CPSC 224 Final Project

PROJECT PLAN
Nov 9th, 2024

Battleship (With a Twist)

Miss Puff's Boating School

Prepared by:

Jaxon Froderberg Aria Wood Damon Lewis

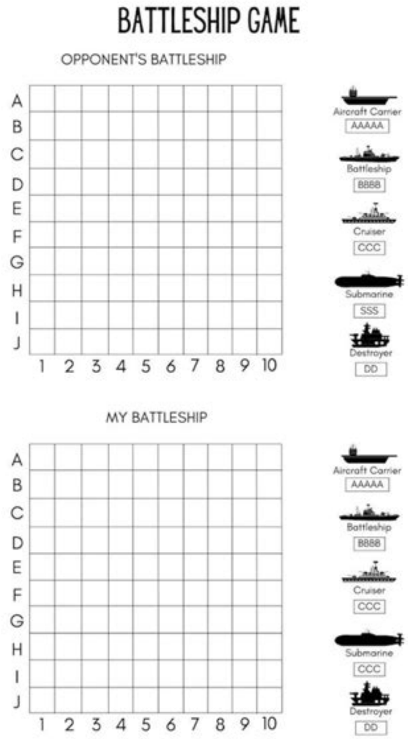
1 Project Overview

1.1 Project Summary

For this project Miss Puff's Boating School is creating a copy of the world-famous game "Battleship"!

Battleship is a traditionally two-player guessing game played on either matching boards or pieces of gridded paper. The game begins with each player setting up several ships of various sizes within a set grid of spaces, which are hidden from the other player by some form of divider. Each player will then take turns calling out individual grid spaces to check on their opponent's grid in order to find and "hit" their ships.

Once all ships belonging to a player are hit and "sunk", the game ends and the player who still has ships remaining unsunk on their board is declared the winner.



In our version of the game, we have added several additional twists and gameplay mechanics to the base gameplay in order to enhance the game's complexity and strategy.

Some of the twists that we are experimenting with include:

- Carriers being able to recon part of the map, potentially revealing enemy ships
- Battleships being able to fire multiple shots at a time
- Submarines being hidden from aerial reconnaissance
- Submarines being sunk in one hit from other submarines and destroyers
- Time limits for play, similar to Chess

2 Project Requirements

2.1 Major Features

Feature	Description
GUI: Main game display	The main game display will show all necessary information for both players during each turn, including both 10x10 boards, player scores, ships (with abilities / current status), and turn order. This GUI will also serve as the main method for user interaction with the game, allowing players to click on spaces to check, setup ships on the board, etc.
GUI: Introduction screen	The Introduction screen will display the name of the game being played, and provide the players an opportunity to adjust settings as needed. This will also help mark when the game actually starts by transitioning to the main game display.
GUI: Ending / winner's screen.	The ending screen serves a similar purpose to the introduction screen, and marks when the game is over. It also will display which player won the last round of battleship and show final user scores.
Game Board	The game board will be built around a class, which will generate battleship boards consisting of 10 by 10 grids of spaces. Each space will store important information on: ● Whether a ship is located on it, and what the ship's status is (type, direction facing, hit, etc.) ● Whether this space has been checked with a "hit" or not. ● Whether this space is visible to a player or not.
Setup phase	Before each game begins, all players will be given a chance to enter their usernames and choose where they would like to place their ships on their own board.
Main Game Loop	The game will go through the setup phase, allow players to make actions (firing, special abilities) and receive feedback on what happened following the action. The game will move

	between phases and screens for setup, gameplay, and the ending.
Settings menu	The game will provide a settings menu, in which players will be able to do the following actions: ● Change gameplay music / sound volume ● Change board visibility ● Enable/disable turn timers ● Enable/disable certain ship abilities (This will have an additional drop-down menu).
Players	Each player will have a unique name that will be displayed during play. The performance of the player will be tracked throughout gameplay and will be displayed in the end screen.
Special Abilities	Ships will have unique characteristics and strengths/weaknesses. Some ships will have special actions that players can use. The game will display each ship and allow players to select and choose what ship to use and what ability to perform, if any.

3 Project Game Design

3.1 Initial User Interface Design

The user interface will be split into 4 main screens / frames: Introduction and ending screens, a main game screen, and a settings screen.

