

10 - 2 - 2025



CANBERRA

GAME DESIGN
DOCUMENT

BON CAO
Software Development 3B





Inspired by fictional events, Project Canberra seamlessly blends a variety of concepts, themes, and ideas into a unique and distinctive vision

Background source: [gentleman.gasmask](#)

FOR THE BEST EXPERIENCE, VIEW THIS DOCUMENT ON A
MOBILE DEVICE.

UNDERSCORED TEXT CONTAINS LINKS TO EXTERNAL
WEBSITES.

FOR MORE INFORMATION, VISIT
HTTPS://WYZARD12.GITHUB.IO/ OR SCAN THE QR CODE
BELOW.



OVERVIEW

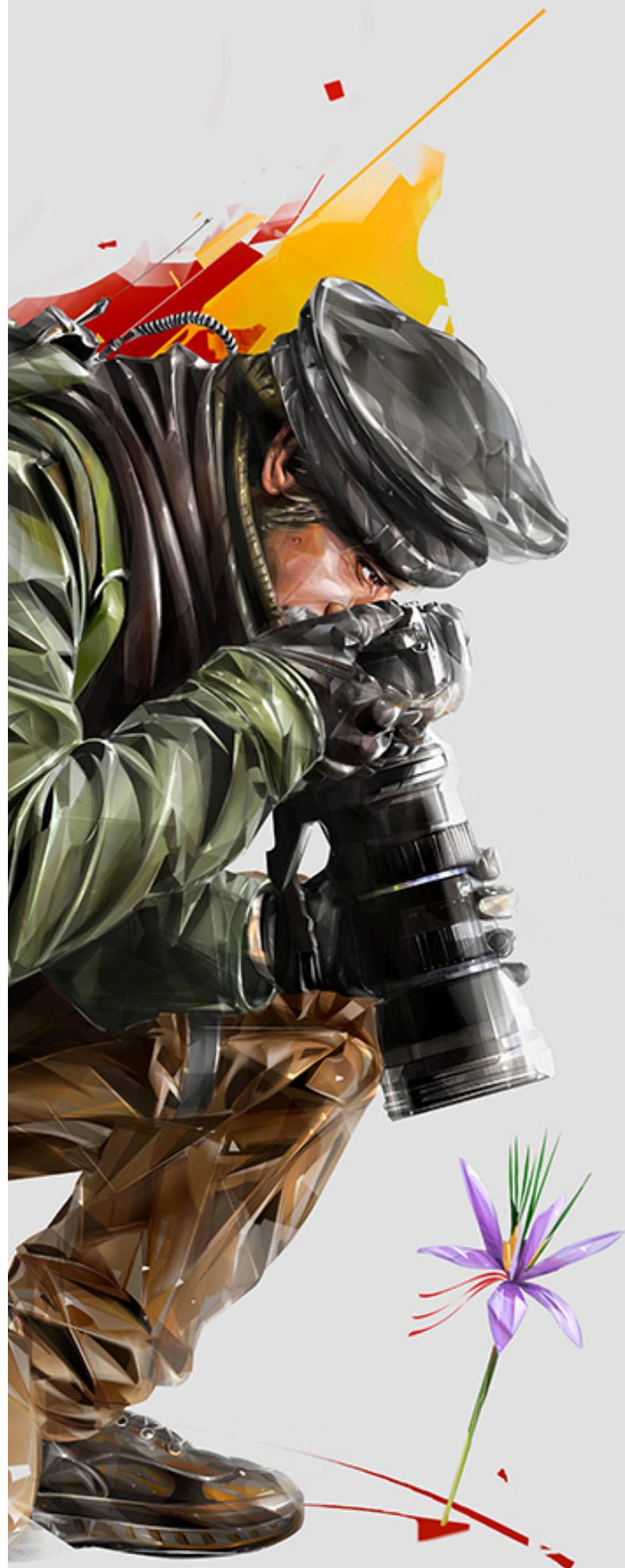
- 01 INTRODUCTION
- 02 IDEAS AND CONCEPTS
- 03 GAMEPLAY MECHANIC ANALYSIS
- 04 USER INTERFACE
- 05 RETROSPECTIVE

FOREWORD

Project Canberra is a first-person shooter I developed during my third year at **SINTLUCAS**. As part of the curriculum, I was given the freedom to create a game based on my own vision. This allowed me to explore my creativity while applying the skills and knowledge I had built up over the years –especially after completing my first internship at the beginning of year three.

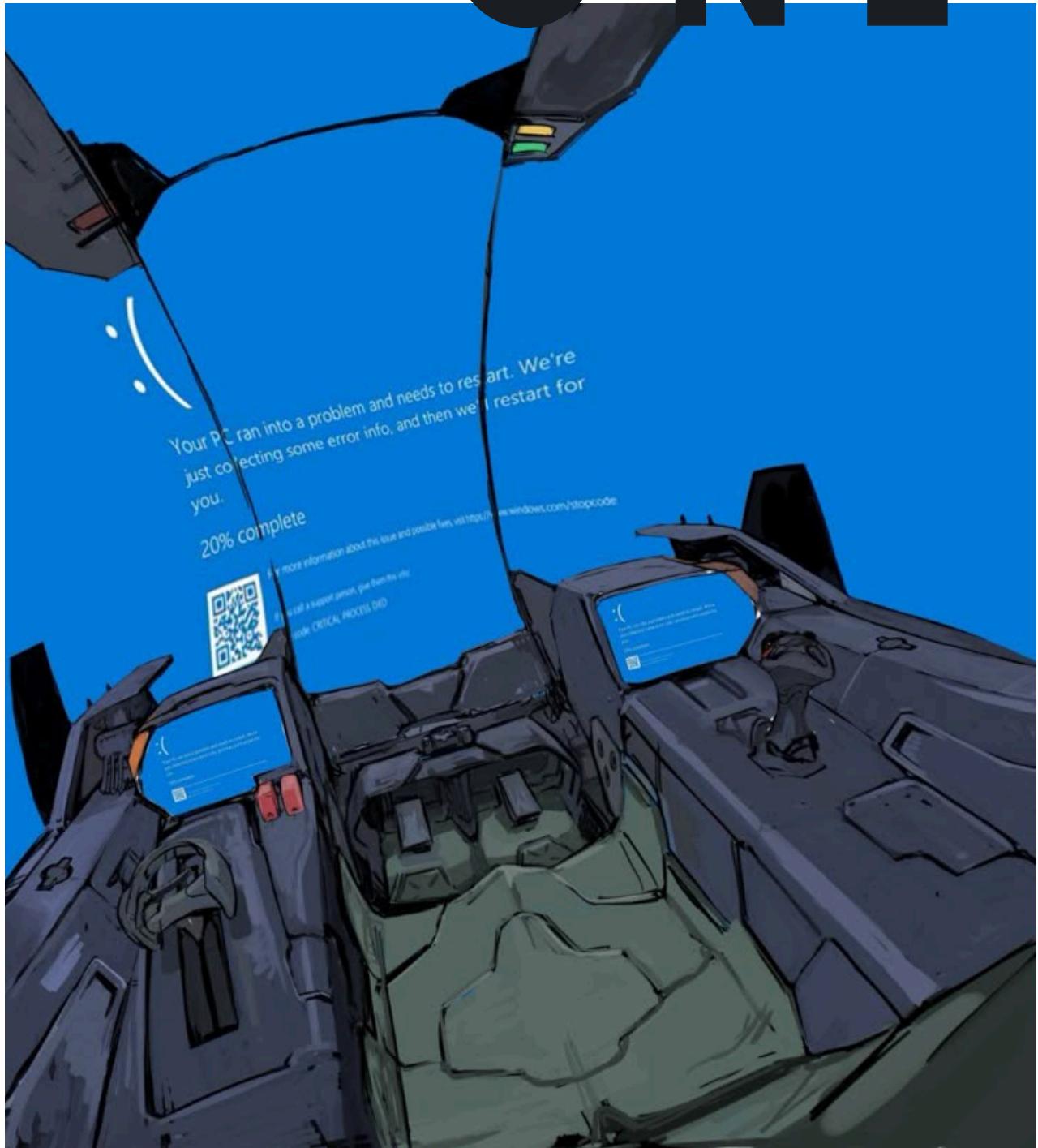
One of my personal goals for this year was to create a **Game Design Document** (GDD). This document outlines the game mechanics, the inspiration behind the design, and how those ideas evolved into the final product. It also offers a behind-the-scenes look at Canberra's development process—from early sketches to the obstacles I encountered along the way.

During the **Software Deepening** and **Game Specialization** modules, I took the opportunity to further refine and technically enhance the project. Explaining game development to people without a background in the field can be challenging. At times, it feels like speaking an entirely different language—full of jargon and concepts that are unfamiliar to outsiders. That's why, alongside telling the development story, I've included a few additional pages to explain these terms and processes. This ensures the document remains accessible to everyone, regardless of their prior knowledge.



Artwork by: Denis Gonchar

Chapter ONE



GET TO KNOW
CANBERRA

IN SHORT

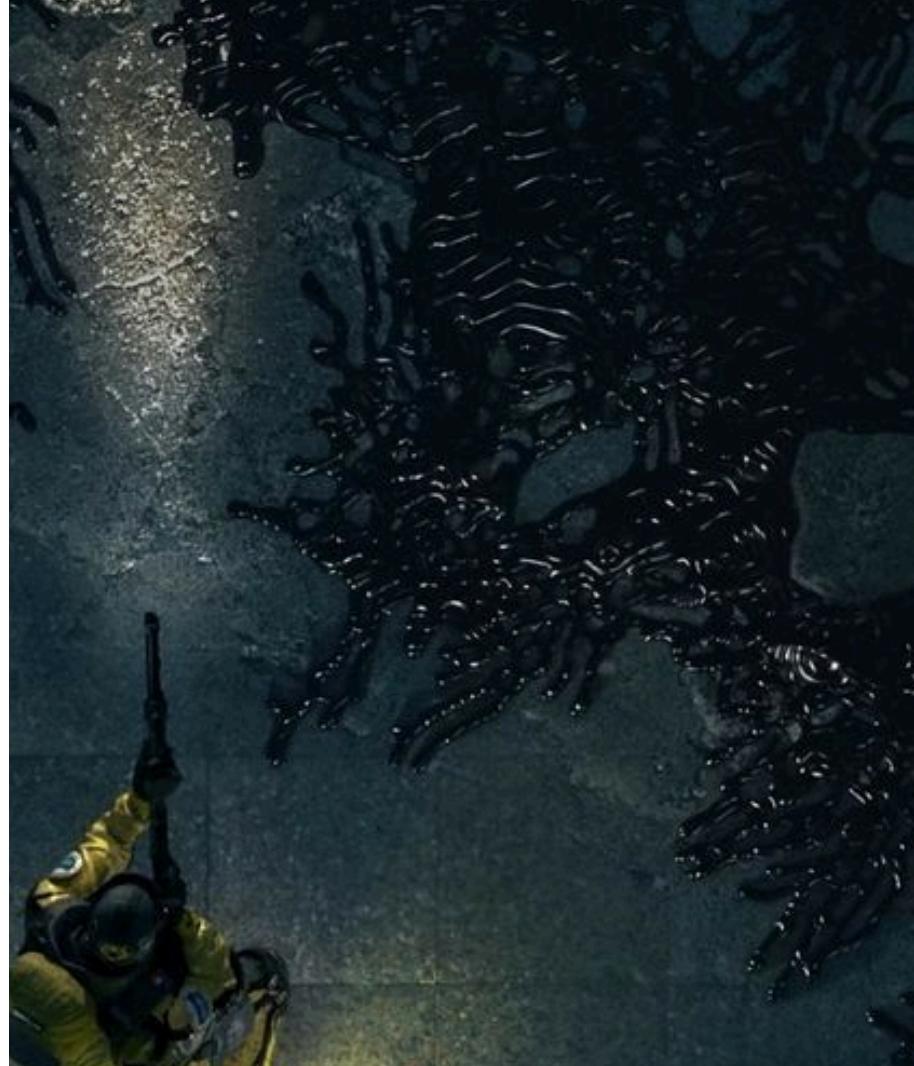
Genre: Horror survival shooter game

Target Audience: Teenagers, ages 13+

Experience Level: Intermediate to experienced

Platform: PC

Unique Selling Points: *Project Canberra* merges key mechanics from several FPS titles, creating a unique and engaging gameplay experience that caters to a wide range of player skill levels.



EXTRA INFORMATION

Short story:

In 2009, a *SARS* pandemic broke out. Those infected mutated into violent, feral creatures known as **Entities**. To bring the situation under control, the government formed a specialized unit: the **Sentinel Tactical Response Unit (STRU)**. Their mission was clear—contain the outbreak and eliminate the threat.

Main objective

As an operative of the *STRU*, you're sent into the heart of the chaos. Your mission: carry out **TOC**-directed (*Tactical Operations Center*) operations to halt the spread and neutralize the threat.

GAMEPLAY PILLARS

Combat

As a member of the STRU, your mission is to fight the '*Entities*' and save the world from a deadly pandemic.

Exploration

You are deployed to unknown and dangerous areas. It's up to you to navigate and explore these zones while carrying out your missions.

Enemies

You will face various types of '*Entities*'. Each has unique behaviors, sizes, and attack patterns—and all are deadly.

