

## Functions documentation:

In our project there are 4 files who responsible for most of the project's functionality

agxgame.js  
public/app.js  
Validator.js

### app.js:

responsible for the logic of the client:

public functions:

- onConnected
- onNewGameCreated
- playerJoinedRoom
- recieveMessage
- recieveTransaction
- GameStarted
- addPlayers
- updateChips
- rejectOffer
- acceptOffer
- beginFaze
- movePlayer
- removeOfferButton
- counterOffer
- reveal
- addRowToHistory
- thereIsAWinner

**onConnected:**

this function is called when a user connects to the server.

The function caches a copy of the client's socket.IO session ID on the App.

**onNewGameCreated:**

calls to function `gameInit(data)` in the client, which sets the new game's parameters such as `gameId`, `socketId`, etc.

**playerJoinedRoom:**

This function is called on each player who joins the game.

The server tells the client that he connected to the server and transfer him to waiting screen until a game is actually starts.

**recieveMessage:**

responsible for handling messages from other players.

Adds row to the table of offers, in which the player can choose if to accept or reject the offer.

**recieveTransaction:**

This function is called when there is chip transaction from one player to another, the function adds row to the history table mentions the transaction made by the two players.

Also transfer the chips from the sender to the receiver.

**addPlayers:**

add the players to list to the right side of the board.

**updateChips:**

change the amount of chips for each player, and also sets their score respectively.

**rejectOffer:**

sets the state of the offer to be "rejected", which means the player will not be able to accept it anymore.

**acceptOffer:**

sets the state of the offer to be "accepted", which means the player will not be able to accept it again.

**beginFaze:**

starts a new phase in the game, this function gets all the required data from the server, the board, the players, the players' roles, the scores etc. This means that on each phase the game may change in a big way.

**movePlayer:**

this function moves a specific player to a new location on the board and decreasing a chips from his stash.

**removeOfferButton:**

removes the button who adds the offers/transaction rows, and all the transaction rows who havn't been sent yet.

**counterOffer:**

If this function is called the player need to make a counter offer to the player who sent him an offer previously, so an offer raw will be shown on screen.

**reveal:**

this function is called when a player chosses to reveal his goals to other players.

It runs on all the player's goals and revel those who are real.

**addRowToHistory:**

Adds another row to the history table, it can be a transaction or an offer.

**therelsAWinner:**

finishes the game, shows the winner and after few seconds changes the screen to the waiting screen.

### **agxgame.js:**

responsible for the logic of the server:

public functions:

- playerDisconnect
- playerJoinGame
- sendOffer
- transferChips
- updateChips
- rejectOffer
- movePlayer
- dontReveal
- reveal

#### **playerDisconnect:**

Identifies the room in which the player disconnected, and ends the game in this room.

#### **playerJoinGame:**

the server adds the player to a list of waiting players, when a game starts the players mentioned in the configuration file and in the room will be pulled out and start play the rest will keep wait.

#### **sendOffer:**

send an offer from one player to another, it specifies the amount of chips to send the other player, and to get from him in response.

#### **transferChips:**

transfer an amount of chips from one player to another in exchange for nothing, regardless of the other player's response.

#### **updateChips:**

sends to all the players in the room the amount of chips that a player has in his stash.

The function is mostly called because there was a transfer in chips between players or an offer who got accepted.

#### **rejectOffer:**

in response to an offer of a player, this function tells the player who offered that his offer got rejected.

**movePlayer:**

verifies if the move is legal, and if the it is a legal move, the player will be transferred to the new location. By the way the function checks if the player reached goal if so, he will be announced as winner and the gae will be over[ according to the rules of the configuration].

**dontReveal:**

tells the server that a player choose not to reveal his goals. So the server tells the client not to reveal the goals.

**reveal:**

tells the server that a player choose to reveal his goals. So the server tells the client to reveal the goals of the player who are real goals.

***Validator.js***

In this file there is a class who responsible on the validation of the configuration file.

Also it inculdes functions who verify the rules and restrictions of the game.

For example: two players can't be placed on the same trail in the beginning of the game.