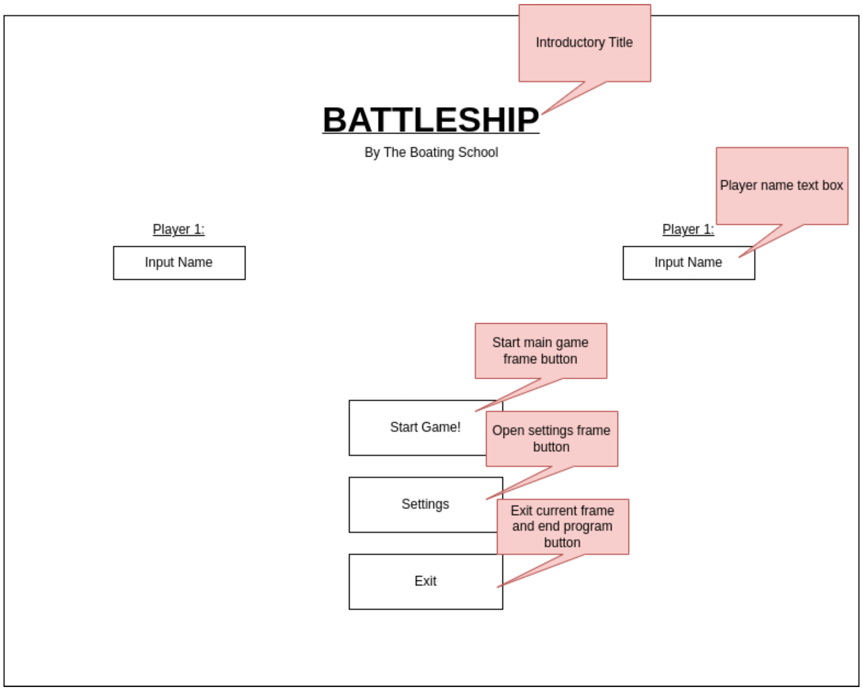
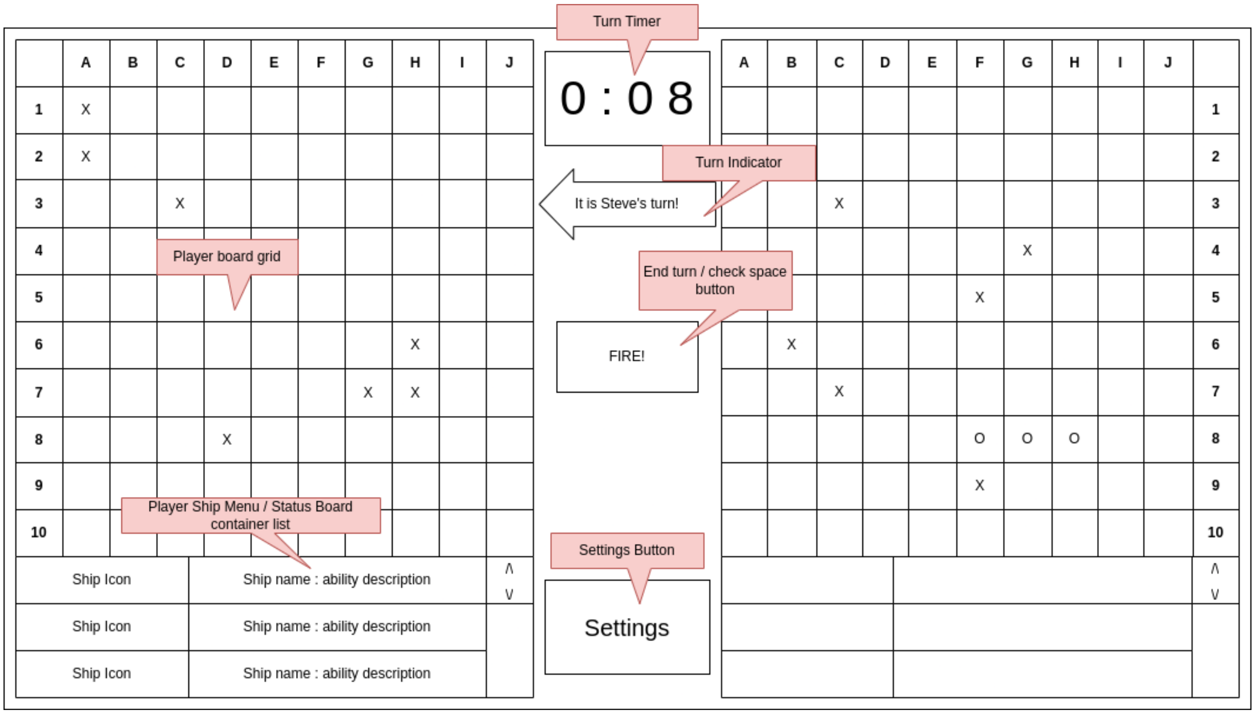


FIgure 1: Introduction frame for Battleship, with various options buttons.

The Introduction screen will display our game's title in bold lettering, and give users a visual indicator as to what game they are playing. Below this will be two text boxes prompting them to enter usernames for the game. The three buttons below, "Start Game!", "Settings", and "Exit", will hold code for opening the main game frame, opening the settings frame, and exiting the whole program respectively.



	A	B	C	D	E	F	G	H	I	J
1	X									
2	X									
3			X							
4										
5										
6								X		
7							X	X		
8				X						
9										
10										

Ship Icon	Ship name : ability description	∧ ∨
Ship Icon	Ship name : ability description	
Ship Icon	Ship name : ability description	

A	B	C	D	E	F	G	H	I	J	
										1
										2
		X								3
						X				4
					X					5
	X									6
		X								7
					O	O	O			8
					X					9
										10

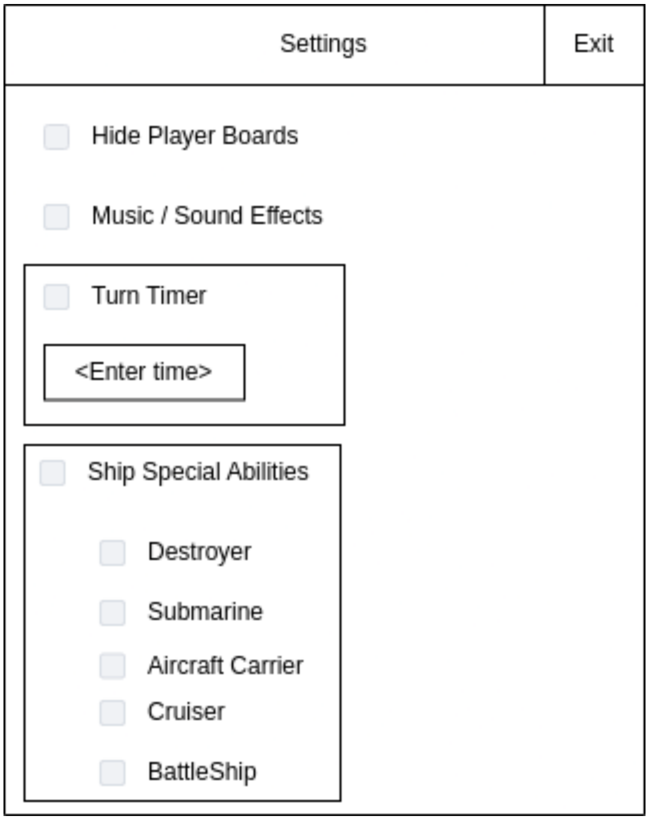
		∧ ∨

FIgure 2: Main game frame of Battleship, with player boards and various buttons.

Once the players have entered their usernames and clicked the “Start Game!” button, the main game frame will focus on displaying the actual battleship game. Both player’s full 10 by 10 grid boards will be displayed, with a drop-down menu placed below they can access to place, use, or otherwise check the status of their ships. Between both players' boards will be the scoreboard, timer (if enabled), and a turn indicator, showing which player’s turn it is.

When one player is having their turn, the opponent’s board and ship menu will be hidden or faded out. In this way the game will be designed around hot-seat gameplay. Once a player has finished selecting a space on the opponent’s board to check, they will confirm their choice by pressing the “Fire!” button, which will check the space, update both boards, and change which player is having a turn.

Once again, a settings button will be displayed as well for ease of access to configurable elements of the game.



□ Hide Player Boards

□ Music / Sound Effects

□ Turn Timer

<Enter time>

□ Ship Special Abilities

□ Destroyer

□ Submarine

□ Aircraft Carrier

□ Cruiser

□ BattleShip

FIgure 3: Settings frame for all battleship settings

The settings screen should be accessible from any of the other windows listed at any time via buttons, and will be responsible for holding toggleable settings for the game’s volume, board-hiding mechanic, special ship abilities, and turn timers.

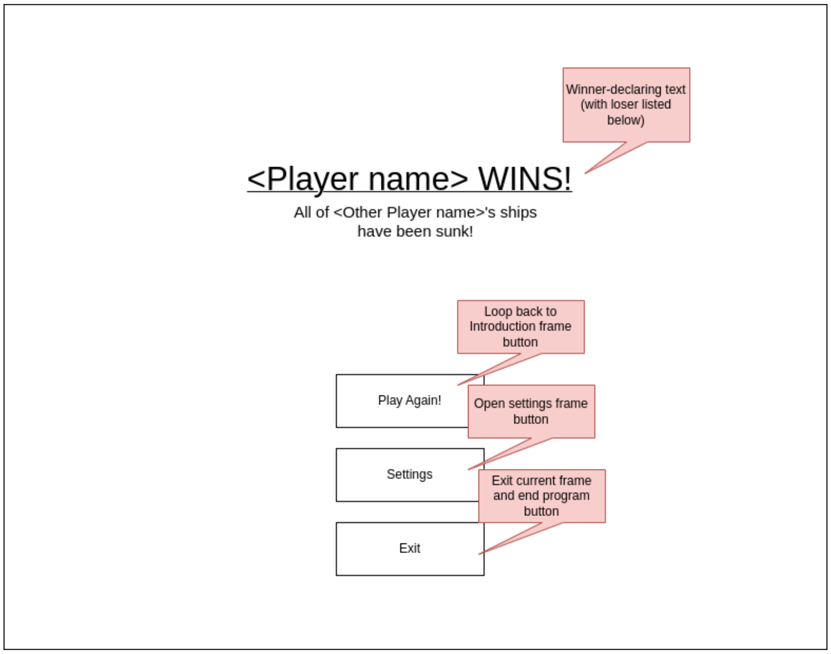

Winner-declaring text (with loser listed below)
<Player name> WINS!
All of <Other Player name>'s ships have been sunk!
Loop back to Introduction frame button
Play Again!
Open settings frame button
Settings
Exit current frame and end program button
Exit

FIgure 4: Ending screen with winning player display and options buttons

Once a player has successfully sunk all ships belonging to their opponent, the main game frame will be replaced by the ending frame, which will declare the player as the winner and prompt them with buttons to play again and loop around to the Introduction frame, access settings, and exit the program.

3.2 Initial Software Architecture

We have divided our project into a few main classes based around ships, boards, with the actual game itself doing most of the work as far as running the game. The settings class is a singleton and will be available globally for all systems. One of the main principles behind our architecture is to separate the data and business logic from the user interface itself. The two interact primarily through the game class, with functions and callbacks being used on either side to communicate state and handle user input.

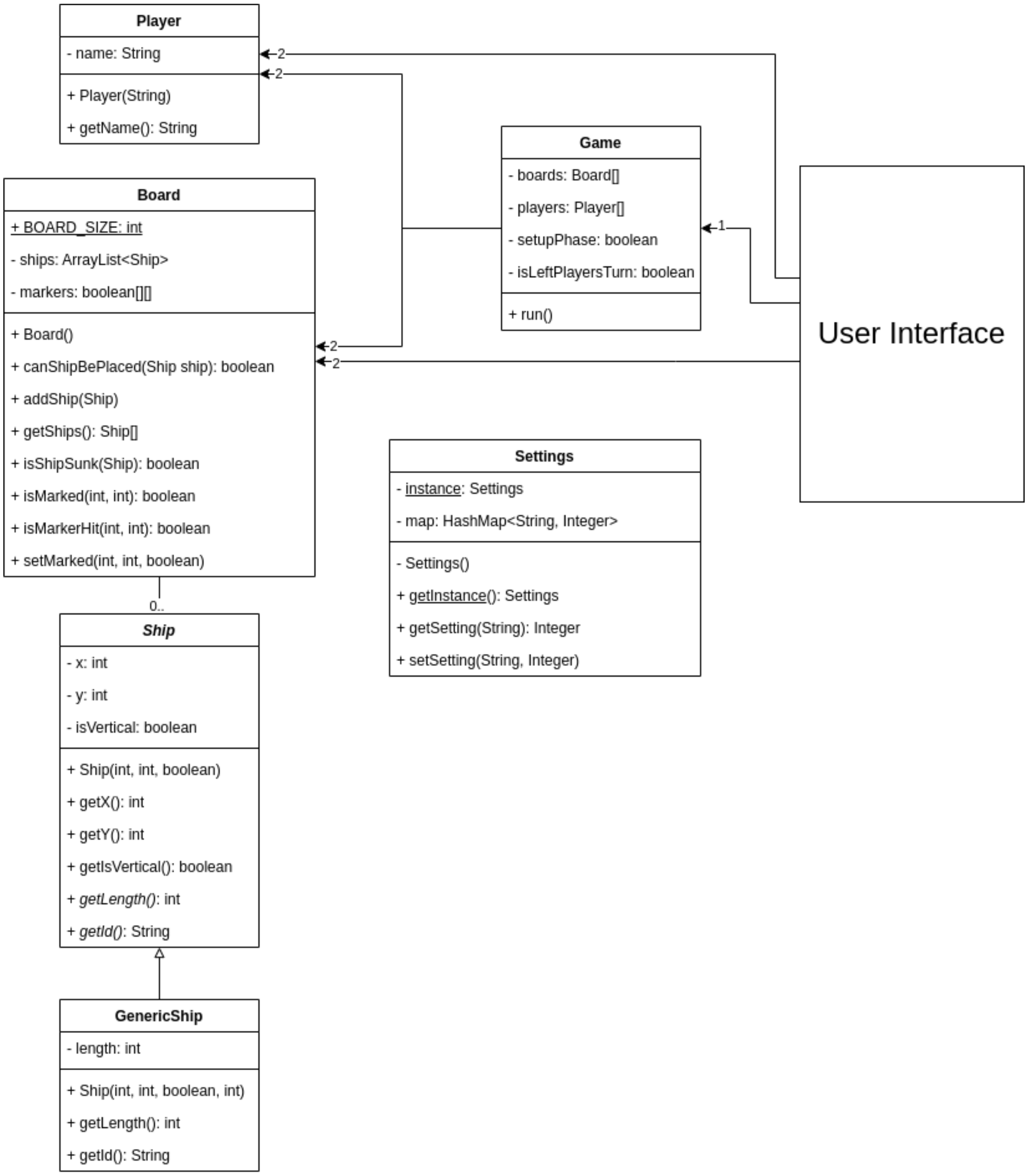


FIgure 5: Battleship UML diagram of all classes / objects

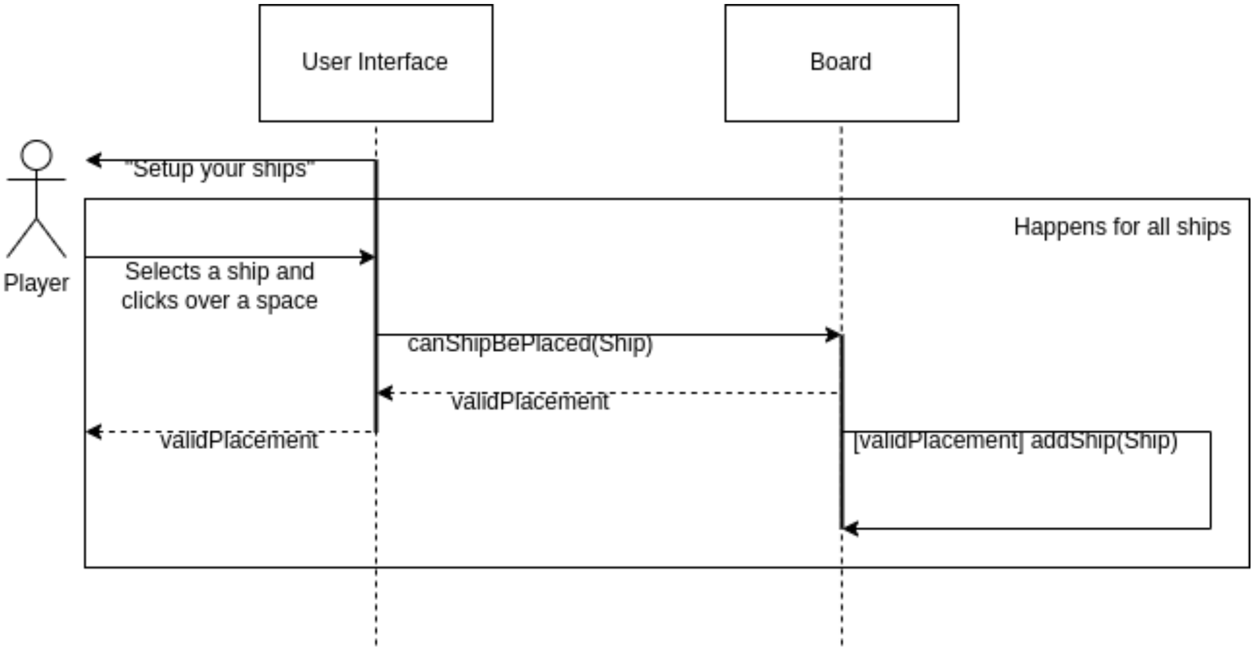


FIgure 6: Sequence Diagram for setup phase of the game

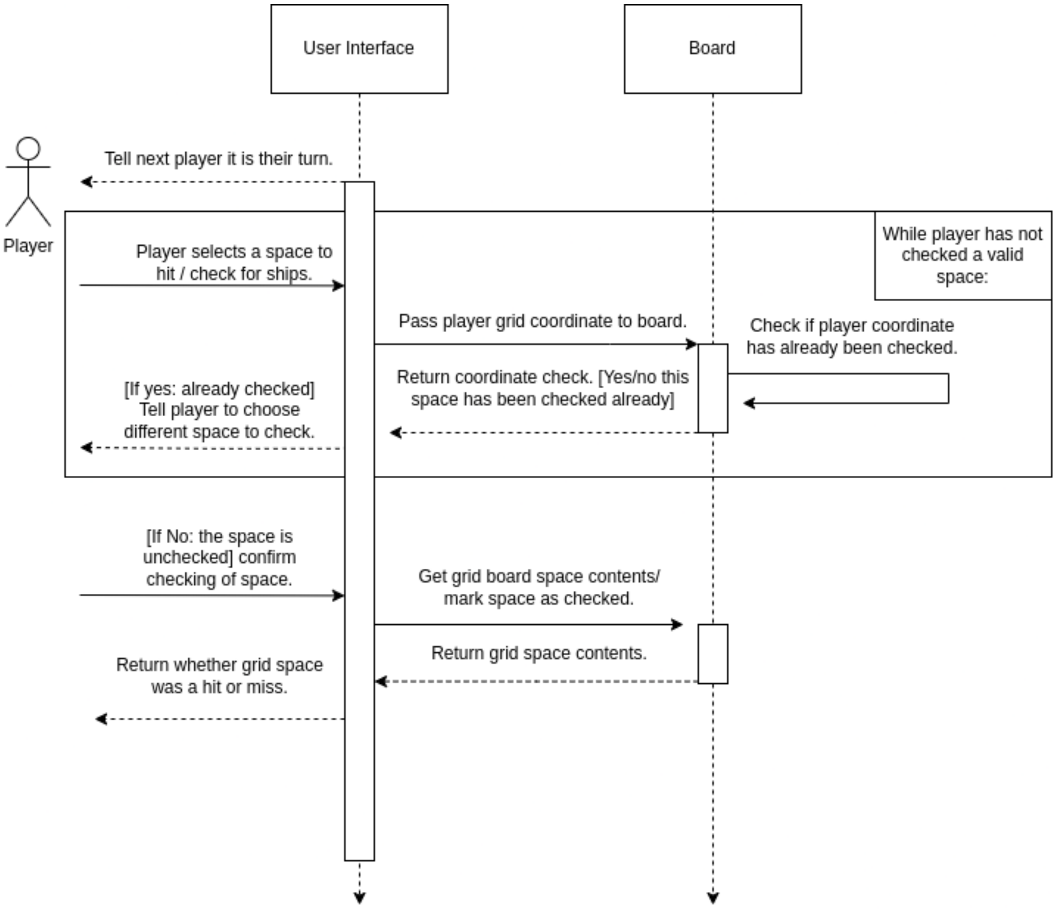

User Interface
Board
Tell next player it is their turn.
Player
While player has not checked a valid space:
Player selects a space to hit / check for ships.
Pass player grid coordinate to board.
Check if player coordinate has already been checked.
Return coordinate check. [Yes/no this space has been checked already]
[If yes: already checked] Tell player to choose different space to check.
[If No: the space is unchecked] confirm checking of space.
Get grid board space contents/ mark space as checked.
Return grid space contents.
Return whether grid space was a hit or miss.

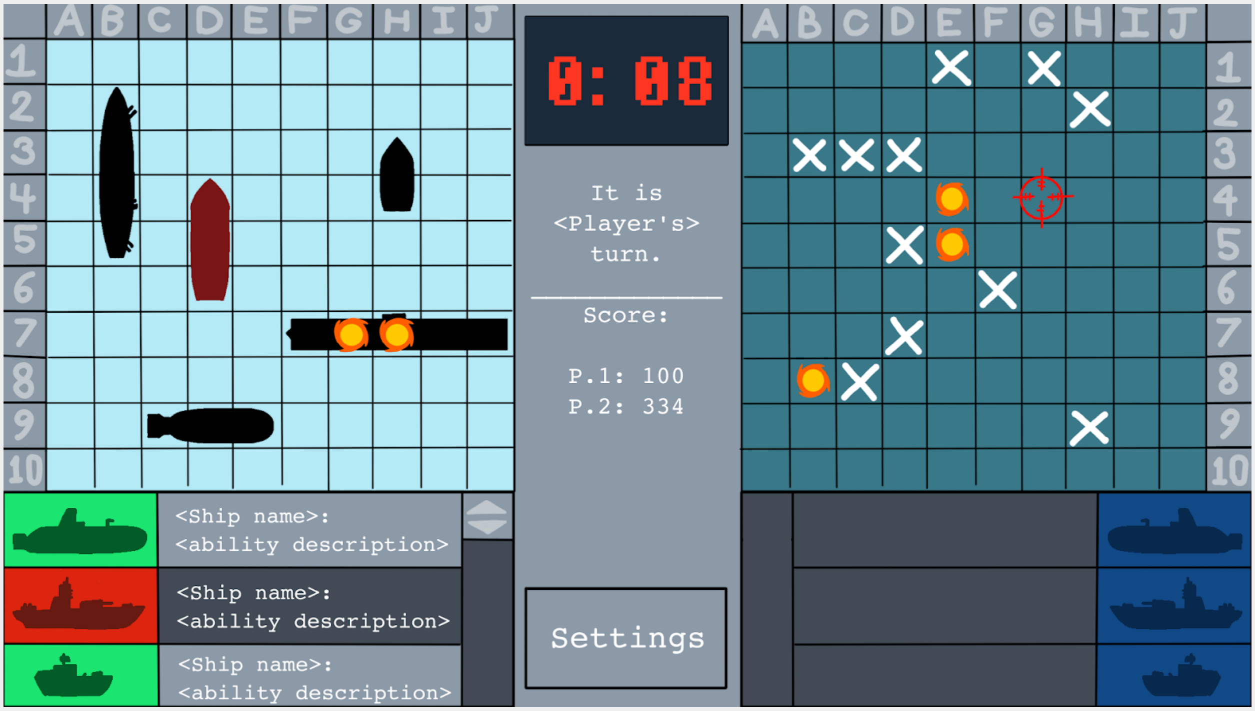
FIgure 7: Sequence Diagram for a single player turn / space-checking phase.

4 Project Schedule

Milestone	**Description**	**Target Completion Date**
Project Plan	The Project Plan will be complete and submitted.	November 9th
Basic Game	The basic components of the game, including players, the game loop, and GUI will be complete. A simple game of Battleship should be playable. There should be few serious bugs and early unit tests should be in place. **Tasks / Issues:** ● Board functionality - Aria - 2 hr ● Player class - Aria - 1 hr ● Settings class - Damon - 1 hr ● Game class - Jaxon - 2 hr ● Game loop - Jaxon - 5 hr ● Setup phase - Aria, Damon - 2 hr ● GUI - Damon - 10 hrs	November 15th
Near-Complete Game	All core features of the game should be playable, including special abilities. A complete game should be playable all the way through. Only a few small features and polish should be missing. **Tasks / Issues:** ● Individual ship classes - Aria - 2 hrs ● Special abilities - Jaxon, Aria - 4 hrs ● GUI selection of abilities - Damon - 2 hrs ● Introduction screen - Damon - 1 hr ● Settings screen - Jaxon, Damon - 2 hrs ● Ending screen - Damon - 1 hr	November 27th
Code Complete	The code for the project will be fully complete in the main branch. The code	December 2nd

	will be fully documented, gone through unit and integration testing, and will be free of nearly all bugs. **Tasks / Issues:** ● Documentation - Damon, Aria, Jaxon - 2 hrs ● Unit Testing - Damon, Aria, Jaxon - 2 hrs ● Application Testing - Damon, Aria, Jaxon - 4 hr s ● Polish - Damon, Aria, Jaxon - 6 hrs	
Presentation Complete	The Presentation will be complete and submitted. Team members will have practiced presentation and know when they will be speaking. **Tasks / Issues:** ● Outline - Damon, Aria, Jaxon - 2 hrs ● Visual Aid - Damon, Aria, Jaxon - 2 hrs ● Practice - Damon, Aria, Jaxon - 1 hr	Week of December 2nd
Final Report Complete	The Final Report will be complete and submitted. **Tasks / Issues:** ● Outline - Damon, Aria, Jaxon - 2 hrs ● Draft - Damon, Aria, Jaxon - 4 hrs ● Draw Figures - Jaxon - 4 hrs ● Revise - Damon, Aria, Jaxon - 2 hrs	Finals Week

Appendix



It is <Player's> turn.

Score:

