

# Poker Project

By Faris, Omar, and Tommy

## Motivation

A fun game to play with friends. Also relatively complicated in terms of different scenarios and end-game situations, which makes it a relatively fun project to code up, and with networking, it should make it a very social project.

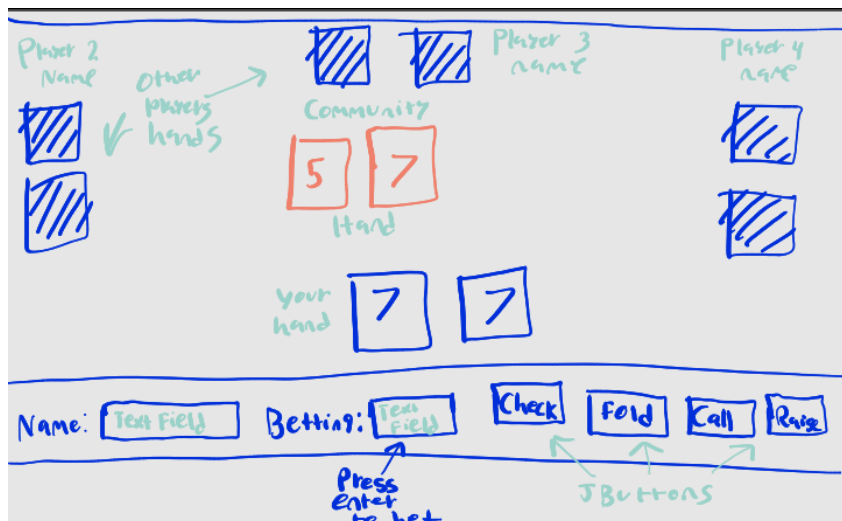
## Problem

The main problems that we have with this project are:

1. How do we implement networking to connect users to each other?
2. How does the server manage the game?
3. Betting amounts, rules, and end-game conditions?
4. What will the front-end look like, and how do we change it for each user?

## GUI

It's going to be a white screen followed by a table with a nice green surface. In the center there are the cards from the community, and there are going to be buttons to raise, call, or fold.



# Networking

Multiplayer game that requires networking to send and receive data from the server for things such as moves, betting amounts, and end-game conditions. Also uses multithreading in order to maintain input/output between the client and the server.

## Challenge

Every hand is a different hand and you will never see the same hand twice, meaning that there has to be algorithms that allow for end-game comparisons to see who wins each hand. You have to be able to keep track of all moves that are made and have them distributed to all clients. Also, with all of the hands, you need to be able to keep track of end game scenarios.

## Testing Plan

In order for the testing to work correctly, we need to be able to play multiple scenarios, with different amounts and different hands, in order to ensure that it works. The first step of testing would probably be ensuring that the connection works and that the players are placed into the array, and if another player tries to join a full table, a pop-up appears that shows the table is full. Other errors to think about is an invalid betting amount, or a betting amount greater than the amount allotted.

Deliverables would include the GUI for the poker table, the server that maintains the clients, and any other threads that run in the background in order to ensure smooth operation of the game.

## Manager's Workload

In order to maintain all of the work, the manager will be handling all of the busywork that includes uploading images and instantiating arrays of the cards, as well as figuring out ways to send the data back and forth. The manager will also figure out mapping between the different images of the cards and the different positions of the array in order for the frontend and the backend to communicate correctly.