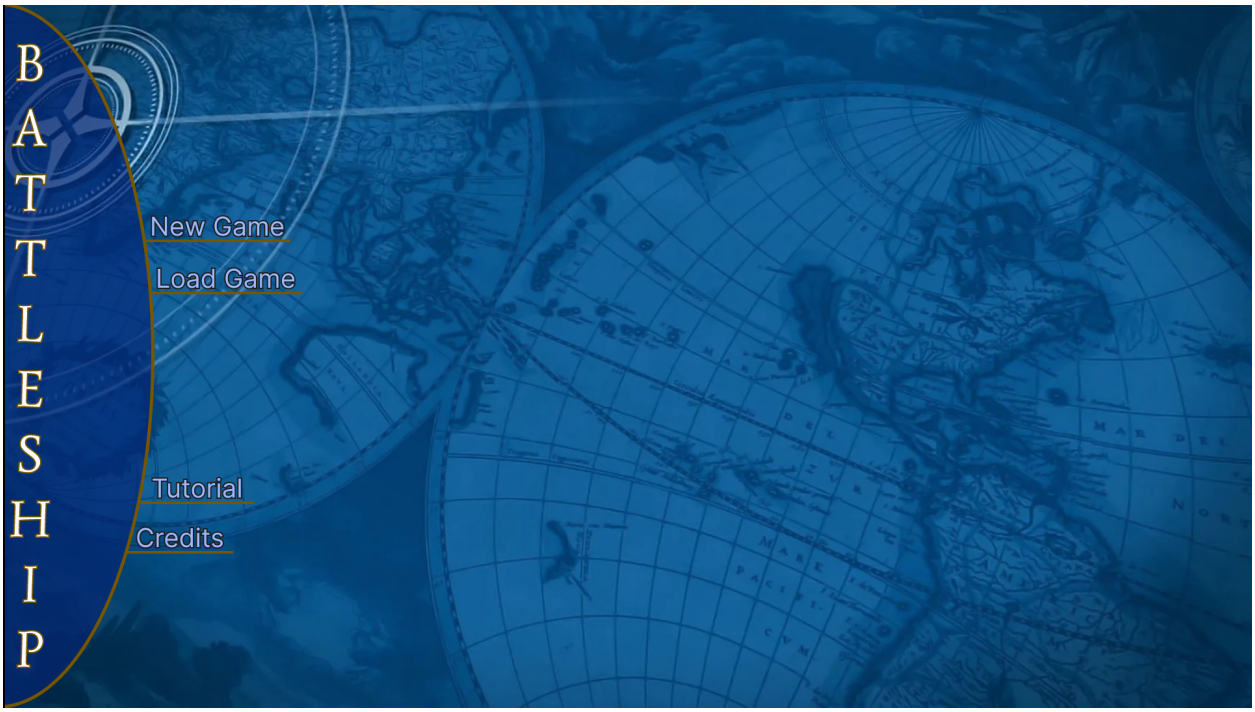
CMPT 276 Project Phase 2

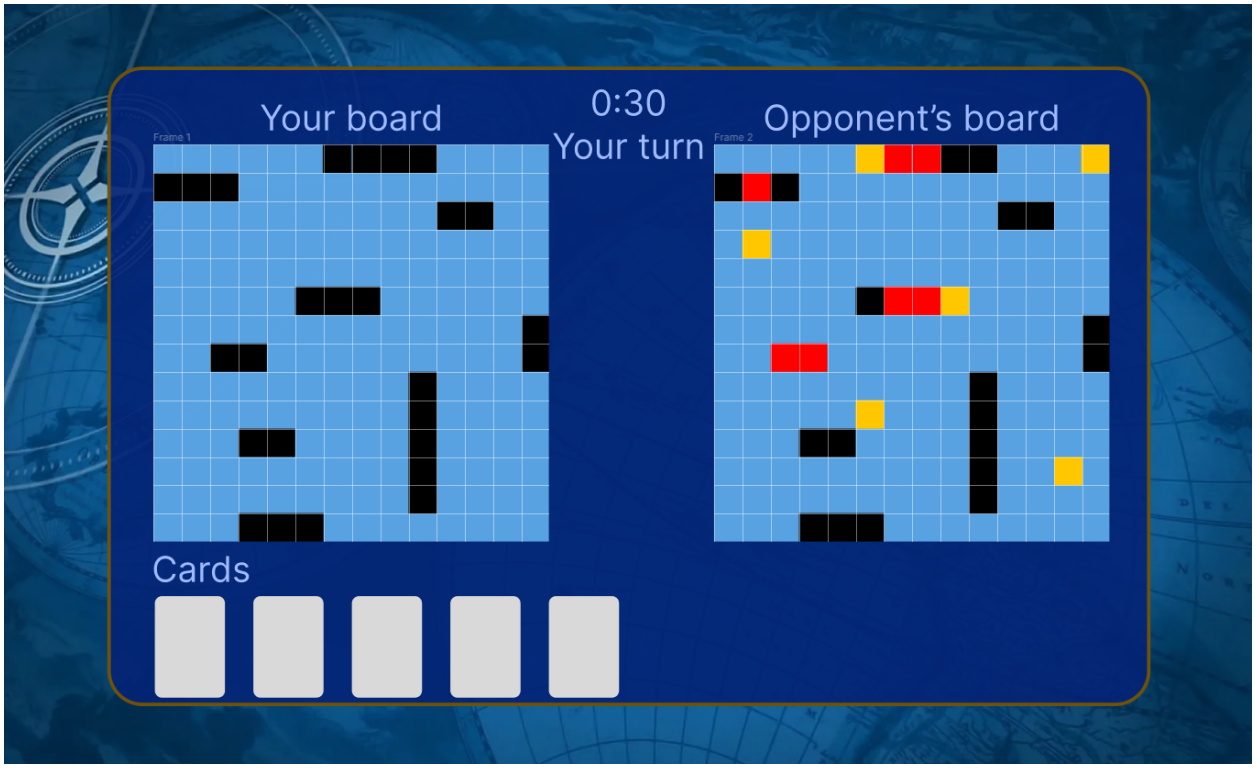
Warthog Group

UI Prototyping

Prototype 1




BATTLESHIP
Difficulty
Easy
Medium
Hard
Back

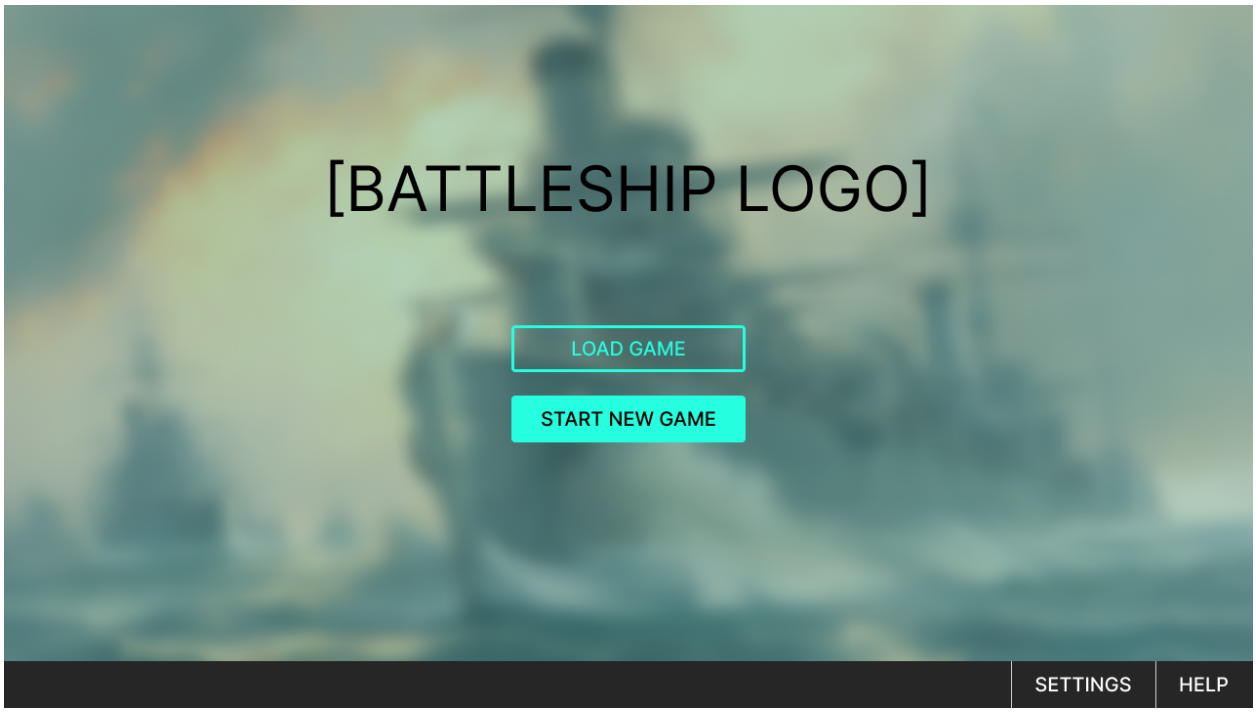

Your board
0:30
Your turn
Opponent's board
Frame 1
Frame 2
Cards

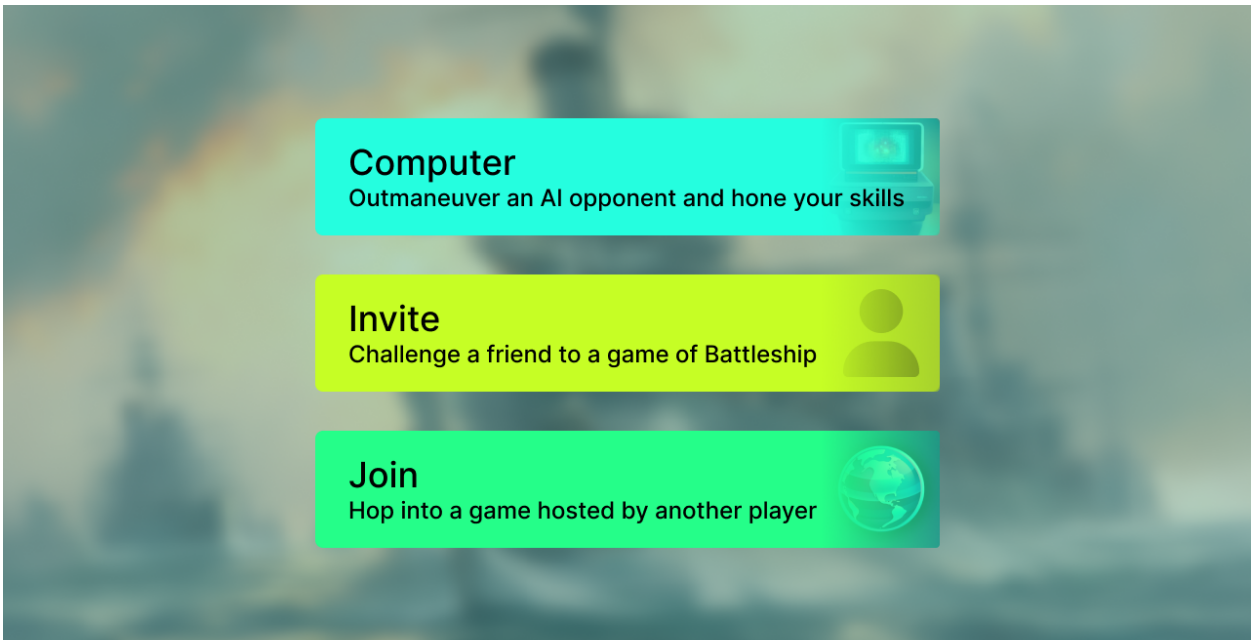
Feedback for Prototype 1

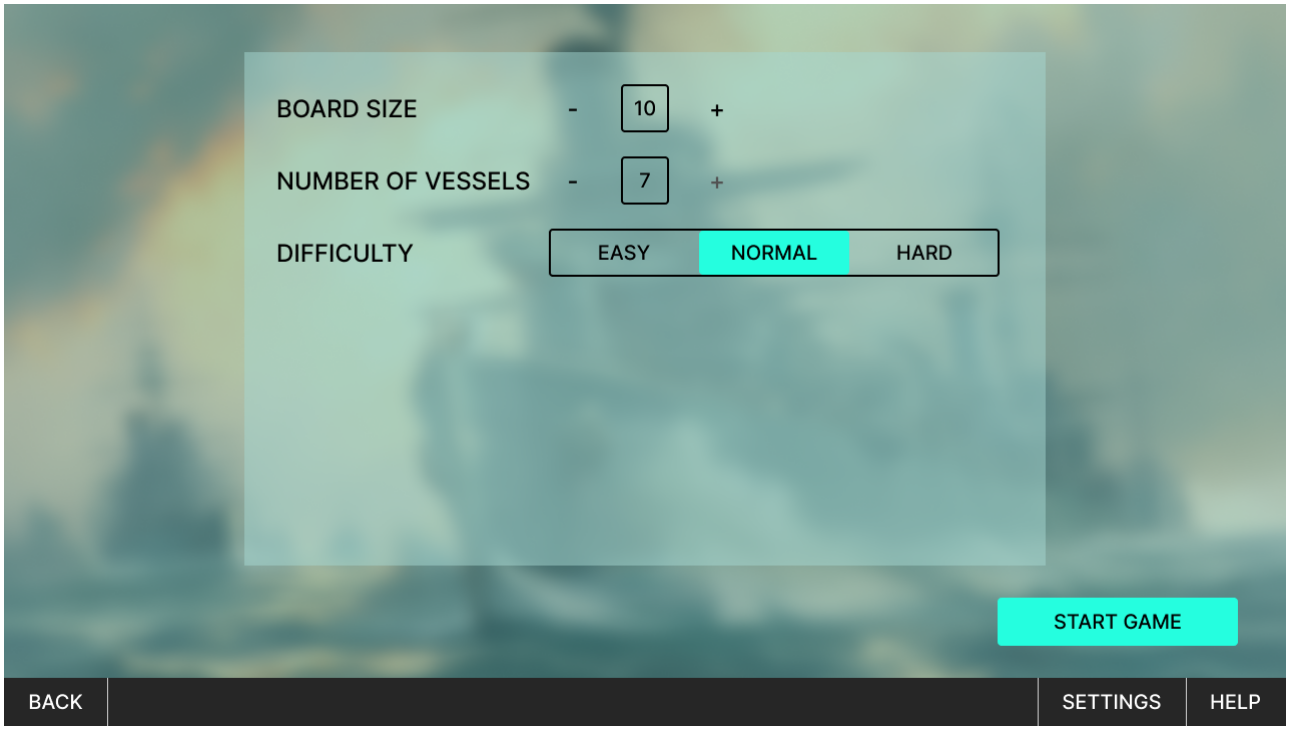
Typeface is a good choice for the theme though in some areas readability is affected by the serif. Contrast is good between text and background. Cohesive colour palate. Maybe make the opponent board the primary board you're looking at and make your board smaller and to the side since that information isn't as prevalent while playing.

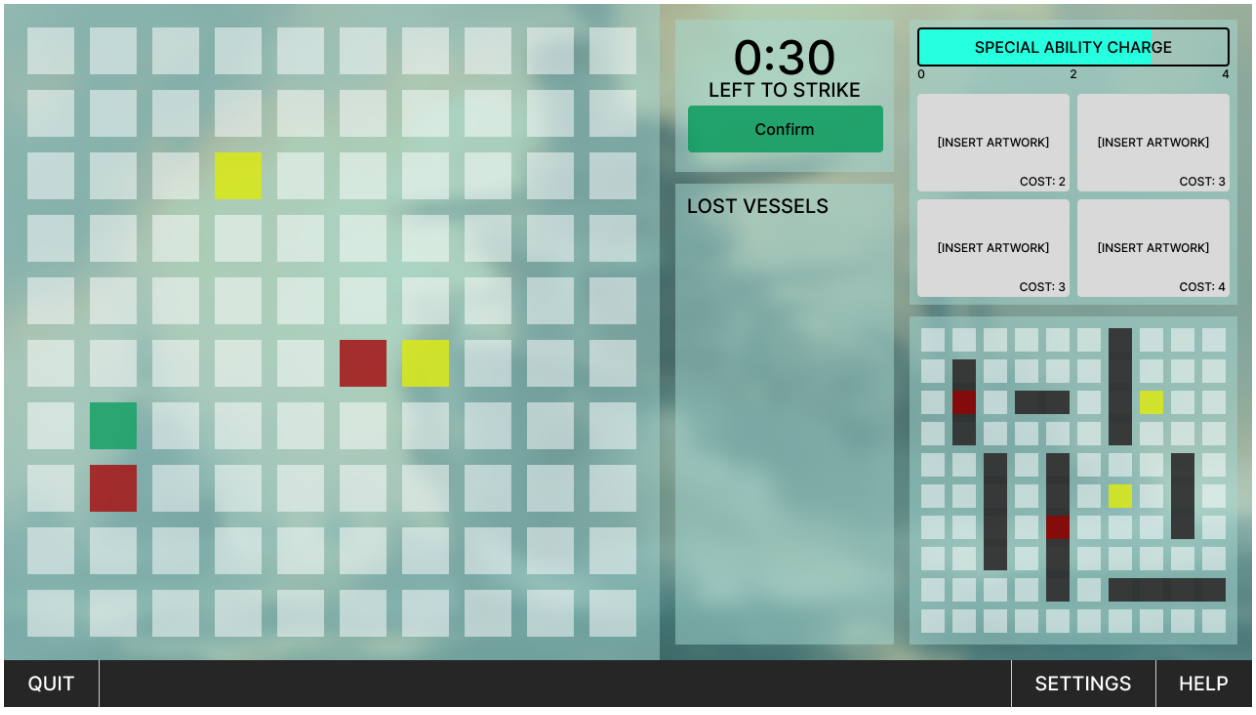
Also, there is good proximity between the New Game and Load Game buttons on the main screen. This is good because of the Gestalt principle of proximity, where more related items should be physically closer together.

Prototype 2






BOARD SIZE
-
10
+
NUMBER OF VESSELS
-
7
+
DIFFICULTY
EASY
NORMAL
HARD
START GAME
BACK
SETTINGS
HELP


0:30
LEFT TO STRIKE
Confirm
LOST VESSELS
SPECIAL ABILITY CHARGE
0
2
4
[INSERT ARTWORK]
COST: 2
[INSERT ARTWORK]
COST: 3
[INSERT ARTWORK]
COST: 3
[INSERT ARTWORK]
COST: 4
QUIT
SETTINGS
HELP

.

Feedback for Prototype 2

The descriptions for the opponents are very helpful. We should incorporate that into our final design. Using different colours emphasizes the difference between the options, while at the same time using analogous colours to show that they belong to the same category (choose opponent). In addition, the icons are a nice touch.

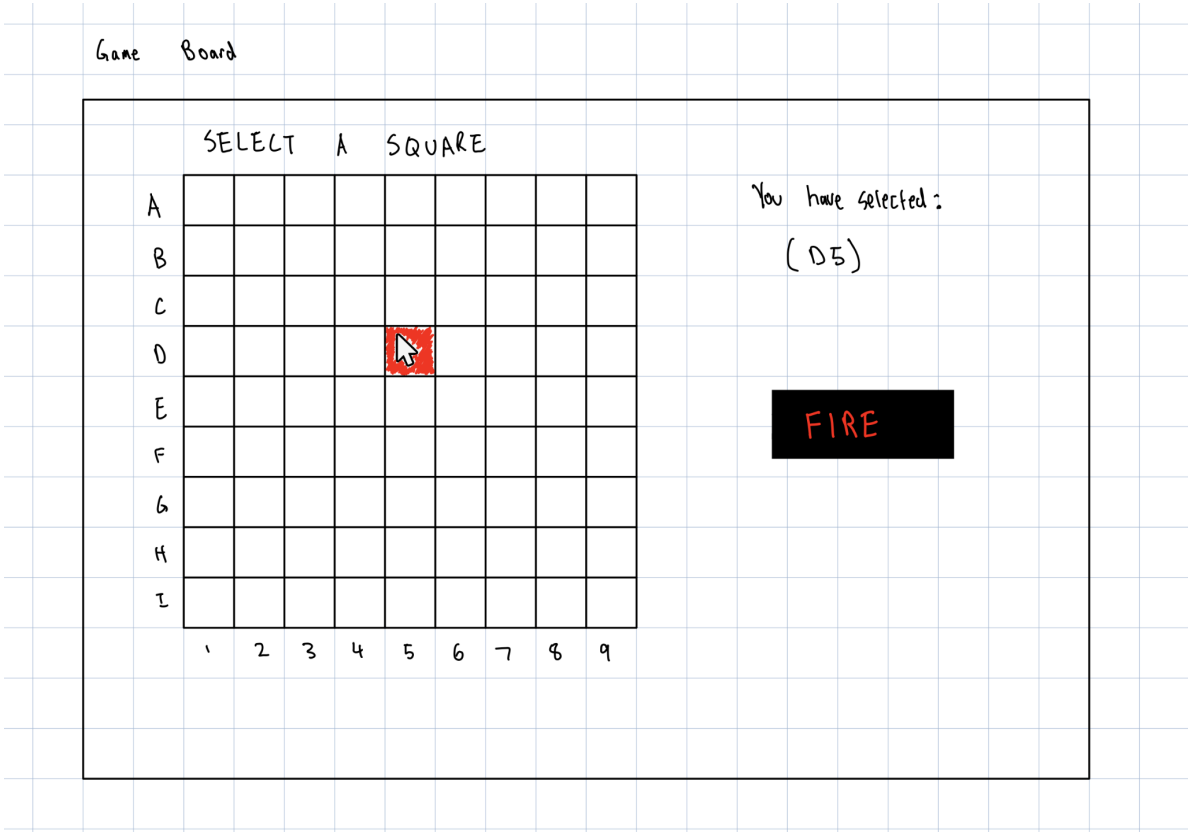
The contrast between the text and the background is low in some areas, such as where the shadow of the ship is. This makes the text hard to read.

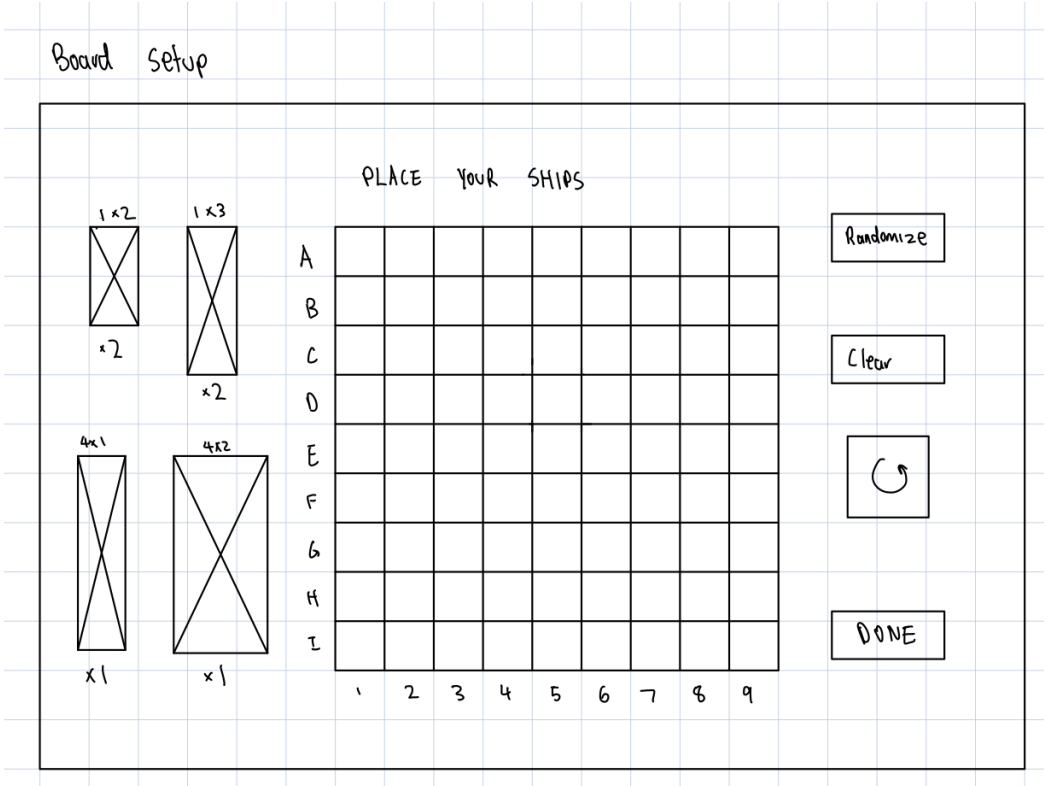
We should make our menu options look like how they do in this prototype.
Colour palate uses a consistent amount of saturation. The layout for the game space is nice showing all needed elements.