Story

The narrative—both the main storyline and background lore—is revealed through documents, flyers, and notes you discover along the way.

Items and Weapons

While exploring, you'll acquire weapons and collect useful items that help you survive and successfully complete your objectives.

WHAT ARE GAMEPLAY DARTS?

*Gameplay Pillars are the core mechanics that define a game's unique identity. They are deeply embedded in the experience and can range from **exploration**, **shooting**, and **survival** to direct **confrontations**.*

Most games feature multiple pillars, which add variety to both gameplay and content. There is no fixed limit to the number of pillars—this is entirely up to the developers.

*A classic example is Pac-Man. Its gameplay pillars include: **navigating and exploring the maze**, **collecting points**, **avoiding ghosts**, **temporarily turning the tables through power-ups**, and **learning enemy behavior patterns**. These pillars make the game challenging, engaging, and highly replayable.*

Chapter TWO

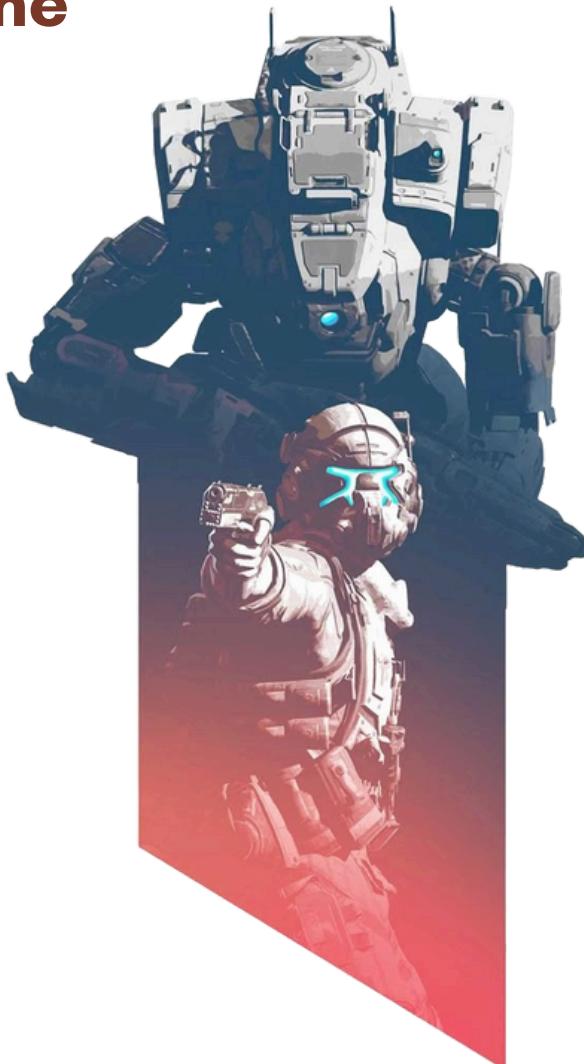


Background source: [gentleman_gasmask](#)

THEY CAN ALL
COME

Before the First Stone Was Laid

Even before the project officially began—yes, *Canberra* was actually in development before the school project kicked off—I already had my own *alpha demo* ready. Initially, that demo was purely a fun side project, but the gameplay mechanics I developed there proved solid enough to build a full-fledged game around. The core was a blend of shooting and parkour, with the standout feature being **wall running**. This mechanic allowed players to run smoothly along walls, jump, and shoot enemies simultaneously, creating a spectacular flow. The inspiration came from *Titanfall 2*, a game by *Respawn Entertainment* that remains a wellspring of inspiration for countless indie developers—including myself. *Titanfall 2* holds a special place in my heart simply because its gameplay and overall experience are exceptional.



Technically, just that one mechanic alone could serve as the foundation for an entire game. But that brought me face to face with the first major challenge during the concept phase: if I wanted to follow *Titanfall 2*'s path of fast-paced, dynamic gameplay, the rest of the game had to match that pace. And that was problematic. From my perspective at the time, I feared that such rapid gameplay would lead to countless bugs and instability. Players would be expected to perform complex and lightning-fast maneuvers, but the system might not keep up. That would compromise the overall experience. Moreover, *Titanfall 2* takes place in a distant future, with humans battling on spaceships or piloting massive *Titans*—super cool, but not fitting within the scope of this project. I needed something slower, more manageable.

Gathering the Building Blocks

So I did what I believe I do best: sought out inspiration. At that time, I was playing two games that would later form the core of *Canberra*: *Tom Clancy's Rainbow Six: Siege* and *Indiana Jones and the Great Circle*. Both titles left a deep impression on me, each in its own way—whether through gameplay, atmosphere, or level design. *Siege* was in the spotlight then because of the *Six Invitational 2025*, the game's largest esports tournament. Watching those professional teams compete, I was struck by their movement and how the firefights unfolded—fast-paced yet incredibly deliberate. One mistake meant harsh consequences. That sense of high stakes and decisive action was exactly what I wanted to bring into *Canberra*.



Tom Clancy's Rainbow Six: Siege



Indiana Jones and the Great Circle

On the other hand, I had completed *The Great Circle*, which initially left less of an impression. However, upon revisiting it later, I noticed how many interesting gameplay elements it included: picking up objects, reading notes, stamina that drains when you run, and so on (more on this later). These small details might have seemed insignificant on their own, but combined with *Siege*'s precision and tension, they gradually formed a clear vision for what *Canberra* could become.

I had a solid collection of gameplay mechanics in hand—but ran into another problem: the setting. *Siege* takes place in the modern day with counter-terrorism units; *The Great Circle* follows *Indiana Jones* in the 1930s hunting treasures around the world. These two contexts were completely incompatible.



The Last of Us

The Arrow That Sealed the Deal

As I continued searching for the right setting, my mind suddenly drifted back to the *COVID-19* period—that confusing, chaotic time when the world came to a standstill. I’m not exactly sure why it struck me then, but that moment unexpectedly sparked a key idea. I remembered *The Last of Us*, by *Naughty Dog*, where an unknown infection transforms humanity into monstrous creatures and the world order collapses entirely. That turned out to be the perfect setting for my concept. Let’s be grateful something like that never really happened, and we don’t have to fight over food or water—but as a concept, it was spot-on: relatable, confronting, and centered on one simple yet powerful question—“*what if?*”—enough to grip players’ curiosity.

In the end, I had gathered enough ideas and mechanics, paired with a setting that not only matched the slower, calculated pace of the gameplay but also allowed for tension and looming threat: a world where every step must be carefully considered, where hordes of zombies can overwhelm you in seconds. It was a shame to let go of *Titanfall 2*’s fast-paced parkour mechanics, which held a lot of potential. Maybe those will resurface in another *Game Design Document* someday. But for this project, I made a conscious choice—**Canberra demanded something different.**





World Design in Video Games

In game development, the game world plays a fundamental role—it is essentially the player’s “playground.” Building a world goes beyond simply drawing a map: it’s a way for developers to tell stories, guide experiences, and evoke emotions. For example, a game designed for children aged five and up would be completely out of place if set in a violent, desolate environment—and vice versa. In *Canberra*, world design is developed concurrently with the story and mission structure. A believable, well-thought-out world enhances the gameplay experience and reflects the game’s overarching theme. There are various forms of world design, each with its own characteristics and purposes. Below is an overview of the most common types, so you can better understand how *Canberra*’s design fits into the bigger picture.

Linear World

The player follows a fixed path predetermined by the developer. This design is common in narrative-driven action games like *Uncharted 4* or *Call of Duty*. The major advantage is that pacing is tightly controlled, creating a directed experience—like living a story through the protagonist's eyes. The downside is the lack of freedom: there's little room for exploration outside the set path.

Open World

Players can freely roam, explore areas, and decide their own activities. Think *The Witcher 3*, *Skyrim*, or *GTA V*. This approach emphasizes freedom and immersion but demands significant development time and resources. Without enough meaningful content, such worlds can quickly feel empty or repetitive.

Sandbox

Similar to an open world but with a focus on systems and freedom: players receive tools to create their own goals and experiences. Examples include *Minecraft* and *Garry's Mod*. Sandbox games foster creativity, but without clear direction, players may feel aimless.

Hub World

The world centers around a hub—a central point from which players select missions and enter different areas. Upon completion, they return to the hub. This structure, used in games like *Bloodborne* or *Monster Hunter*, strikes a balance between linearity and openness. It keeps progress manageable, though the segmented structure can occasionally disrupt immersion.



Fun Fact: *Hades* was the very first video game to win a Hugo Award for literature—a historic milestone proving that games can excel in storytelling.

Semi-Open / Metroidvania

Players start with limited access to the world and must acquire new abilities or items to advance. Seen in titles like *Metroid Prime*, *Hollow Knight*, and *God of War (2018)*. This design encourages exploration and provides a strong sense of progression but can frustrate players without clear guidance.

Procedurally Generated World

The world—or parts of it—are randomly generated by algorithms each playthrough. This ensures unique, unpredictable sessions and boosts replayability. Examples include *Hades*, *No Man's Sky* en *Enter the Gungeon*. It saves development time and keeps the game fresh but can sacrifice narrative depth and carefully crafted moments.

Arena-Based Design

The game world is divided into enclosed zones where players face challenges like combat or survival. Think *DOOM Eternal*, *Valorant*, or *Rainbow Six: Siege*. This structure makes gameplay balance and combat focus straightforward. However, it often lacks the feel of a living, cohesive world—unless well-integrated within a strong narrative context.

Many choices, but at the same time, not so many

There are countless ways to design a game world—ranging from open worlds to linear structures, arenas, sandbox environments, procedurally generated worlds, or hub-based designs. Each model has its own strengths; there is no universal “best” design. As a solo developer working within a tight three-month development window, I had to choose an approach that fit my capabilities and resources.

Open or semi-open worlds offer freedom and scale but demand enormous effort: building maps, populating worlds with content, events, and systems. Within my time constraints, this simply wasn’t feasible. Linear worlds are appealing for their control over narrative and pacing but require precise scripting and timing—areas where I currently lack sufficient experience.

Sandbox and arena setups might seem simpler but bring their own challenges. Sandbox requires advanced systems and player creativity tools, while arenas without clear rewards or progression can quickly become repetitive.

Procedurally generated worlds looked attractive for their replayability, but I have no prior experience with procedural generation. While I’m open to new techniques, it wouldn’t be wise to experiment with entirely new methods without room for a learning curve in this project. Perhaps in a future project—or in a limited form here (such as randomized enemies or loot).

The final choice (which I already knew)

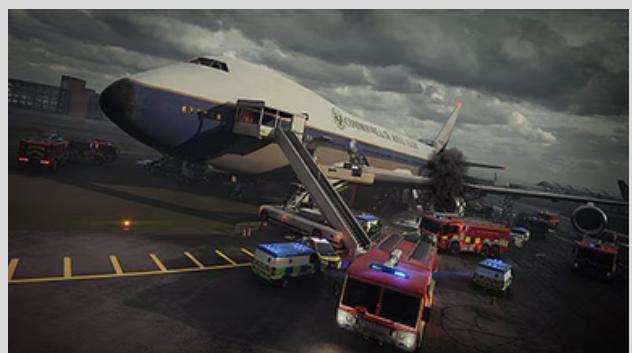
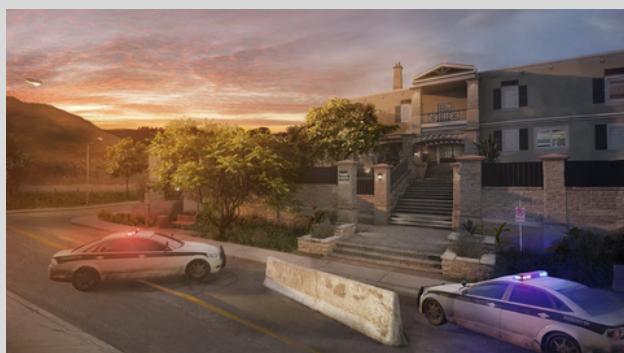
After careful consideration, I settled on a **hub-based world design**—a structure where the player travels from a central point (the “hub”) to various regions to complete missions and then returns. This model gives me, as a developer, more control over pacing, content, and world-building. At the same time, it preserves the player’s sense of freedom and adventure.

A hub-based approach has already proven successful in *The Great Circle* and *Rainbow Six: Siege*. Both games implement it differently—*Siege* focuses on compact, tactical combat in vertical environments, while *The Great Circle* features larger zones with semi-open paths and localized missions. Yet, they share the same core philosophy: the player departs from a familiar home base, completes a task in a specific region, and returns. This structure balances developer manageability with spatial variety for the player.

This same approach will form the backbone of *Canberra*’s world: modularly designed areas with their own atmospheres, challenges, and stories that together create a believable and explorable game world.



Where *The Great Circle* takes you back to the threatening 1930s, marked by technological limitations, *Siege* drops you right into a harsh, modern reality—with maps based on real-world events.



Presidential plane, based on Air France vlucht 8969



Oregon, based on Waco siege

Chapter

THREE



LOOK, AN
ANALYSIS PHASE

3.1 CORE OF THE GAME

The central mechanic in Canberra revolves around a gameplay loop where the player is deployed into a specific area and receives various assignments from the **TOC** (*Tactical Operations Center*). Since the game is still in development, no formal reward systems have been implemented yet. The “reward” for the player currently lies mainly in the experience gained and the lessons learned from mistakes that lead to a game over.

