**Master’s Capstone Project Guide: Unreal Engine Game Development**

**1. Project Overview**

**A. Objective**

This capstone project aims to develop a modernized recreation of Sub-Terrania using Unreal Engine. It will incorporate physics-based spaceship movement, enemy AI, procedural level generation, and visual enhancements while staying true to the original gameplay mechanics.

**B. Learning Goals**

* Develop an understanding of Unreal Engine's physics and AI systems.
* Gain experience in designing modular and procedural game levels.
* Implement shader and visual effects to enhance the gaming experience.
* Learn optimization techniques to improve performance in real-time environments.
* Deploy and package the game for multiple platforms.

**2. Development Plan**

**A. Game Mechanics Implementation**

1. **Physics-Based Spaceship Movement**
   * Utilize Unreal Engine's physics engine to replicate multidirectional thrust mechanics.
   * Implement gravity and environmental forces to create realistic movement challenges.
   * Design intuitive player controls with smooth acceleration and deceleration.
2. **AI Enemy Behavior**
   * Use Unreal's Behavior Trees to program enemy AI decision-making.
   * Implement different enemy movement patterns and attack strategies.
   * Design adaptive AI that reacts to player actions.
3. **Combat System**
   * Create projectile-based combat mechanics with hit detection.
   * Implement power-ups and special weapons for strategic gameplay.
   * Balance damage scaling and difficulty progression.

**B. Level Design and Procedural Generation**

1. **Level Structure and Objectives**
   * Design multi-layered levels with obstacles, enemy placements, and environmental hazards.
   * Implement mission-based objectives such as rescue, resource collection, and survival.
2. **Procedural Content Generation**
   * Use procedural generation techniques to create randomized enemy placements.
   * Implement modular level design for easy expansion and scalability.

**C. Graphics and Visual Effects**

1. **Shaders and Material Effects**
   * Use Unreal’s Material Editor to create dynamic shaders.
   * Implement lighting and post-processing effects to enhance immersion.
2. **Particle Systems**
   * Develop thruster effects, explosions, and destruction animations.
   * Use Niagara or Cascade particle systems for enhanced visual fidelity.

**D. Sound Design and UI Implementation**

1. **Audio Integration**
   * Implement background music and adaptive sound effects.
   * Design spatial audio to enhance in-game immersion.
2. **User Interface (UI)**
   * Develop a HUD displaying health, score, fuel levels, and mission objectives.
   * Create an interactive main menu and settings panel.

**3. Deployment and Optimization**

**A. Performance Optimization**

1. **Rendering Optimization**
   * Implement LOD (Level of Detail) techniques to reduce GPU load.
   * Optimize texture streaming and asset management.
2. **Physics and AI Optimization**
   * Reduce unnecessary physics calculations.
   * Use efficient pathfinding algorithms to improve AI performance.

**B. Deployment and Platform Support**

* Package the game for PC and potentially other platforms.
* Test performance across different hardware configurations.

**4. Documentation and Showcase**

**A. Project Report**

**A comprehensive report detailing:**

* Abstract: High-level summary of the project.
* Introduction: Problem statement and project goals.
* Methodology: Breakdown of development choices and implementation.
* Results & Challenges: Performance evaluations, testing, and lessons learned.
* Conclusion & Future Work: Summary and potential expansions.

**B. Presentation & Demo**

* Live Demonstration: Show gameplay and core mechanics.
* Screen Recording: Create a video walkthrough of the project.
* GitHub Repository: Share source code and documentation for review.