BACK2REVENGE STUDIO

MACRO GAME DESIGN DOCUMENT

THE LAST FLOOR

STAFF :

El Mehdi Hamte – Game Designer and Developer
Otmane Aayachi – Sound Designer
Abdellah Hannache – Game Designer and Artist 2D

Game Presentation :

"V-GAS: The Last Floor" is a 3D first-person survival horror and stealth puzzle game.
The game is set inside a sealed, high-rise research tower of **Aura Corp**, where a catastrophic explosion in the main pipeline has leaked a toxic experimental gas known as **V-GAS** and flooded the area with a lethal **Pigmented Substance**.
The player takes on the role of a **specialist Doctor** and virus expert who awakens after the blast. The objective is to climb five contaminated floors, solve environmental puzzles to fix the disabled electrical elevators, and reach the main control unit to disable the lockdown and escape.
Core gameplay focuses on **3D immersion**, managing **gas mask oxygen resources**, and avoiding mutated former colleagues who carry a high viral load—where a single touch results in immediate infection and failure. The project aims to deliver an intense, claustrophobic survival horror experience, emphasizing tension and strategic movement within a compact playtime.

Plateformes : PC (Windows), WebGl

Minimum System Requirements (PC) :

- OS : Windows 10 (64-bit)
- CPU : Intel Core i3 / AMD Ryzen 3
- RAM : 4 GB
- GPU : Carte graphique compatible DX10 (Intel HD Graphics 4000)
- Espace disque : 500 Mo

Development Environment:

- **Game Engine : Unity 2022.3 LTS.**
  Reasoning: *This is a stable Long-Term Support version. Unity provides robust native 3D tools and the* ***Universal Render Pipeline (URP),*** *allowing for high-quality dynamic lighting and post-processing effects (like the mask vignette) within a 6-week development cycle*

- **IDE :** Visual Studio 2022 or Visual Studio Code (with C# extensions).
- **Version Control:** Git (hosted on GitHub or GitLab).
- **Client:** GitHub Desktop or Sourcetree

1. Risques Techniques (Version 3D)

Risque	Impact	Solution Technique
Pathfinding 3D & Verticalité	**Les ennemis (ex-collègues) se bloquent dans les couloirs ou ne montent pas les escaliers.**	**Utilisation du NavMesh 3D natif d'Unity. Découpage des zones de navigation par étage et utilisation de *NavMesh Links* pour les passages complexes.**
Performance (Lumières & Gaz)	**Le brouillard volumétrique et les lumières dynamiques 3D font chuter les FPS sur WebGL.**	**Utilisation de l'URP (Universal Render Pipeline). Optimisation via le *Light Culling*. Remplacement du brouillard volumétrique par des "Particle Sprites" 3D moins coûteux.**
Contact et Collision (Infection)	**Difficulté à détecter le "frôlement" avec les infectés en 3D.**	**Utilisation de Spheres Colliders avec des triggers de proximité. Implémentation d'une fonction OnTriggerEnter pour déclencher l'infection immédiate.**
Mal des transports (Motion Sickness)	**La vue subjective (FPS) avec le masque peut donner la nausée.**	Ajustement du *Field of View* (FOV) et ajout d'une option pour désactiver le "Head Bobbing" (balancement de la tête)

2. Pipeline de Production 3D

Graphismes (Graphismes (3D & Textures) :

Logiciels : Blender (Modélisation) / Adobe Substance Painter ou Photoshop (Textures).

Export : .FBX pour les modèles, .PNG ou .TGA pour les textures.

