**BattleShip**

**Required constants:**

• numOfBattleShips(5)

• boardSize(28X8)

• Number of spaces for each type of battleShip ( take up two divs each)

**required variables used to track the state of the game:**

• Location of battleships (inside object)

• Which part of battleship is hit/what is available to be hit(inside objectClass)

• How many battleships are dead/alive(let totalAlive)

• Who is the winner(let winner)

• Whose turn(let playerTurn = bool)

• Name(let playerName = String)

• Score(let playerScore=0, let compScore =0)

• Time for turn(limit time to 10 seconds… will use setTimeout function)

**Upon loading the app should:**

• Initialize the game board and set all variables to null or zero

• Enter name screen to save name

• Name will be passed to player name for scoreboard and num of ships subheading

• Player gets to decide where ships go

• Computer uses math.random to place ship

• Math.random to decide who goes first

• First player chooses a div, if its a hit increase score by 10 points div replaced with fire pic/ if not hit same fire pic but no points

• When a total ship is destroyed, playerShips says destroyed or grayed out.

**WIN LOGIC**

• When all ships on one side are destroyed, the player wins.

• Win gets displayed on screen, Board goes to hidden

• New animation is added

**Computer AI**

• Computer uses Math.floor(Math.random()\*numOfSquares+1) to determine the first move. All ids of divs are pushed into an array.

• If hit ai tries one div on top, if hit keeps going until no more hit. If no battle ships are destroyed and hits were made vertical then the next hit will be below initial value.

• If hit and top no hit? Then try bottom

• If hit && !topHit && !botHit the rightHit

• If hit && !topHit && !botHit & !rightHit then leftHit

• If No hit then next turn value of previous choices removed from possible choices and math.random again.

**6) Handle a player clicking the replay button:**

Reset all values and start from initial