



## Unreal Engine 5.3 Game: Programmer Evaluation

### Objective:

This document outlines a third-person stealth shooter game designed in Unreal Engine 5.3 and aims to evaluate the technical skills and problem-solving abilities of Unreal Engine programmers. The focus is on AI interactions and the strategies the player will use while progressing through a secure facility.

### Focus:

The evaluation will primarily cover programming skills, AI behaviors, interaction systems, and player stealth mechanics.

### Deadline:

You must submit the following documents and files within 7 days of receiving this task.

### Deliverables:

#### 1. Documentation

Answer the following questions:

- Which design patterns did you choose and why?
- What challenges did you encounter while implementing the desired features and how did you solve them?

Describe your code, your approach, and your implementation. Draw a UML diagram that defines each class and the relationships between classes. Use a free website like draw.io to create your UML diagram.

#### 2. Project Files

Zip the project using "File -> Zip Project" and upload it to a folder on Google Drive along with your document.

## **Gameplay Description:**

### **Player:**

Third-person character with stealth elements (crouching, walking, running), capable of interacting with the environment, aiming, and shooting.

The player can switch between different weapons.

Simple inventory system: The player can carry a limited number of bullets and health packs in their inventory. These items should be actively displayed on the UI (Inventory UI).

### **Enemies:**

Can move around. They can respond to triggered alarms and search the area.

Can shoot at the player upon detection, and take cover.

Objectives: Navigate through a highly secure area and reach the exit point without triggering alarms, or reach the area by eliminating enemies after triggering alarms.

### **Levels:**

#### **Level 1: Stealth and Detection**

Two security cameras detect activity. Cameras must rotate around the Z-axis at specific angles and dynamically scan the area. Security cameras can be deactivated by a mechanism.

Visual detection: Cameras should be able to see the player.

Sound detection: Cameras can hear the player. Depending on the type of surface (e.g., grass, snow, or wood), different sounds are produced. Louder areas can be heard more clearly.

Security cameras, upon hearing a sound, will turn towards the source and attempt to detect the player. If the player remains in the field of view for more than 3 seconds, an alarm will be triggered.

Players must use the environment and stealth tactics to avoid being seen by cameras.

#### **Level 2: Alarm System and AI Response**

An alarm is triggered if a player is detected by a camera for more than 3 seconds.

Upon an alarm, AIs will advance to the player's location and engage in combat.

#### **Level 3: Combat and Health Management**

Modular health system should be displayed on the UI for both the player and AI.

The player can carry 2 weapons and switch between them.

Combat allows the player to either neutralize AIs or avoid combat to maintain stealth.

**Evaluation Criteria:****Programming Proficiency:**

Code readability, reusability, and efficiency.

Usage of appropriate design patterns and best practices.

Effective resolution of technical challenges.

**Problem Solving Skills**

Creative and efficient solutions to implement game mechanics.

Ability to troubleshoot and debug encountered issues.

Understanding of performance optimization techniques by applying efficient algorithms.

**Unreal Engine Expertise:**

Solid understanding of Unreal Engine's core concepts and tools.

Proper folder structure and asset naming conventions.

Extensive documentation and explanation of code.

Effective use of engine features.

Proficiency in Blueprint scripting and/or C++ (if applicable).

Performance-critical parts should be coded in C++; otherwise, continue with Blueprints.

**Additional Notes:**

You may use additional assets from Unreal Engine Marketplace (e.g., character models, animations) or create your own.

Focus on code quality and efficiency over artistic elements.