

Designing a Software Solution for Battleship

Assignment 1

Part 1: Requirement Gathering by an Interview Questions

Question 1: What is the main objective of the Battleship game?

The main challenge of Battleship is that it is one player guessing the locations of the opponent's ships and sinking them. Players then call out to hit or miss the opponent's ships, taking turns in turn.

Question 2: Who will be the targeted users?

Casual players of various age groups, including both beginners and more experienced players, would form the target users overall. The interface would have to be easy to understand and play, yet also competitive

Question 3: What will be the position of ships on the grid/ How it is placed on the grid board?

Each ship has multiple grid cells and is placed on a 10x10 grid. The ships can be placed horizontally or vertically, never permitting the diagonal placement. Aircraft Carrier, Battleship, Submarine, Destroyer, and Patrol Boat respectively are each specified a different number of cells that it occupies: Aircraft Carrier (5 cells), Battleship (4 cells), Submarine (1 cell, 3 cells, 5 cells), Destroyer (3 cells), and Patrol Boat (2 cells).

Question 4: what does the process of the player turn such as how they win or lose?

Players take turns. In turn, they choose one coordinate (like A5 or F7) and another player must either guess "hit" if corresponding to the location of a part of one of their ships, or "miss." In the game, the game continues until all the ships of one player are sunk.

Question 5: what if the ship of the player is hit by the instruction of another player?

In the hit, the ship notifies the opponent. An opponent must announce if a ship has been fully hit (if all its coordinates are hit) and has sunk. When the ship is hit the player who uses the player tracking the damage to match the strategy.

Question 6: Discuss the important features that must be included in the software version of the Battleship?

The goal for a digital version of Battleship is a 10x10 interactive grid per player, the ability to place ships of different orientations and tracking hits and misses. And good checking should also give a simple way to verify that a ship has sunk and have clear visual feedback as to which coordinates were hit or missed.

Part 2: User Stories and their Acceptance criteria

User story 1: As a player, I want to place my ships on the grid so that I can prepare for the game.

Acceptance criteria

- A player should be able to place ships on a 10x10 grid.
- It can only be placed horizontally or vertically, not diagonally; ships cannot overlap.
- Prior to hitting place, players will be able to reorient ships.
- It should be such that ships do not overlap with each other or go outside the grid boundaries.
- To be able to place with precise placement, the grid must be displayed clearly.

User story 2: As a player, I want to call out coordinates to attack the opponent's ships so that I can try to hit them.

Acceptance criteria

- The player should attack, and it allows that player to enter a grid coordinate such as A1 or E7 or even F3.
- The player should be informed by the game whether the shot was a 'hit' or 'miss.'
- The opponent's grid should update after each shot showing a visual indication of hits and misses.

User story 3: As a player, I want to see which of my opponent's ships have been hit so that I can adjust my strategy accordingly.

Acceptance criteria

- The software should present hits and misses on the opponent's grid in a clear, visual sense.
- Upon being fully hit – all segments of a ship being hit – they should be marked as 'sunk' and, therefore, shown on the screen as such.
- The player should be informed if the entire ship has been sunk by the game.

User story 4: As a player, I want to track my own hits and misses so that I can keep track of my progress and plan my next move.

Acceptance criteria

- It should be written in such a way that the player's grid clearly marks the coordinates that he has previously placed his shots referred to as hits or misses.
- The player should visually be able to see which areas have been attacked via the grid.

User story 5: As a player, I want the game to end when one player has no ships left so that I know when the game is over.

Acceptance criteria

- Once one player's ships are all sunk, the game should terminate automatically.

- In case of a clear victory, a winner needs to be notified.
- The game ends and then the software should be prompted to prompt the user to decide if he should continue with a new game or if he should dare exit.

***User story 6:** As a player, I want to see a visual representation of my ships on the grid so that I can track their positions and the progress of the game.*

Acceptance criteria

- Before the game starts, we want to have the grid showing each ship's coordinates having its own color or symbol.
- During the game, position ship markers should not be removed from the marked position without being updated with hit and miss markers.

***User story 7:** As a player, I want to be able to reset my game if I make a mistake during ship placement so that I can start over without restarting the entire game.*

Acceptance criteria

- It should be ensured that the player can reset their ship placements before committing them.
- It should be confirmed before reset, so that they are sure to begin again.
- The grid should be cleared, so the player can redeploy his ships on the grid.

***User story 8:** As a player, I want to receive a notification when it's my turn to play so that I know when to take my next action.*

Acceptance criteria

- The game should have something that notifies the player when it's their turn (no matter if they press the button or not).
- Make the notification clear and noticeable to this, so the players are aware it is their turn to make a move.
- The player's turn at four shields would call it on screen, then the notification disappears, and the opponent's turn begins.

Conclusion

In this document there are initial steps of building a software version of Battleship. By interviewing the Battleship expert, the core mechanics of the game were explained, which will help develop the software, as player needs will be met throughout gameplay experience using user stories created in Part 2.