

Intermission Needed – Technical Documentation

Project Summary

Intermission Needed is a AAA-grade first-person shooter (FPS) game built on **Unreal Engine 4**. The game features **networked multiplayer mechanics**, **physics-based projectile systems**, and an **AI-driven enemy bot framework**. Designed with realism and immersive gameplay in mind, it omits common FPS abstractions (like crosshairs) to emphasize manual aiming and physical bullet simulation.


This document provides a technical overview suitable for developers, technical testers, and UE4-based game engineers.

System Requirements

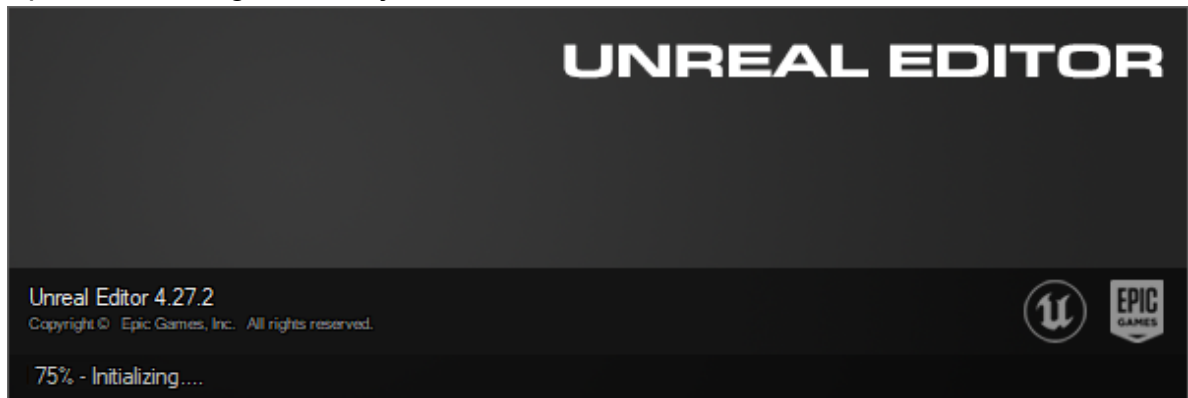
Software Dependencies

Component	Version
Unreal Engine	4.x (recommended: 4.27)
Git	Latest Stable
Git LFS	≥2.0
Windows/Linux	x64

