In [8]:

```python
#KETHARNATH R 111723102089 CSE C EX.NO. 3

import numpy as np
import random
from time import sleep

def create_board():
    """Create a 3x3 board initialized with zeros."""
    return np.zeros((3, 3), dtype=int)

def possibilities(board):
    """Returns available positions on the board."""
    return [(i, j) for i in range(3) for j in range(3) if board[i, j] == 0]

def random_place(board, player):
    """Places player's move randomly in an available spot."""
    selection = possibilities(board)
    if selection:
        current_loc = random.choice(selection)
        board[current_loc[0], current_loc[1]] = player  # Fixed indexing
    return board

def check_win(board, player):
    """Check if the given player has won the game."""
    return (
        np.any(np.all(board == player, axis=0)) or  # Check columns
        np.any(np.all(board == player, axis=1)) or  # Check rows
        np.all(np.diag(board) == player) or         # Check main diagonal
        np.all(np.diag(np.fliplr(board)) == player) # Check secondary diagonal
    )

def evaluate(board):
    """Evaluate game state: returns 1 if player 1 wins, 2 if player 2 wins, 0 if
    for player in [1, 2]:
        if check_win(board, player):
            return player  # Player wins
    return -1 if np.all(board != 0) else 0  # Draw if board is full, else contin

def play_game():
    """Simulates the game with two players randomly placing moves."""
    board = create_board()
    winner, counter = 0, 1
    print("Initial Board:\n", board)
    sleep(1)

    while winner == 0:
        for player in [1, 2]:
            board = random_place(board, player)
            print(f"\nBoard after {counter} move(s) by Player {player}:")
            print(board)
            sleep(1)
            counter += 1
            winner = evaluate(board)
            if winner != 0:  # Stop game if we have a winner or a draw
                break

    if winner == -1:
        print("\nGame ended in a draw!")
    else:
        print(f"\nWinner is: Player {winner}!")
    return winner

# Driver Code
if __name__ == "__main__":
    play_game()
```

```
Initial Board:
 [[0 0 0]
 [0 0 0]
 [0 0 0]]

Board after 1 move(s) by Player 1:
[[0 1 0]
 [0 0 0]
 [0 0 0]]

Board after 2 move(s) by Player 2:
[[0 1 0]
 [0 0 2]
 [0 0 0]]

Board after 3 move(s) by Player 1:
[[0 1 0]
 [1 0 2]
 [0 0 0]]

Board after 4 move(s) by Player 2:
[[0 1 0]
 [1 2 2]
 [0 0 0]]

Board after 5 move(s) by Player 1:
[[0 1 0]
 [1 2 2]
 [1 0 0]]

Board after 6 move(s) by Player 2:
[[0 1 2]
 [1 2 2]
 [1 0 0]]

Board after 7 move(s) by Player 1:
[[0 1 2]
 [1 2 2]
 [1 1 0]]

Board after 8 move(s) by Player 2:
[[2 1 2]
 [1 2 2]
 [1 1 0]]

Board after 9 move(s) by Player 1:
[[2 1 2]
 [1 2 2]
 [1 1 1]]

Winner is: Player 1!
```

In [ ]: