Project 2 Code Documentation

Quick Start

To run the program, run **python main.py** on Windows 11.

Tasks:

Initial Refactoring Changes

1) Changed recursive logic for input validation to iterative logic
2) Abbreviated “horizontal” and “vertical” to “H” and “V” for better UX
3) Added turn counter for Game class to keep track of number of turns. Individual player plies can be calculated from this number and information about who the current player is

Implementing AI

- Enable users to select between 1 or 2 human players in the beginning
- If 1 human player, then the opponent will be an AI
- One among 3 difficulties (Easy, Medium, Hard) can be chosen for the AI opponent
- AI opponents are modeled as Player objects with is_ai set to True and an ai_difficulty field
- Gameplay loop has been updated such that all functions requiring player interaction can incorporate the AI opponent too
- The selection algorithm for each of the difficulties can be found in player.py under the get_shot_placement function

Implementing Custom Addition

- Players can view their individual stats using the “P” command during their turn
- The view_stats function in the Player class handles displaying the player stats. It makes use of a get_stats function to get the stat values
- Players can display the overall scoreboard using the “S” command during their turn
- display_scoreboard function handles displaying the scoreboard. It relies on the get_stats output from both player objects in the game

Files:

game.py

This program utilizes the Player class, and the functions from validifier. It brings everything together in order to run a game of Battleship. The Game class manages player turns, ship placement, and shot processing. It handles input validation through validifier.

main.py

This file implements the Game and Player classes from their respective files. This creates a 2-player battleship, letting players input coordinates to both place and sink ships.

player.py

This file is utilized by the game file in order to run Battleship. It focused on creating a lot of needed values for the game file, like the boards of ships and shots. It also checks the state of the game, with is_sunk, and is_all_sunk, as well as printing the board.

validifier.py

This file is used by the Game class in order to validate the placement of ships. It also determines the validity of shots.