```
# coding=utf-8
"""
tic tac toe
dave koning
2015
todo:
1. bij winst wordt de winnende letter vervangen door een smiley, pas het programma aan dat alleen de letters op de 'win-lijn' smiley's wo
2. de functie bedenk_computer_zet doet nu een willekeurige zet, kun je het programma zo aanpassen dat hij wat beter speelt?
"""

import random
import readline


def bedenk_computer_zet(board, computer_letter):
    """
    @type board: list
    @type computer_letter: str
    @return: None
    """
    return doe_willekeurige_zet_van_de_lijst_met_mogelijkheden(board, [1, 2, 3, 4, 5, 6, 7, 8, 9])


def controleer_bord_vol(board):
    """
    @type board: list
    @return: None
    """
    for i in range(10):
        if i > 0:
            if vrije_plek(board, i):
                return False

    return True


def doe_willekeurige_zet_van_de_lijst_met_mogelijkheden(board, moves_list):
    """
    @type board: list
    @type moves_list: list
    @return: None
    """
    possible_moves = []

    for i in moves_list:
        if vrije_plek(board, i):
            possible_moves.append(i)

    if len(possible_moves) != 0:
        return random.choice(possible_moves)
    else:
        return None


def doe_zet(board, letter, move):
    """
    @type board: list
    @type letter: str
    @type move: int
    @return: None
    """
    board[move] = letter


def feest():
    """
    feest
    """
    print()
    print(b'\xf0\x9f\x8e\x88 \xf0\x9f\x92\xa5  \xe2\x9c\xa8 \xf0\x9f\x8e\x88 \xf0\x9f\x92\xa5  \xe2\x9c\xa8 '.decode())
    print()


def invoer_speler_letter():
    """
    invoer_speler_letter
    """
    letter = ''

    while not (letter == 'X' or letter == 'O'or letter == 'S'):
        welkom = '\033[0;34mWilt u X \033[34mof\033[0m O \033[0;34mzijn:\033[0m'
        print(welkom.replace("X", "\033[0;36mX\033[0m").replace("O", "\033[0;95mO\033[0m"))
        try:
            letter = input().upper()

            if letter.strip() == "S":
                exit(1)
        except KeyboardInterrupt:
            exit(1)

    if letter == 'X':
        return ['X', 'O']
    else:
        return ['O', 'X']


def is_er_een_winnaar(bo, le):
    """
    @type bo: list
    @type le: str
    @return: None
    """
```

```
    # bekijk alle win mogelijkheden
    return ((bo[7] == le and bo[8] == le and bo[9] == le) or  # across the top
            (bo[4] == le and bo[5] == le and bo[6] == le) or  # across the middle
            (bo[1] == le and bo[2] == le and bo[3] == le) or  # across the bottom
            (bo[7] == le and bo[4] == le and bo[1] == le) or  # down the left side
            (bo[8] == le and bo[5] == le and bo[2] == le) or  # down the middle
            (bo[9] == le and bo[6] == le and bo[3] == le) or  # down the right side
            (bo[7] == le and bo[5] == le and bo[3] == le) or  # diagonal
            (bo[9] == le and bo[5] == le and bo[1] == le))  # diagonal


def maak_kopie_bord(board):
    """
    @type board: list
    @return: None
    """
    dupe_board = []

    for i in board:
        dupe_board.append(i)

    return dupe_board


def main():
    """
    main
    """
    text_naar_scherm(b"\xf0\x9f\x98\x8e\033[95m  Boter Kaas en Eieren! \033[0m\xf0\x9f\x98\x8e".decode())
    text_naar_scherm()

    while True:

        # reset
        het_bord = [' '] * 10

        # teken_bord(het_bord)
        player_letter, computer_letter = invoer_speler_letter()
        turn = wie_gaat_eerst()

        if turn == "player":
            turnvis = "U"
        else:
            turnvis = "de computer"

        text_naar_scherm(turnvis + ' mag beginnen.')
