Technical Documentation - Emergency Response Simulation

1. Technology Stack

Game Engine: Unity

Programming Language: C#

AI & Navigation: Unity NavMesh & NavMeshAgents

UI: TextMeshPro (TMP)

Audio: Unity AudioSource for alarm sounds and footsteps

Target Platform: Windows

2. Architecture Overview

The simulation follows a realistic emergency response flow:

1. Fire starts and spreads to nearby flammable objects
2. Alarm triggers with visual and audio alerts
3. AI responders and player work together to put out fires
4. Civilians panic and run to safety
5. Mission completes when all fires are out and civilians are safe

3. Main System Components

1. 🔥 Fire System

- FireSource.cs - Controls individual fires (start, spread, extinguish)
- FireManager.cs - Tracks all fires in the scene and coordinates events

2. 🧍 Civilian System

- InjuredNPC.cs - Controls civilian panic and escape behavior
- InjuredManager.cs - Tracks all civilians and their status

3. 🤖 AI System

- AIResponder.cs - AI helpers that follow and assist the player
- VehicleAutoResponder.cs - Fire truck AI that finds fires

4. 🧯 Player System

- PlayerInteraction.cs - Handles player input (E to interact, H to help)

5. 🎮 Mission System

- MissionManager.cs - Controls the overall mission flow
- FlashingVolumeController.cs - Creates emergency visual effects

6. 📢 Audio System

- SoundController.cs - Manages alarm sounds and other audio effects

4. How Everything Works

1. Fire Behavior

- First fire starts after 5 seconds
- Fire spreads to nearby flammable objects
- Players and AI can put out fires by interacting with them

2. Civilian Behavior

- Civilians panic when their building is on fire
- They run to safe spots around the map
- Players can help injured civilians by pressing H

3. Mission Flow

- Fire breaks out → alarms sound, screen flashes red
- Player and AI work to put out fires
- Fires spread if not contained quickly
- Civilians escape and need help
- Mission ends when all fires are out and all civilians are safe