At present, the game contains only one level. Completing this mission equates to successfully finishing the game.

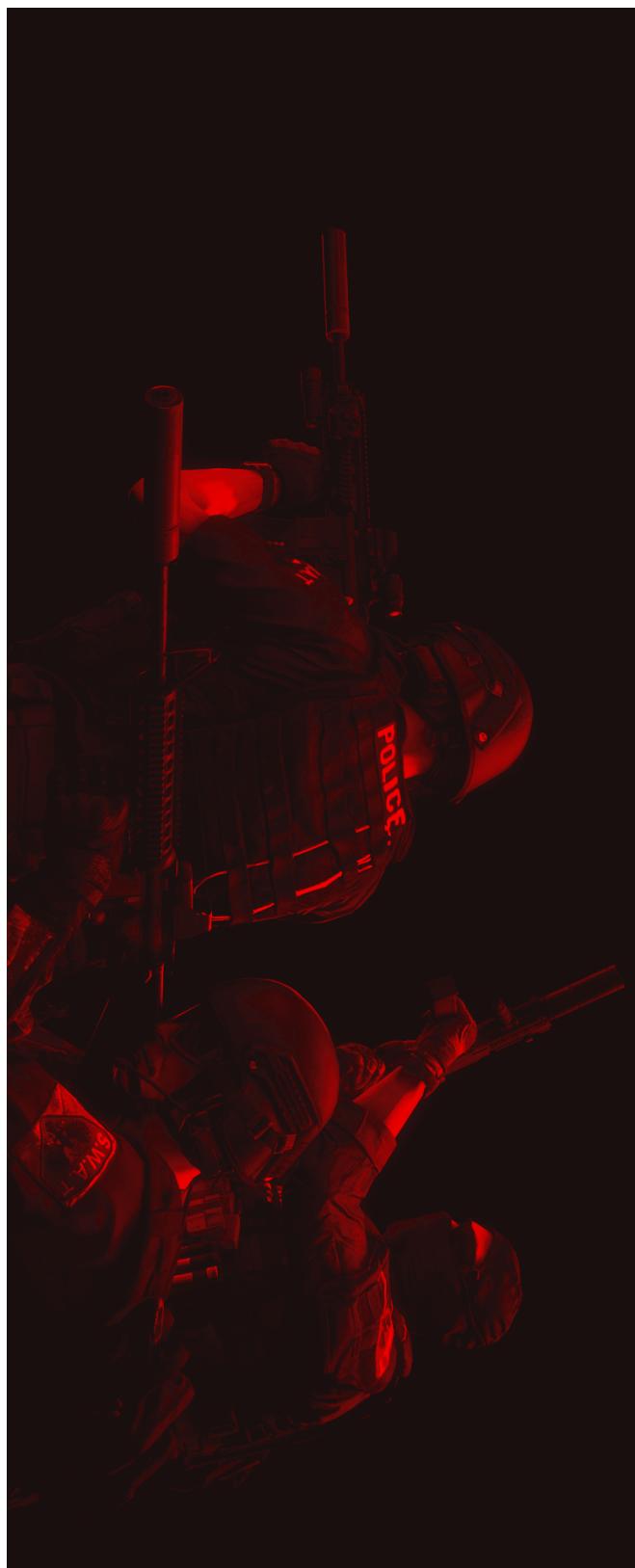
3.2 OBJECTIVES

As a member of the *STRU* resistance unit, the player’s task is to penetrate deep into the heart of the outbreak zone. The mission is to stop or completely eliminate the infected entities, with the ultimate goal of saving the world from catastrophe.

3.3 PROGRESSION DETAILS

In the first mission, the player is deployed to a residential area where the virus outbreak has occurred. The objective is to eliminate a set number of hostile entities and collect enough *DNA* samples. These *DNA* samples are then sent back to the resistance camp for further research.

Ready or not (Void Interactive)



3.4 GAME MECHANICS

Canberra is a *First-Person Shooter (FPS)* centered around its core shooting mechanic. While shooting forms the foundation of the gameplay, several additional elements have been incorporated to enrich the overall experience:

Exploration: Players are encouraged to thoroughly explore the environment to find hidden items and important clues.

Resource Management: Careful management of ammunition and survival tools is essential to stay alive.

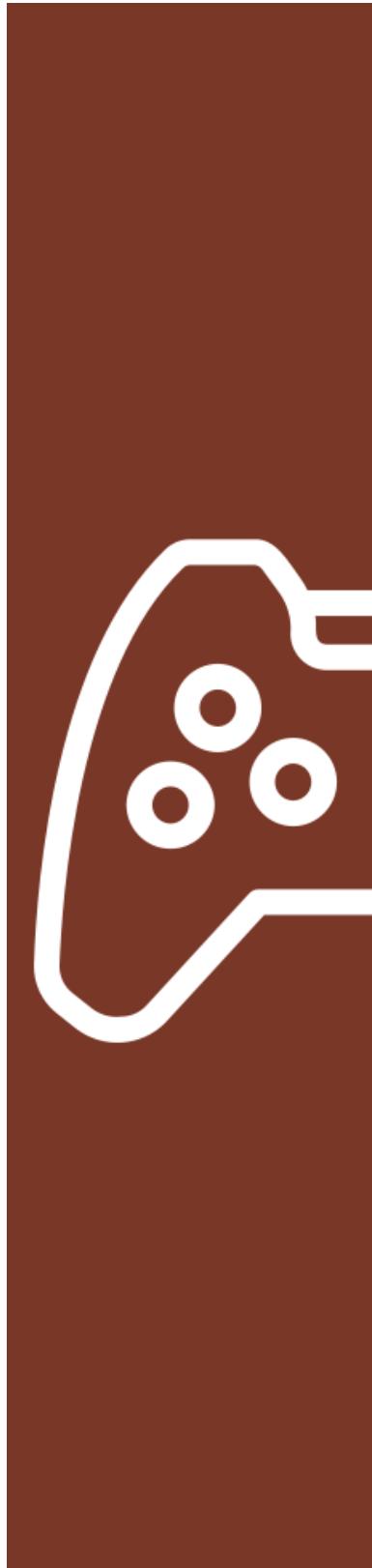
Gas Mask System: Players must strategically decide when to enter hazardous zones where toxic air or contaminants pose a threat.

Lighting: Light plays a crucial role. Enemies react strongly to light sources, so players must use their flashlight wisely to avoid drawing unwanted attention. This encourages moving cautiously in the dark and avoiding direct confrontations.

Lean Mechanic: Players can lean left or right to peek around corners without fully exposing themselves, adding a tactical layer to engagements.

Permadeath: Failure in a mission results in a game over, requiring the player to restart the level from the beginning.

Together, these mechanics create an intense and strategic gameplay loop where every decision counts.





3.5 CONTROL MECHANICS

The movement mechanics in *Canberra* draw inspiration from several sources, with a primary focus on *Siege* and *The Great Circle*. The majority of the player movement is modeled after *Siege*, while the system for picking up and using items leans more towards the approach taken in *The Great Circle*. A few mechanics are hybridized with elements from *Siege* to shape *Canberra*'s unique control scheme. Notably, this project marks the first time *Canberra* utilizes the [new Input System](#), replacing the legacy Input Manager that has been the industry standard for years. The main improvements lie in the system's increased flexibility and finer control. With the *new Input System*, you can:

1. Easily support multiple input devices (keyboard, mouse, gamepad, touchscreen).
2. Decouple actions from specific keys, enabling seamless button remapping.
3. Trigger input based on events rather than continuously polling keys, improving performance.

In short, this new system facilitates the creation of sophisticated and modern controls—perfectly suited for *Canberra*'s tactical movement, leaning mechanics, and complex interactions.

Below is the list of controls for movement, combat and using objects in *Canberra*:



Movement:

- Move mouse – Look around
- WASD – Move in respective directions
- WASD + mouse movement – Move in the direction you're looking
- WASD + Left Shift – Sprint (stamina consumption applies)
- Press C – Crouch
- WASD + C – Move while crouched (slower movement)

Leaning:

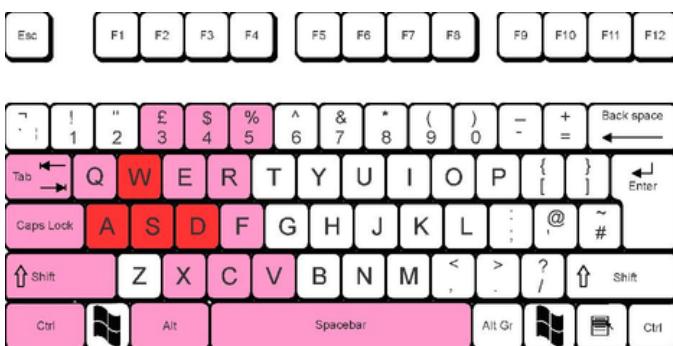
- Hold Q – Lean left
- Hold E – Lean right
- WASD + Q/E – Move while leaning
- WASD + Q/E + Left Shift – Sprint while leaning
- WASD + Q/E + C – Move slowly while leaning

Interaction & Items:

- Press F – Interact with objects (items, notes)
- Press 1, 2, 3 – Use items (if available)
- Press M – Toggle gas mask on/off (requires filter)

Combat:

- Hold Left Mouse Button (LMB) – Aim
- Press/Hold Right Mouse Button (RMB) – Shoot (single shot per press or automatic while held, depending on weapon)
- RMB + LMB held – Shoot while aiming
- Press R – Reload
- Press T – Toggle flashlight on weapon on/off
- Mouse wheel – Switch weapons



WASD has become the standard for PC controls because it gives the left hand efficient movement control, while the right hand operates the mouse. The keys are also conveniently close to other function keys, which ensures ergonomic use and quick access during gameplay.

3.6 Enemies and Failure Conditions



During their mission, the player encounters various obstacles, with the greatest threat coming from the so-called *Entities*. Individually, these enemies may not seem particularly powerful, but in groups they form a deadly force. Since the player operates solo and has access to only a limited amount of ammunition, a constant tension builds: the longer you remain in one place, the greater the enemy pressure becomes. Eventually, the growing swarm will surround and overwhelm you.

At this stage of development, the game features three main types of anomalous entities. Two of these types each have three variants to introduce behavioral and threat-level diversity, while the third type represents a unique, exclusive entity. These enemies move freely across the map, forming a dynamic and unpredictable challenge for the player:

Undead - These entities appear most frequently and pose the lowest threat level. They move slowly and their attacks are individually weak. However, they can become dangerous in large numbers.



Undead Punk



Undead Teenager



Undead female officer

Feral - One level above the *Undead*. They appear less frequently but are faster and more alert. They can detect the player from a greater distance and pursue relentlessly. Taking down a *Feral* is considered a moderate challenge.



Feral female flower shop owner



Feral office worker



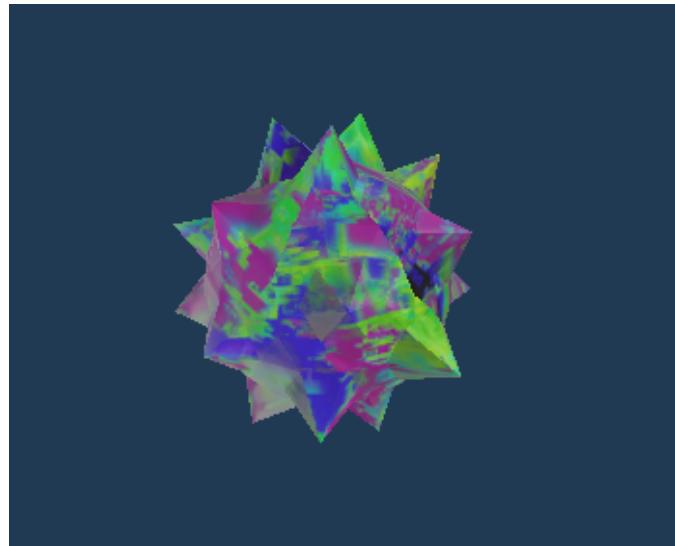
Feral homeless man

Brute – The most formidable and powerful entity in the game. They are physically imposing and deal the highest amount of damage. Within the game world, the origin of these monsters remains unknown, and their rare appearances make them all the more terrifying.



In addition to the grotesque entities, there are other dangers present in the game – namely a strange kind of **thorny orb**. These objects **do not move or actively attack the player**, but they still pose a very real threat. The *SRTU* still has no idea where these orbs come from or who created them, but one thing is certain: they are always **located in hard-to-see places**, often in **dark or narrow corners** where players can easily overlook them if they're not paying close attention.

When a player walks past one, the orb immediately explodes. The explosion temporarily disorients the player through **visual distortion (a blurry or blacked-out screen)** and a **strange, unsettling sound**. They are designed to induce confusion and frustration – and while **not deadly** on their own, they can play a critical role in tense situations, especially when enemies are nearby.



In addition to the troublesome trap orbs, **radioactive contamination** also poses a serious threat to the player. Entering these zones without proper protection almost certainly results in death if you remain there for too long. These areas are visually marked by a dense, deadly green haze – a clear signal that radiation is present and that you should avoid the area unless you're equipped with the proper gear.

