Tic Tac Toe game

Game description: Tic-tac-toe, or Xs and Os is a game for two players who take turns marking the spaces in a three-by-three grid with X or O. The player who succeeds in placing three of their marks in a horizontal, vertical, or diagonal row is the winner. Tic-tac-toe is played on a three-by-three grid by two players, who alternately place the marks X and O in one of the nine spaces in the grid.

Example: In the following example, the first player (X) wins the game in seven steps:

Implement Tic-tac-toe in python against computer (based on simple artificial intelligence) according to the following rules:

- Use **Tkinter** to create a **3x3 board**, a **game status label** and **a button for quitting** the game.
- Determine randomly which player goes first. Announce it using the game status label.
- Ask player to select unoccupied positions for the next move. This should be done using the game status label. The human player **selects the position by clicking**. Make sure the selected board position is available.
- The computer should use the following AI to choose its next move:
    - ❖ First, check if there exists a single move such that the computer can win the game. Otherwise, go to the second step.
    - ❖ Second, check if there exists a single move for the player that will cause the computer to lose the game. If there is, the computer should move there to block the player. Otherwise, go to the third step.
    - ❖ Third, check if any of the corner spaces (spaces 0, 2, 6, or 8) are free. If no corner space is free, then go to the fourth step.
    - ❖ Fourth, check if the center is free. If so, move there. If it isn't, then go to the fifth step.
    - ❖ Fifth, move on any of the side pieces (spaces 1, 3, 5, or 7).
- Continue till a player wins or the game ends in tie.
- Depending on the outcome, announce the winner or the tie.

Important! The game should be implemented using OOP concepts. Procedural implementations are not accepted.