The modern design looks very interesting. This design could be improved by placing some of the buttons in different places, like the bottom panel at the top and the “start game” button within the settings field.

Having a stationary bar at the bottom helps the user know where to look for options such as navigation and help, complying with the Nielsen Norman Group usability heuristic of consistency.

Prototype 3


Game Board
SELECT A SQUARE
A
B
C
D
E
F
G
H
I
1 2 3 4 5 6 7 8 9
You have selected:
(D5)
FIRE


Board Setup
PLACE YOUR SHIPS
1x2
x2
1x3
x2
4x1
x1
4x2
x1
A
B
C
D
E
F
G
H
I
1 2 3 4 5 6 7 8 9
Randomize
Clear
DONE

Feedback for Prototype 3

Having confirmation of a coordinate before firing is good for clarity.

This prototype could improve by communicating the state more clearly, which was one of the usability heuristics of the Nielsen Norman Group. On the first screen, it is not really clear which cells have already been fired on, which cells have been hit, etc. Although it appears this screen depicts the user's first move, it would be good to include that.

Also, this prototype could improve with consistency, which is another one of the usability heuristics. In the second screen, 3 buttons (randomize, clear, done) are labelled with text while a third button is indicated with an icon. It might be better to make all buttons consistent.

Prototype 4

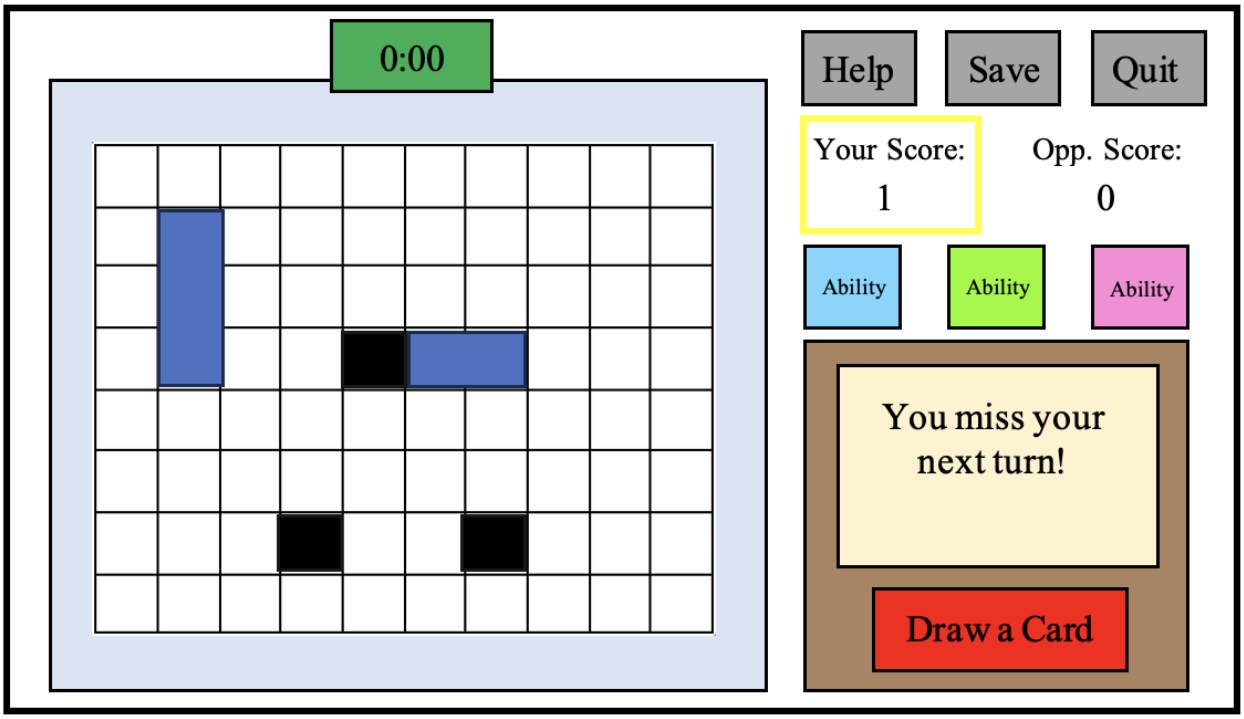

MOVES

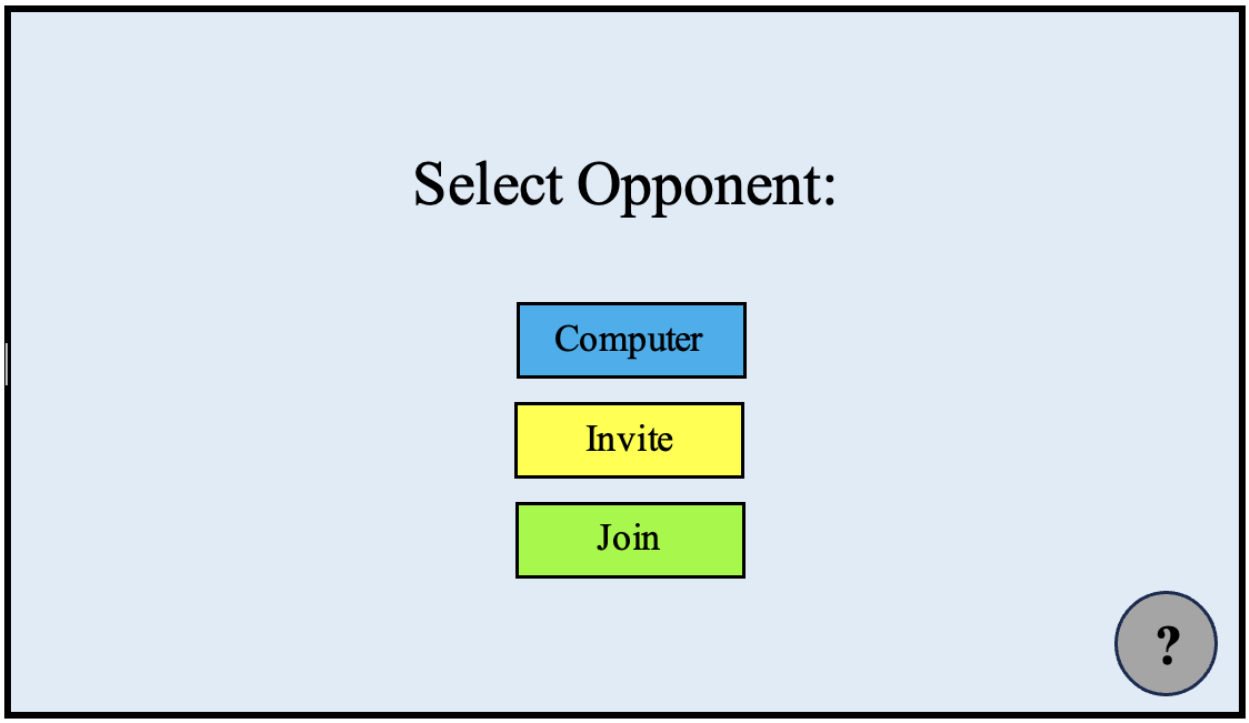

AUDIO & VIDEO
SAVE & QUIT
QUIT

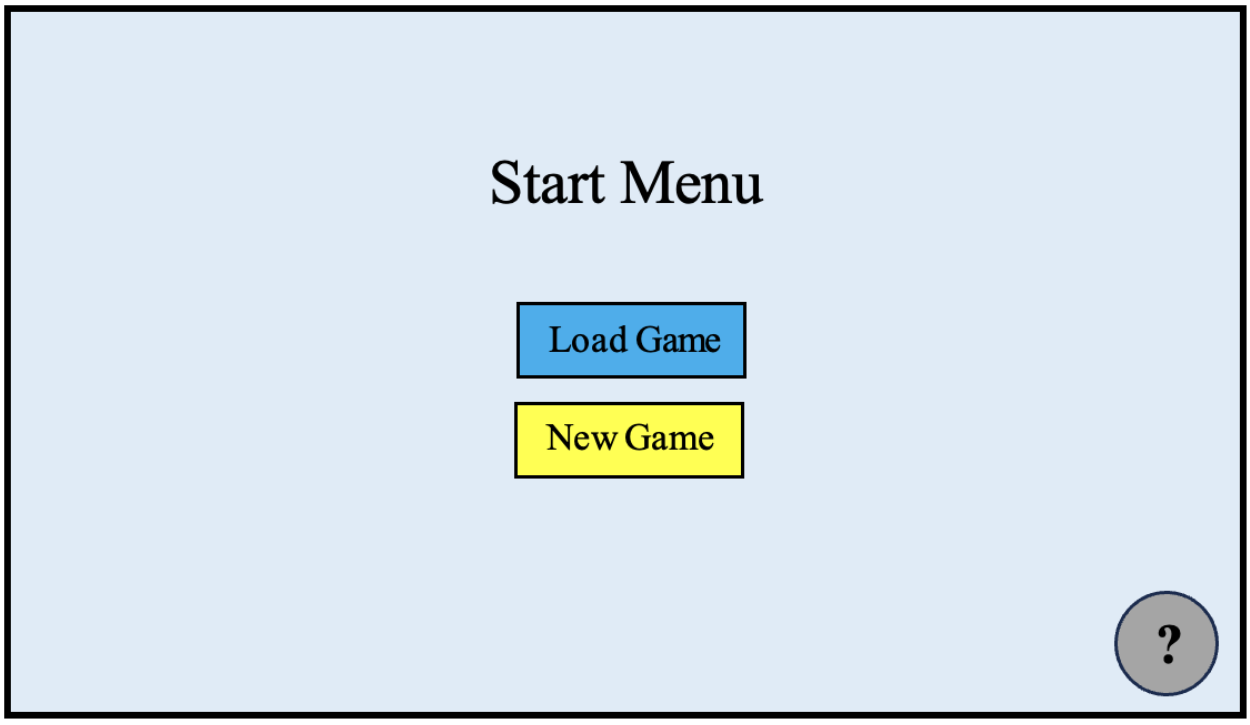
Feedback for Prototype 4

The contrast between the background image and the menu options makes the text easy to read. Furthermore, the layout follows the psychology concept of an F-shaped model.
Having the “back” arrow in the upper right goes against the convention of having the “go back” option at the top left and the “close” option at the top right. Not following convention goes against the UI design principle of consistency with common standards. However, the back button does well at giving the user control, allowing them to navigate easily between screens; another usability heuristic.

Prototype 5




Select Opponent:
Computer
Invite
Join
?


Start Menu
Load Game
New Game
?

Feedback for Prototype 5

The idea of a question mark for a help button is good, maybe move it to the top right and keep it on the board screen too; it would help with consistency. The yellow around the score is a little harsh and doesn't contrast enough with the white background, reducing the saturation of the outline would help.

One thing this prototype does well is it communicates the state of the game very clearly, which is a usability heuristic. From the game screen, the user is easily able to see their score, the opponent's score, who's turn it is, etc.

Final prototype

- Took the color palate and general styling from prototype 1.
- Took the button styling from prototype 3
- From prototype 2, took the idea to make the opponent's board large more prevalent and my board smaller in the corner, since the user primarily interacts with the opponent's board and my board is just there for reference.
- From prototype 3, we take the idea of confirming the user's action before we execute it
- From prototype 5, take the feature of clearly communicating the state of the game, including the score, time left in the turn, who's turn it is, etc.
- Most of the prototypes relied on colour to relay information in terms of ship status. This is a major accessibility design issue. Instead, we will use markings to display ship status:
    - No marking: cell has not yet been guessed
    - "?": cell is selected for guessing
    - "X": cell has revealed a ship
    - "-": a cell revealed it has no ship
    - Ship icon: your hidden ship

- The tutorial will help users familiarize themselves with any part of the game.
- Used a typeface with less extreme serifs and no outline to increase readability.
- All interactable buttons follow a consistent styling.

Design Patterns

1. Mediator