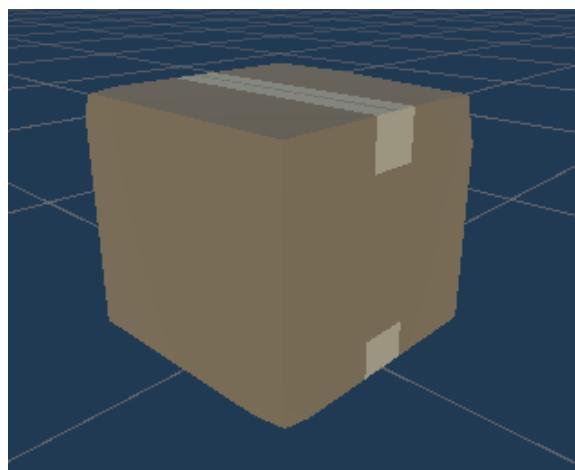




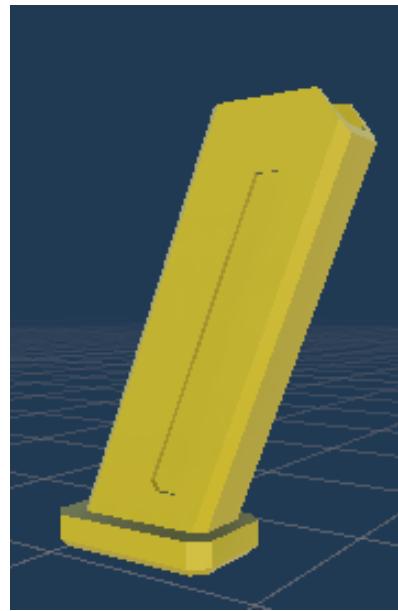
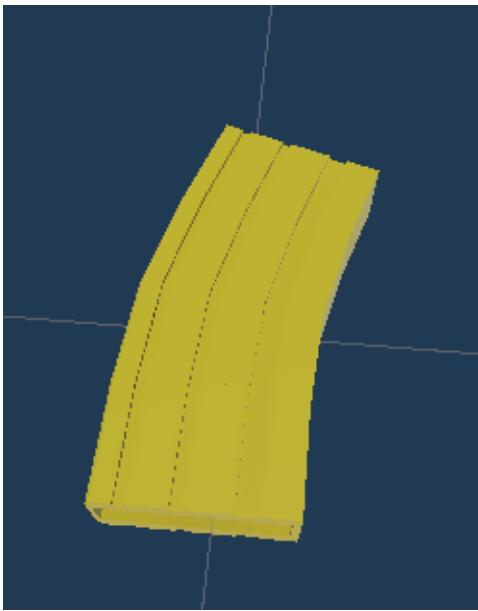
3.7 Equipment and Items

In the game, the player can collect various support items, such as **bandages**, **medication**, **gas mask filters**, and different types of **ammunition**. These items are scattered across the map; some are immediately visible and easy to pick up, but most are found in so-called **loot boxes**. When a player opens one of these boxes, its contents are randomly generated – the outcome is different each time.

This mechanic increases the element of surprise and keeps the gameplay fresh: players never know exactly what they'll get – it might be the item they desperately need, or something that could come in handy later. At the same time, this randomness prevents players from memorizing the locations of the most valuable items, discouraging repetitive looting of the same containers and encouraging exploration of other parts of the map.



Let's start with the most critical tool: ammunition. Without it, the player is essentially a defenseless outsider, completely at the mercy of the *Entities* – perfect prey for their attacks. To prevent players from falling into a hopeless situation too quickly, the game begins with a base supply of 325 rifle rounds and 127 pistol rounds. This ammunition is marked with a bright yellow color to ensure better visibility in the dark game world, where lighting is scarce and players can quickly lose their sense of direction. Moreover, the system continuously monitors the amount of ammo the player is carrying: once the cap of 300 rifle rounds or 120 pistol rounds is reached, no additional ammunition can be picked up until enough has been spent.

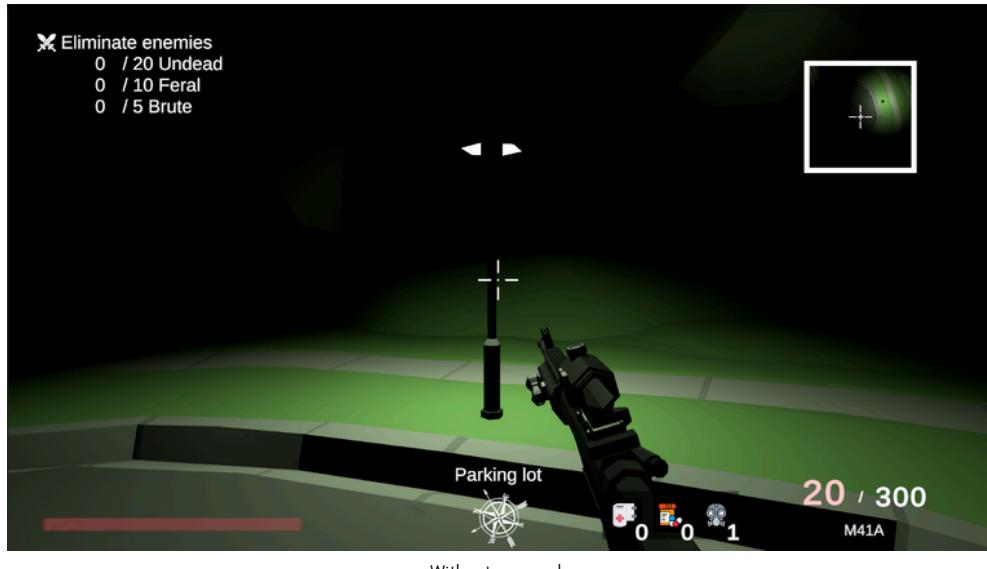
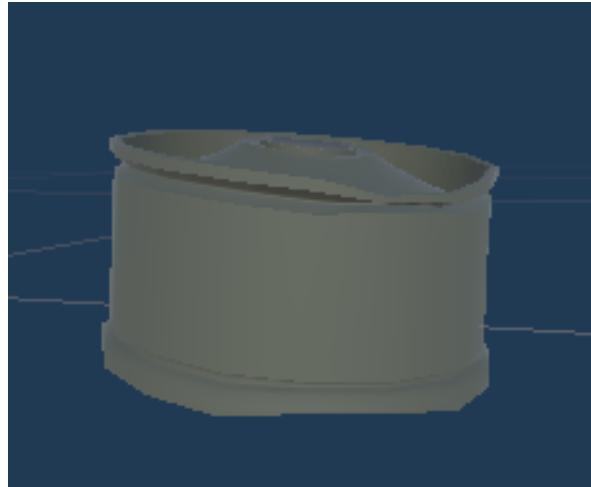


If ammunition allows you to defend yourself against threats, then **bandages** and **medicine** are the essential backbone for recovering from injuries and extending your chances of survival. Without these tools, every mission becomes significantly more difficult. Bandages restore a fixed amount of health instantly, while medicine provides gradual healing over a short period of time. While both serve the same purpose, their use scenarios differ: medicine is ideal for treating minor wounds, while bandages are more effective when the player is in critical condition.



Last but not least: gas mask filters. These items aren't constantly needed but become invaluable the moment the player must enter radioactive or toxic zones. Each filter provides roughly three minutes of protection, allowing access to otherwise deadly areas.

However, using one comes with a clear drawback: the player's visibility is significantly reduced. This creates a claustrophobic sensation – almost as if you're really wearing a gas mask – enhancing the tension and realism of the experience.



3.8 LEVEL DESIGN

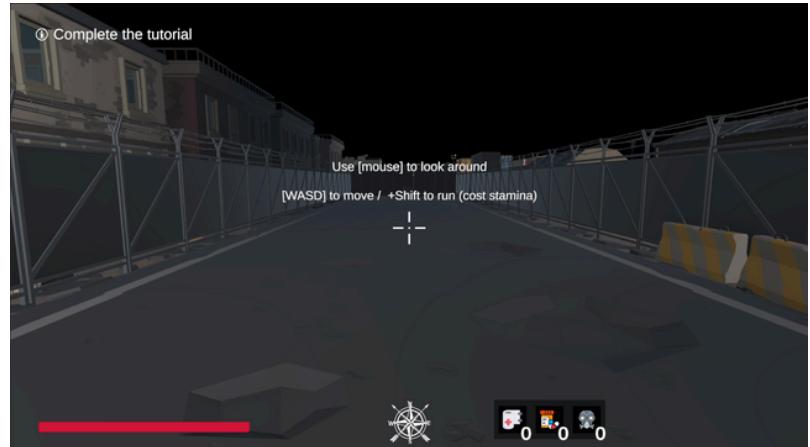
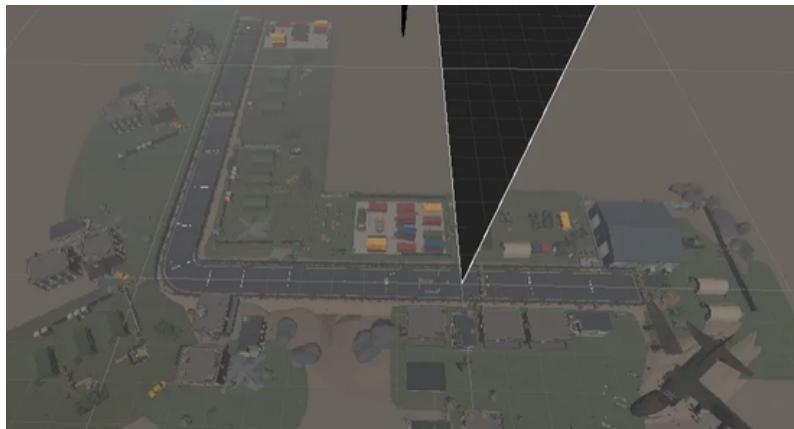


3.8.1 Tutorial Level

Before the game truly begins, the player must first go through a specially designed tutorial level. In this introductory phase, the player's movement is intentionally limited – they can only move forward. This approach allows the player to gradually become familiar with the game's core mechanics without being overwhelmed by too many options at once.

As the player progresses through the linear environment, on-screen prompts appear. These provide clear explanations about the controls, the interface, and key gameplay concepts. Each instruction builds logically upon the last, guiding the player in a structured and intuitive manner to master the controls and understand the overall gameplay structure.

Once all steps have been completed and the player has tested all core mechanics, the tutorial level ends automatically. The player is then returned to the main menu, fully prepared to begin the actual mission – equipped with foundational knowledge and a sense of familiarity with the game world.





3.8.2 Firing Range

The instructional portion of the game primarily focuses on introducing the core mechanics. However, it quickly becomes apparent that this segment significantly restricts the player – especially during the shooting tutorial.

Once the player has grasped the shooting mechanics and all training targets are taken down, the targets do not respawn. As a result, there's little left for further practice or experimentation. To address this limitation, the idea for a dedicated firing range was introduced – an additional game environment that was implemented shortly thereafter.

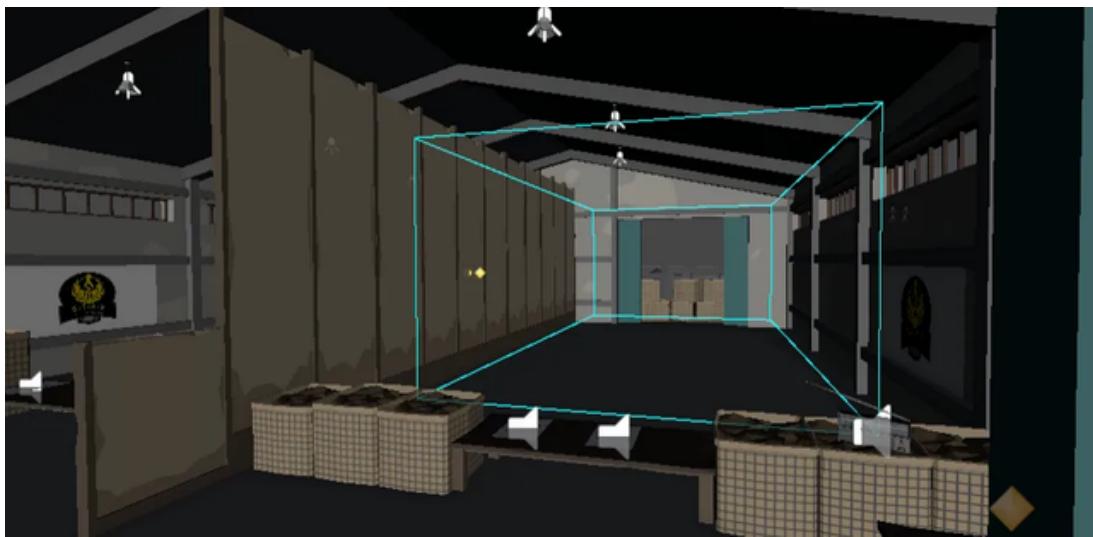
As the name suggests, the firing range provides players with the opportunity to freely practice using various weapons. The firing range is designed as a space where players can refine their skills, improve their aim, and familiarize themselves with the unique characteristics of each weapon. It's a valuable supplement to the tutorial and helps instill the confidence and control necessary to begin the real game well-prepared.

