Assignment 2

Team Battle Chips (CSE2 56)

N. Cohen, 4836650

T. Brogtrop, 9131094

# 2.1

Setting up the game:

* Press start game loads the – randomly generated - player set-up from the server
* Server informs player if it his move
  + If it is the player move, then wait for the move to be selected
  + If it is the adversary move then PlayerBoard elements are disabled

Playing the game:

* Mouse click on div element with id “…” in the section adversaryBoard. Send move to server. Since already used div elements are disabled there is no need to check the move validity on the server.
* Server checks if the move is a “HIT” or a “MISS” and checks if the game is over
* Server informs client 1 of:
  + HIT or MISS
  + Update game status (player hits vs adversary hits
* Server informs client 2 of:
  + Move made by player
  + Update game status
* In case game is over, Client 1 and Client 2 update screen with “Game Over” message and a button is enabled “Play Again”
* Client 1 updates the div element selected by marking it red or gray, and disabling the div
* Client 1 disables the board to wait for the opponent move
* Client 2 updates the div element and enables the player board to select a move

# 2.2

Main objects to consider are:

GameStatus – this is the ‘Model’ part of the MVC Design Pattern. GameStatus will need the following attributes:

* Player Grid (Grid is a separate object)
* Adversary Grid
* Initialize game (load player and adversary positions from server, decide who goes first)
* Who has won? (function returning P (Player) A (Adversary) or null
* Send Player has Lost – send message to server that game was lost
* Whose turn is it?
* playerShotWasHitOrMiss – function called by a message from the server with the information whether shot was hiot or miss
* adversaryShotWasHitOrMiss – function called by adversaryMove to send to the server feedvack if adversary shot was hit or miss
* playerMove – function called by View when player has made a move. This function will forward the move to the server.
* adversaryMove – function triggered by a message received from the server. If it is a “Game Lost” message then update game accordingly.
* Update Grid with (1) adversary move retrieved from server or (2) player move from HTML

Grid – helper object for GameStatus 🡪 strings gebruiken

2 dimensional matrix with:

0 – no ship, no misses yet

1 – no ship, and a miss

2 – ship, no hit yet

3 – ship, hit

Function numberOfHits (returns number of hits – counts # of ‘3’s)

View – this is the “View’ part of the…

* adversaryMOve – update view based on adversary move, enable player screen
* mouseclick – gameStatus update, disabled player screen