        game_is_playing = True

        while game_is_playing:
            if turn == 'player':
                teken_bord(het_bord)
                move = vraag_speler_zet(het_bord)
                doe_zet(het_bord, player_letter, move)

                if is_er_een_winnaar(het_bord, player_letter):
                    teken_bord(het_bord, player_letter)
                    feest()
                    text_naar_scherm(b'U heeft gewonnen! \xf0\x9f\x98\x8e'.decode())
                    feest()
                    game_is_playing = False
                else:
                    if controleer_bord_vol(het_bord):
                        teken_bord(het_bord)
                        text_naar_scherm(b'Gelijkspel! \xf0\x9f\x98\x90'.decode())
                        break
                    else:
                        turn = 'computer'
            else:
                move = bedenk_computer_zet(het_bord, computer_letter)
                doe_zet(het_bord, computer_letter, move)

                if is_er_een_winnaar(het_bord, computer_letter):
                    teken_bord(het_bord, computer_letter)
                    text_naar_scherm(b'U heeft verloren \xf0\x9f\x98\xa2'.decode())
                    game_is_playing = False
                else:
                    if controleer_bord_vol(het_bord):
                        teken_bord(het_bord)
                        text_naar_scherm(b'Gelijkspel! \xf0\x9f\x98\x90'.decode())
                        break
                    else:
                        turn = 'player'

        if not nogmaal_spelen_vraag():
            text_naar_scherm(b"Tot ziens. \xf0\x9f\x98\x80".decode())
            break


def nogmaal_spelen_vraag():
    """
    nogmaal_spelen_vraag
    """
    print()
    try:
        text_naar_scherm('Nog een keer? (ja of nee)')
        return input().lower().startswith('j')
    except KeyboardInterrupt:
        pass


def teken_bord(board, winner=None):
    """
    @type board: str
```

```
    @type winner: str, None
    @return: None
    """
    num = []

    for i in range(10):
        num.append("\033[37m" + str(i) + "\033[0m")

    boardstring = num[7] + '  |' + num[8] + '  |' + num[9] + '\n'
    boardstring += ' ' + board[7] + ' | ' + board[8] + ' | ' + board[9] + '\n'
    boardstring += '   |   |\n'
    boardstring += '__________\n'
    boardstring += num[4] + '  |' + num[5] + '  |' + num[6] + '\n'
    boardstring += ' ' + board[4] + ' | ' + board[5] + ' | ' + board[6] + '\n'
    boardstring += '   |   |\n'
    boardstring += '__________\n'
    boardstring += num[1] + '  |' + num[2] + '  |' + num[3] + '\n'
    boardstring += ' ' + board[1] + ' | ' + board[2] + ' | ' + board[3] + '\n'
    boardstring += '   |   |\n'
    boardstring = boardstring.replace("_", "\033[0;38m-\033[0m")
    boardstring = boardstring.replace("|", "\033[0;38m|\033[0m")

    if winner is None:
        boardstring = boardstring.replace("X", "\033[0;36mX\033[0m")
        boardstring = boardstring.replace("O", "\033[0;95mO\033[0m")
    elif winner is "X":
        boardstring = boardstring.replace("X", b"\xf0\x9f\x98\x8e".decode())
        boardstring = boardstring.replace("O", "\033[0;95mO\033[0m")
    else:
        boardstring = boardstring.replace("X", "\033[0;36mX\033[0m")
        boardstring = boardstring.replace("O", b"\xf0\x9f\x98\x8e".decode())

    print(boardstring)


def text_naar_scherm(s=None):
    """
    @type s: str, None
    @return: None
    """
    if s is None:
        print()
    else:
        print("\033[0;34m" + str(s) + "\033[0m")


def vraag_speler_zet(board):
    """
    @type board: list
    @return: None
    """
    move = ' '

    while move not in '1 2 3 4 5 6 7 8 9'.split() or not vrije_plek(board, int(move)):
        text_naar_scherm('Wat is uw volgende zet? (1-9, s=stoppen)')
        try:
            move = input()
        except KeyboardInterrupt:
            exit(1)

        if move.strip().lower() == "s" or move.strip().lower() == "stoppen":
            exit(1)

    return int(move)


def vrije_plek(board, move):
    """
    @type board: list
    @type move: int
    @return: None
    """
    return board[move] == ' '


def wie_gaat_eerst():
    """
    wie_gaat_eerst
    """
    if random.randint(0, 1) == 0:
        return 'computer'
    else:
        return 'player'


readline.parse_and_bind("tab: complete")

if __name__ == "__main__":
    main()
```