3.8.2.1 Zone 1 – Fundamentals

In this first training zone, players are introduced to the basic principles of shooting. Around five hostile *entities* continuously appear here, each with limited mobility, allowing players to focus entirely on precision and reaction time.

The zone is designed as an accessible entry-level environment, free from time constraints or ammo restrictions. Players are encouraged to shoot freely and experiment with different weapons. Once an entity is neutralized, a new one spawns instantly at a random location, ensuring the training loop remains fluid and uninterrupted. This setup enables players to practice endlessly, building confidence and becoming fully comfortable with the weapon mechanics and combat feel before progressing to more demanding challenges.





3.8.2.2 Zone 2 – Accuracy Training

This zone is specifically designed to help players improve their aim. Instead of hostile entities, golden orbs float in the air, scattered across the training area. Every time a player successfully hits an orb, a new one instantly spawns at a random location.

Due to the unpredictable nature of the spawn positions, players are constantly challenged to stay alert, react quickly, and shoot with precision. This environment not only encourages sharp shooting but also trains aim speed and control under light pressure. It's a valuable space for anyone looking to sharpen their accuracy before facing real danger on the field.





3.8.2.3 Zone 3 – Hostile Environment

This final training zone simulates a hazardous environment heavily contaminated with toxic gas – a realistic preview of what players will encounter later in the game. To safely enter this zone, wearing a gas mask is mandatory. Without proper protection, the player begins taking damage from poisoning within seconds, which will eventually prove fatal.

Zone 3 is more than just a training ground for firefights. It emphasizes the importance of preparation, situational awareness, and effective resource management. Here, players not only learn to handle hostile entities, but are also forced to think strategically about their equipment, timing, and positioning. It's a stress test that combines physical skill with tactical thinking – essential for surviving the later, more dangerous parts of the game.

Have you ever wondered how I create a shooting system in Canberra? It's not what you think!

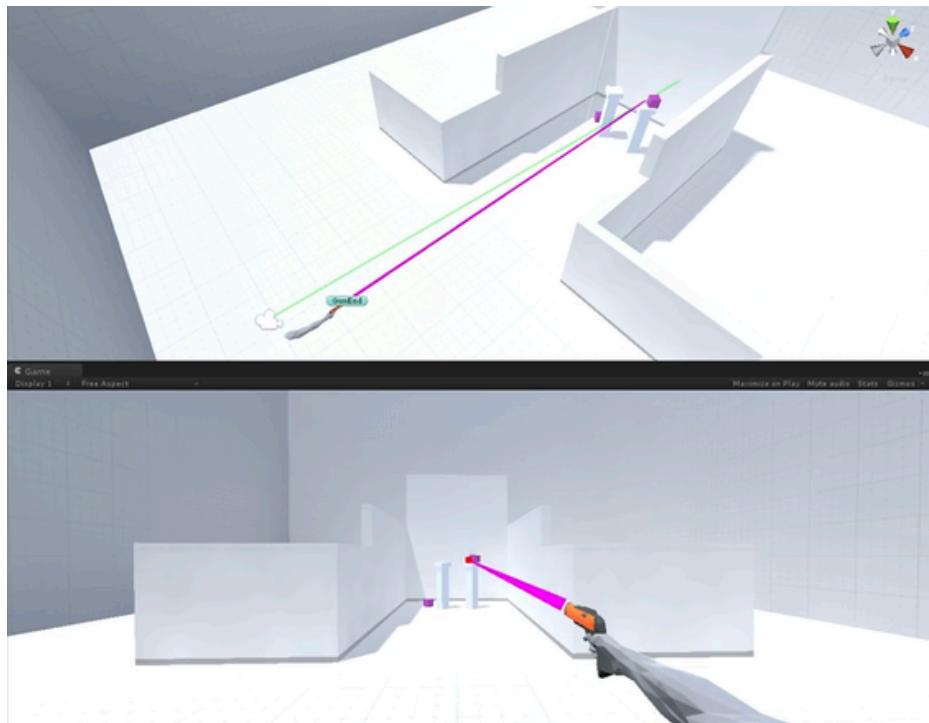
Lots Of Pew Pew, No Bullet In Sight

The shooting system in Canberra is based on **Raycasting**.

Instead of physical bullets flying through the barrel like in real life, the game uses an invisible **laser beam** to determine where the shot lands. This technique is commonly used in modern FPS games due to its efficiency and performance benefits.

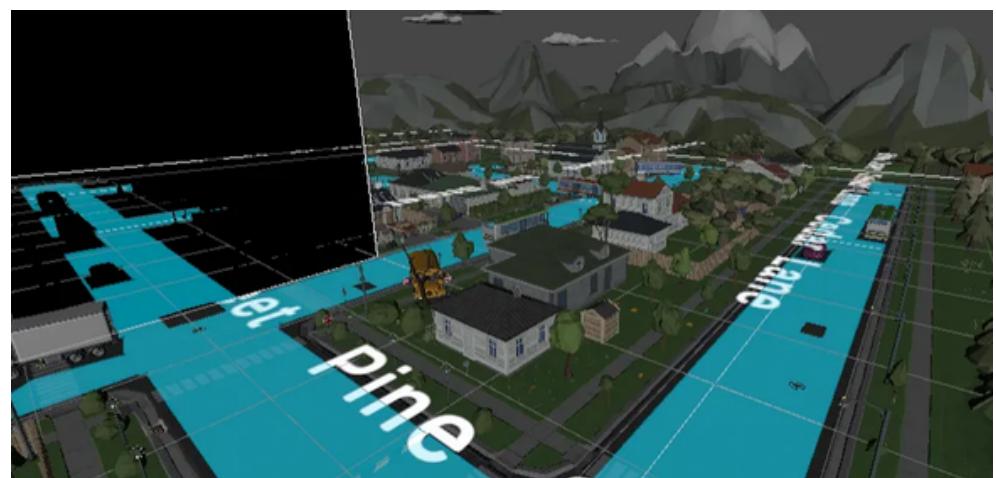
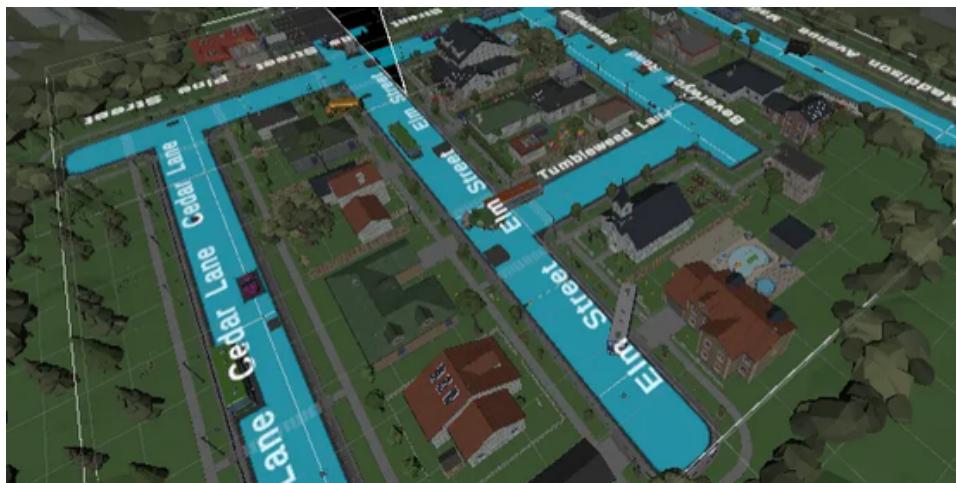
When physical bullets are used, the system must generate each projectile individually, including force, direction, and collision detection. This requires significant processing power, especially when players fire continuously. Moreover, most bullets are not even noticed by the player or disappear immediately after impact.

Raycasting bypasses all these steps. The system only needs to calculate the shot and damage instantly, regardless of how many bullets are fired. The result is a smoother gameplay experience with more stable performance, without sacrificing the realistic feel of shooting.

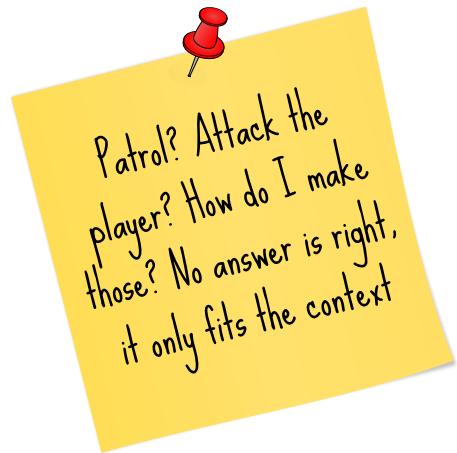


3.8.3 First Level

The first level takes place in an abandoned town, inspired by urban warfare. Players navigate narrow alleys, dilapidated buildings, and war-torn streets, remaining constantly alert to hostile entities. During exploration, valuable items must be collected to survive and complete the mission. Strategic decisions regarding route, cover, and timing are crucial. Every corner can hide both danger and resources, creating constant tension and a tactical challenge.



An Element That Didn't Even Fulfill Its Intended Purpose.



A problem encountered during the development of the **spawn mechanism**: when enemies spawn repeatedly at fixed locations, players quickly learn and exploit these spots to easily eliminate foes. This results in unfair gameplay and reduces the challenge. Therefore, this method had to be replaced.

The chosen solution was the use of ***NavMesh Surface***. This is a Unity feature that determines the areas where AI is allowed to move by analyzing the environment and generating a navigation mesh (bake). NavMesh is traditionally used to:

- Calculate optimal paths for AI (*pathfinding*).
- Allow AI to move smoothly over complex surfaces such as stairs and slopes.
- Limit permitted movement zones to safe areas.

In *Canberra*, *NavMesh* is creatively employed as an alternative spawn mechanism. Instead of fixed spawn points, enemies now appear at random, valid locations on the map – as long as those locations fall within the *NavMesh*. This approach offers two major advantages:

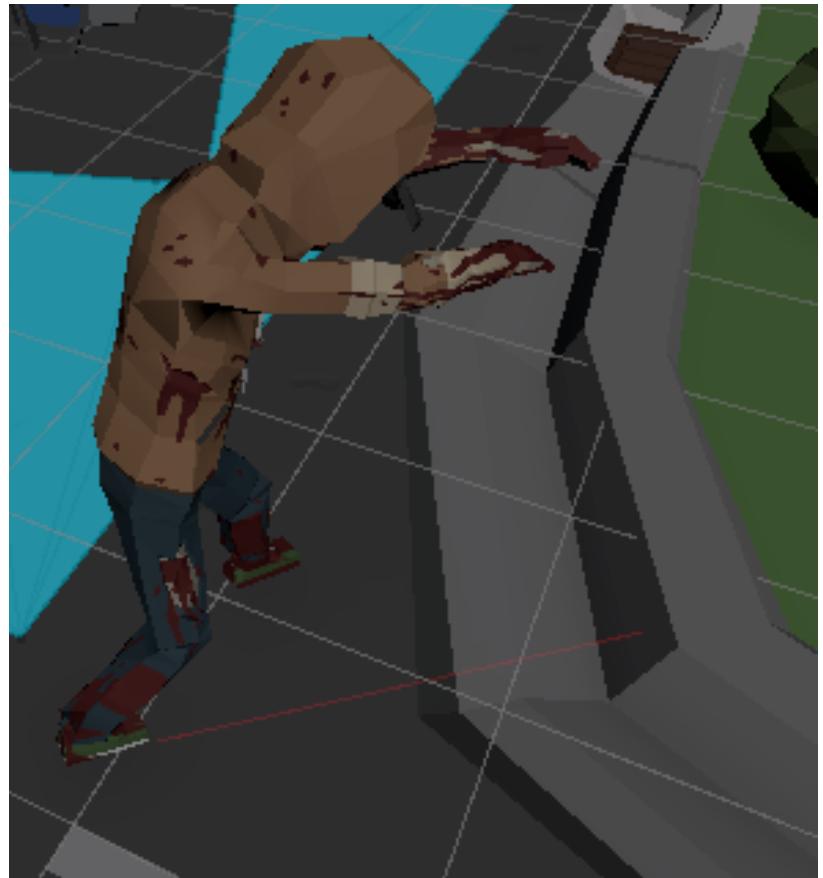
1. Players cannot predict spawn locations, increasing tension.
2. It prevents errors where enemies spawn in unwanted places like rooftops or blocked areas.

That's why you see a blue-colored walkable surface in the design scene – it's the visual result of the baked *NavMesh Surface*.

Where are we going?

While moving, an enemy AI projects a red line in its viewing direction, visible in the game view. This line functions as an obstacle detection system. When the line hits an object, the system interprets this as an impending collision, and the AI automatically turns away—randomly choosing a direction away from the obstacle.

If the AI moves unobstructed for a certain period without detecting obstacles, a random direction change is forced anyway. This prevents the enemy from traveling too long in a single direction, making its behavior more dynamic and less predictable.



You Missed Something – No, I Didn’t

Initially, the *entities* in my game were nothing more than gigantic, walking pills. At the time, I hadn't yet considered using 3D models for enemies simply because I had little experience with animation and wasn't sure I could handle it. I thought these three-part, colorful capsules would be enough to build a frightening shooter. Of course, I knew this came with compromises to the gameplay experience—especially since the rest of the game world consisted of fully 3D models—but I accepted that.

