# Project Battleship

## Client

### Init – folder

#### Config.js

* Globale variabele config + info over game

#### Initializer.js

A screen shot of a computer program

Description automatically generated with low confidence

* HandleSettings
  + UI/Setting-Handler
* RandomizeUniqueCode
  + Helper-functions/helper-functions
  + Code aanmaken
* testConnection
  + Helper-funcitons/helper-functions
* updateSliders
  + UI/ui-handler
  + Board size + audio slider
    - handleInputChange
  + rendering/render-range-of-size
* serverEventListeners
  + ui/ui-handler
    - events voor server interactie
* dialogHandler
  + ui/dialog-handler
  + tutorial voor het spel te spelen
* closeDialog
  + tutorial message sluiten
* initShips
  + init/ship-init
    - grote schip bepalen
    - lengte op het board geven
    - cells op bord geven aan dat schip
  + ships renderen op sidebars
* handleShipPlacement
  + ship-placing
    - ship-placing/drag-drop
    - plaatsen van schepen
      * richting van schepen veranderen
    - checken of alle schepen geplaatst zijn.
    - Submitten van form wanneer je de unique code ingeeft
    - Checken of een schip geplaatst is
* validShipPlacement
  + ship-placing/restrict-ship-placement

### Game-matcher

#### Matching-engine.js

* Connecting naar game
* Checken voor andere spelers
* Game starten
* Game refreshen
* Update game modus
* startGame
  + game-matcher/matching-engine
* set salvo size
  + in-game/gameplay
* sidebarUpdateCommander
* History aanspreken
  + Via server fetchen en informatie uithalen
    - In-game/Visualize-information

#### Gameplay.js

* Cellen selecteren voor te schieten
* Volgende cel aanduiden voor salvo schots
* Move ships
  + checken adhv. Regels om een schip te moven
* coördinaten van schepen krijgen
* shots aanduiden
* kijken of je een schip geraakt hebt
  + hit => css style voor hit + sound
  + mis => css style voor miss + sond
* Checken of het jouw beurt is
  + **SetTimeOut**
    - **wachten voor response**
* Tonen van gezonken schepen op de sidebar

computeCellsForSalvo

* berekenen hoeveel keer je mag schieten

##### serverEventListeners

A screen shot of a computer code

Description automatically generated with low confidence

## Server

* openApiBridge
  + Requests naar server
* FireSalvoRequest
  + Salvocoordinates
  + getDefendingcommander
  + getGames
  + setResponse back
* FireSalvoRequestBodyRequest
  + getSalvo
    - salvo terugkrijgen
* GetFleetDetailsRequest
  + updatePathParameters
    - gameId
    - commander
  + setGame
    - Aanmaken van pregames
* GetGameByIdRequest
  + setGame
    - aanmaken van een game
* GetGamesRequest
  + getPrefix van game
  + getStatus van game
  + getType van game
  + getRows van game
  + getCols van game
  + updateQueries
    - updaten van gegevens
  + setFilteredGames
    - filteren van games
      * type
      * status
      * prefix
      * toList
    - logic/GameList
* JoinGameRequest
  + getGameType
  + getCommader
  + getFleet
  + getPrefix
  + setGameCredentials
    - gameId
    - token
* MoveShipRequest
  + updatePathParameters
    - ship
    - movingCommander
    - gameId
    - startedGame
* exceptions/BatlleshipException
  + throwIfNotAllShipsArePresent
  + IdenticalCommanderName
  + WrongTypeGame
  + WrongGridSize
  + MisMatchingGame
* BaiscGameCredentials
  + Token
  + gameId
* Game – Interface
  + getId
  + getPrefix
  + getType
  + getStarted
  + getSize
  + getCommanderByName
* GameList
  + getFilteredRunningGames
  + getPreGameList
  + addGame
  + deleteGameById
  + getGameByPrefix
  + getSize
  + deleteGameList
  + getGamesByRowsAndCols
  + getGamesByType
  + getGamesByStatus
  + getGamesByPrefix
  + toList
* PreGame
  + getId
  + getType
  + getCommander
  + PreGame
    - Prefix
    - gameId
    - Type
    - Player
    - Size board
  + getStarted
  + getSize
  + getPrefix
  + getCommanderByName
* StartedGame
  + - gameId
    - prefix
    - gameType
    - attackingCommander
    - defendingCommander
    - size
  + getSalvoSize
  + getSunkShips
  + setWinner
  + getWinner
  + getHistory
  + getLastSalvoCoordinates
  + addSalvoToHistory
  + getAttackingCommander
  + getDefendingCommander
  + getStarted
  + getId
  + getType
  + getSize
  + getCommanderByName
  + getPrefix
  + moveShips
  + switchCommanders
  + checkGameOver
* helper/coordinate
  + - row, col
  + getRow, getCol
  + toString
  + computeNext ?
  + isOnGrid ?
* Direction – ENUM
  + computeDirection
* Fleet
  + - Size
    - Ships
  + isDestroyed
  + getRows
  + getCols
  + getShips
  + moveShip
  + getSize
  + calculateMaxSalvoSize
  + getSunkShips
  + shootSingleShotAtAllShips ?
* GridSize
  + Rows en cols
* Player
  + - Name
    - Fleet
  + getName
  + getFleet
  + getGridSize
  + moveShip
  + isLost
  + getSalvoSize
  + getSunkShips
* MoveAction
  + - attackingCommander
    - ship
    - move => MoveActionResponse
  + getResult
* MoveActionResponse
  + - Ship
    - Coordinates
  + getShips
  + getNewLocation
* Action
  + - attackingCommander
    - coordinate
    - shipName
  + getAttackingCommander
  + publicAbstract getResult => **STARTEDGAME**
* shoot/shootAction
  + extends => Action
    - allShots
    - attackingCommander
  + getResult
* shootActionResponse
  + - shots
  + getShots
* ships/Ship
  + - coordinates
    - hitCoordinates
    - direction
    - name
  + isDestroyed
  + getName
  + moveShipForwardsByOne
  + fireSingleShotAtShip
* ships/ShipName - ENUM
  + Carrier, BattleShip, Cruiser, Submarine
  + Size
  + ShipName
  + getSize
  + getName