DS-GA-1007 Programming for Data Science

Assignment 5

Instructions

You are free to use whichever development environment you wish to create and submit the assignment answers.

1. Fork the `assignment5` repository from the ds-gs-1007 user on GitHub.
2. Clone this repository to your local system.
3. The `assignment5` repository has a directory called `adventure` that contains Python program that plays the game of “Adventure”. Instructions for running the game are located in the `README.txt` file.
4. The game is written for Python 2.7 so will not run using Python 3. Try running the game to see what happens. Familiarize yourself with the structure and organization of the code.
5. Using online or other resources, identify and correct the problem with the game. Use the tests to ensure that you have fixed all problems.
6. Once you have fixed the problem, experiment with the game by playing it.
7. Create a directory using your Net ID as the directory name under the `assignment5` directory.
8. Place your answers to questions 1 through 12 below in a file called “`answers.txt`” in this directory.
9. Add the directory to the repository and commit it.
10. Push your changes to your forked repository on GitHub.
11. Submit a pull request to the repository owner (ds-ga-1007).

Questions

Note: Use the file name and line number when referring to locations in the code.

1. Describe the changes you made to make the game work with Python 3
2. Describe three main techniques that the author used to structure the program.
3. Has the author used meaningful names? Give some examples of meaningful names used and what you think they mean. Give some examples of where the author has not used meaningful names.
4. Do the functions used in the code do one thing? Give some examples of functions that only do one thing. Give some examples of functions that do more than one thing.
5. Do any of the functions cause side effects? If so, which ones?
6. Can you find any repeated code that could be made into a function?
7. Does the program use exception handling? Can you find any input that causes the program to terminate abnormally? *Hint: run the program from the terminal prompt. The invalid input may not be normal text.*
8. Do any of the classes have responsibility over more than one piece of functionality. If so, which ones?
9. Are all the classes cohesive? List any that aren't.
10. Describe the author's approach to commenting the code. Provide examples of good and bad comments that have been used in the code.
11. Provide an example of where vertical formatting has been used to make the code clearer.
12. Run the tests provided with the program. Do they pass or fail? Do you consider the tests meet the F.I.R.S.T. criteria? Provide details of why they do or do not meet the criteria.