

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import random

# tworzenie tablicy gry oraz zmiennych
def generate_board(size):
    # funkcja generująca tablicę gry
    item = []
    board = []
    x = 65 + size
    for n in range(65, x):
        for i in range(1, size + 1):
            item.append(chr(n) + str(i))
        board.append(item)
        item = []

    choice = []
    for row in board:
        for item in row:
            choice.append(item)
    return (board, choice)

board = generate_board(3)[0]
choice = generate_board(3)[1]
turn = 1
max_turn_count = 9
o = "O "
x = "X "

def print_board(tab):
    # funkcja wyświetlająca tablicę
    for row in tab:
        print(" ".join(row))

def check_win(board, sign):
    # funkcja sprawdzania wygranej

    win = [sign, sign, sign]
    size = len(board)

    for row in board:
        if row == win:
            return True

    a = []
    b = []
    c = []
    for j in range(0, size):
        for row in board:
            c.append(row[j])
        b.append(board[j][j])
        a.append(board[j][size-1-j])
        if c == win or b == win or a == win:
            return True
    c = []

def remove_used_move(move, sign):
    # funkcja usuwająca wykonany ruch z możliwych ruchów (ze zmiennej choice)
    for i1, j in enumerate(board):
        for i2, item in enumerate(j):
            if item == move:
                board[i1][i2] = sign
                choice.remove(move)

# początek gry
print("Witaj w grze kółko i krzyżyk!")
print_board(board)

while turn < max_turn_count:
    # gracz stawia kółko
    g_move = input("Postaw kółko wpisując nazwę pola: ").upper()

    while g_move not in choice:
        g_move = input("Błędne pole, postaw kółko jeszcze raz: ").upper()

    remove_used_move(g_move, o)
    print_board(board)
    check_win(board, o)

    if check_win(board, o):
        print("Wygrałeś!")
        break

    if len(choice) == 0:
        print("Remis!")
        break

    # komputer stawia krzyżyk
    print("Komputer stawia krzyżyk:")
    c_move = random.choice(choice)
    while c_move == o:
        c_move = random.choice(choice)

    remove_used_move(c_move, x)
    print_board(board)
    check_win(board, x)

    if check_win(board, x):
        print("Przegrałeś!")
        break

    turn += 1

print("Koniec gry!")
```