Only when I had others playtest my game did I get feedback that I really needed to consider proper 3D models with animations for the enemies. At first, I dismissed the advice—it seemed like too much hassle. But after some time, I started to doubt myself and eventually decided to take the plunge: I would give my ‘entities’ real models with animations, even though I wasn't thrilled about it.



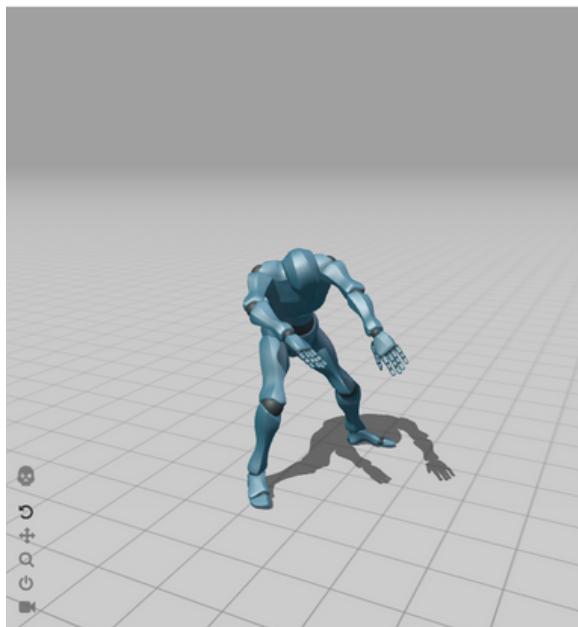
Luckily, I already had access to suitable 3D models, so that wasn't an issue. The real challenge was the animations. Creating animations myself quickly proved to be very time-consuming, especially for models with multiple limbs and fleshy structures. During my search for a solution, I stumbled upon [Mixamo](#), a platform that offers free ready-made animations. Not everything I needed was available, but the selection was broad enough to cover most of my goals.



... This Actually Turns Out Quite Fun

I began importing the *Mixamo* animations into my game. At first, it wasn't without challenges—I had little experience and struggled to understand exactly how animation works within the game engine. Yet, after enough experimentation, I successfully applied my very first walking animation. Shortly after, I added animations for detection, pursuit, and finally attacks, bringing the enemies to life as I had envisioned.

Once the system was up and running, I decided to expand it further. I developed multiple enemy variants to add more variety and challenge to the game.

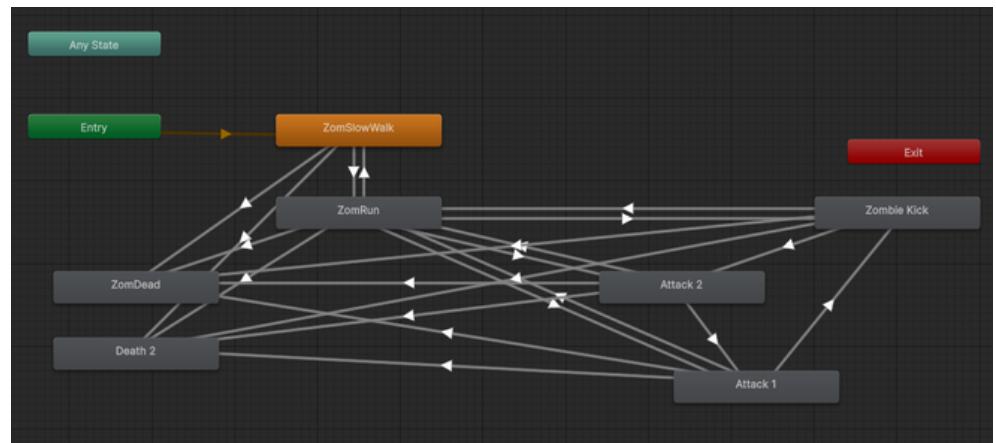


Mixamo animatie

The image below shows the **Animator** – an indispensable component in Unity for managing animations. With this tool, developers can precisely control the sequence in which animations play and how they transition into one another. For any project involving animation, the Animator is essential.

Although powerful, the *Animator* has a clear drawback: with a large number of animations, the overview quickly becomes a tangled web of connections. The whole thing starts to resemble a spider's web, making it difficult to maintain clarity and implement changes.

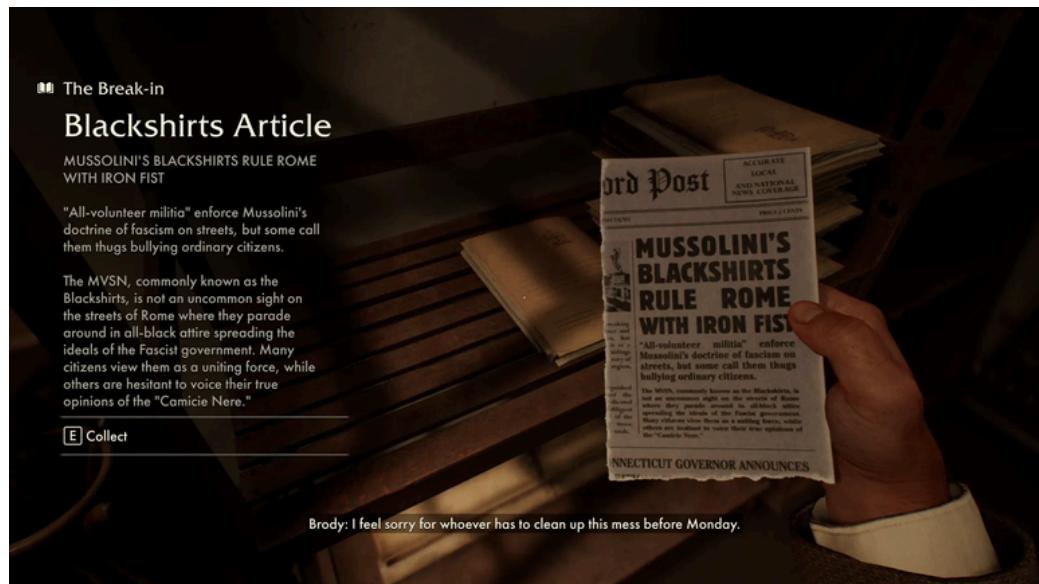
In my case, the number of animations isn't extreme, but even then, the system already looks quite complex.



Chapter FOUR



INFORMATION > GOLD



The **user interface** in Canberra draws inspiration from multiple games, primarily *Tom Clancy's Rainbow Six: Siege* and *Indiana Jones and the Great Circle*. Most mechanics related to **movement** and **gunplay** are directly derived from *Siege*. This includes how players navigate the environment, check corners, and handle weapons with precision.

On the other hand, **interactions with the game world** – such as reading **notes**, **picking up items**, and **exploring levels** – take more influence from *The Great Circle*. These elements ensure that players engage not only in combat but also in narrative and exploratory aspects of the game.

The UI plays a crucial role in virtually every game by providing essential information that players rely on to respond to various situations. Think of details like remaining ammunition or potential enemy locations. In Canberra, where the information the player receives is intentionally limited and filtered, the UI becomes even more vital. In dangerous or chaotic moments, on-screen **information** can be **more valuable than gold**. It is precisely this tension between what the player knows and what they suspect that can mean the difference between survival and failure.

4.1 Analysis of UI Components in Canberra

4.1.1 The Basics

The UI in *Canberra* is crafted with precision, made up of distinct elements designed to support the player effectively throughout the gameplay.



The core UI element is the **health bar**. Positioned bottom-left on the screen, it's a prominent red bar signaling the player's vitality. If this bar hits zero, the player dies and must restart the level—no second chances.

Interestingly, the health bar isn't always visible. It smartly **disappears when the player neither takes damage nor heals**, maintaining a clean and minimalist UI. This automatic vanishing reduces visual clutter, aligning perfectly with the game's realistic design ethos. Only active, relevant elements show up; the rest stay out of sight.

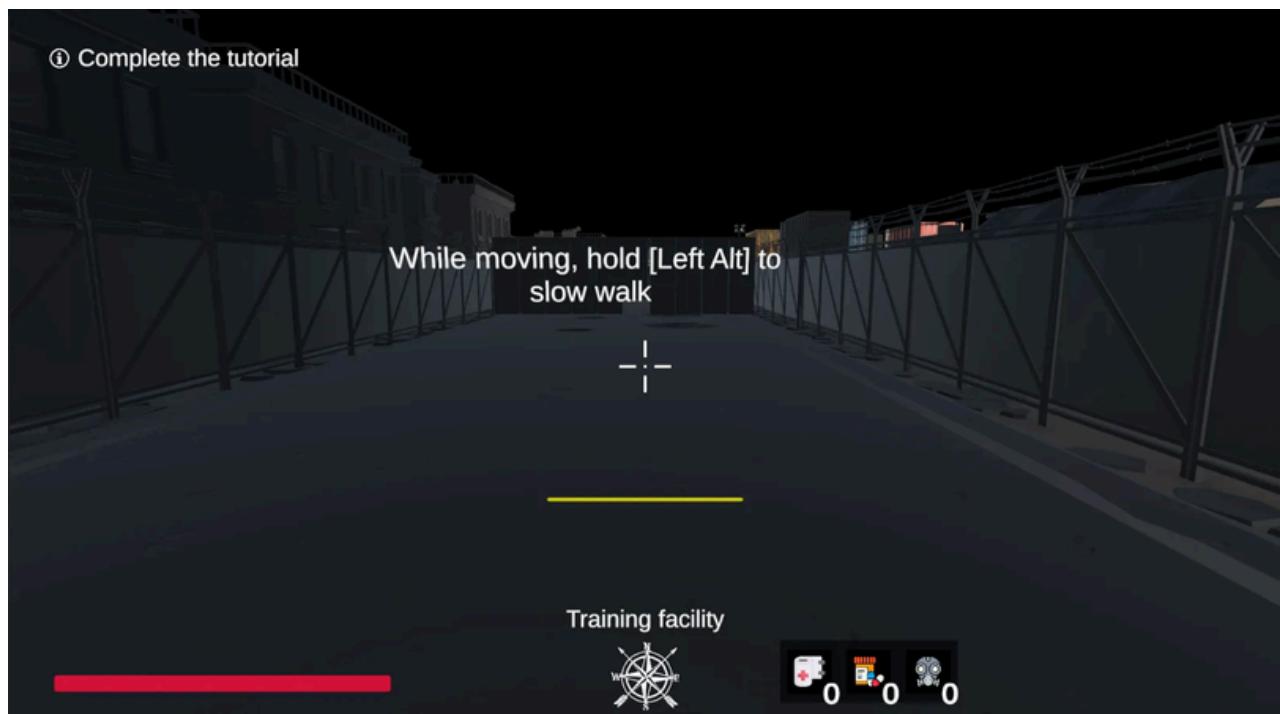
At the top-left, **mission information** is displayed. This continuously informs the player of the current objective until it's completed. Keeping this front and center ensures players never lose track of what they're working towards—an essential element in maintaining focus and immersion.

At the bottom center of the screen, **textual feedback** appears indicating the player's current location. This is crucial in dark and confusing environments where the limited range of the flashlight makes orientation difficult. Thanks to this location indicator, players can better assess their position and make informed decisions about their next steps.

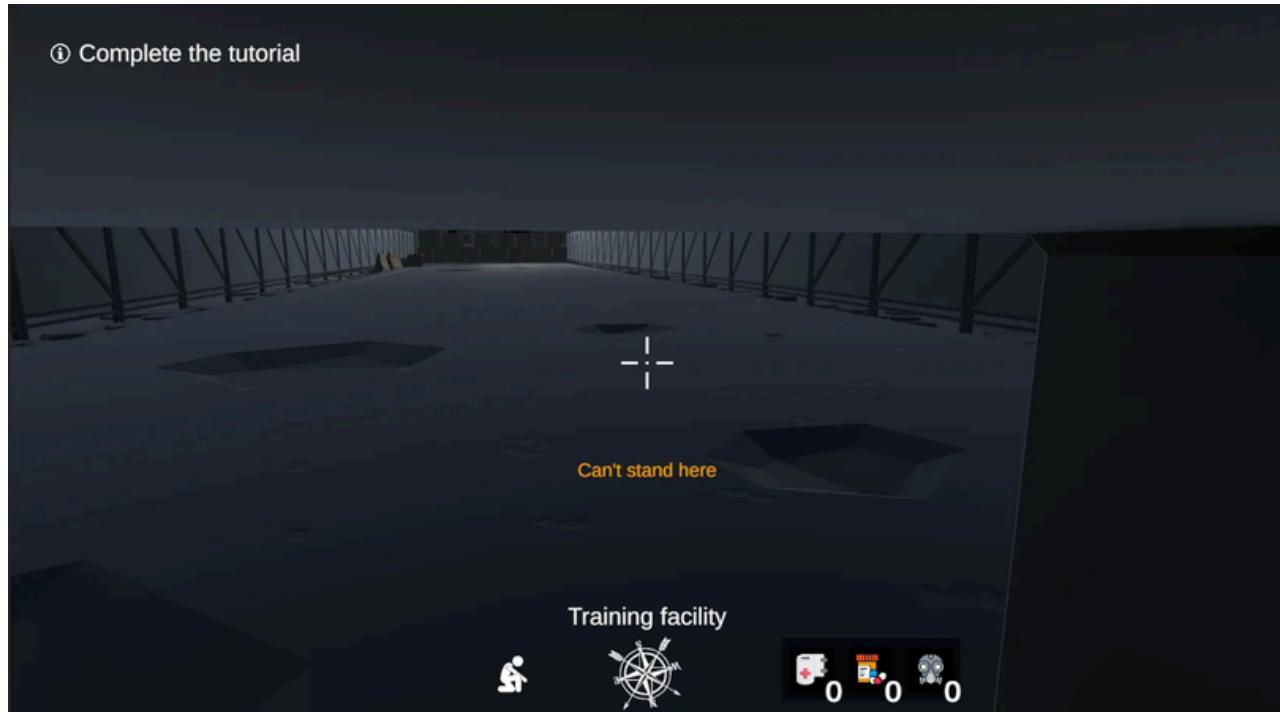
Directly below this location display is a **compass bar**. This navigation aid helps players orient themselves within the game world and assists them in retracing their steps, which is especially important in large or labyrinthine levels.

