Poker Face

Kathryn Wang TP Design Proposal

Description

This TP project will be combining the anonymity of online poker with the crucial facial expression aspects of poker, by using facial recognition to display the emotions of the other players and/or drawing a simple caricature of their expression.

Competitive Analysis:

In the Term Project Gallery, I saw in Fall 2019, Etan Cohn created Program Poker which used a computer AI to program single player poker and calculate the probability of winning for the player at each point in the game. My program will also be creating online poker and my MVP may include using a computer AI to play single player, but the focus of this project is more on the facial recognition aspects. I haven't seen a project online yet that incorporates facial expressions into online poker, so I think it will be different from what I have seen.

Facial recognition seems to be pretty common in term projects, where other students have used facial cues to determine if someone is lying and as shown in class, trying on different glasses. I think my project includes some elements of all of these, but since I don't know how OpenCV really works, I'm not sure how algorithmically similar mine will be.

Structural Plan

- HighestHand File
  - ➢ Contains the functions needed for determining the highest hand
  - ➢ Takes in a list of cards (hand)
  - ➢ Each type of poker hand (Straight flush, four of a kind, full house, etc.) will have its own function that determines if the given cards contains that hand
  - ➢ The HighestHand function will determine a player's highest 5-card hand based on the given cards
  - ➢ Each "level" of hand is given a number and that number is returned so that it can later be compared amongst different hands. The "kickers," as in the cards that determine tie breakers of a hand, will also be returned. The kickers will be different data types depending on the hand. For example, a straight flush will return the rank of the lowest card in the straight, where as a pair will have to return the ranks of the other three highest cards.
- PlayingCard class
  - ➢ Contains the different types of cards in the deck
  - ➢ Shuffling of cards

- Player class
  - ➢ Will contain the cards of each player, their player “number”/username, and potentially the OpenCV information
  - ➢ Contains the amount of dinex each player has for “betting” purposes
- GetWinner File
  - ➢ Takes in a list of players
  - ➢ Will call the HighestHand function on the hands of each of those players and store the values returned
  - ➢ It will first compare the first value returned by each call because it will tell us which hand is the highest “level”
  - ➢ If different players have the same level, then the returned kickers will be compared.
  - ➢ Includes a tie breaker method
- Betting file
  - ➢ The Betting method will be called after every round of cards is revealed
- Splitting the Pot file
  - ➢ Based on the values returned by the GetWinner, will tell us if there’s a tie and how the pot will be split
- App file
  - ➢ Controls the graphics
  - ➢ Also calls the necessary functions to determine what to display

Algorithmic Plan:
Some of the algorithmic details are described above, but before any modules are used, the most algorithmically complex part is determining the winner of the poker game since there are a lot of cases and edge cases. I wrote out a diagram explaining what each hand functions returns and how those return values will be compared to determine the winner.

Hand:	Returns:
Straight Flush	1, lowest card rank
Four of a kind	2, (four rank, single rank)
Full House	3, (triple rank, double rank)
Flush	4, (5 cards)
Straight	5, lowest card rank
Three of a kind	6, (remaining two)

Determining winner will be comparing the first number returned. If it's the same

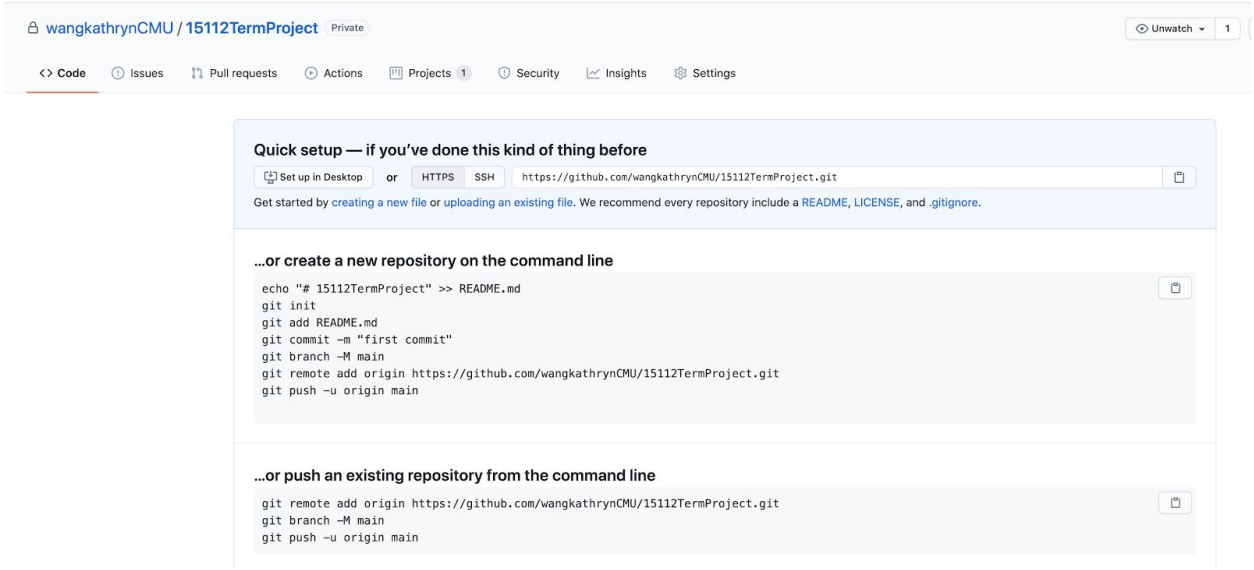
↓

Tie Breaker Function

→ takes multiple lists of same number of single cards and determines the highest. If highest are the same, goes to second highest and so on.

