Project 4: Game Center

July 5, 2022

Description: You are a developer wanting to get more people involved in the python coding language. To get more people interested, it is a good idea to merge it with something everyone enjoys – video games! Your goal for this project is to code a game that people can interact with! You can choose to code one of the three projects below. Click on the name of the game to see how to play.

- War – Moderately Easy to code
- 21 (Blackjack) – Moderately difficult to code
- Wordle – Moderately hard to code

If you choose War: War is a game that is usually played between 2 people. Your implementation of the program should play between at **most** two people. This game plays automatically until one person has all the cards. Thus, you should include a command that acts as a turn. Think: "Press Spacebar to draw the next card". This is to prevent the game finishing within seconds.

If you choose 21: 21 (blackjack) is played between a dealer and a player. To make things simpler, the dealer can be (and should) be automated. The game should be able to be played with **at least** two players (not including the dealer). Once all players have "stayed" on their cards, then the dealer should have their turn and report the winner. This will be the game demo'd in class on Wednesday so if you choose 21, please come with questions on how you would like to implement this.

If you choose Wordle: Wordle is an automated game that is not played between multiple players. Thus, it can be considered easier than the other. If you choose war or 21, there are some limitations that are included. I.e., if two players have the same value in War, the rank can decide the winner. Further, if the players tie in value, they can continue playing cards until there is not a tie. For 21, the game usually includes betting. These implementations that can be made to make the game more robust but at a bare level, the game should be playable and can exclude some features. For wordle, every rule, feature, and feedback must be implemented. I.e. there are characters that show a word is invalid, a letter bank that shows useable and unusable letters, etc. I would suggest playing wordle a few times to see how the game plays. This game may not seem difficult at first but you will find that there are plenty of minor rules that can introduce bugs you have never seen before!

Project: Since this is a longer and more difficult project than before, you will have more time to complete it. To make sure you are making meaningful progress, there will be two phases for the project. There will be three classes to work on the project – July 6, July 13, and July 20.

- By **July 12**, the first draft of the project should be completed. This should include asking the user for input, having 1 turn be able to take place, and minor game logic included.
- By **July 20,** the final draft of the project should be completed. This should include full game logic, multiple turns taking place, and any extra features you want to include (betting, helpful print statements, or any other flourishes you find useful).

There is no starter code posted on GitHub. Code will be given to everyone after we have gone through the code in class. Between now and class, please look through the project, rules, and how-to-play videos on YouTube to help influence your decision on what game you want to code. Think about what data structures may be useful, how you can store player information, how would a game take place, etc. Before any code is written, making helpful outlines on what you PLAN to do can give you an idea on where to start coding. I am excited to play your games and seeing your progress! If you have any questions, please reach out!