MEETING 1 MINUTES:

* repo:
  + join
  + clone
  + commit/merge standards
    - commit to your own branch
    - push and merge when it's working
    - master branch must always compile
  + create empty files, pull requests
* planning
  + classes
  + interface/display
  + division of tasks
    - Connie: Ship class, Fleet class
    - Ethan: Exec class
    - Archana: HTML/CSS (index.html)
    - Malena: Admiral class
    - Kyle: Grid class
  + variable naming convention:
    - CONST\_NUM: allcaps separated by underscores
    - variableName: camelCase
    - functionName(): camelCase
    - Class: capitalized

RESOLVED:

* Class/object structure (see below)
* Everyone looks into JS classes, modules, event listeners, etc.
* Current division of tasks. This can be modified later as needed.

PROPOSED:

* The “stuff” file is now config.js and will contain constants, assuming we need them. [Pending asking Gibby for clarification on project requirements.]
* Using events and event listeners to prompt the running of methods within different objects/classes.

CLASSES/OBJECTS. The specifics of what methods are needed for each class are WIP and can be modified as needed. Diagrams and details below. Our original whiteboard notes are included in the meeting folder.

* Executive
* Admiral (formerly “Player”)
* Grid
* Fleet (newly added, contains the Ship objects)
* Ship

ADMIRAL

* METHODS/FUNCTIONALITY NEEDED
  + Place ships
  + Update Grids based on player interaction

GRID (firing map)

* MEMBER VARIABLES:
  + 2D array of sectors
* METHODS/FUNCTIONALITY:
  + getters, setters
  + point to shipMap
  + read coordinates passed from Ship
  + Fleet shooty thing?

GRID (Shipmap)

* MEMBER VARIABLES:
  + 2D array of sectors
* METHODS/FUNCTIONALITY:
  + getters, setters
  + read coordinates passed from Ship
  + Fleet shooty thing?

ADMIRAL [same as left]

GRID (firing map) [same as left]

GRID (Shipmap) [same as left]

FLEET

* MEMBER VARS:
  + numShips
  + numShipsAfloat
  + Ship (as many as specified)
* METHODS/FUNCTIONALITY:
  + Create Ships
  + sinkShip
  + populate Ship coords

SHIP

* MEMBER VARS:
  + Bool afloat
  + Size
  + numHits
  + array cords
* METHODS:
  + setters & getters
  + setShipCoords

FLEET [same as left]

SHIP [same as left]

EXEC

* METHODS/FUNCTIONALITY NEEDED
  + Run
  + Initialize game: create 2 Admirals; take in player names and # of ships; etc.
  + Place ships
  + Handle turn switching
  + Handle player inputs/interaction
  + Display alerts to players
  + Modify the HTML/CSS when the display needs to change