

**class User**

- socketID: null  
+AcceptGame(user): boolean

**class Game**

- player1: new Player(nick)  
- player2: new Player(nick)  
  
+ isGameOn() ---> boolean  
+ initGame()  
+ setTurn(player)  
+ isWon()  
+ resolveGuess(user);

**class Player**

- gameId: Game object  
- user: reference to user object  
- opponent: null  
- hitCount: number  
- board: []  
  
+ setOpponent(Player object)  
+ initBoard(board)  
+ getCellContent(index)

gameManager

+class Game  
+class Player  
  
games: {gameID: game object...}

I/O Router

+ajax ---> users.getUserList()  
  
+socket.on("connection") ---> establishConnection()  
  
+socket.on("disconnect") ---> pauseGame()  
  
+socket.on("guess", {clickX, clickY, nick})  
  
+socket.on("acceptGame") ---> game.player.initBoard()\*2  
  
+eliminateConnection()  
  
+registerGame(socketID1, socketID2)  
  
+socket.on("gameRefused") ---> fwdDeclinedRequest(user)  
  
+AnnounceTurn(player) emits("turn") to player socket

Users Module

-registeredUsers: {nick: {}, nick: {}, nick: {}}  
  
+getUserList(registeredUsers)  
  
+addUser()  
  
+setOnline(user)  
  
+setOffline(user)  
  
+class User

## Client Side

### class Cell

-status: string  
- x: num  
- y: num

+ draw()  
+ setHit()  
+ setEmptyGuess()

### class Board

- boardCells: [100 Cell objects]  
+ drawBattleship(x,y, direction, length)

### boardDrawer Module

+ class Board  
+ class Cell  
- messageBoard: object  
- messageBoard.displayMessage(msg);

### Game Manager Module

- localBoard: new board  
- targetBoard: new board  
- event listener: board.click  
- validateCell(x,y, board)  
- turn: boolean  
+ initTargetBoard(board array);

### Connection/Authorizarion Module

+ displayLoginWindow() (onLoad)  
+ sendLogin() --> event listener on button - sync  
+ validateNick(nick)

### User Input Module

+ getCellClicked() ---> returns {x:x, y:y} - sync  
+ moveShip(ship, x,y)

### Server Communicator Module

+ on("GameRequest", (user)) --> emit true/false  
+ sendRequest(user) ---> emit "GameRequest" (user)  
+ Ajax - getOnlineUsers();  
+ connectUser() :socket.connect ---> sendLogin()  
  
+ sendGuess(): emit("guess")  
  
+ on("turn") --> Game Manager.turn = true;  
+ sendReponse(response): emit("acceptGame") or  
emit("gameRefused")