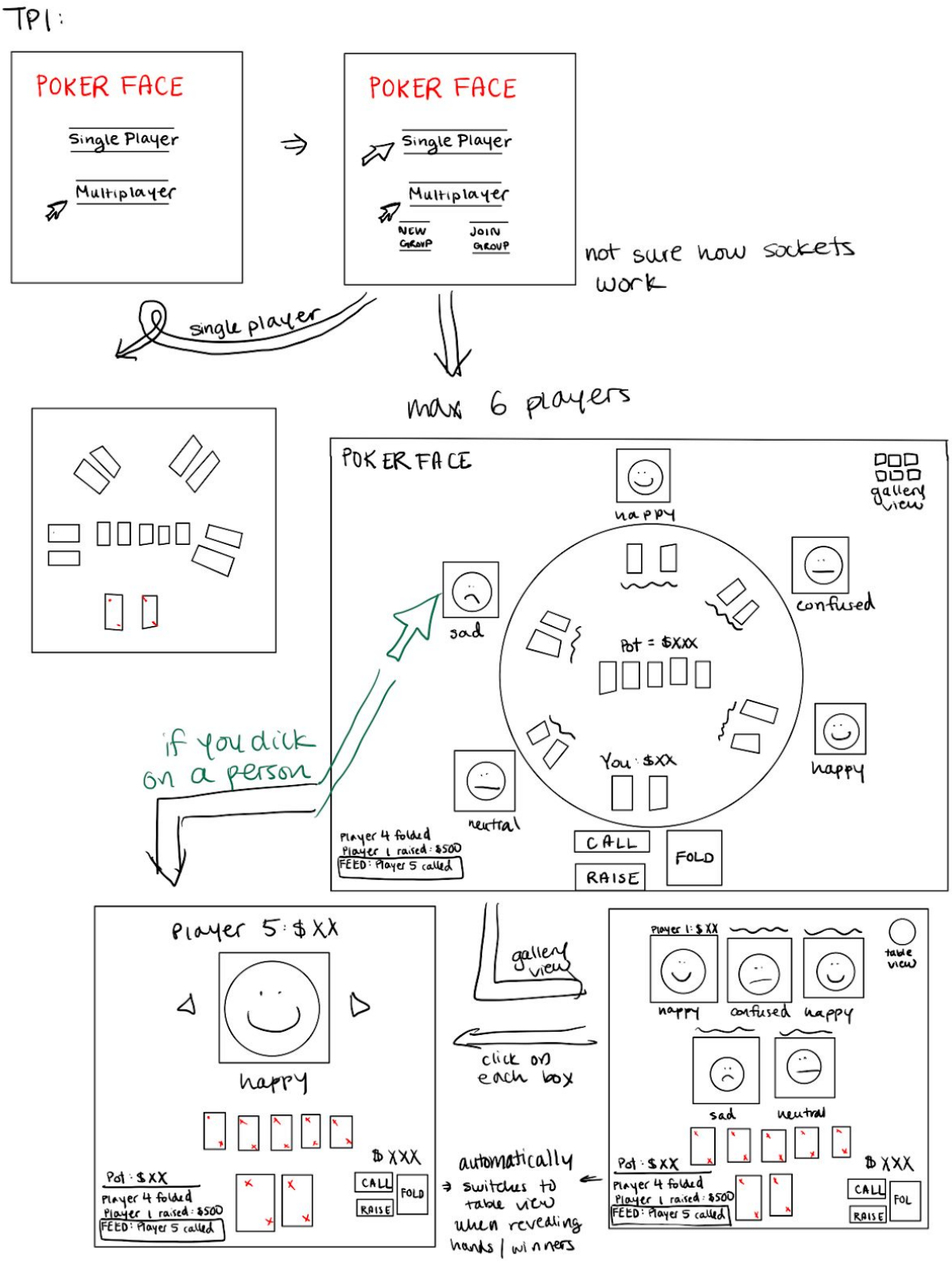
Timeline Plan:
Thursday: Finish entire basic poker game with betting and graphics
Saturday (TP2): Do basic tech demo for openCV and sockets after figuring out how to use them. Add computer AI or other basic single player functionality for MVP.
Wednesday (TP3): Add computer vision aspects and sockets

Version Control Plan: I'm planning on backing up my code using github.



Module List: OpenCV, sockets

Storyboard


TP1:
POKER FACE
Single Player
Multiplayer
POKER FACE
Single Player
Multiplayer
NEW GROUP
JOIN GROUP
not sure how sockets work
single player
max 6 players
POKER FACE
gallery view
happy
confused
sad
Pot = $XXX
You: $XX
happy
neutral
if you click on a person
Player 4 folded
Player 1 raised: $500
FEED: Player 5 called
CALL
RAISE
FOLD
Player 5: $XX
happy
$ XXX
Pot: $XX
Player 4 folded
Player 1 raised: $500
FEED: Player 5 called
CALL
RAISE
FOLD
gallery view
click on each box
automatically switches to table view when revealing hands / winners
Player 1: $XX
table view
happy
confused
happy
sad
neutral
Pot: $XX
$ XXX
Player 4 folded
Player 1 raised: $500
FEED: Player 5 called
CALL
RAISE
FOL

TP2 Updated Design: