Designing a Software Solution for Battleship

Part 1: Requirements Gathering via an Interview

Interview Report

Interview Questions and Responses

Q1: What are the core rules of Battleship?

A: Battleship is a two-player strategy game where each player places ships on a grid and takes turns guessing coordinates to "hit" the opponent's ships. The game continues until one player has sunk all the opponent's ships.

Q2: What are the key challenges in designing a digital version?

A: The main challenges include developing an intuitive user interface, handling player interactions (e.g., placing ships, attacking), and implementing game logic for checking hits and misses. Additional challenges arise if an AI opponent is included.

Q3: How should the grid be represented?

A: The grid is typically a 10x10 board where players place their ships and track their hits/misses. Each cell should be able to store ship locations, hit markers, and misses.

Q4: What actions should a player be able to perform?

A: Players should be able to place ships, reposition them before finalizing, select attack coordinates, view a log of previous moves, and receive feedback on hits/misses.

Q5: How do players know when they win?

A: A player wins when they have successfully hit and sunk all of their opponent’s ships. The game should provide a message indicating victory or defeat.

Q6: Should the game have an AI opponent or be multiplayer?

A: The game can support both. A multiplayer mode allows two users to play against each other, while an AI mode lets a player compete against a computer-controlled opponent.

Q7: What should the user interface look like?

A: The UI should have two grids: one for the player’s ships and another for tracking attacks on the opponent’s grid. It should display ship placement options, attack results, and a game status message.

Part 2: User Stories with Acceptance Criteria

User Stories

1. Ship Placement

User Story:

As a player, I want to place my ships on a grid so that I can set up my game before playing.

Acceptance Criteria:

- The player should be able to place all ships on a 10x10 grid.
- Ships should not overlap.
- The player should be able to reposition ships before confirming placement.

2. Attacking an Opponent's Grid

User Story:

As a player, I want to select a coordinate to attack so that I can try to hit my opponent's ships.

Acceptance Criteria:

- The player should be able to select a grid cell to attack.
- The system should indicate whether the attack was a hit or a miss.
- The attack should be recorded, preventing duplicate moves.

3. Viewing Game Status

User Story:

As a player, I want to see the current state of the game so that I can track my progress.

Acceptance Criteria:

- The UI should display which ships have been sunk.
- Players should see a history of their previous moves.
- A message should indicate when the game is won or lost.

4. Multiplayer Mode

User Story:

As a player, I want to play against another person so that I can compete in real-time.

Acceptance Criteria:

- The system should allow two players to connect and play against each other.
- Turns should be taken sequentially.
- The game should notify players when it is their turn.

5. AI Opponent (Optional Feature)

User Story:

As a player, I want to play against an AI opponent so that I can enjoy the game without needing another player.

Acceptance Criteria:

- The AI should place its ships randomly.
- The AI should attack the player's grid with a logic-based or random strategy.
- The AI should follow the same turn-based system as a human opponent.

6. Game Over Notification

User Story:

As a player, I want to be notified when the game is over so that I know if I have won or lost.

Acceptance Criteria:

- The system should display a "You Win" or "You Lose" message at the end of the game.
- The final board state should be visible to both players.
- Players should have an option to restart the game.