Project 2: Initial Design Components
Group: Coruscant

Functional Requirements

Client:

- Allow players to create an account and log in
- Display the game battlefield (16x16 grid)
- Allow players to control their tank (move, turn, fire)
- Implement "shake to fire" functionality
- Display player's account balance and owned items
- Poll server for game updates every 100ms
- Store game state information locally for replay feature
- Implement game replay functionality (real-time or n times faster)

Server:

- Manage player accounts and authentication
- Manage the game battlefield and enforce game rules
- Handle client requests (join, move, turn, fire, leave)
- Implement game constraints:
    - Tank can move once every 0.5 seconds
    - Tank can fire once every 0.5 seconds
    - Maximum of 2 bullets from a given tank can be in the game simultaneously
    - Tank can make only one 90-degree turn per step
    - Tank can only move forward or backward relative to its current direction
- Maintain a 30-second event history
- Initialize game board using Builder pattern
- Provide REST API for client-server communication

Main Success Scenario: Join and Play Game

1. Player opens the BulletZone app on their Android device.
2. Player logs in with their account credentials.
3. System authenticates the player and displays the main menu.
4. Player selects "Join Game" option.
5. System sends a join request to the server.
6. Server assigns a tank to the player and sends initial game state.
7. System displays the game battlefield with the player's tank.
8. Player uses on-screen controls to move their tank.
9. System sends move request to the server.

10. Server validates the move, updates the game state, and sends confirmation.
11. System updates the display to show the new tank position.
12. Player turns their tank using on-screen controls.
13. System sends turn request to the server.
14. Server validates the turn, updates the game state, and sends confirmation.
15. System updates the display to show the new tank direction.
16. Player shakes the device to fire a bullet.
17. System detects the shake gesture and sends a fire request to the server.
18. Server validates the fire action, creates a bullet, updates the game state, and sends confirmation.
19. System updates the display to show the fired bullet.
20. Steps 8-19 repeat as the player continues to play the game.
21. Player chooses to leave the game.
22. System sends a leave request to the server.
23. Server removes the player's tank from the game and sends confirmation.
24. System returns to the main menu.

Use-Case Diagram