

Theodore\_vs\_Alvin()

break

elif player1 == 3 and player2 == 2:

Theodore\_vs\_Simon()

break

elif player1 == 3 and player2 == 4:

print("Player 2 has chosen secret character Dave! He has two turns on his first appearance!")

Theodore\_vs\_Dave()

break

elif player1 == 3 and player2 == 3:

print("This character is taken. Please choose another character.")

player2 = int(input("Choose your character Player 2. Alvin [1], Simon [2], Theodore [3]:"))

continue

elif player1 == 4 and player2 == 1:

print("Player 1 has chosen secret character Dave! He has two turns on his first appearance!")

Dave\_vs\_Alvin()

break

elif player1 == 4 and player2 == 2:

print("Player 1 has chosen secret character Dave! He has two turns on his first appearance!")

Dave\_vs\_Simon()

break

elif player1 == 4 and player2 == 3:

print("Player 1 has chosen secret character Dave! He has two turns on his first appearance!")

Dave\_vs\_Theodore()

break

elif player1 == 4 and player2 == 4:

print("This character is taken. Please choose another character.")

player2 = int(input("Choose your character Player 2. Alvin [1], Simon [2], Theodore [3]:"))

continue

else:

print("Error. Characters of Player 1 or Player 2 not found.")

player1 = int(input("Choose your character Player 1. Alvin [1], Simon [2], Theodore [3]:"))

player2 = int(input("Choose your character Player 2. Alvin [1], Simon [2], Theodore [3]:"))

continue

1. **Work Breakdown**

| Student Name | Tasks Assigned | Percentage of the Work Contributed |
| --- | --- | --- |
| **Stanley Ng** | Made the IPO chart, pseudocode, flowchart, and discussion of results. | **33%** |
| **Enzo Ong** | Wrote the Code, User’s Manual, Objectives | **33%** |
| **Anton Roldan** | Wrote the background of the study, problem statement, significance of the project, review of related literature, Analysis, Conclusion, Further Directives, and References. | **33%** |

1. **Personal Data Sheet**

Personal data sheet of each member with 2x2 ID picture, you may opt to limit the information

to appear in your datasheet to those which you are comfortable sharing.