The GameManager class is a mediator that mediates interactions between boards, players and the ai. The GameManager keeps track of the players, their moves, and the boards. We chose to implement this design pattern because it allows for all interactions between players and boards to be in one place. Rather than the players communicating with the board class directly, we delegate this responsibility to the GameManager. This improves our code by reducing the number of responsibilities on the game class which now is focussed on taking in user input and drawing the screens. It also reduces the coupling of classes

This is implemented in the file game_manager.py

2. Abstract Factory

The BoardFactory class is an abstract factory that assists in the construction of Board objects. We chose to do this because constructing the board involves setting up ships and cells, which is a very complicated process needing multiple methods. Rather than clutter up the Board class with all this object creation code, we delegate the responsibility to the BoardFactory class. This improves our code by adhering to the single responsibility principle, allowing the Board class to just focus on the logic behind the board rather than the object creation.

This is implemented in the files board.py and board_factory.py

3. Decorator

We have a base implementation of Button, which only sets its location and loads a background image that can be clicked on. We introduce 2 decorators that add functionality to this Button: TextDecorator and ReactiveButton. TextDecorator enables buttons to have text on top of their background image. When the user hovers their mouse over the button, the text changes colour to indicate that the button is clickable. ReactiveButton enables the button to change its background image when the user hovers their mouse, also indicating that the button is clickable. We chose to implement this design pattern to extend the functionality of our base Button class, without having to rewrite it. This improves our code because we can customize the types of buttons we want (ex. Basic button with background image only, button with text only, etc.) without having to write a separate class for each one. Furthermore, this improves usability because it allows us to better communicate to the user that buttons are clickable, since they visually change when you hover over them.

This is implemented in the file button.py