In the bottom right corner of the screen is the **inventory section**. Here, up to three usable items that the player can carry during the mission are shown. Each item has its own “slot” in the interface and can be quickly activated via corresponding hotkeys. This ensures swift access without interrupting gameplay.

While running, a **yellow stamina bar** appears, indicating the player's remaining stamina. When this bar is depleted, the player temporarily cannot sprint until the stamina recovers. A clever nuance is that stamina **recharges more slowly during movement and faster when standing still**. This design discourages excessive sprinting and increases tension during dangerous situations, as players must carefully weigh when to sprint and when to conserve energy.



4.1.2 Crouch & Safety Checks



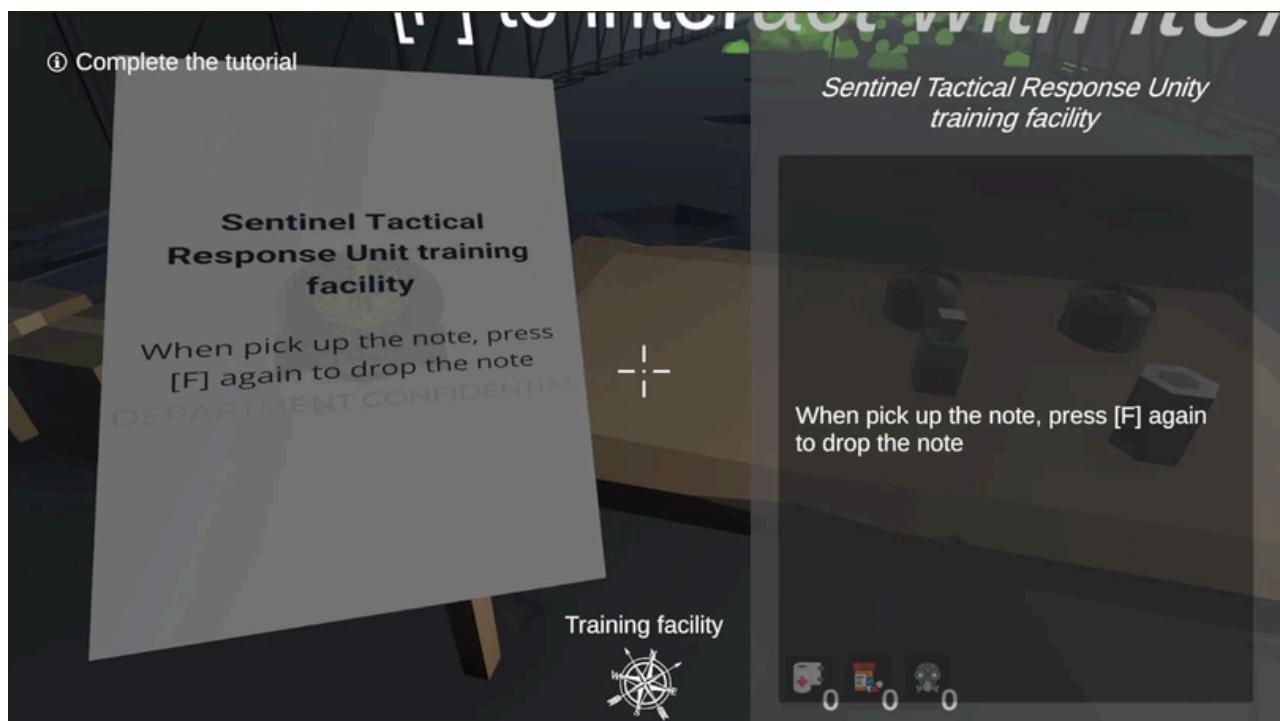
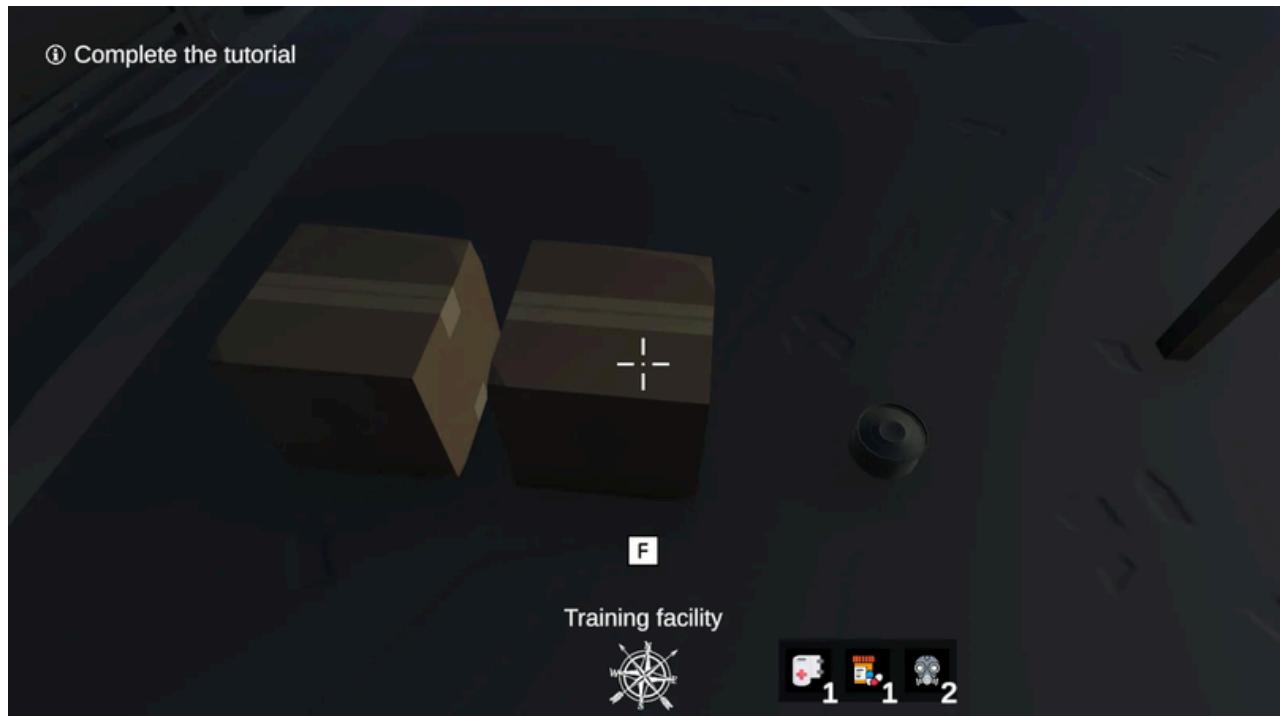
When the player crouches, an icon appears on the screen indicating that the character is in a crouched position. This feedback is vital because certain parts of the game can only be navigated while crouching. Since much of the game takes place in poorly lit environments, it is crucial that the player always knows whether they are standing or crouching. Because the player controls a character via a keyboard rather than directly seeing their own body, this visual confirmation is essential for orientation and control.

To prevent the player from accidentally hitting their head on an object—something that could cause **physical glitches** such as clipping or even falling out of the game world—a **safety mechanism** is implemented. This system checks if there is an obstacle above the player's head before allowing them to stand up. If something is blocking the way, standing up is prevented.

This is considered a critical **error prevention** measure, designed to avoid frustrating bugs and maintain a consistent and professional gameplay experience. A player falling out of the map not only loses progress but can also create a negative impression of the game's technical quality.

4.1.3 Interaction with Objects and Storytelling through Notes

When a player approaches an interactive object, an **icon featuring the letter 'F'** automatically appears on the screen. This serves as a clear visual cue indicating that the object can be picked up or used. This UI prompt helps players intuitively identify which elements in the environment are interactive, without breaking immersion.

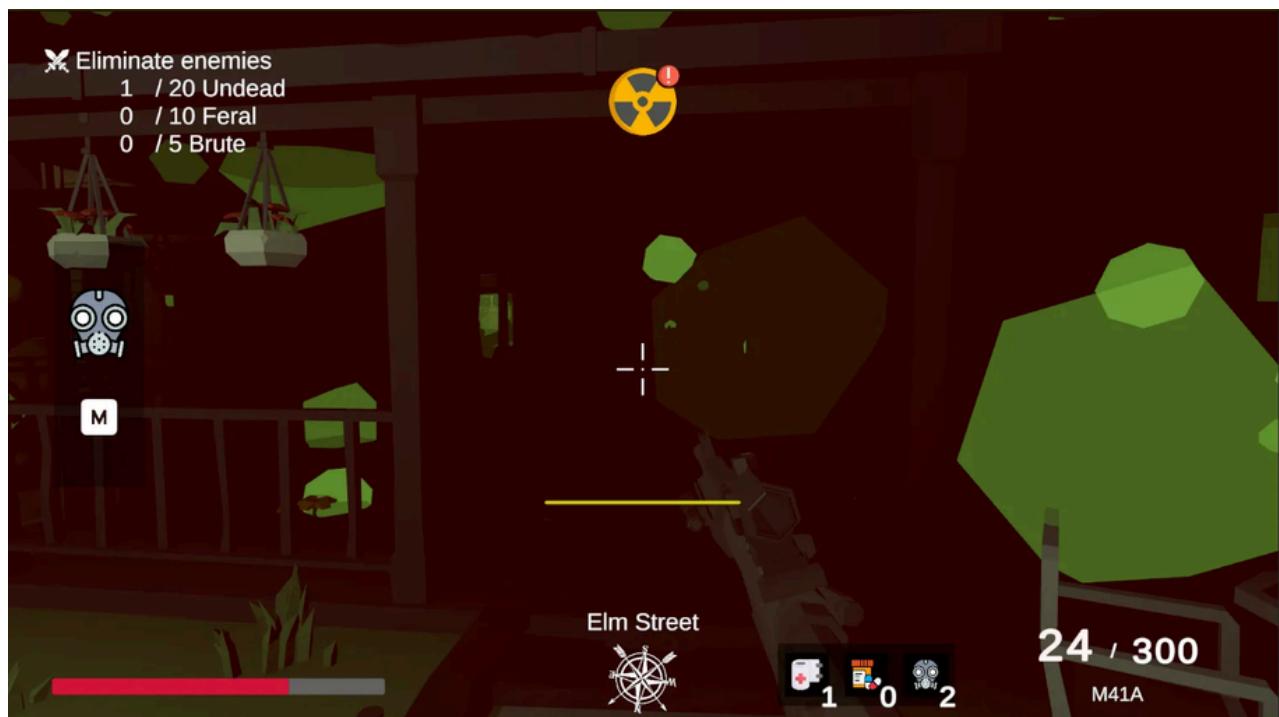


Besides the use of consumable items as mentioned in previous chapter, *Canberra* also incorporates a narrative layer delivered through **notes**. Since the game **contains little to no voice acting**, the world's backstory is conveyed via flyers, newspaper clippings, and other written documents scattered throughout the levels. These notes provide information about the player's location, contextual details of the situation, or even satirical messages reminiscent of tabloid newspapers. This mechanic is not new – games like *Indiana Jones and the Great Circle* use it effectively – but it fits perfectly with the tone of *Canberra*. Players can choose to pick up these notes or ignore them, but those who do are rewarded with a deeper understanding of the setting and the overarching story.

4.1.4 Radioactive Zones and Protective Mechanisms

In *Canberra*, there are zones heavily contaminated with radioactive material. When a player enters such an area without proper protective gear, they are gradually exposed to lethal radiation.

Once a player steps into a contaminated zone, two UI elements appear on the screen. At the top center, a **radiation icon** displays as an immediate warning. Additionally, in the upper left corner, a **mask icon** appears with the letter '**M**' underneath – reminding the player to put on their gas mask. The mask icon disappears once the mask is worn, while the radiation icon only vanishes when the player completely leaves the contaminated area.



