Unity FPS Sniper Contracts Prototype

IEEE-Style Project Documentation and Case Study

Muhammad Haider Abbas | MSc Computer Science & Technology | Ulster University Birmingham Campus



Figure 1. Project visual identity used in the portfolio website.

Abstract

Unity FPS Sniper Contracts Prototype is documented as a professional portfolio case study by Muhammad Haider Abbas. The project demonstrates practical capability in Game Development through a structured problem-solution design, implementation direction, testing plan and future development roadmap. The report explains the business or academic motivation, system requirements, architecture, workflow, quality considerations and expected professional value of the project. It is prepared for use as downloadable portfolio evidence for recruiters, clients and academic reviewers.

Keywords: Game Development, Prototype, Game Developer, Unity, C#, AI, Level Design

I. Project Profile

Field	Details
Project Title	Unity FPS Sniper Contracts Prototype
Category	Game Development
Year	2026
Role	Game Developer
Status	Prototype
Tools	Unity, C#, Blender, AI Systems, Vehicle Controller
Portfolio Slug	unity-fps-sniper-contracts-prototype

II. Introduction and Problem Context

This document presents the technical and professional documentation for Unity FPS Sniper Contracts Prototype. The project is positioned as a prototype in the Game Development category. The work highlights the author's ability to combine technical implementation, documentation discipline and outcome-focused project thinking. The project summary is: A tactical FPS sniper prototype with AI states, missions, vehicles, weapons, ADS and large terrain design.

A. Problem Statement

Tactical shooter prototypes require multiple systems to work together: AI, movement, weapons, mission logic and level design.

B. Proposed Solution

The project builds modular systems for missions, weapons, vehicles and enemy AI states.

III. Aim and Objectives

To build a tactical FPS vertical-slice prototype in Unity combining mission logic, weapon systems, AI states, vehicle interaction and large terrain design.

- Create a playable sniper/FPS controller.
- Build modular weapon, ADS and scope systems.
- Implement AI state behaviour for patrol, alert, investigate and combat.
- Design mission objectives and extraction flow.
- Prototype vehicle movement and terrain gameplay zones.

IV. Requirements Analysis

ID	Requirement	Type	Priority
FR-01	FPS movement	Functional / project requirement	Required
FR-02	Weapon firing and ADS	Functional / project requirement	Required
FR-03	AI state machine	Functional / project requirement	Required
FR-04	Mission manager	Functional / project requirement	Required
FR-05	Vehicle controller	Functional / project requirement	Required
FR-06	Terrain and objective zones	Functional / project requirement	Required
FR-07	Pause/options UI	Functional / project requirement	Required
NFR-01	Usability and clear portfolio presentation	Non-functional	Required
NFR-02	Maintainability and future extensibility	Non-functional	Required
NFR-03	Professional documentation and evidence quality	Non-functional	Required

V. System Architecture

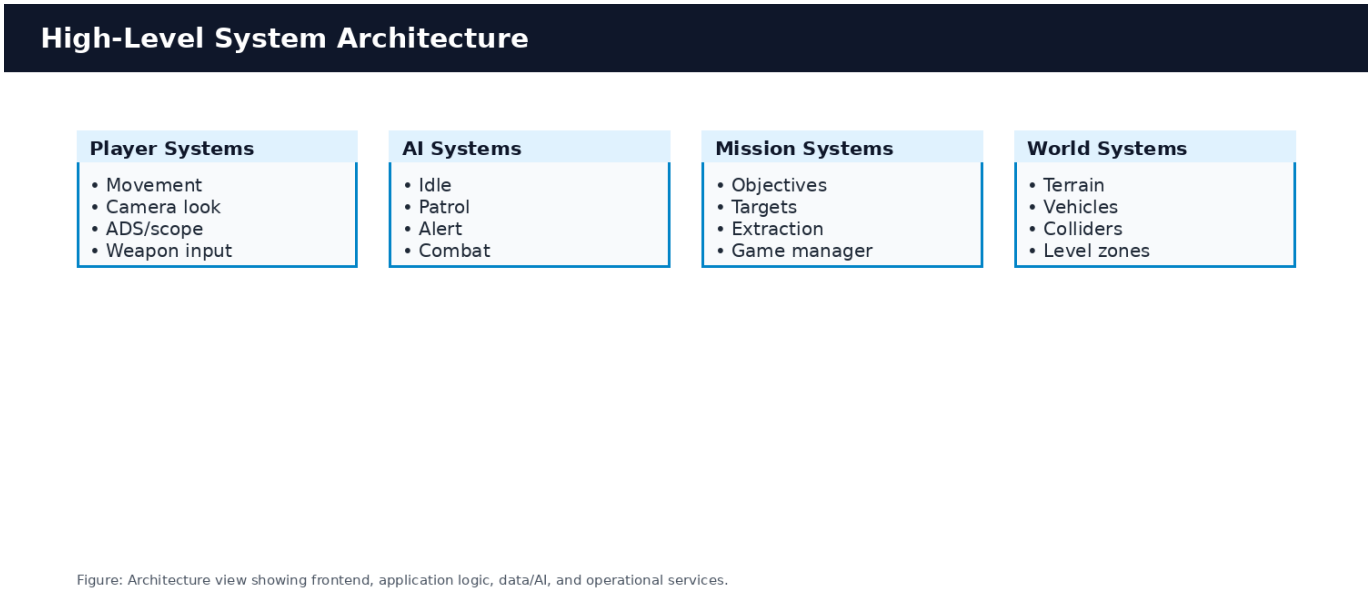


Figure: Architecture view showing frontend, application logic, data/AI, and operational services.

