

TicTacToe.py

```

1  # This is a game of Tic-Tac-Toe Played against the Computer
2  # The Human Player uses the symbol X and the computer uses 0. For extra clarity each of these symbols are colored
3  coded
4  # who ever gets consecutive 3 symbols in spots next to ech other horizontally, vertically or diagonally, wins!
5
6  # Usage: python TicTacToe.py
7  # -----
8
9  import random
10 import os
11 from copy import deepcopy
12
13 # Declares constant
14 COMPUTER = '0'
15 HUMAN = 'X'
16 COLSYM = ' | '
17 ROWSYM = '- '
18 TOTSPOT = 9
19 MAXCOL = 3
20 RED = '\033[91m'
21 BLUE = '\033[94m'
22 BOLD = '\33[1m'
23 UNDERLINE = '\33[4m'
24 ENDCODE = '\033[0m'
25
26 def clean_slate():
27     # Clear the screen
28     if os.name == 'nt':
29         _ = os.system('cls')
30     else:
31         _ = os.system('clear')
32
33
34 def draw_slate(mat):
35     # Clear the screen and redraw the current game with pipe separating columns and hyphen separating row.

```

```

36 # Format the messages with extra formatting
37 clean_slate()
38 print(BOLD + UNDERLINE + "This is a game of Tic-Tac-Toe where You play against the computer" + ENDCODE)
39 print("The Computer selects a position randomly.\n")
40 print("You are "+ BLUE + BOLD + "X" + ENDCODE +" and the Computer is " + RED + BOLD+ "O\n" + ENDCODE)
41
42 # We want the X to be printed in blue color and the 0 in red
43 # But we don't want to change the original values in the list because it will effect it's comparison
functionality
44 # Hence we are making a deep copy of the game mat and changing only the copy with color codes
45 color_mat = deepcopy(mat)
46 # Color code the X and 0
47 for i in range(MAXCOL):
48     for j in range(MAXCOL):
49         if color_mat[i][j] == HUMAN:
50             color_mat[i][j] = BLUE + HUMAN + ENDCODE
51         elif color_mat[i][j] == COMPUTER:
52             color_mat[i][j] = RED + COMPUTER + ENDCODE
53
54 # Print the colored game mat
55 i = 0
56 for row in color_mat:
57     i += 1
58     print("\t" + COLSYM.join(row))
59     if i < len(color_mat):
60         print("\t" + ROWSYM * TOTSPOT)
61
62
63 def check_game_over(mat):
64     # Check if every spot on the game mat has been filled.
65     # Returns True if all spots are filled and indicates that the game is over
66
67     return all(col != " " for line in mat for col in line)
68
69 def make_a_play(mat, player):
70     # This function lets the player choose a spot on game mat and places an X
71     # OR generates a random move for the computer to place a 0
72     # If the spot is already occupied or out of range, an error message is generated

```

```

73 while True:
74     try:
75         if player == COMPUTER:
76             spot = (random.randint(1, TOTSPOT) - 1 )
77         else:
78             spot = int(input(f"\nPlease enter your move (1-9): ")) - 1
79             i, j = divmod(spot, MAXCOL)
80             if 0 <= spot <= TOTSPOT - 1 and mat[i][j] == " ":
81                 mat[i][j] = player
82                 return
83         else:
84             print("Invalid choice. Try again.")
85     except ValueError:
86         print("Invalid choice. Please enter a number between 1 and 9.")
87
88 def check_game_winner(mat, player):
89     # Check all the winning combinations
90
91     # 1. Horizontal
92     for i in mat:
93         if all(j == player for j in i):
94             return True
95     # 2. Vertical
96     for j in range(MAXCOL):
97         if all(mat[i][j] == player for i in range(MAXCOL)):
98             return True
99     # 3. Diagonal
100    if all(mat[i][i] == player for i in range(MAXCOL)) or all(mat[i][2 - i] == player for i in range(MAXCOL)):
101        return True
102    return False
103
104 def get_next_player(cur_player):
105     # Get the next player based on the current player
106     if cur_player == COMPUTER:
107         next_player = HUMAN
108     else:
109         next_player = COMPUTER
110     return next_player

```

```
111
112 def match():
113     # Creates a 3 x 3 game mat and plays the game
114     # Randomly generates a spot for COMPUTER
115     # Accepts a spot from user
116     # Checks if a winner can be declared in the game.
117     mat = [[" " for _ in range(MAXCOL)] for _ in range(MAXCOL)]
118     player = HUMAN
119     while True:
120         draw_slate(mat)
121         make_a_play(mat, player)
122         if check_game_winner(mat, player):
123             draw_slate(mat)
124             if player == HUMAN:
125                 dec_winner = "Congrats!! YOU WIN THE GAME AGAINST THE COMPUTER!!"
126             else:
127                 dec_winner = "Sorry you lost!! THE COMPUTER WINS THE GAME!!"
128             print("\n" + dec_winner + "\n")
129             break
130         if check_game_over(mat):
131             draw_slate(mat)
132             print("\nYou Tied the game with the Computer!!\n")
133             break
134
135         player = get_next_player(player)
136
137
138 if __name__ == "__main__":
139     match()
140
```

Terminal Output Samples

Python ./TicTacToe.py

1

This is a game of Tic-Tac-Toe where You play against the computer
The Computer selects a position randomly.

You are **X** and the Computer is **O**

```
  |  |  
--|--  
  |  |  
--|--  
  |  |
```

Please enter your move (1-9): █

2

This is a game of Tic-Tac-Toe where You play against the computer
The Computer selects a position randomly.

You are **X** and the Computer is **O**

```
X | O | X  
--|--  
O | X | X  
--|--  
O | X | O
```

You Tied the game with the Computer!!

3

This is a game of Tic-Tac-Toe where You play against the computer
The Computer selects a position randomly.

You are **X** and the Computer is **O**

```
X | O |  
--|--  
O | X |  
--|--  
  |  | X
```

Congrats!! YOU WIN THE GAME AGAINST THE COMPUTER!!