

Homework 6 — Jeopardy Dice

Due February 6th, 2018 at 11:55pm

(50 points)

Objectives:

- Implement a “real” application (a game that you can play) and conquer the challenges associated with it
- Understand the mapping of “real world” game elements to program structure
- Understand how while loops work and how they are different than for loops
- Use a module (`random`) and functions from it
- Use if `__name__ == "__main__"` to prevent your program from running if it is imported anywhere

Turn In:

- `jeopardy_dice.py`

You should turn this file in on moodle under the “Homework 6 - Jeopardy Dice” link.

Instructions:

Your job is to write a program that plays a two player version of a game called “Jeopardy Dice”. Your user will be one player, and the computer will be the other player. Your program will be robust against users who aren’t always cooperative. They may type a number when you ask for a letter, they may type in UPPER CASE when you ask for lower case, etc.

How to play Jeopardy Dice:

General:

This is a two player game in which the players alternate taking turns until one of them wins by scoring at least 25 points. In this game, it is possible that the players do not get the same number of turns. For instance, if player 1 has 26 points at the end of their 5th turn, the game will end and player 2 would only have gotten 4 turns.

In your implementation, the user will always go first.

One turn:

- 1) The player whose turn it is rolls a 6 sided die.
 - a) If they roll a 1, their turn is over and 1 point is added to their score.
 - b) If they roll any other number, that number is added to their total and they are given the following choice: roll again, and return to step #1 or end their turn and add their current turn total to their score.
 - i) If it is the computer’s turn, the computer decides whether to roll again or not by comparing their current turn total to a predefined number. If the computer’s total is at or above that number, it stops rolling and ends its turn. Otherwise, it continues rolling.



Game end condition:

The game ends, if, at the end of a player's turn, any player has a score that is greater than or equal to 25 points.

Phrased another way: the game continues while no player has a score greater than or equal to 25 points.

Program Structure:

You must implement the following functions:

- `get_user_answer`: this function takes no parameters. It asks the user whether or not they want to roll again until the user inputs a properly formatted answer. It then **returns** the user's response. **Case insensitive**.
- `get_next_player`: this function should take one parameter that indicates if the player who just finished their turn is or is not the computer. If the computer just took its turn, this function should **return** False, if the computer did not just take its turn, this function should return True.
- `take_turn`: this function should take two parameters, `is_computer`, that indicates whether or not the current player is the computer player, and `hold_value`, that indicates the value at which the computer decides to stop rolling the die. This function should **return** the number of points that the current player scored.
- `report_points`: this function should two parameters, the user's current score and the computer's current score. It will then print out the current scoreboard.

You must define two variables, `game_end = 25` and `hold_value = 10` in your `main` function. If you wanted to change your game so that it is played to 50 points and the computer stops rolling at 5 points, these are the only two variables whose values you should change.

Output & Testing:

It is your responsibility to test your program. You can use an online difference checker tool (like <https://text-compare.com/>) to compare and contrast the contents of the output files with the output of your program. If your program output does not exactly match the files we provide, you will not receive full credit.

Development:

Your code should always be in a "runnable" state. **Do not attempt to develop your entire program without running it.**

- 1) Before you start, get a die and play the game. Make sure you understand how it works.
- 2) Start by writing your `main` function. Greet the user.
- 3) Next, write your `take_turn` function. Call your function from `main`. Make sure that you can run one complete turn as the user player before moving on.



- 4) Next, make it so that the computer player can also play one complete turn.
- 5) Now, write a while loop that runs the game until it has finished, alternating turns between the user and the computer.
- 6) Finally, test your code a lot.
- 7) Lastly, test your code with a variety of user inputs. Test to see if you can easily change your program to play a game to 50 instead.

Style & Structure (7 points)

Naming (1 point)

- Your variable names should be meaningful and concise
- Your variable names should be formatted like: `user_birthday` (words start with lowercase letters and are separated with underscores)
- Function names should be formatted the same way as your variables: `get_today()` (words start with lowercase letters and are separated with underscores)

Proper Variable Usage (1 point)

- You should not have variables outside of functions.

Functions (5 points)

- You **must** have a `main` function. The only code in your program that is not inside a function should be a single call to `main()`. Your main function may contain: function calls, variable declarations, one `while` loop, and `if/else` statements. It should not print more than 4 lines of text.

REQUIRED: use `if __name__ == "__main__":`

```
def main():  
    print('This is the main function')  
  
main()
```

Becomes:

```
def main():  
    print('This is the main function')  
  
if __name__ == '__main__':  
    main()
```

- All non-main functions **must** be defined at the zero indentation level, as in the example lecture code. Do not attempt to define a function inside another function.
- Do not define two main functions.