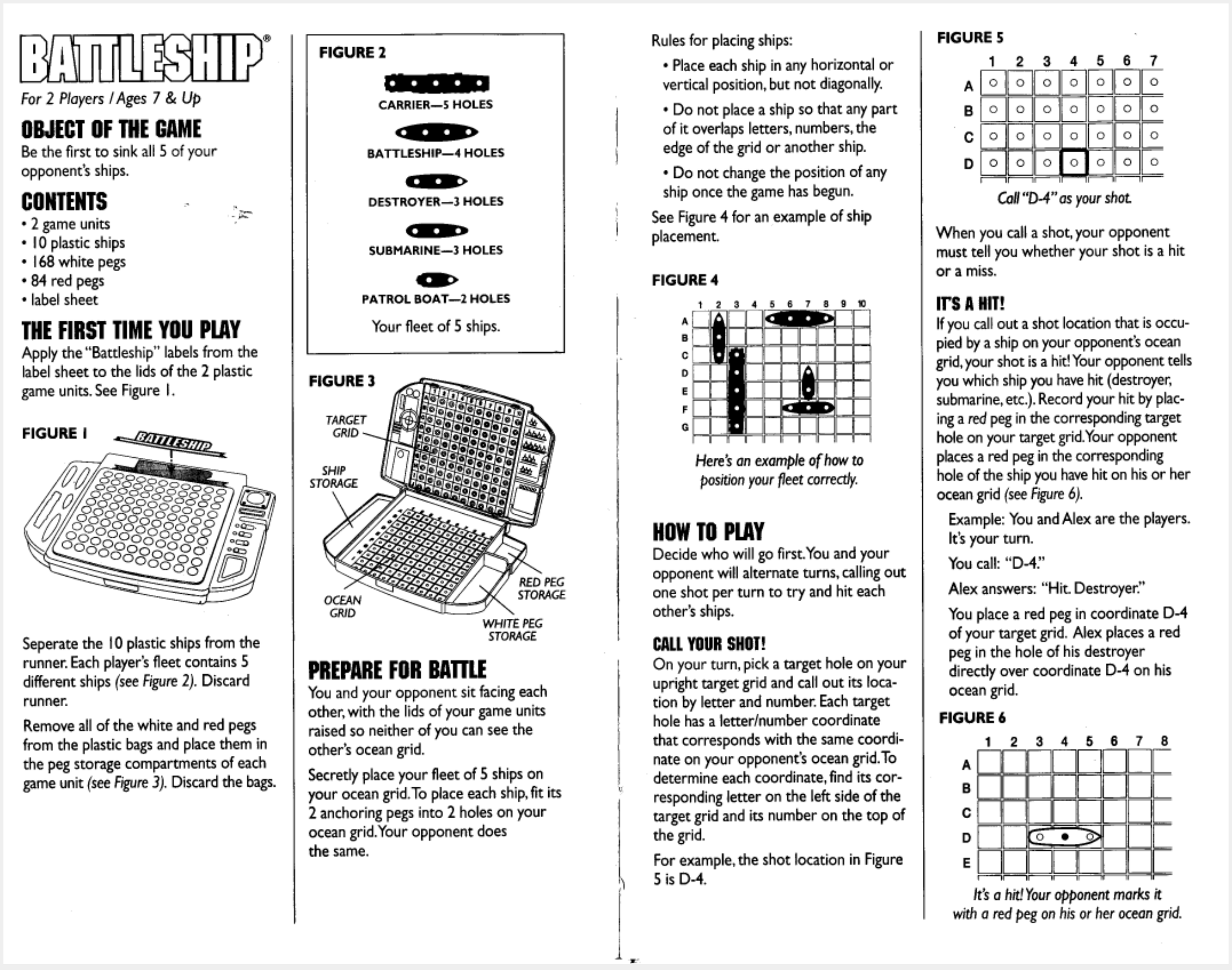
P.1: 100
P.2: 334

<Ship name>:
<ability description>

<Ship name>:
<ability description>

<Ship name>:
<ability description>

FIgure 8: More stylized version of what the complete Battleship game will look like



BATTLESHIP®

For 2 Players / Ages 7 & Up

OBJECT OF THE GAME

Be the first to sink all 5 of your opponent's ships.

CONTENTS

- 2 game units
- 10 plastic ships
- 168 white pegs
- 84 red pegs
- label sheet

THE FIRST TIME YOU PLAY

Apply the "Battleship" labels from the label sheet to the lids of the 2 plastic game units. See Figure 1.

FIGURE 1

BATTLESHIP

Seperate the 10 plastic ships from the runner. Each player's fleet contains 5 different ships *(see Figure 2)*. Discard runner.

Remove all of the white and red pegs from the plastic bags and place them in the peg storage compartments of each game unit *(see Figure 3)*. Discard the bags.

FIGURE 2

Your fleet of 5 ships.

FIGURE 3

PREPARE FOR BATTLE

You and your opponent sit facing each other, with the lids of your game units raised so neither of you can see the other's ocean grid.

Secretly place your fleet of 5 ships on your ocean grid. To place each ship, fit its 2 anchoring pegs into 2 holes on your ocean grid. Your opponent does the same.

Rules for placing ships:

- Place each ship in any horizontal or vertical position, but not diagonally.
- Do not place a ship so that any part of it overlaps letters, numbers, the edge of the grid or another ship.
- Do not change the position of any ship once the game has begun.

See Figure 4 for an example of ship placement.

FIGURE 4

Here's an example of how to position your fleet correctly.

HOW TO PLAY

Decide who will go first. You and your opponent will alternate turns, calling out one shot per turn to try and hit each other's ships.

CALL YOUR SHOT!

On your turn, pick a target hole on your upright target grid and call out its location by letter and number. Each target hole has a letter/number coordinate that corresponds with the same coordinate on your opponent's ocean grid. To determine each coordinate, find its corresponding letter on the left side of the target grid and its number on the top of the grid.

For example, the shot location in Figure 5 is D-4.

FIGURE 5

Call "D-4" as your shot.

When you call a shot, your opponent must tell you whether your shot is a hit or a miss.

IT'S A HIT!

If you call out a shot location that is occupied by a ship on your opponent's ocean grid, your shot is a hit! Your opponent tells you which ship you have hit (destroyer, submarine, etc.). Record your hit by placing a *red* peg in the corresponding target hole on your target grid. Your opponent places a red peg in the corresponding hole of the ship you have hit on his or her ocean grid *(see Figure 6)*.

Example: You and Alex are the players. It's your turn.

You call: "D-4."

