Chinese Checkers

Libraries and Frameworks:

* Libraries: node.js, express.js
* Frameworks: EaselJS
* Languages used and other: JavaScript, HTML 5, JSON

Game board:

* One per game instance that is manipulated by all clients.
* Contains:
  + A list of Marbles that are moved around on the board. The board must handle between 20-60 Marbles (2-6 players).
  + 7 Arrays of Points. 1 for the points in the center and 6 that contain the points in each of the 6 starting areas
  + A function that determines if an attempted new location of a marble on the board is valid or not.
  + Some kind of button or some such object that it can place marbles on, and which marbles can be moved to.
  + A function that draws the board and all its contents.
  + A function to determine if the game has ended (someone won).
  + The ability to recognize what kind of space the marble is on {a home space of X color, or a field space}.
  + Check\_Valid\_Move(Marble m, Point old, Point new).
  + Gameboard(int players, string gamemode) – constructors.
  + Player\_Lost(Player player).

Marbles:

* Between 20-60 in a game for Classic/Super. 60 in a game for Capture.
* Contains:
  + An image to represent the Marble.
  + A color, chosen from red, green, blue, yellow, black, or white.
  + The opposite color of the chosen color of the marble, which is the color whom the chosen color wants to get into the home of.
  + Current Position
  + A function that allows the user to click on the Marble and drag it to a new location.
    - Press\_Move\_Handler() – when clicked on and dragged, the marble’s will follow the mouse.
    - Press\_Up\_Handler() – When the mouse is let up on this marble, if the position is valid, then the servers update\_Position() function is called. If the position is not valid then the marble is reverted to its starting point.
  + Remove() – Removes marble from the game, increases current player’s score in if capture mode.

