International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

BattleZone- Online Mobile Multiplayer Battle Royale Game

Nandan Yadav & Vighnesh Thakur
Department of Computer Science
S. S. & L. S. Patkar College of Arts & Science & V. P. Varde College

Abstract
BattleZone is an online multiplayer battle royale game designed for Android devices using Unity and C#. The project focuses on real-time competitive gameplay in which players enter a large game map, collect weapons and resources, engage other players, and survive until the final player remains. The proposed system combines online matchmaking, responsive player control, weapon and inventory management, combat, health and elimination, a dynamically shrinking safe zone, and match results into one integrated gameplay experience. Multiplayer communication is supported through Photon Fusion or Photon PUN 2, while Firebase can provide authentication and persistent player data. The overall design follows a gameplay loop of Matchmaking → Spawn → Explore and Collect → Engage Opponents → Survive the Safe Zone → Eliminate Players → Final Battle → Results. The project aims to provide a responsive, competitive, and expandable mobile battle royale experience while demonstrating the integration of networking, game mechanics, user authentication, data management, and player-centered testing.

Keywords— battle royale game, multiplayer game, Unity, C#, Android game development, Photon, Firebase, matchmaking, combat system, shrinking safe zone, player survival.

1. Introduction
Modern mobile games combine real-time interaction, graphics, networking, artificial intelligence, physics, user interfaces, and persistent player data. Multiplayer action games add another level of complexity because player actions must be synchronized across connected devices while maintaining responsive controls. Battle royale games are particularly suitable for demonstrating these challenges because many players compete within the same match, resources are limited, the playable area changes over time, and the final outcome depends on survival and combat decisions.

BattleZone is proposed as an Android-based online multiplayer battle royale game. Players join a real-time match, move through a large environment, search for weapons and resources, fight opponents, and remain inside a shrinking safe zone. As the match progresses, the playable area becomes smaller, increasing encounters and creating a final survival challenge. The project connects networking, player movement, combat, inventory, health, matchmaking, and results rather than treating them as isolated features.

2. Problem Statement
Developing a multiplayer battle royale game is not simply a matter of adding shooting mechanics to a 3D environment. The game must coordinate player actions over a network, manage match state, maintain responsive controls, handle weapon and inventory data, and keep the experience fair for all participants. Poor synchronization can result in inconsistent player positions or combat outcomes, while weak matchmaking can make it difficult for players to enter a match. Similarly, an ineffective safe-zone system may fail to create sufficient pressure for players to move and engage.

Therefore, the project addresses the problem of designing an integrated mobile battle royale system in which authentication, matchmaking, player control, weapons, combat, health, safe-zone behavior, elimination, and results work together as a consistent real-time gameplay experience.

3. Objectives
- To design and develop an Android-based online multiplayer battle royale game using Unity and C#.
- To implement player registration and login through a suitable authentication service.
- To provide online matchmaking for creating and joining multiplayer matches.
- To implement responsive player movement, jumping, aiming, and shooting.
- To develop weapon collection, inventory, firing, ammunition, and combat mechanics.

- To implement health management, damage processing, elimination, and survival states.
- To create a shrinking safe zone that applies damage outside the playable area.
- To display leaderboard information, winner details, and match statistics.
- To evaluate usability, responsiveness, network behavior, and gameplay balance through structured testing.

4. Primary and Secondary Research

4.1 Primary Research

Primary research for BattleZone can be conducted through player questionnaires, observation, playtesting, interviews, and feedback forms. The research can examine whether players understand the controls, whether combat feels responsive, whether matchmaking is clear, whether the safe zone creates appropriate pressure, and whether weapons and resources provide meaningful choices.

Question	Response Type
How engaging is the battle royale concept?	1–5 rating
How easy is player movement and aiming?	1–5 rating
How enjoyable is combat?	1–5 rating
How clear is matchmaking?	1–5 rating
How useful is the inventory system?	1–5 rating
How challenging is the shrinking zone?	1–5 rating
How fair does the multiplayer experience feel?	1–5 rating
Would you continue playing the game?	Yes/No

Note: Actual numerical results should be collected during playtesting rather than assumed.

4.2 Secondary Research

Secondary research examines established multiplayer game-design concepts, player motivation, networking, competitive balance, and user experience. Achievement can be supported through eliminations, survival, and match results; competence can be supported through responsive controls and skill-based combat; autonomy can be supported through movement, resource selection, and tactical positioning; and social interaction can be supported through online multiplayer competition.

5. Literature and Design Review

Game research commonly examines achievement, immersion, social interaction, mastery, autonomy, and competence as important aspects of player motivation. These concepts are applicable to BattleZone because the player must make decisions under pressure while adapting to changing opponents and a shrinking play area. The game can support achievement through survival and winning, competence through combat and movement mastery, and autonomy through choices about routes, weapons, resources, and engagement timing.

From a technical perspective, multiplayer games also require synchronization and authoritative game-state management. Networking architecture should reduce inconsistent states between clients, while latency and packet loss should be considered during testing. The proposed use of Photon Fusion or Photon PUN 2 provides a framework for real-time multiplayer communication, while Firebase can support authentication and persistent data.

