16.0 hours
Project Status Failed

Project 3

Phrase Hunters

Needs Work

Your project was reviewed and needs some work. Take time to look over what was marked as needs work.

If you get your project back and it still needs work, you'll have to wait 1 day before you can resubmit your project.

Resubmit for Review

Here's how you were graded on the individual requirements for this project. These requirements are combined with your reviewer's overall assessment to make up your overall grade.

Thank you for rating this project review!
How satisfied were you with your project review feedback?

Elements that Need Work

- **Gameplay**

  Missing one or more of the following:

  - Prior to the player's first attempt at guessing the phrase did not output to the screen hidden, showing only _ placeholders for each character
  - The did not use strictly Object-Oriented approaches: Instance creation and method calls for the entire length of one or multiple plays of the game.

  Reviewer Comments:

  So so so close! There is not a noticeable space between each character. There is a noticeable space between each word though. You want both. In your display method you want to have end='' not end = '' in the else condition it should look like this: else: print('_', end=' '

Elements that Met Expectations

- **Script execution**

  - The script should not crash due to uncaught exceptions. Raised exceptions should be handled appropriately so the program can continue or exit without a crash
  - Instance creation, method calls, print statements, or any calculated execution logic should be wrapped inside a Dunder Main statement for the entry point script of `app.py`

- **Phrase initialization**

    - The initializer method `def __init__` was implemented
    - A parameter was defined on `def __init__` to allow a phrase argument to be passed in and stored as an instance attribute

    Reviewer Comments:

    Great job! An **init** method is properly used in the Phrase class!

- **Game initialization**

    - Includes an initializer that will set a `phrases` instance attribute to a List of at least five `Phrase` objects
    - Initializes an instance attribute to store and track the number missed. This should start at 0.
    - Initializes an instance attribute to store the "active phrase", or the phrase that will be guessed by the player
    - Initializes an instance attribute to store the user's guesses

    Reviewer Comments:

    Great job! An **init** method is properly used in the Game class!

- **Entry Point app.py**

    - The entry point script was named properly as: `app.py`
    - Running `app.py` creates a Game instance and calls an instance method that handles starts the game.

    Reviewer Comments:

    Awesome! The program has the properly named entry point and a game instance is properly created!

Elements that Exceeded Expectations

- **Correct character guess**

    - Validate the player's guess by ensuring the guess is 1 character in length and that it is only letter character: a through z *(uppercase or lowercase)*

    - Any errors/exceptions should be handled

- **Incorrect character guess**

    - Validate the player's guess by ensuring the guess is 1 character in length and that it is only letters: a through z *(uppercase or lowercase)*

    - Any errors/exceptions should be handled

    NOTE: It is entirely up to you if you want to remove a turn/life for these types of incorrect guesses.

    Reviewer Comments:

    Awesome! Again the messaging is a little off, but I'm not able to guess more than one letter at a time! I noticed that you have unused raise statements on lines 64 and 69 if you run your program through

flake8 or some kind of linting program it will be really easy to stop these types of errors. Again great job!

- **Player wins**

  After winning, the player was prompted to play again:

  - if they agree a new game instance should be created OR the current game instance should have its attributes reset to a NEW game state
  - if they disagree, the game should end with a message

- **Player loss**

  After losing, the player was prompted to play again:

  - if they agree a new game instance should be created OR the current game instance should have its attributes reset to a NEW game state
  - if they disagree, the game should end with a message

Overall Comments

So so close! I'm really sorry this doesn't quite make it! Overall this is a great project it's really just the lack of spacing between characters. If you're not using an IDE like Atom, VScode, or PyCharm. Personally I like VSCode best of those three, there are a lot, I recommend finding one that you like and sticking to it. I also recommend running you projects through flake8 or another linting program just to check for Pep8 errors. Your program has a lot of little things like lines with whitespace, bare except and unused raise statements. There are also a number of indentations that don't use multiples of four. All of these Pep8 'error' are really, really tricky to see and flake8 or pylint or any other linting program will really help out to correct these issues. I wouldn't have noticed most of them if I didn't have a linting program running while I was looking at your code. Your doing great so far. I really liked that you used sets to check if the game was over, very clever. I also really liked that you included notes in the file for each of you methods it's a lot easier to figure out what is going on. Good luck! If you get stuck reach out on Slack everyone is really friendly there!