Rebekah Blazer, Ryder Gallagher, Jack Nealon

CPSC224 – Dr. Aaron Crandall – Gonzaga University

04/05/2023

----- Final Project pt 2 -----

1) Project description

The objective of this group project is to develop a program to run a game of Battleship. First, we will develop a written plan on how we want to approach our program. This includes UML charts, sequence diagrams and other functional requirements identified before starting to code. This will help with smooth development and minimal errors throughout the project.

The main structure of the game will have two players play against each other on a 10x10 game board. Each player will have 5 ships: one carrier (size 5), one battleship (size 4), one cruiser (size 3), one submarine (size 3) and one destroyer (size 2). The ships will be placed on the game board and each player will take turns guessing where the opposing player's ships are. Each player will take turns selecting coordinates to attack. If the player's guess hits one of the opposing player's ships, the coordinate square will be marked with an 'X'. If the attack misses, the coordinate square will be marked with an 'O'. A battleship is sunk when there is an 'X' on every coordinate square that the ship occupies. The game will continue until one player has sunk all of the opposing player's ships. At the end of the game, the winner will be displayed and the option to play again will be given to the user.

The program will utilize an external GUI, which will display each player their respective game boards and provide an interactive experience for the players. The game will be developed using the java programming language and will utilize git and GitHub for collaborative file sharing.

2) Functional Requirements

Player can Take Turn	
Priority	High
Purpose	This application shall allow the user to guess the coordinates of other player’s ships
Inputs/Needs	• Guessed coordinates • Ship locations of other players • Already guessed coordinates
Operators/Actors	• Player • Ship • Board
Outputs	• X or O depending on whether they ‘hit’ other player’s ship or miss it • If the user tries to guess already guessed coordinates, told to try again

Game Start	
Priority	High
Purpose	This application shall start the play. It will allow users to place their ships and then start taking turns.
Inputs/Needs	• Board layout • Players selection of where to place ships • Who Player 1 is (take turn first)
Operators/Actors	• Player • Ships • Board
Outputs	• All ships placed with no listed hits • Board is blank and clear of any hit-or-miss markers

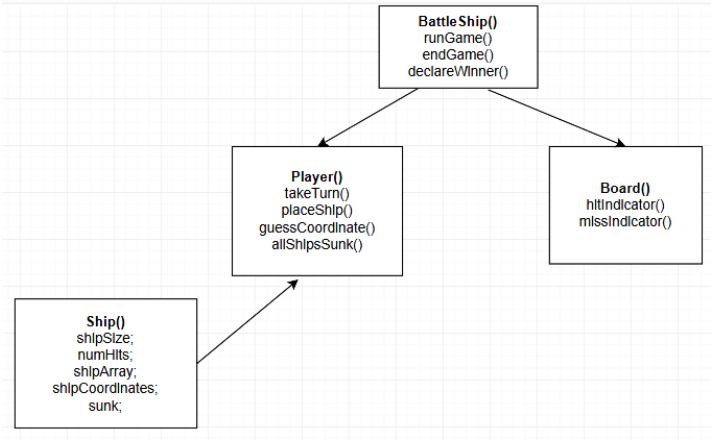
Ship Sinks	
Priority	Med
Purpose	This application shall remove the ship from play and subtract the number of ships in play from Player who lost the ship
Inputs/Needs	• Number of ships in play for Player • Number of hits ship received • Ship size
Operators/Actors	• Ship • Player
Outputs	• Number of ships the Player has on board decreases by 1

Declare Winner	
Priority	Low
Purpose	This application shall end the game and declare a winner
Inputs/Needs	• Number of ships each player has on the board
Operators/Actors	• Player
Outputs	• Game ended when a player has 0 ships in play on the board • Player with more than 0 ships at end of the game declared winner

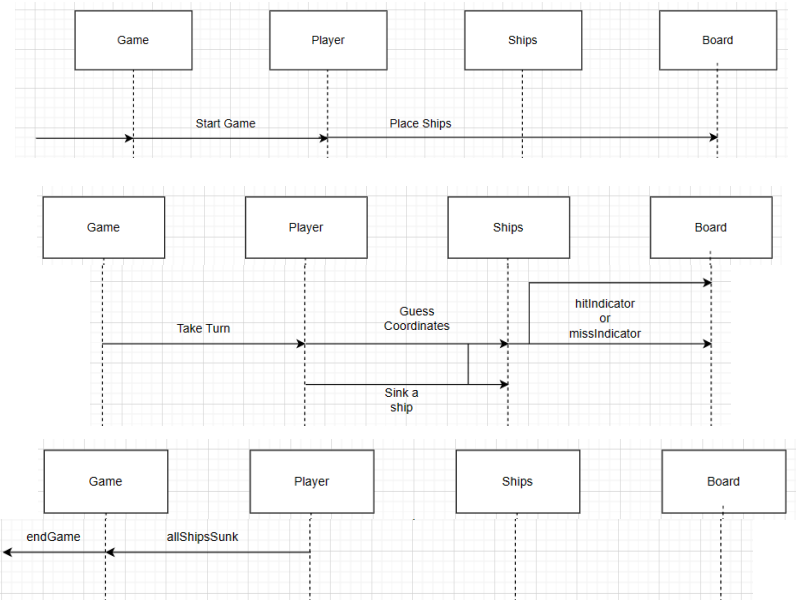
- 2 individual players
- Players take turns
    - Players can guess coordinates
- Players can place ships
- Players can see already guessed coordinates
- Players can see when they ‘hit’ a ship
- Ships can be ‘hit’
- Ships can be placed
- Ships can ‘sink’
- Players can win or lose