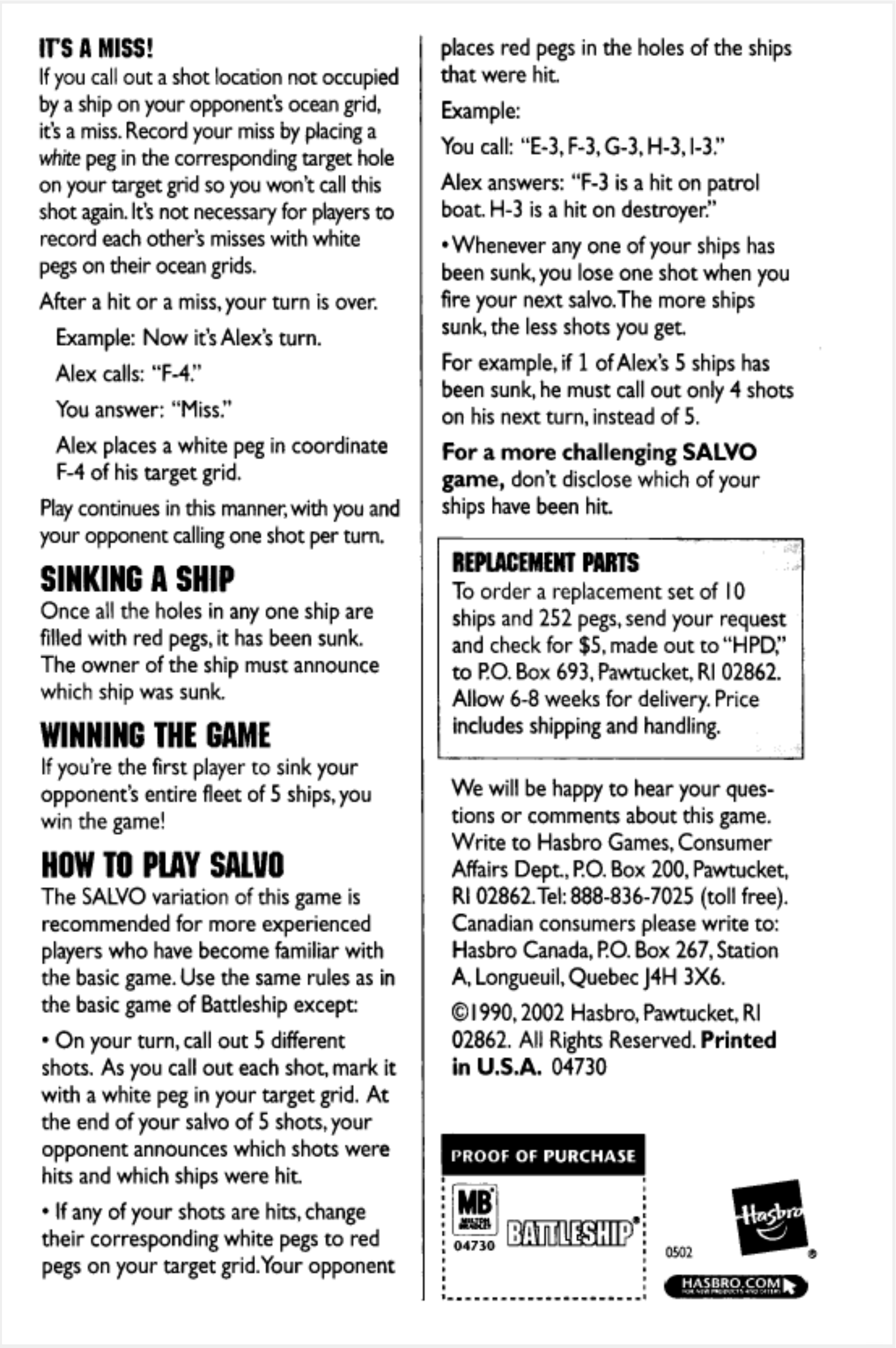
Alex answers: "Hit. Destroyer."

You place a red peg in coordinate D-4 of your target grid. Alex places a red peg in the hole of his destroyer directly over coordinate D-4 on his ocean grid.

FIGURE 6

It's a hit! Your opponent marks it with a red peg on his or her ocean grid.

FIgure 9: Official Hazbro battleship rule set page 1



IT'S A MISS!

If you call out a shot location not occupied by a ship on your opponent's ocean grid, it's a miss. Record your miss by placing a *white* peg in the corresponding target hole on your target grid so you won't call this shot again. It's not necessary for players to record each other's misses with white pegs on their ocean grids.

After a hit or a miss, your turn is over.

Example: Now it's Alex's turn.

Alex calls: "F-4."

You answer: "Miss."

Alex places a white peg in coordinate F-4 of his target grid.

Play continues in this manner, with you and your opponent calling one shot per turn.

SINKING A SHIP

Once all the holes in any one ship are filled with red pegs, it has been sunk. The owner of the ship must announce which ship was sunk.

WINNING THE GAME

If you're the first player to sink your opponent's entire fleet of 5 ships, you win the game!

HOW TO PLAY SALVO

The SALVO variation of this game is recommended for more experienced players who have become familiar with the basic game. Use the same rules as in the basic game of Battleship except:

- On your turn, call out 5 different shots. As you call out each shot, mark it with a white peg in your target grid. At the end of your salvo of 5 shots, your opponent announces which shots were hits and which ships were hit.

- If any of your shots are hits, change their corresponding white pegs to red pegs on your target grid. Your opponent places red pegs in the holes of the ships that were hit.

Example:

You call: "E-3, F-3, G-3, H-3, I-3."

Alex answers: "F-3 is a hit on patrol boat. H-3 is a hit on destroyer."

- Whenever any one of your ships has been sunk, you lose one shot when you fire your next salvo. The more ships sunk, the less shots you get.

For example, if 1 of Alex's 5 ships has been sunk, he must call out only 4 shots on his next turn, instead of 5.

For a more challenging SALVO game, don't disclose which of your ships have been hit.

REPLACEMENT PARTS

To order a replacement set of 10 ships and 252 pegs, send your request and check for $5, made out to "HPD," to P.O. Box 693, Pawtucket, RI 02862. Allow 6-8 weeks for delivery. Price includes shipping and handling.

We will be happy to hear your questions or comments about this game. Write to Hasbro Games, Consumer Affairs Dept., P.O. Box 200, Pawtucket, RI 02862. Tel: 888-836-7025 (toll free). Canadian consumers please write to: Hasbro Canada, P.O. Box 267, Station A, Longueuil, Quebec J4H 3X6.

PROOF OF PURCHASE

MB

04730

BATTLESHIP®

0502

HASBRO.COM

FIgure 10: Official Hazbro battleship rule set page 2