Title: Dice-Life Game Documentation

# 1. Introduction

Dice-Life is an interactive game designed to allow players to roll a die, accumulate points, and compete against each other. The game's primary objective is to accumulate more points than the opponent. The game's rules, player registration, die-rolling, and score tallying are facilitated through Python functions, allowing for a structured and interactive gaming experience.

# 2. Functions Description

## 2.1 `print_rules()`

This function displays the game rules to the players, providing a brief introduction to the game's structure and objectives.

## 2.2 `no_of_players()`

This function prompts the user to input the number of players and their names. It returns the names of the players, which are stored in a list called `player`.

## 2.3 `roll()`

The `roll` function simulates the rolling of a die by generating a random number between 1 and 6. The generated value is returned as the result of the die roll.

## 2.4 `roll_option()`

This function seeks the player's consent to roll the die again. The user can enter 'Y/y' to roll again or 'N/n' to pass the turn to the next player.

## 2.5 `comp_move()`

This function simulates the computer's move by calling the `roll_option` and `roll` functions. Points are accumulated based on the die roll, with a roll of six resetting the score to zero.

## 2.6 `human_move()`

Similar to `comp_move`, this function simulates the human player's move, accumulating points based on the die roll.

## 2.7 `current_game_status()`

This function evaluates the current game status by comparing the total points of the computer and the human player, returning the point difference.

## 2.8 `show_final_result(comp_total, hum_total)`

This function checks the total scores of the computer and human player to determine if either has surpassed 50 points, declaring the winner accordingly.

# 3. Main Execution (`main()`)

The `main` function orchestrates the game flow by calling the defined functions in a structured manner. It also writes the game behaviors and results to a text file named `template_behavior_{dice}.txt` for record-keeping.

### 3.1 File Writing

The `main` function opens a file `template_behavior_{dice}.txt` in append mode, writing the game's behaviors and outcomes to this file for each run of the game.

### 3.2 Game Flow

The game starts by displaying the rules using `print_rules`, followed by player registration through `no_of_players`. The game then enters a while-loop where the `comp_move`, `human_move`, and `current_game_status` functions are called in sequence to facilitate the game play. The loop continues until the `run` variable is set to False, at which point the final result is displayed using `show_final_result` and the game concludes.

### 3.3 Execution Trigger

The script's execution is triggered by the `if __name__ == "__main__":` statement, ensuring that the `main` function is called when the script is run directly.

## 4. Conclusion

The Dice-Life game script provides a simple, yet engaging, gaming experience where players can interact with the game, rolling a die to accumulate points in a competitive setting against a computer opponent. Through a series of well-structured functions, the script facilitates the game flow, player interaction, and score tallying, providing a fun and interactive gaming experience.