6. Existing System

Existing battle royale games typically combine real-time multiplayer combat, large maps, resource collection, inventory management, player elimination, and shrinking playable areas. However, implementing a smaller academic project with these features presents several challenges: network synchronization may be inconsistent, mobile controls may be difficult to optimize, matchmaking may introduce waiting time, weapon balance may be unfair, and large environments can affect performance.

For this project, the existing approach is treated as a reference for identifying required features and design challenges. BattleZone focuses on a manageable modular implementation that can be tested and expanded without requiring the complete feature set of commercial-scale battle royale games.

7. Proposed System

The proposed system is an online multiplayer battle royale game in which authenticated players enter a shared match, spawn on a large map, collect equipment, fight opponents, and attempt to remain alive until the end of the match. The major modules are User Authentication, Online Matchmaking, Player Control, Weapon System, Battle Royale Zone, Health and Survival, and Leaderboard and Results.

Core gameplay loop: Login → Matchmaking → Spawn → Explore/Collect → Equip Weapons → Engage

Opponents → Move with Safe Zone → Survive/Eliminate → Final Players → Winner → Match Results.

8. Game Architecture

- 8.1 User Layer: registration, login, profile, session information.
- 8.2 Player Layer: movement, camera, aiming, shooting, health, inventory, interaction.
- 8.3 Multiplayer Layer: room creation, matchmaking, player synchronization, network events, match state.
- 8.4 Gameplay Layer: weapons, damage, pickups, safe zone, elimination, respawn or spectating rules.
- 8.5 Data Layer: authentication records, player statistics, match results, leaderboard data.

9. Game Modules

9.1 User Authentication

The User Authentication module manages player registration and login. Firebase Authentication can be used to create and validate player accounts. Successful authentication allows the player to access the multiplayer lobby and game-related services.

9.2 Online Matchmaking

The Online Matchmaking module creates or joins multiplayer rooms. Photon Fusion or Photon PUN 2 can be used to manage network sessions and synchronize gameplay information. The system should provide clear feedback for room creation, joining, loading, and disconnection.

9.3 Player Control

The Player Control module manages walking, running, jumping, camera movement, aiming, shooting input, and player interaction. Controls should be optimized for Android touch input and should remain responsive under normal network conditions.

9.4 Weapon System

The Weapon System manages weapon pickup, inventory, ammunition, firing, reload behavior, weapon switching, and damage values. Different weapons can have different ranges, firing rates, magazine sizes, and damage values to support tactical decisions.

9.5 Battle Royale Zone

The Battle Royale Zone controls the playable area. The safe zone gradually shrinks according to match phases. Players outside the safe zone receive periodic damage, encouraging movement toward the active area and increasing the probability of player encounters.

9.6 Health & Survival

The Health and Survival module manages player health, incoming damage, healing or recovery if supported, death, elimination, and the transition to spectating or match-end states.

9.7 Leaderboard & Results

The Leaderboard and Results module displays player placement, winner information, eliminations, survival time, and other match statistics. Persistent statistics may be stored using Firebase.

10. Multiplayer Processing

The multiplayer system should maintain a consistent match state across connected players. Player movement, firing, damage, item pickups, eliminations, and zone updates must be synchronized. Network authority and validation should be designed so that important gameplay decisions are not controlled only by an untrusted client.

A typical match state can be represented as: Waiting → Match Ready → Players Spawn → Active Match → Zone Phase Changes → Final Players → Winner Determined → Results → Return to Lobby.

11. Algorithms and Processing Techniques

11.1 Safe Zone Damage

Distance = $\sqrt{((x_2-x_1)^2 + (y_2-y_1)^2)}$. If Distance > Safe Zone Radius → Apply Zone Damage.

11.2 Combat Damage

Final Damage = Weapon Damage − Defensive Modifier. The implementation can later include headshot multipliers, armor, range falloff, weapon-specific behavior, and other balancing factors.

11.3 Elimination

IF Player Health ≤ 0 → Player State = Eliminated → Update Remaining Players → Record Statistics → Check Winner.

11.4 Match Result

IF Remaining Active Players = 1 → Winner = Remaining Player → End Match → Save Result → Display Leaderboard.

12. System Implementation

The implementation is organized as interconnected modules rather than one monolithic script. Principal Unity components can include Player Controller, Network Manager, Matchmaking Manager, Weapon Controller, Inventory Manager, Health Manager, Safe Zone Manager, Pickup Manager, Match Manager, UI Manager, Authentication Manager, and Results Manager. This modular structure allows new weapons, maps, statistics, or gameplay rules to be added without redesigning the entire application.

The software environment consists of Unity 6 or Unity 2022 LTS, Visual Studio for C# development, Android Studio for Android SDK and emulator support, Photon

Fusion or Photon PUN 2 for multiplayer networking, and Firebase for authentication and database services. The target platform is Android.