Comments (3 points)

Your code must be commented. You must include a file comment (your name, your section, the homework name, and a brief description of what the program does), inline comments (explaining any complex code), and function comments. An example function comment is shown below:

```
# This function converts a temperature in fahrenheit to celsius
# and prints the equivalence.
# Parameters: fahrenheit - int or float degrees fahrenheit
# Return: float - equivalent degrees celsius
def fahrenheit_to_celsius(fahrenheit):
    celsius = (fahrenheit - 32) * (5/9)
    return celsius
```

Output

Example Output (user input <u>bold underlined</u>)	Example Output (user input <u>bold underlined</u>)
Welcome to Jeopardy Dice! It is now the user's turn You rolled a 4 Your turn total is 4 Do you want to roll again (Y/N)? <u>Y</u> You rolled a 2 Your turn total is 6 Do you want to roll again (Y/N)? <u>hi</u> Do you want to roll again (Y/N)? <u>n</u> Computer: 0 User: 6 It is now the computer's turn You rolled a 1 Computer: 1 User: 6 It is now the user's turn You rolled a 2 Your turn total is 2 Do you want to roll again (Y/N)? <u>sandwich</u> Do you want to roll again (Y/N)? <u>7</u> Do you want to roll again (Y/N)? <u>y</u> You rolled a 2 Your turn total is 4 Do you want to roll again (Y/N)? <u>y</u> You rolled a 3 Your turn total is 7 Do you want to roll again (Y/N)? <u>y</u> You rolled a 3	Welcome to Jeopardy Dice! It is now the user's turn You rolled a 2 Your turn total is 2 Do you want to roll again (Y/N)? <u>n</u> Computer: 0 User: 2 It is now the computer's turn You rolled a 1 Computer: 1 User: 2 It is now the user's turn You rolled a 3 Your turn total is 3 Do you want to roll again (Y/N)? <u>N</u> You rolled a 2 Your turn total is 5 Do you want to roll again (Y/N)? <u>n</u> Computer: 1 User: 7 It is now the computer's turn You rolled a 3 Your turn total is 3 You rolled a 5 Your turn total is 8 You rolled a 1 Computer: 2 User: 7 It is now the user's turn You rolled a 3 Your turn total is 3



<p>Your turn total is 10</p> <p>Do you want to roll again (Y/N)? n</p> <p>Computer: 1</p> <p>User: 16</p> <p>It is now the computer's turn</p> <p>You rolled a 5</p> <p>Your turn total is 5</p> <p>You rolled a 2</p> <p>Your turn total is 7</p> <p>You rolled a 1</p> <p>Computer: 2</p> <p>User: 16</p> <p>It is now the user's turn</p> <p>You rolled a 2</p> <p>Your turn total is 2</p> <p>Do you want to roll again (Y/N)? y</p> <p>You rolled a 3</p> <p>Your turn total is 5</p> <p>Do you want to roll again (Y/N)? y</p> <p>You rolled a 1</p> <p>Computer: 2</p> <p>User: 17</p> <p>It is now the computer's turn</p> <p>You rolled a 1</p> <p>Computer: 3</p> <p>User: 17</p> <p>It is now the user's turn</p> <p>You rolled a 3</p> <p>Your turn total is 3</p> <p>Do you want to roll again (Y/N)? y</p> <p>You rolled a 2</p> <p>Your turn total is 5</p> <p>Do you want to roll again (Y/N)? y</p> <p>You rolled a 4</p> <p>Your turn total is 9</p> <p>Do you want to roll again (Y/N)? n</p> <p>Computer: 3</p> <p>User: 26</p> <p>Congratulations! User won this round of jeopardy dice!</p>	<p>Do you want to roll again (Y/N)? n</p> <p>Computer: 2</p> <p>User: 10</p> <p>It is now the computer's turn</p> <p>You rolled a 5</p> <p>Your turn total is 5</p> <p>You rolled a 2</p> <p>Your turn total is 7</p> <p>You rolled a 6</p> <p>Your turn total is 13</p> <p>Computer: 15</p> <p>User: 10</p> <p>It is now the user's turn</p> <p>You rolled a 5</p> <p>Your turn total is 5</p> <p>Do you want to roll again (Y/N)? n</p> <p>Computer: 15</p> <p>User: 15</p> <p>It is now the computer's turn</p> <p>You rolled a 4</p> <p>Your turn total is 4</p> <p>You rolled a 2</p> <p>Your turn total is 6</p> <p>You rolled a 1</p> <p>Computer: 16</p> <p>User: 15</p> <p>It is now the user's turn</p> <p>You rolled a 1</p> <p>Computer: 16</p> <p>User: 16</p> <p>It is now the computer's turn</p> <p>You rolled a 4</p> <p>Your turn total is 4</p> <p>You rolled a 5</p> <p>Your turn total is 9</p> <p>You rolled a 5</p> <p>Your turn total is 14</p> <p>Computer: 30</p> <p>User: 16</p> <p>Congratulations! Computer won this round of jeopardy dice!</p>
---	---

As a reference, our solution, including empty lines and comments, is 100 lines long.

