SI 507
Homework 4: Github

You will need a github account to start and complete this homework assignment.

Homework Objective:

Demonstrate the ability to:

- Use git and github to maintain project history and collaborate
- Use branching and merging to support parallel collaborative work
- Create a program for Magic Ball 8

Supporting Material:

- Magic Eight Ball Possible answers: https://en.wikipedia.org/wiki/Magic_8-Ball
- Magic 8-ball simulator: http://www.8-ball-magic.com/index.php#anchor

Readings:

- http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1/
- https://readwrite.com/2013/10/02/github-for-beginners-part-2/
- https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control
- https://git-scm.com/book/en/v2/Getting-Started-A-Short-History-of-Git
- https://git-scm.com/book/en/v2/Getting-Started-Git-Basics
- https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging

Part 1: Using Github and Creating Magic Eight Ball code (20pts)

Find a partner and sign up your pairing here, so we know who's working with whom.

There may be an odd number of people if someone is absent. If you don't have a partner, you can join a group of two and follow the same process with some slight differences as a Person C -- it's not a problem to be a 3rd party in a group and won't make much of a difference. See the bottom of this document if you are in a group of 3! (Could even be more fun.)

If you are encountering other difficulties and are unable to find a partner, please let your GSI know so you can come up with another way of fulfilling similar objectives.

There are (at least) two roles:
Person A
Person B

You're encouraged to work on this together, side-by-side, and talk through both/all parts!

The instructions here are written in terms that are sometimes general and sometimes git-specific. For the general instructions, you will need to figure out how to translate the common-sense description into the appropriate git commands (of course taking advantages of all of the usual channels for getting help).

PERSON A:

- Create a directory on your personal computer and have git start tracking it.
- Create a magic_eight.py file
- Create a function that asks users “What is your question?” (or something similar) and saves the response.
- Create a repository on GitHub for your work, called `SI507-HW04-(uniquename).`
- Add Person B (and C) as a collaborator
- Connect your GitHub repository to your local

PERSON B:

- Get the repository
- Create a new branch called “add_questions”
- In the magic_eight.py file add code to pick an answer at random from the 20 possible 8 ball answers.
- Push your new code *and branch* to GitHub.

PERSON A:

- Create a new branch called “check_question”
- Write code that checks if user input is a question (i.e., ends in a ‘?’) and, if not, prints “I’m sorry, I can only answer questions.” (or something similar)
- Edit the program so that it keeps asking for user input until the user inputs “quit”
- Get your changes onto GitHub

PERSON B:

- Get the latest code from GitHub
- Merge your branch into the master branch
- Get your changes back onto GitHub

PERSON A:

- Get the latest code from GitHub
- Merge your branch into the master branch (after Person B has completed merge)
- Get your changes back onto GitHub

PERSON A+B:

- Pull the master branch and verify that everything works

IF YOU HAVE 3 PEOPLE IN YOUR GROUP:
Add a Person C, who will follow the same steps as Person B, but both B and C will create different branches (say, “add_questionsB” and “add_questionsC”) and each add only 10 of the total answers.

BEFORE TURNING IT IN:
Edit the README.md file to include the names, uniqnames, section number, and github IDs of the 2 or 3 people who worked on the project.

Extra Credit 1 (2 pts)

For the first extra credit, you are going to implement a simple version of the classic card game *Go Fish* with a partner. You will need to create a new repo on Github called `SI507-HW04-gofish-(uniquename)` containing a file named *go_fish.py* (which you will later submit on canvas). For the first extra credit, you will simply create a version of the game which asks for input from two human players taking turns (no computer players will be involved).

The goal of this part of the exercise is to familiarize yourself with collaborative coding and also give you some more experience/practice with concepts you have learned earlier in the semester. (*HINT:* Feel free to use your Card, Deck, and Hand classes from Homework 2).

Go Fish Rules: https://en.wikipedia.org/wiki/Go_Fish

Adjustments To These Rules:

1. You do not need to worry about selecting a dealer, ‘cutting the deck’, or ‘deal direction’ when dealing out the cards at the beginning of the game. Simply start each human player with a hand of 7 cards.
2. You have creative license in terms of how you structure the game, prompt the user for input, etc. That being said, please include directions as to how to play in your output (ie. on someone’s turn, you might print the following when asking for input: ‘*Please choose a card rank you would like to ask the other player if they have (between 1-13):’)*

Once you have completed this part of the assignment, please submit a link to your repository. You and your partner are expected to have each made ***at least two commits*** each. The README for your repo should list the names, uniqnames, section number, and github IDs of everyone who contributed to the project.

Extra Credit 2 (2 pts)

For the second extra credit, you are going to implement an autoplay feature for *Go Fish* which will allow you to run a game simulation between 2 and 4 computer players. This implementation can simply have each computer player play randomly or you can implement a simple ‘intelligent play’ system should you choose to do so. For this part of the assignment, you will create a branch of the repository you created for Extra Credit Part 1. The branch should be called ‘autoplay’. You will *not* merge this branch back into master, but keep it separate (it’s a different *version* of your Go Fish game).

The rules for this part of the assignment are rather open ended as well, however, please provide ample output which displays the actions and final outcome of the game clearly in the terminal. At the beginning of the game, prompt the user to input how many computer players they would like to take part in the simulation (between 2 and 4). For this portion of the assignment, you may deal each computer player 7 cards to start with.

As with the first extra credit, you and your partner are each expected to make ***at least 2 commits each*** (they should be real commits, with several lines of code added in each case) to this part of the assignment. You don’t need to submit anything different for this. If we see that there is an autoplay branch in your repo, we will test it out to determine credit for this part of the assignment.

What to turn in:

All code must be executable. Any code that does not run in **Python3** will be **given a score of 0.**
You can receive partial credit for working programs.
Submit following python module(s):

- A link to your GitHub repository for Part 1
- Make sure the two (or three) of you have "registered" that you're working together on the spreadsheet, including placing your GitHub links in it.
- (Optional) a link to your GitHub repository for the Extra Credit problems.