Battleship Project 2
Software Requirements Document

Team 42

Project Overview

Administrative

Instructor: Prof. David Johnson
GTA Supervisor: Tianxiao Zhang
Semester: Fall 2024
University: University of Kansas
Project Owners: Team 19

Project Contributors:

- Manoj Turaga
- Connor Forristal
- Ceres Botkin
- Henry Marshall
- Elizabeth Chanel

For further information about this project's development, please contact the contributors listed above.

Refactoring

The initial implementation of Team 19's Battleship code was refactored to have logical decisions in main.py and anything ancillary in board_util.py. This is because we determined the code was not modular enough to allow us to implement new features. The following modules are now the modules in the codebase

Main

The main module is the launching point of the entire program. In this case, the main module is also the executive module, controlling the flow of execution of the code. The flow of logic is presented below.

1. Main
    a. See Bullet Information
    b. Start Game
        i. Get number of ships
            1. Select if playing AI
                a. Place Ships
                    i. Make attack
                        1. If all ship destroyed, end game
                    ii. See Info
                    iii. Concede

Scoreboard

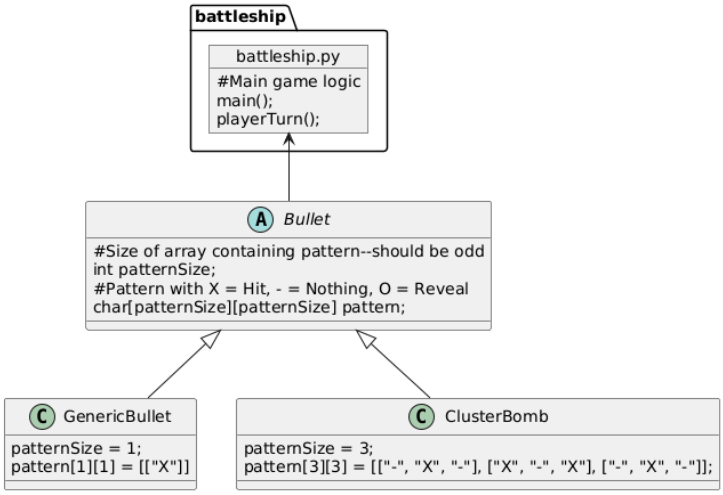
The Scoreboard module is just a module that allows the game instance to keep track of winners across many different instances of the battleship game

Events

The Events module is a standard location of key events that can occur during the context of the game's runtime. These include events such as displaying message, getting user input, and clearing the terminal

Bullets

The bullets are what we are claiming as the required feature to the project. The bullet module is a configurable module that can create new bullets that can attack an N x N grid with custom action taken on each cell. The following image shows the UML diagram for this implementation.



Visuals

The Visuals module is an ancillary module to the bullet module. The visuals module will display general information about how the new bullets work and what bullets each player currently has.

AI

The AI module encapsulates the logic of the AI player, allowing the user to play against a computerized player.

Additions

The following additions were made to Team 19's Battleship implementation

AI

The game shall feature an AI with 3 levels of difficulty that the player can opt to use instead of playing against another player.

The AI shall place ships onto the board at random.

The AI shall be constituted with the following difficulty levels:

1. Easy: The AI shall select special bullets by their lowest rank and make attacks at random
2. Medium: The AI shall select both special bullets and attack locations at random. If a random attack hits a ship, then the AI will continue attacking orthogonal coordinates until a ship is sunk.
3. Hard: The AI shall select special bullets by their highest rank and make attacks knowing exactly where the opponent's ships are

In essence, the AI does anything that a regular player can do

Special Bullets

Special Bullets are a way of enhancing the way players can attack the opponent. Our game offers a variety of bullets that can be used but the logic for special bullets is as follows

Special Bullets are composed of an N X N grid where N % 2 = 1. Each cell is then independently able to execute an attack on the other player. Each cell can take the following types of attacks

- The bullet can choose to fire a real attack at the coordinate
- The bullet can choose to reveal the location of the coordinate
- The bullet can choose to not do anything to this coordinate

There are variations of these three rules presented in the bullets that are provided, but since it is fully configurable, we will not define the bullets that are implemented