Figure 2. High-level architecture or production pipeline for the project.

The architecture separates user-facing presentation, project logic, data or model processing, and operational output. This structure makes the project easier to explain, test and extend inside a professional portfolio environment.

VI. Workflow Design

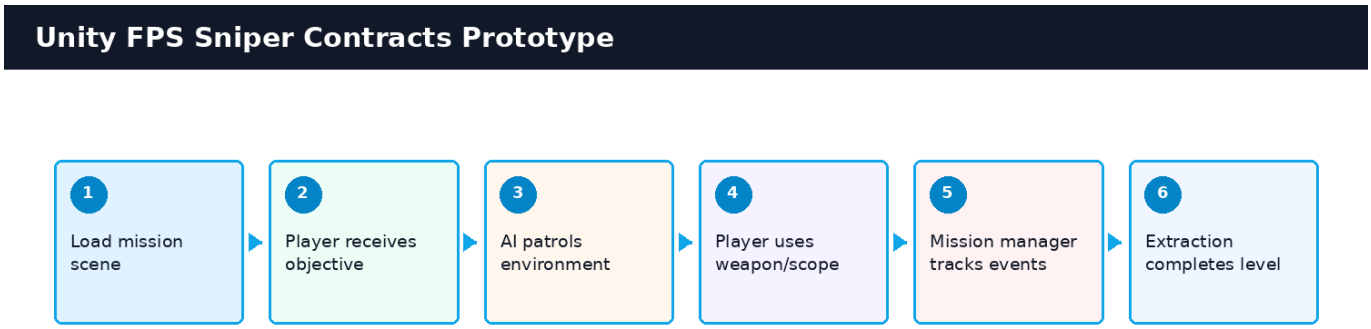


Figure: Generated project workflow / architecture artefact for portfolio documentation.

Figure 3. Project workflow from input or user action to final output.

VII. Data, Components and Implementation View

Component	Layer	Responsibilities	Notes
C-01	Player Systems	Movement, Camera look, ADS/scope, Weapon input	Core component group
C-02	AI Systems	Idle, Patrol, Alert, Combat	Core component group
C-03	Mission Systems	Objectives, Targets, Extraction, Game manager	Core component group
C-04	World Systems	Terrain, Vehicles, Colliders, Level zones	Core component group

Implementation should be documented through screenshots, code snippets, database schema extracts, test outputs or demonstration media when available. For deployed or live systems, screenshots and logs should be attached as appendices to increase evidence strength.

VIII. Testing and Validation Strategy

Test Area	Test Method	Expected Result
Player movement	Move/jump/look	Controller responds
Weapon system	Fire/ADS/reload	Expected weapon state
AI state machine	Trigger alert/combat	AI changes behaviour
Vehicle controller	Drive/brake/turn	Stable physics result

The testing strategy focuses on validating the main user or system journey, confirming that core requirements behave as expected, and recording limitations clearly. This supports honest portfolio evidence and avoids overclaiming features that are still prototypes.

IX. Security, Ethics and Professional Considerations

- Respect third-party asset licences.
- Keep production files organised and reusable.
- Optimise performance for the target platform.

X. Results, Outcome and Future Work

A. Outcome

A serious game development prototype that demonstrates systems thinking and Unity engineering.

B. Future Enhancements

Add polished AI, UI, audio, cinematic intro and Steam-ready vertical slice.

Roadmap Stage	Enhancement Direction
Short term	Add missing screenshots, demo links and downloadable supporting files.
Medium term	Connect live/source URLs and replace placeholder evidence with real artefacts.
Long term	Use analytics, user feedback or formal evaluation results to strengthen the project case study.

XI. References

[1] Unity Technologies, "Order of execution for event functions," https://docs.unity3d.com/Manual/ExecutionOrder.html

Appendix A. Portfolio Download Mapping

Item	Value
Recommended PDF path	docs/unity-fps-sniper-contracts-prototype-project-documentation.pdf
Recommended Word path	docs/unity-fps-sniper-contracts-prototype-project-documentation.docx
Admin field: Documentation	docs/unity-fps-sniper-contracts-prototype-project-documentation.pdf
Admin field: Extra Project File	docs/unity-fps-sniper-contracts-prototype-project-documentation.docx