# WEB405 Project 2

## Multi-Player Client/Server Game

Create a new game server and client or extend your previous one if it can be multi-player. This one will be a multi-player game. The computer will not play the game but will control the game play of two or more players. Write a server and client in JavaScript run in node.

Choose a game that requires at least two players. Don't do a game that you know someone else is doing. You are free to help each other with this assignment unless you are both doing the same game.

Same as project 1, include with your files a plain text (or markdown) document that outlines the rules of the game. Include one or more links to websites you used to research the rules. Note any rules that you ignore in order to simplify your game. Also, make a note of any known bugs in your code.

### Start of game

1. Each player will start a client that will connect to the server. You can determine when the game starts. For example, if the game has four and only four players, the server could start the game when the forth player connects. If the number of players is variable, then a player could start the game.
2. Once the game starts, players will not be able to join the game but they should get a message. If you are really ambitious, you could start a new game for additional players, but this is not a simple task!

### Game Play

1. The server will need to keep track of the game play such as cards delt, score, etc.
2. Prompt a player for their move.
3. When the game is complete, show the score and who won.
4. Ask the user if they would like to play again.

### Coding

1. The client and server program will both be written in JavaScript and run with Node.
2. Use the zeromq messaging module. A recommended approach would be to use PUB/SUB for general announcements and REQ/REP for individual players interaction.
3. Write external modules where they would help simplify your main program. For example, you could have a module for a deck of cards, dice, player, etc.

### Submitting

1. Demonstrate your game to the class.
2. Submit a zip file including your server, client, and a text file describing the game play. Include any other files such as modules.

# Comment

Add a comment at the top of each file (after `"use strict";`) with your name, date, and the project name.

```
"use strict";
/*
*Name of My Game*
WEB405 Project 2
Author: *YOUR_NAME*
Date: *DATE*
*/
```

Everyone’s code will be shared with the class. That way you can learn from each other.