13. Methodology

- Stage 1 – Requirement Analysis: Identify multiplayer, combat, authentication, and mobile gameplay requirements.
- Stage 2 – Game Design: Define map structure, weapons, player rules, safe-zone phases, UI, and match flow.
- Stage 3 – Prototype Development: Build player movement, camera, map, basic weapons, and shooting.
- Stage 4 – Multiplayer Integration: Add networking, matchmaking, synchronization, and match state.
- Stage 5 – System Integration: Integrate authentication, inventory, safe zone, health, leaderboard, and results.
- Stage 6 – Playtesting: Test controls, combat, network behavior, matchmaking, performance, and difficulty.
- Stage 7 – Refinement: Fix defects, optimize mobile performance, balance gameplay, and improve usability.

14. Primary Testing and Evaluation

Parameter	Evaluation
Controls	Ease and responsiveness of movement, aiming and shooting
Combat	Damage behavior, weapon feel and fairness
Networking	Synchronization, latency and disconnection handling
Matchmaking	Room creation/joining and waiting experience
Safe Zone	Clarity, timing and difficulty of zone phases
Performance	Frame rate, loading time and Android stability
Results	Correct winner, placement and statistics

A five-point feedback scale can be used: 1 – Very Poor, 2 – Poor, 3 – Average, 4 – Good, 5 – Excellent. Actual scores should be collected from participants during testing rather than assumed.

15. Results and Discussion

The proposed evaluation focuses on whether BattleZone successfully combines multiplayer networking with core battle royale mechanics. Player engagement can be assessed by observing whether players understand the match flow and remain interested in surviving. Combat quality can be assessed through responsiveness, clarity of damage, and weapon balance. Networking can be assessed through synchronization and connection stability. The safe-zone system can be evaluated by checking whether it creates appropriate pressure without making survival unfair.

The final numerical results should be replaced with actual values collected from playtesting. The project should not claim performance, engagement, or satisfaction results before structured evaluation has been completed.

16. Advantages of the Proposed Game

- Integrated multiplayer gameplay combining matchmaking, combat, survival, and results.
- Mobile-focused controls designed for Android devices.
- Modular architecture that supports adding weapons, maps, modes, and statistics.
- Real-time competition creates strategic decision-making and replay value.
- Shrinking safe zone encourages movement and prevents matches from remaining static.
- Authentication and persistent data provide a foundation for player accounts and progression.
- Photon-based networking provides a practical framework for real-time multiplayer development.

17. Limitations

- Real-time multiplayer development introduces networking and synchronization complexity.
- Mobile devices have different performance capabilities and screen sizes.
- Large maps, many players, and network traffic can increase resource usage.
- Weapon and safe-zone balancing requires repeated playtesting.
- Reliable multiplayer gameplay depends on network quality and server-side behavior.
- The scope may need to be limited compared with commercial battle royale games.

18. Future Scope

- Additional multiplayer maps and game modes such as team battle royale.
- More weapons, attachments, vehicles, items, and character customization.
- Advanced matchmaking based on skill or player statistics.
- Ranked seasons, achievements, rewards, and expanded leaderboards.
- Voice or team communication features.

- Advanced anti-cheat and server-authoritative validation.
- Dynamic weather, day/night cycles, events, and interactive environments.
- Cloud-based analytics for monitoring matches and improving balance.

19. Conclusion

This project presents the design and development plan for BattleZone, an online mobile multiplayer battle royale game built using Unity and C#. The system combines user authentication, online matchmaking, player control, weapons, inventory, combat, health and survival, a shrinking safe zone, and leaderboard-based results. The project emphasizes the integration of these systems into a consistent real-time gameplay loop rather than treating them as independent features.

The proposed architecture provides a practical foundation for an expandable Android game. Photon Fusion or Photon PUN 2 can support multiplayer networking, while Firebase can provide authentication and database services. Through structured playtesting and refinement, BattleZone can be evaluated for usability, performance, synchronization, combat quality, and overall player engagement.

20. Software and Hardware Requirements

Category	Requirement
Programming Language	C#
Game Engine	Unity 6 or Unity 2022 LTS
IDE	Visual Studio
Android Tools	Android Studio, SDK and Emulator
Networking	Photon Fusion / Photon PUN 2
Backend	Firebase Authentication and Database
Operating System	Windows 10/11
CPU	Intel Core i5 / AMD Ryzen 5 or better
RAM	8 GB minimum; 16 GB recommended
Storage	20 GB free storage
Graphics	Integrated or dedicated GPU supporting Unity

21. References

[1] Unity Technologies, Unity documentation and game development resources.

[2] Photon Engine, Photon Fusion / Photon PUN 2 multiplayer networking documentation.

[3] Firebase Documentation, Authentication and Cloud database services.

[4] Android Developers, Android SDK and application development documentation.

[5] Ryan, R. M., Rigby, C. S., & Przybylski, A., “The Motivational Pull of Video Games: A Self-Determination Theory Approach,” Motivation and Emotion, 2006.

[6] Yee, N., “Motivations for Play in Online Games,” CyberPsychology & Behavior.