Project Proposal

Operation Clean Sweep – Tactical Shooter Game (TPS/FPS Hybrid)

Submitted By:

Muhammad Affan bin Aamir – Project Lead, Gameplay & Systems Developer

Umar – Environment & Assets Coordinator, Blueprint Support

Course: Data Structures & Algorithms (DSA) Project

Semester: Summer 2025

Date of Submission: July 3, 2025

Contents

1 Overview

Operation Clean Sweep is a tactical third-person shooter game (with a toggle to first-person mode), built using Unreal Engine 5. The game places the player in command of a 4-member squad tasked with assaulting an enemy base defended by AI enemies. It aims to provide a rich blend of gameplay and algorithmic depth by embedding Data Structures and Algorithms concepts directly into its systems.

This game is both a technical and creative challenge, helping the team understand game development fundamentals while applying computer science theory.

2 Objective

To build a playable shooter experience that demonstrates:

- Team-based AI combat
- Smooth player mechanics
- Real-time stat tracking and UI
- Direct implementation of Data Structures like Arrays, Queues, Trees, Maps, Graphs

3 Key Features

- TPS to FPS toggle (camera + combat mode switch)
- Combat mechanics (shooting, health, scope)
- AI-controlled enemies with patrol & attack behavior
- Teammates (bots or players) with individual stat tracking
- Custom UI for health, ammo, team stats (TAB View)
- Dynamic environment with death zones, HQ assault zone

4 Tools & Technology

- **Game Engine:** Unreal Engine 5 (C++ + Blueprints)
- **Languages:** C++, Blueprint Scripting
- **Assets:** Quixel Bridge, Mixamo, Sketchfab (for map, characters, weapons)
- **Version Control:** GitHub
- **Documentation & Planning:** Notion

5 Learning Goals

- Master basic to intermediate game development in Unreal Engine
- Learn gameplay programming and animation handling in C++/Blueprints
- Apply DSA concepts practically in game mechanics
- Improve project planning and team coordination using industry tools

6 DSA Concepts to be Implemented

Concept	Use Case
Arrays/Lists	Storing player, enemy, bullet data
Queues	Healing cooldown, spawn timers
Trees	Enemy AI behavior trees
Graphs	Pathfinding using Navigation Mesh
Maps	Tracking player stats (kills, damage)
State Machines	Player actions, Game state transitions

7 Development Timeline

Sprint	Dates	Focus
Sprint 1	July 4–9	UE5 Setup, Character Controls
Sprint 2	July 10–17	Player Combat & Scope
Sprint 3	July 18–25	Map Setup & AI Behavior
Sprint 4	July 26–Aug 2	Game Systems & DSA Integration
Sprint 5	Aug 3–10	Testing, Polish, Documentation

8 Team Roles

- **Affan:** Lead Developer, Game Logic, Combat, DSA Implementation
- **Umar:** Asset Research & Integration, Blueprint Setup, Visual Polish

9 Expected Deliverables

- Playable Level 1 Demo with 4-player squad & enemy HQ
- Source code with DSA-implemented gameplay logic
- Game Design Document (GDD)

- Technical Documentation (DSA code explanation)
- Sprint Logs & Weekly Progress
- GitHub Repository

10 Success Criteria

- All core mechanics functional and tested
- Minimum one playable level with enemy AI
- Clear implementation of at least 4 DSA concepts in gameplay
- Clean and professional project documentation
- Completion of team coordination via GitHub and Notion