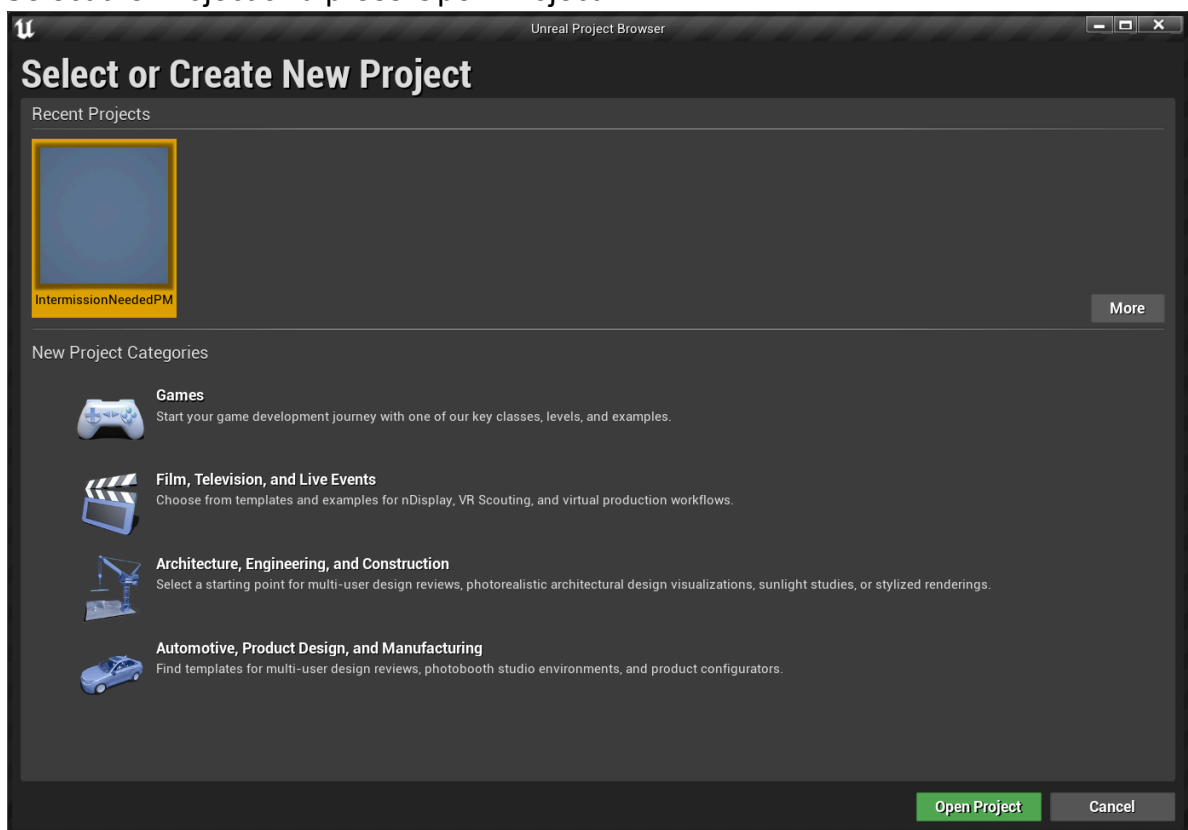
Installation & Project Setup

1. Download Project files using the following links:
 - a. Git Repo: `git clone https://github.com/Zayn-qxzxq/apm.git`
 - b. Git LFS:
 - i. `git lfs install`
 - ii. `git lfs pull`
 - c. Google Drive:  INF 395 Sufyan Mustafa

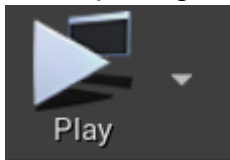
2. Install Unreal Engine 4
3. Open Unreal engine in Project Browser



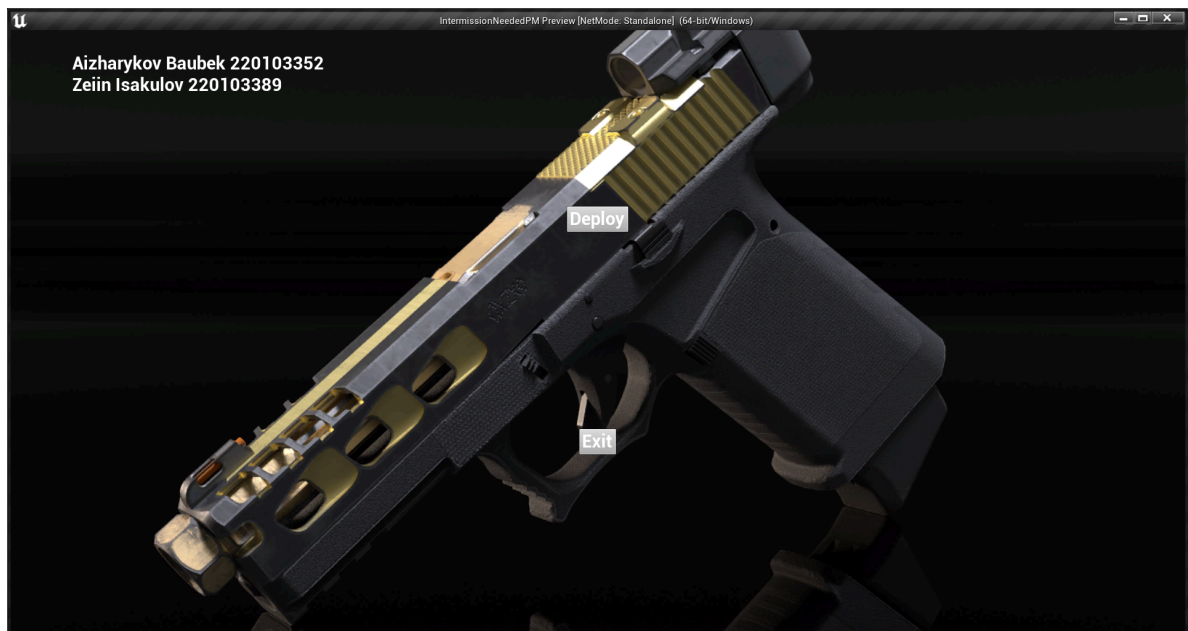
4. Select the Project and press Open Project



