Group Name: Cranberry

Christen Holmes
One aspect of this project that I am not entirely sure how to accurately implement is establishing the cards that each player has in their hand. Essentially, each player is dealt 10 cards. Their objective is to make combinations of three cards of the same rank or runs of the same suit. I realized that one way we could do this is by creating a Hand class. Within that class, we could create a method to sort their individual decks in hand.

James Joseph:
The problem that I will attempt to solve for this assignment is how player points are scored each turn. As in this game the turns are continuous until a player's hands meet the specific conditions to stop the turn and determine who wins this round. And the game completely ends when a player's score reaches 100. It will be tricky to connect this class with a player, opponent turn class to assign score specifically to a player. To solve this problem I would like to create a scoring class with three method which details the specific ways a player can win a round in gin rummy.

Chloe Root:

The problem that I set out to solve for this first check in is to figure out how to create the player class and then a method that would create the instance of each individual player and establish the order of their turn to determine how the game will be played. This was definitely a bit challenging for me to figure out how to separate the players and determine who would go first, second, etc. in the round of the game. I learned that in order to define the separate players and their order, it would be easier if I imported a text file that had a list of all of the information, something similar to the one we used for the battle aardvarks assignment.

John Spurrier

The problem I am solving for this assignment is creating a card class and in that class creating methods that can check whether or not cards are the same suit or checking that cards could be part of a straight. One of the things I realized while working on this portion was that Kings,Queens,and Jacks had to have their own values that were not equal to 10. In most card games those values are 10, but for this game they had to have separate values so that we can determine straight, and three/four of a kind. Also, one of the rules in gin rummy is that in order to count a straight, the cards must be the same suit. My solution accounts for that technicality. My solution is the Card.py file and then the only thing that is needed to run the program is to type “python3 Card.py” and the program will run. My solution creates three cards and then tests whether some of those cards could be part of the same straight or be part of a three/four of a kind.