- **Import Unity :**
  - **Mesh Compression : Medium**
  - **Texture Type : Default (avec Normal Maps Metallic/Smoothness maps)**
  - **Rigging :** Humanoid (pour utiliser les animations d'ex-collègues infectés)

Level Design (Espace 3D):

- **Outil : Unity ProBuilder pour le prototypage rapide (Greyboxing) des couloirs et de l'ascenseur.**
- **Structure : Utilisation de Prefab Variants pour les modules de laboratoire (murs, portes, consoles électriques)**

Audio

- **Formats:** .WAV for short SFX (heavy breathing, heartbeats, gas hissing) to be uncompressed in
- **memory;** .OGG for ambient music and long soundscapes (streaming).
- **3D Spatialization**: Full use of Unity's 3D Audio Spatializer.
- This is crucial so the player can hear the footsteps and mutated moans of "former colleagues" through walls and floors to determine their position.

Custom Tools to Develop

Given the short 1.5-month deadline, we focus on high-impact tools:

- 3D Waypoints Editor: A script using Gizmos (visual lines in the editor) to allow Level Designers to create patrol paths for mutated enemies in 3D space via simple drag-and-drop nodes.
-
- **V-GAS & Substance Manager (3D):** A component attached to 3D Box/Mesh Colliders (Triggers).
-
- **It allows designers to:**
  - Define "Gas Zones" (affects Oxygen mask levels).
  - Define "Pigmented Substance" zones (floor-based triggers that deplete Health HP on contact).

Software Architecture

- **GameManager:** The central orchestrator. It manages game states (Main Menu, Active Gameplay, Death/Infection, and Floor Transitions).
-
- **DoctorController (FPS):** Handles 3D movement (WASD), mouse-look, and interaction logic. It triggers events like OnOxygenEmpty or OnContamination.
-
- **GasMaskSystem:** A modular script attached to the First-Person camera. It manages the toggle for the mask, the oxygen timer logic, and links to the visual "fogging" effects.

Graphic Rendering & Complexity

- **Pipeline:** URP (Universal Render Pipeline) 3D.
- **Shaders (Shader Graph):**
  - ***Mask Visor Shader:*** A custom screen-space shader that

simulates glass reflections and accumulating condensation/steam as oxygen runs low.
    - ***Toxic Pigment Shader:*** A scrolling, animated texture for the substance on the floor to make it look lethal and "alive."
- **Post-Process Stack:**
    - **Vignette:** Simulates the restricted view from inside the gas mask.
    - **Chromatic Aberration:** Increases as Health (HP) drops or when touching the Pigmented Substance.
    - **Bloom:** Used for the glowing eyes of mutants and the eerie green glow of the V-GAS.

Technical Implementation of Gameplay

1. Artificial Intelligence (Mutated Friends)

The AI uses a 3D Finite State Machine (FSM) integrated with Unity NavMesh.

- **States:** Idle -> Patrol -> Suspicious (Hears noise) -> Chase (Sees Doctor).
- **Visual Detection:** Physics.Raycast from the enemy's eyes to the player, blocked by walls/obstacles.
- **Sound Detection:** When the Doctor moves fast or interacts with objects, they emit a Physics.OverlapSphere. If an enemy is within the radius, they move to investigate the source.
- **Infection Logic:** If the enemy's collider touches the player, the OnInfected event triggers an immediate Game Over.

2. Oxygen & Health

- **Logic:** A float oxygenLevel variable decreases in Update().
- **Safe Zone (Elevator):** A specific 3D trigger zone where
- OxygenMaskSystem stops depleting and starts a "Refill" state, providing a moment of rest.

Performance Optimization

- **Goal:** Constant 60 FPS.
- **Occlusion Culling:** Essential in a 3D tower to avoid rendering rooms behind closed doors or on floors above/below the player.

- **Object Pooling:** Used for gas particles and impact effects to prevent CPU spikes.

  Code Conventions & Organization
- **Language:** English for variables/functions; French/English for comments.
- **Serialization:** Use [SerializeField] private to keep variables encapsulated while remaining editable in the Inspector.
- **Version Control:** GitHub with Git LFS (Large File Storage) is mandatory now to handle 3D FBX models and high-res textures.
- **Communication:** Discord and Trello (To Do / Doing / Done).