3) UML for:

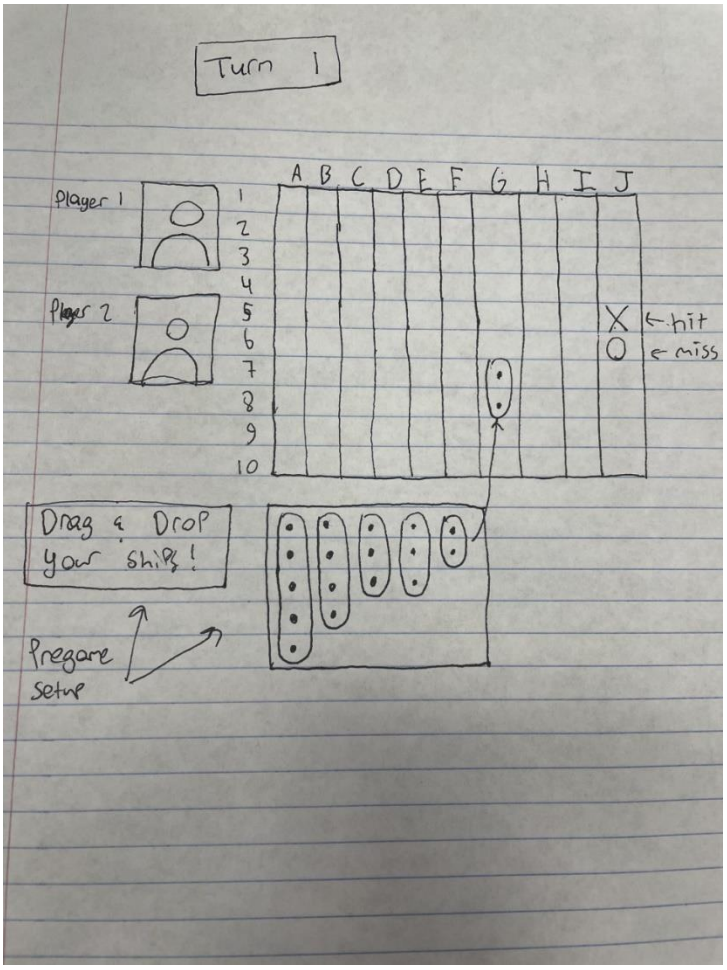
a) System objects you’ll be making



b) Primary 2-3 sequence diagrams for the game interface



4) Sketch of your user interface design



5) Overall plan and project schedule (*subject to change as project develops)

* Have these things (or at least the first version) done by the listed date.

* Teammate assignments are subject to change throughout the project.

a. March 28th –
    a. Setup Group Communication and Initial Meetings
    b. Create discord group chat/exchange numbers
    c. Create and test navigation of the GitHub teams page and the repository.
    d. Begin working on the project
b. April 5th –
    a. Work on and complete the starting schedule to plan out the project before coding
    b. Project Description
        i. Assigned to Jack
    c. Functional Requirements
        i. Assigned to Rebekah

    d. UML and Sequence Diagrams
        i. Assigned to Rebekah
    e. UI Sketches
        i. Assigned to Ryder
    f. Figure out major classes and determine some variables and methods.
        i. Using the noun-verb methods (nouns: classes/variables, verbs: methods)
    g. Overall plan and project schedule
        i. Assigned to Jack
    h. Basically, this assignment (Final Project part 2)
        i. Figure out our plan before beginning to code
c. April 10th –
    a. Turn in plan to implement system testing
        i. Determine system behaviors and plan out how to test each behavior
            1. Rebekah – Player class test cases
            2. Ryder – Ship class test cases
            3. Jack – Board and Battleship class test cases
d. April 15th –
    a. Begin coding the project
        i. Initial Assignments:
            1. Rebekah – Player class
            2. Ryder – Ship class
            3. Jack – Board and Battleship classes
        ii. Additional issues will be assigned as they arise
    b. Submit individual peer evaluations #1
e. April 28th –
    a. Initial versions for coding and testing are completed (majority of program works)
    b. Revise code and tests fixing bugs and errors as they arise
f. May 1st –
    a. Submit final version of code
    b. Develop final report
    c. Start on final presentation
        i. Describe the project timeline
            1. Issues and Delays
            2. Project Milestones and when they were hit
            3. Etc.
        ii. Use weekly meetings to rehearse presentation
g. May 9th –
    a. Present final project during finals block (1-3pm)
h. May 11th –
    a. Submit final report
    b. Submit individual peer evaluations #2
    c. Submit final project presentation