Algorithm for Battleships.

Aim

- The produce a battleships game to be played against the computer in the command line.
- The computer should display two grids. The players ships should be visible.
- The player takes a shot by entering x and y coordinates.
- The computer takes a shot by choosing a grid square at random.

Symbols used:

^ - grid square (It's a wave...).
x - Hit
. - miss

Imports:

import random

class GameSize:

Regulates the size of the game, depending on player input.

method choose_game(player choice):

Outputs values for the below variables based on player choice.
Alternatively has code to quit the game if the player chooses.

x_axis, y_ axis, number_of_ships.

method generate_grid(x_axis, y_ axis):

Takes x and y axis outputs from above and generates a list of lists

return list_of_lists

method add_ships(list_of_lists, number_of_ships)

Takes the list output of the second method and adds a number of 'ships' at random places. the number of ships is determined by the first method.

return game_grid

function take_shot(game_grid, x, y):

Function to resolve player shot.
Takes player guess of x and y axis and compares them with the grid.
Will update the list with either a 'x' or '.' to indicate a hit or miss.
NB. will need to deduct from one each of the player inputs to account for list indexing.

return game_grid, hit/miss

function enemy_shot(game_grid, x_axis, y_axis):

Function for enemy shooting. Picks a random grid location
calls the take_shot function and passes the random grid location.

return game_grid, hit/miss

function display_battlespace(game_grid):

Displays the grid in a viewer freindly format.
Adds a newline character to the end of each x axis list.
Extracts the contents of the list and concatinates them together in a string.

return battlespace

function hide_fleet(battlespace):

Takes the battlespace generated for the enemy by the display_battlespace function above.
replaces any ship characters with wave characters.

function combat(fleet_size, enemy_ships, friendly_ships, x_axis, y_axis):

Loop for combat
Runs until one side looses all their ships.

Player is asked for input for x and y axis coordinates.
Checks input is a number between 1 and va;ue of x/y axis.
Calls take shot function, passes player unputs as parameters

Calls enemy shot function.

function main_game():

The main game function.
Prints rules, options for game size or option to quit to screen.
Validates player option and calls methods in the game size class to generate game size.
Calls the combat function
Takes combat function outputs and updates player and computer grid.
Deducts 1 from ship numbers if hit.
Prints feedback and updated battlespaces to screen.
Gives player option to quit or play again once game ends.

Start of game. main_game() called here.