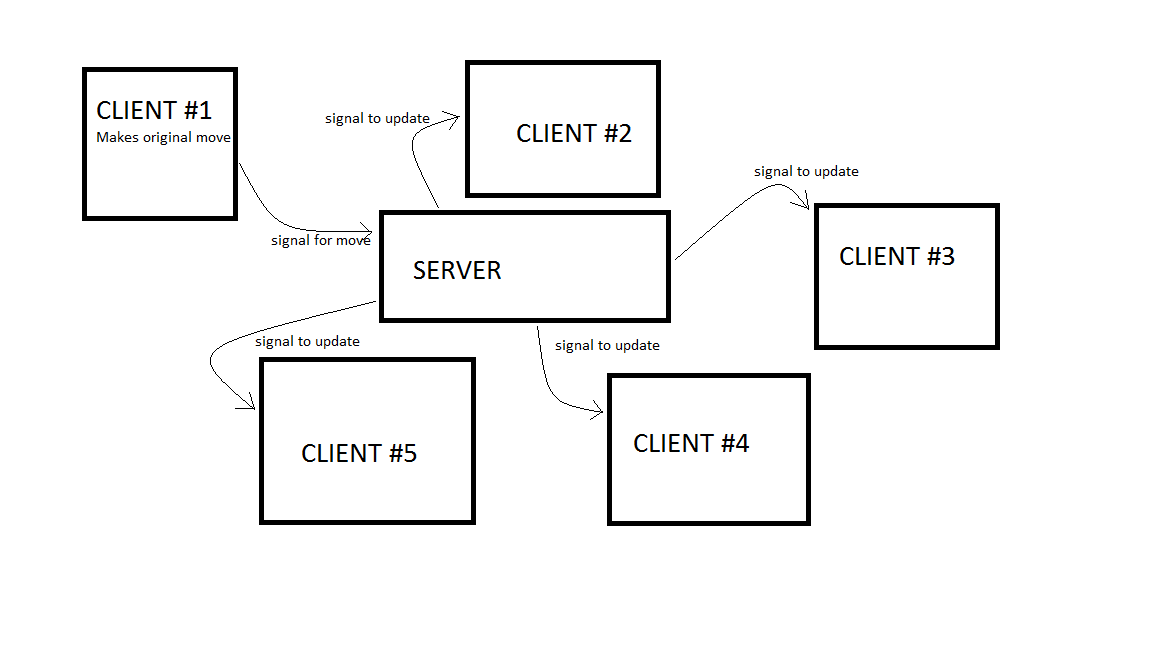
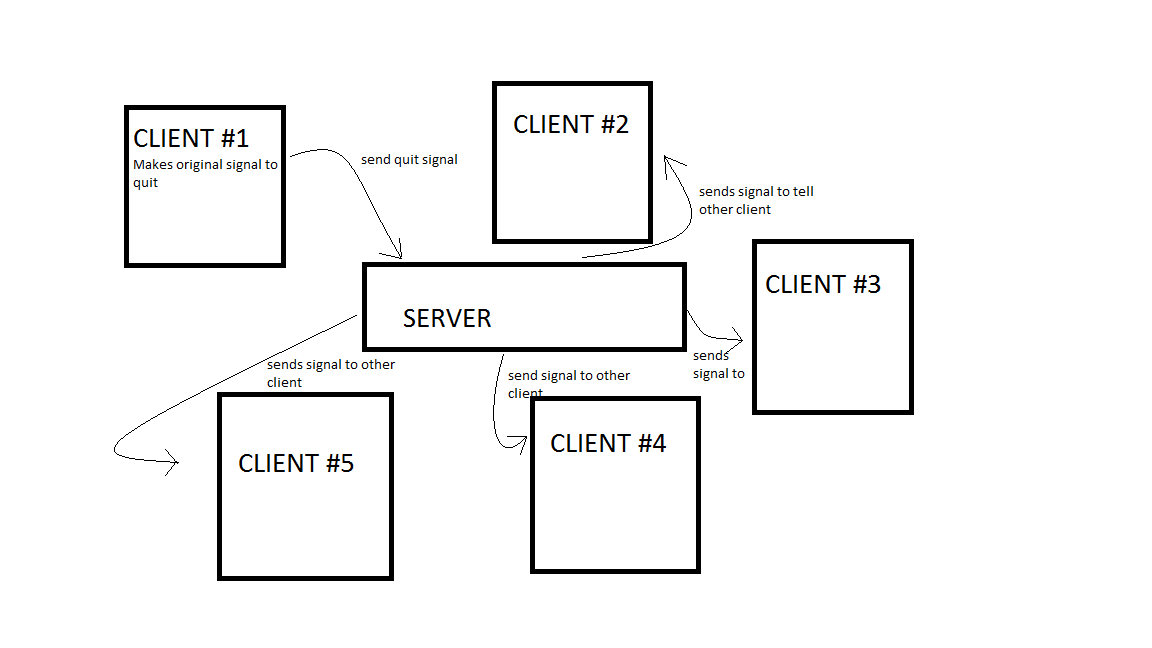
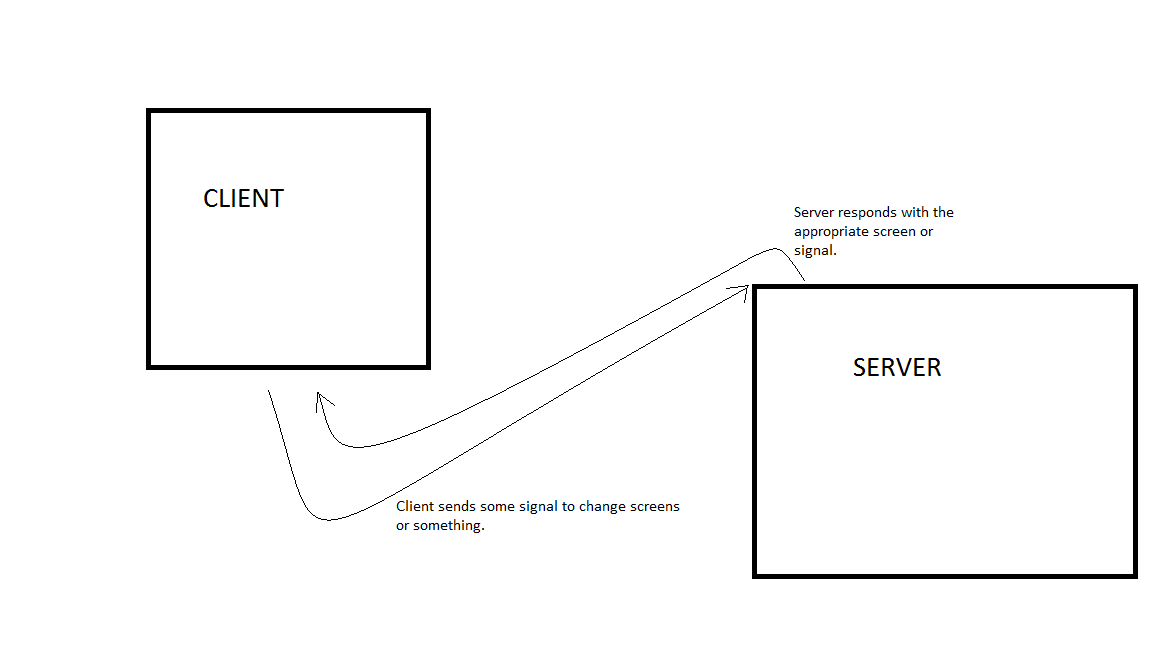
Player:

* Between 2 to 6 players in a game.
* Contains:
  + A list of same colored marbles that they are allowed to manipulate.
  + A score, if they are playing Capture.
  + A way to check with the Game Board to see if they have won or lost.

Point:

* Color
* Screen Point
* Game board Point
* Marble

Screen:

* 5 different game screens throughout application.
* Introduction Screen:
  + A title that reads “Chinese Checkers”.
  + A button labeled “Instructions” that allows the Instruction screen to pop up in a different screen.
  + A button labeled “Play” that sends the user to the Choice Screen.
  + A button labeled “Quit” that quits the application all together.
* Instruction Screen:
  + Text that contains information about the game, and explains how to play it.
  + Images/GIFS that illustrate some of the explanations.
  + A way to make this screen pop up separate from the main screen.
  + A button labeled “Quit” that quits the instruction screen.
* Choice Screen:
  + A text box that allows the player to enter their name.
  + A drop down menu that allows the player to select how many players they wish to play against {1, 2, 3, 4, 6, any}.
  + A drop down menu that allows the player to choose the game mode they wish to play in {Classic, Super, Capture, any}.
  + A drop down menu that allows the player to select if they wish for the game to be timed or not {Yes, No, any}.
  + A button labeled “Continue” that sends the user to the Waiting Screen.
  + A button labeled “Back” that sends the user back to the Introduction Screen.
  + A button labeled “Quit” that quits the application all together.
* Waiting Screen
  + A message informing the player that other players are being located, and how many have thus far been located.
  + A button labeled “Continue” that sends the user to the Game Screen.
  + A button labeled “Back” that sends the user back to the Choice Screen and pulls them out of the running for searching for other players.
  + A button labeled “Quit” that quits the application all together and pulls them out of the running for searching for other players.
* Game Screen
  + A Game Board object, with its associated marbles.
  + Labels that tell the user who they are, who the other players are (with identification as to their color game piece and score if the game mode is Capture), and information about the game they have selected (game mode and such).
  + A timer if they chose the timed version of the game.
  + A button labeled “Instructions” that allows the Instruction screen to pop up in a different screen.
  + A button labeled “Surrender” that causes an Are You Sure? Screen to pop up.
  + Game\_Over(Player winner)
  + Update\_Turn(Player current\_player)
  + Instructions\_Button\_Handler()
  + Surrender\_Button\_Handler()
  + Time\_Handler() – Called every second. If in timed mode and it is this players turn, decrement the timer.
* Are You Sure? Screen
  + A button labeled “Yes” that brings the user back to the Introduction screen and takes them out of the game.
  + A button labeled “No” that exits the Are You Sure? Screen and allows the user to continue playing the game.
* Communication Between Server and Client
  + When a marble is moved the client that is moving the marble sends a signal to the serve which then in turn sends a signal to the rest of the clients about the marble’s new position
    - 
  + When a player quits it sends a signal to the serve which then in turn sends a signal to the other clients saying that it quit.
    - 
  + Most communication between the server and client dealing with the actual game play behaves in much of the same manner as the two previously mentioned aspects.
    - Other actions that communicate in the same way: Game Over (when a player wins the server sends the signal to all clients after the client that won sends a signal to the server saying it won)
  + The other ways the serve may communicate to the clients is when the server only has to communicate to one of the clients. The best examples of this are probably the switching of the screens, such as: going from the intro. Screen to the choice screen which goes to the waiting screen. Also, the player may send a signal for the instructions screen during many periods of the game. A diagram of this communication is as follows:
    - 
    - The other times a client and server may communicate in this way are through the “are you sure?” screen, or if the move is invalid.
  + The server and client are going to communicate through:
    - Node.js
    - Socket.io
  + Much in the same way the chat room example was given to us on Piazza.
  + We will be hosting the game on “compute.cse.tamu.edu” using any high port number (greater than 10000). Also through Http this way it should be accessible through any web browser if you know the port number.