**Project Description:**
The project that I will be making is titled Coup, and it is a role playing card game which heavily uses the concepts of bluffing alliances and deception in order to cause other players in the game to lose their cards. There are 5 cards in the game each with their own unique functions. The cards are Duke, Captain, Ambassador, Assassin, Contessa.

Duke: Has the ability to take 3 coins and Block Foreign Aid
Captain: Has the ability to steal 2 coins from another player and block steals
Ambassador: Has the ability to pick up 2 cards from the deck and exchange and can also block steals.
Assassin: A player can pay 3 coins to make another player lose a card
Contessa: Can block the Assassin

Additionally there are 3 built in actions that players may do:

Coup: Pay 7 coins to make a player lose his card and nothing can be done to stop this
Income: Take 1 coins
Foreign Aid: Take 2 coins

Blocking can occur without being a player's turn but offensive actions may only be executed when it's the player's turn.

The player with the last card is the winner.

**Competitive Analysis:**
The only similar project that I have seen online is the coup implementation by Mr.Valdez on github. My project is somewhat similar to his in that our classes are structured similarly but the respective User Interfaces will be very different as well as the logic behind the function. There are only one set of rules that coup can follow and I have in this respect designed my game similarly. There are no other implementations of this game that have been made so there are not any other cases for me to compare my game to.

**Structural Plan:**
- The project will be organized as follows:
- The Game Class and Player Class are organized in a file called Deck Operations
- The card Actions are split into a primary Actions and secondary Actions classes and they are in a file called Deck
- The last file will contain the AI Class from which the AI makes its decisions
- Then finally there will be the draw file where the user interface is drawn and sockets are implemented

**Algorithmic Plan:**
The toughest part of my project is making the AI, so the plan for approaching it is as follows:

- I will first make a class for the AI
    - The class will inherit from the regular player class and contain and Init and Run
        - The Init will possess all of the typical attributes: Including cards left, the hand, and whether the player is alive or not
        - The run function will be how the AI decides what action to play
            - This function will make decisions off of the expected value of actions. The expected value will be calculated using numpy.
    - This is the detailed algorithmic part for my project

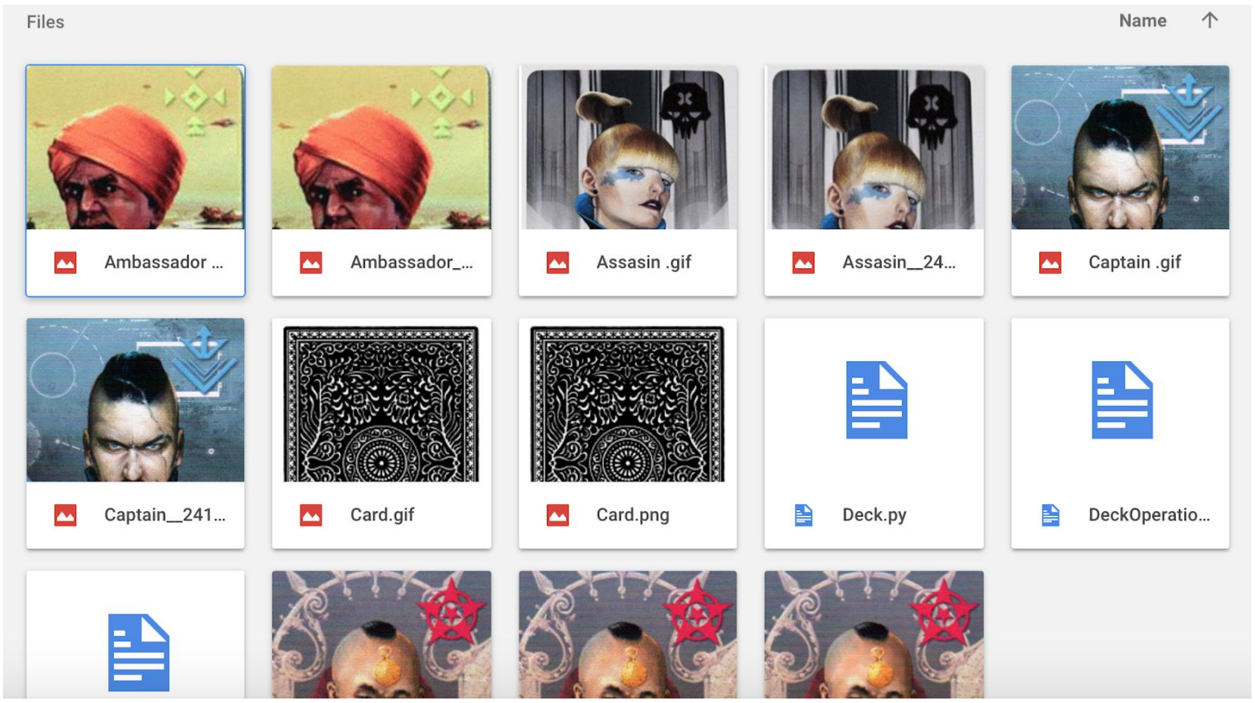
**Timeline Plan:**
Tuesday April 17: Finished Build for Multiplayer
Thursday April 19: Finish implementation for scoet
Tuesday April 24: Finish Building User Interface for the Game
Tuesday May 1: Implement AI

**Design Proposal TP2:**
There is no design that I have changed for the project

**Design Proposal TP3:**

There is no design changes that I've made.

**Version Control:**
I am just uploading every version of my project to Google Drive.



**Module List:**
Sockets
Numpy