MINI PROJECT – ROLLING DICE SIMULATION

Group Members:

Aditya Singh (7106)

Dhruv Raj Singh (71)

Gagan Raghav (71)

Problem Statement:

To make a Tic-tac-toe game. **Tic-tac-toe** (American English), noughts and crosses (British English), or Xs and Os is a paper-and-pencil game for two players, X and O, who 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 is the winner.

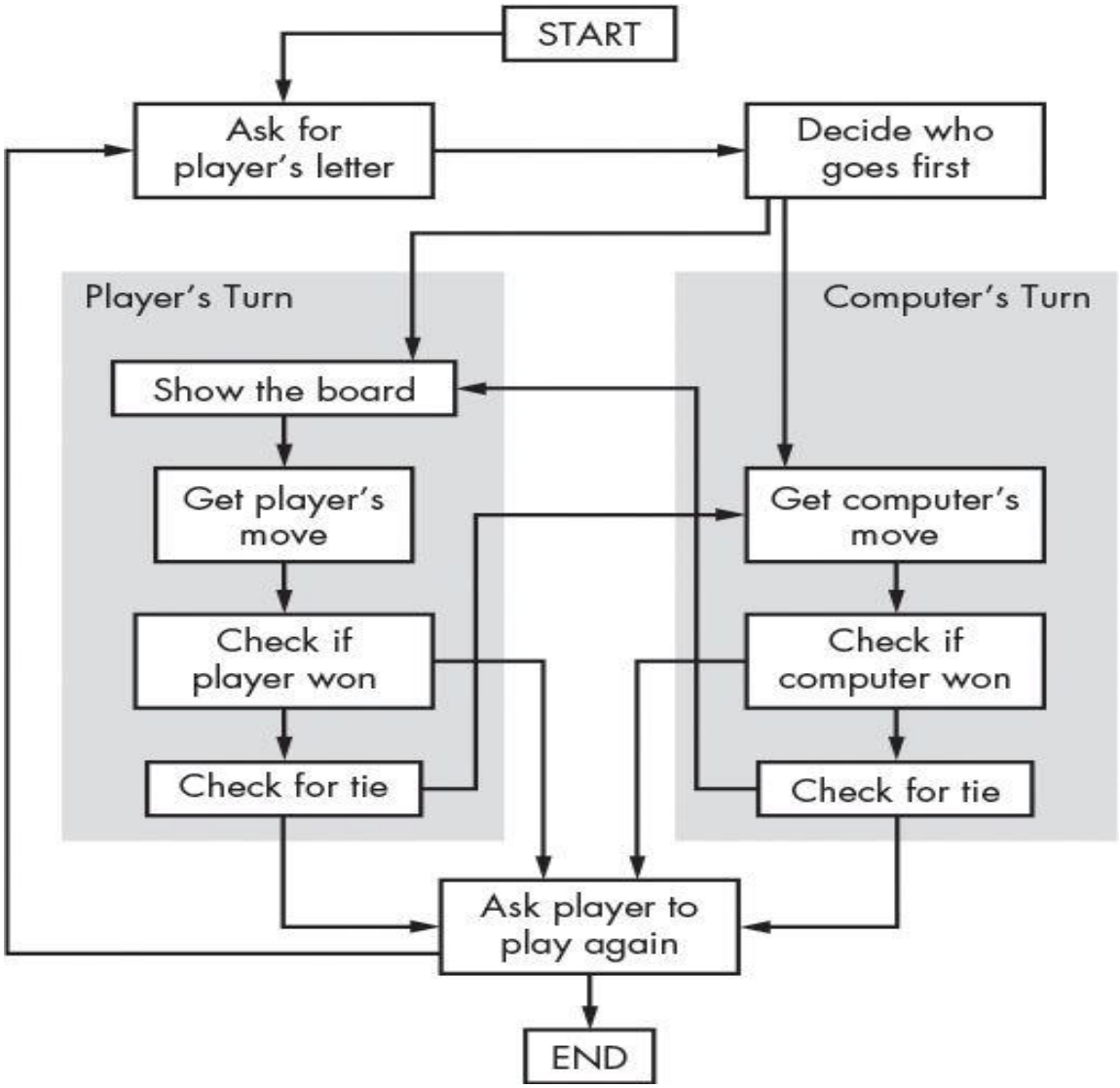
Approach:

- Create a landing page containing selection buttons: Single-player or multiplayer.
- Create a game board containing nine tiles to play the game along with other details (i.e. playing with a system or another player, whose turn etc.).
- Allow the player to press the tile and check the status of the game (i.e. Tie game, any one of the players won the match or the game is still running).
- Display the message, of who won the match.

Pre-requisites:

The major topics that must be refreshed before creating and building the tic tac toe Python project are as follows:

- Basic concepts of Python
- User define functions in Python
- Loops in Python (for loop, if else, if elif else)
- try-exception block in Python
- while loop of Python



Code Explanation:

1. The code starts by importing the tkinter package.
2. This is a library that allows us to create graphical user interfaces in Python.
3. Next, we import the messagebox function from this same library.
4. The messagebox function creates a window with an OK button and text input field for the player to enter their move.

5. The next line of code sets up our variables: Player1 = ‘X’, stop_game = False.
6. These are global variables which means they can be accessed anywhere in the program without having to use parentheses or any other special syntax (e. g., if Player1 == “X” and states[r] == 0).
7. Global variables are often used when you want your program’s logic to be able to access all parts of it without needing extra lines of code for each part (e.g., checking if Player1 == “O” and states[r] == 0).
8. Next, we have two functions: clicked(r,c) and check_if_win().
9. The first one checks whether player X has won or not based on what state r is currently in; if so, then it changes state r back into X; otherwise, it changes state r into O depending on who won last time.
10. The code is a function that checks if the player has won or lost.