5. After opening the Project press the Play Button



6. Now You can Deploy or Exit



7. After pressing, the Deploy button game starts.

Project Directory Structure

apm/

```

|
|— Config/           # Engine and editor configuration files (INI)
|— Content/          # All game assets (maps, BPs, materials, etc.)
|  |— Characters/    # Player and enemy blueprints, animations
|  |— Weapons/       # Projectile systems, weapon blueprints
|  |— Maps/          # Game level(s)
|  |— UI/            # HUD assets (minimal use)
|  |— AI/            # Behavior trees, AI controllers
|  |— FX/            # VFX and audio placeholders
|
|— Source/ (if available) # C++ source (project appears mostly Blueprint-based)
|— weaponTesting/      # Sandbox for validating shooting and reload mechanics
|— .gitattributes      # Git LFS file types and filters
|— .gitignore          # Excluded files
|— IntermissionNeededPM.uproject # Main UE4 project file

```

Gameplay Mechanics (Engine-Level Behavior)

Mechanic	Implementation Detail
Movement	WASD input mapped via InputAction bindings
Aiming	Mouse Look; no HUD crosshair is rendered
Shooting	Spawns physics-enabled projectile actors (e.g., BP_Projectile)
Bullet Physics	Projectiles use ProjectileMovementComponent; dropoff occurs over ~50–100m
Reloading	R key triggers reload animation; interruptions (shoot/aim) invalidate reload state
AI Bots	Spawn via blueprint spawners; use behavior trees and perception systems
Health System	Player health capped at 50 HP; damage via projectile overlap/collision
Timer System	Tracks session duration; displayed in HUD widget
Gun Realignment	Post-shot recoil and offset requires player to re-aim each time

Note: Manual aiming required after every shot due to lack of aim reticle and deliberate recoil displacement.

AI System Architecture

- **AIController:** Manages individual bot logic using AIController_BP
- **Behavior Trees:** Control decision-making—seeking, attacking, patrolling
- **Perception Component:** AI uses sight sensing via AIPerceptionComponent

- **Spawn System:** AI spawners instantiate enemies at runtime via placed blueprint actors

Weapon System Details

- Weapons are implemented as **Blueprint Actors** (no C++ weapon base).
- Bullets are physical **actor projectiles** with collision detection.
- Reloading is implemented as an **animation state sequence**:
 - Interruption (e.g., aiming/firing during reload) leaves weapon in an unusable state.
 - Full reload animation must complete to restore weapon fire state.

QA & Testing Strategy

Testing Type	Tools/Methods
Manual Testing	Editor play-in sessions, testing weapon flow, AI logic
Blueprint Debugging	UE4 built-in visual debugger for node execution tracing
Unit Testing	N/A – Project is primarily Blueprint-based
Regression Checks	Manual replays of known use cases (weapon/AI behaviors)

Known Issues / Limitations

Issue	Description
-------	-------------

Collision Boxes	Some props use large or default collision volumes; causes projectile misfires
Smoothing Groups	Certain static meshes lack smoothing groups; causes shading artifacts
Missing SFX	Reload sounds are not implemented; require recording + asset integration

Design Principles

- **No Crosshair Design:** Reinforces player skill and physicality of gunplay
- **Physics-Driven Combat:** Bullets simulate real trajectories (with drag, gravity)
- **Minimalist HUD:** Excludes clutter; relies on player memory, awareness
- **Modular AI:** Easy to extend via UE4 behavior trees and blueprint-injected traits

Future Directions

- Integrate dedicated multiplayer support (Server authority + replication)
- Add audio feedback and VFX for enhanced realism
- Optimize asset import pipeline (especially for props and environmental assets)

Repository

GitHub: github.com/Zayn-qxzxq/apm

For feature requests, issues, or contributions, submit via GitHub Issues or Pull Requests.