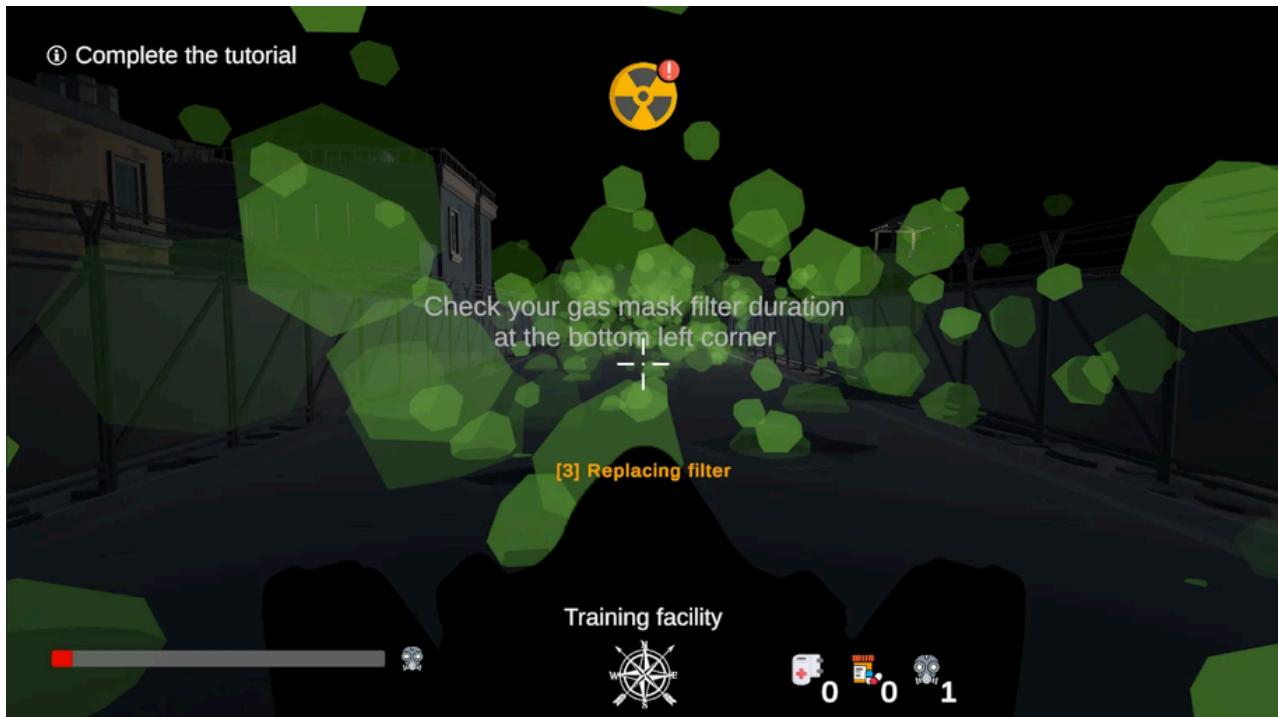
When the player presses ‘M’, they **put on a gas mask**. Just like in real life, wearing a gas mask is not comfortable. The wearer experiences a constant **lack of oxygen, a claustrophobic feeling, and a limited field of vision**. In *Canberra*, it’s no different: as soon as the gas mask is put on, the screen **darkens slightly** and an **image of the mask appears at the bottom of the screen**, partially blocking the player’s view and creating a sense of discomfort.



4.1.4.1 Filter System

To prevent players from using the mask indefinitely, a **filter mechanism** is integrated. Each gas mask uses a **temporary filter**, and as it becomes clogged, its **effectiveness decreases**. The player sees this status represented by a **green bar** above the health bar, accompanied by a mask icon for quick recognition. Over time, the bar’s color **shifts from green to red** and eventually disappears when the filter is completely dirty.

A clever addition is that the **filter status is saved** when the mask is taken off. This means a partially used **filter is not lost** upon removing the mask – a crucial feature in a game where resources are scarce and every detail counts.



4.1.4.2 Warnings and Failsafes

Because players aren't always fully attentive to the UI while exploring, a clear warning is displayed when the filter is nearly depleted: a text message appears just below the crosshair, informing the player that it's **time to replace the filter**. This warning is positioned centrally on the screen so the player can immediately take action.

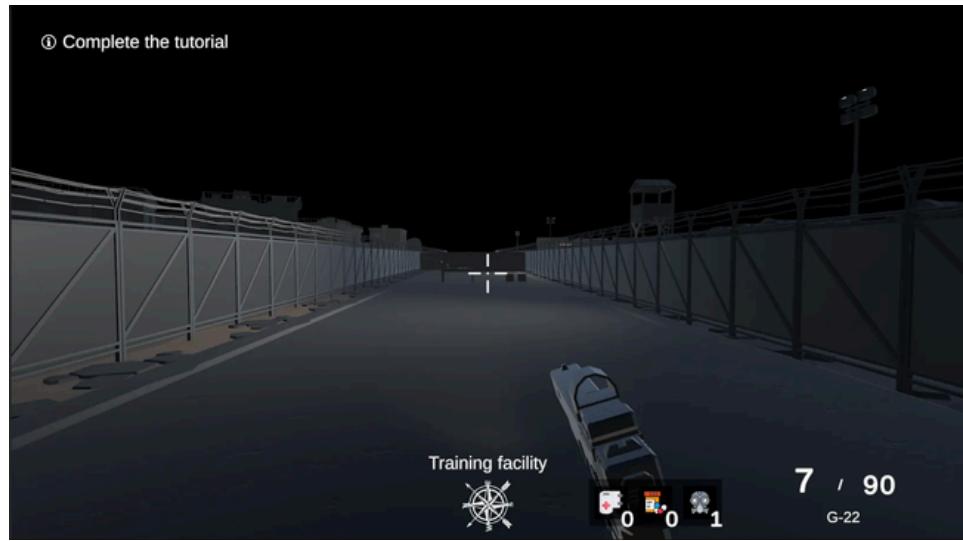
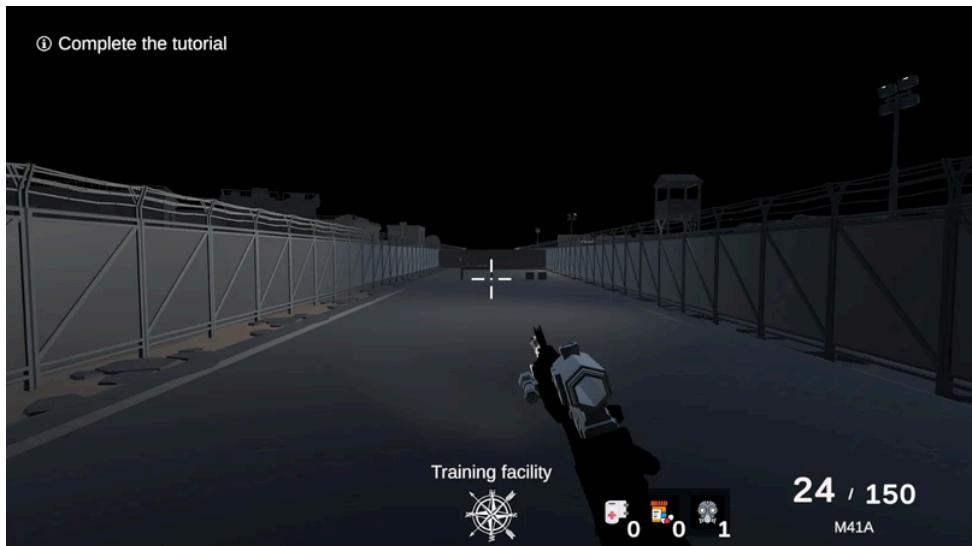
Replacing the filter is **entirely manual** and **requires time** – the system does not do this automatically. This forces the player to think ahead and make strategic decisions during dangerous exploration missions.



4.1.5 Firearms and Shooting Mechanics

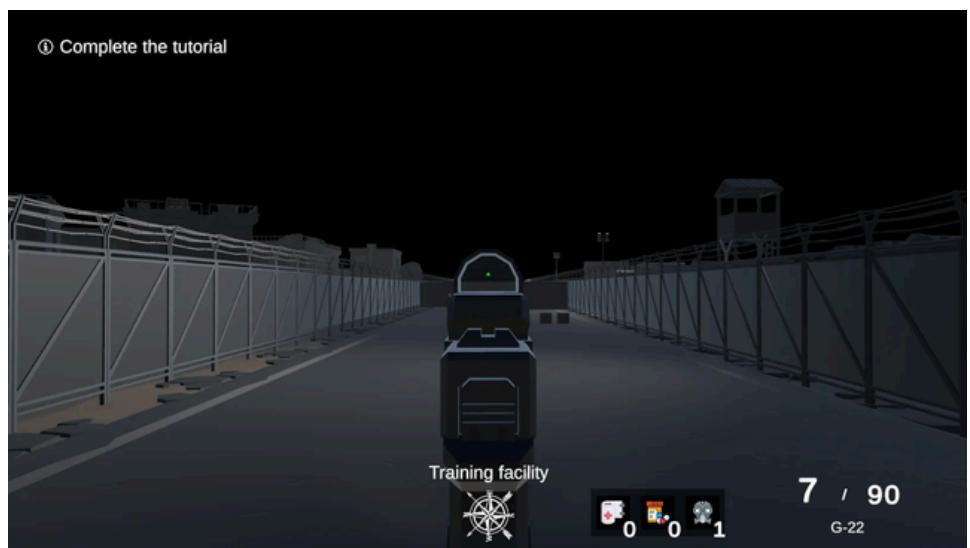
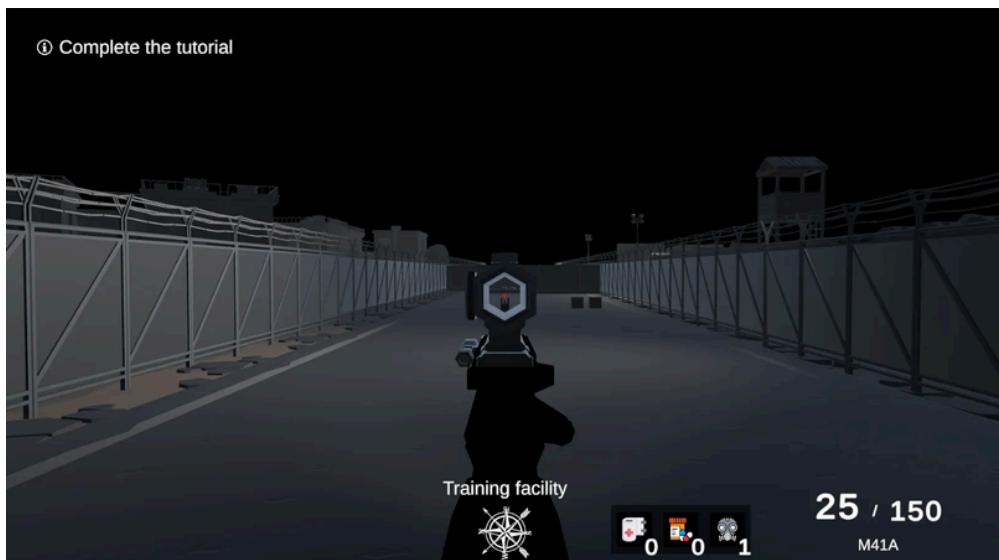
Since *Canberra* is a **First Person Shooter (FPS)**, the weapon system plays a central role in the gameplay experience. Shooting is one of the core mechanics of the game.

The player has access to **two primary weapons**: the **M41A** and the **G-22**. Both weapons are designed with a minimalist approach to their user interface. Only the essential elements are displayed to minimize visual distraction and ensure a clean, clear screen.



At the bottom right of the screen, the UI shows the current number of bullets in the magazine along with the weapon's name. In the center of the screen is a **crosshair** – a fixed element that assists with orientation in the world and aiming during combat. While players are encouraged to aim through the sights for more precise shots, the crosshair provides a constant visual reference for orientation.

When the player **aims down sights** (ADS), the standard crosshair **temporarily disappears** and is replaced by the reticle of the equipped sight—such as a holo or reflex sight. Once the player stops aiming, the standard crosshair automatically returns to the screen.



During weapon use, the **number of bullets** is continuously displayed in the **bottom right corner of the screen**. The UI always **accurately shows** how many rounds remain in the magazine—whether the player is firing in a torrent or carefully counting each bullet.

An additional helpful feature is that the bullet count changes color from **white to red** depending on the remaining ammunition. **White** indicates a full magazine; **red** signals that the magazine is nearly or completely **empty**. When only three bullets remain, a notification appears below the crosshair to remind the player to reload.

The **reload mechanic** in *Canberra* is relatively **deliberate** compared to other games, where speed usually takes priority. Reloading takes approximately **2 seconds** if the **magazine isn't empty**, and about **3 seconds** if it is **completely empty**. This means players must not only shoot **accurately** but also think **strategically** about when and where to reload—ideally in a safe zone, to avoid being caught off guard from a blind spot.



4.1.6 Damage Indicator and Blood Screen

When the player takes damage, two UI elements signal that health has been lost. The most prominent is the screen **turning red as soon as damage is received**. If the player sustains light damage—for example, when around **80%** health remains—the red effect **gradually fades away**. However, if the damage is severe, especially when health drops **below 40%**, the screen **intensifies to a deep red**. This effect is designed to create a sense of discomfort and serves as a visual warning that the player is seriously injured. The only way to fully clear the blood screen is by using a healing item. Thus, the blood screen plays a crucial role in **signaling health loss and urging the player to heal as quickly as possible**.



In addition to the red screen, there is another important UI component: the **damage indicator**. This mechanism is common in many *FPS* games and is vital for situational awareness. The indicator appears as a **semicircle with a triangle pointing toward the player**, showing the **direction from which the attack originates**.

Since the first level takes place in a dark environment with limited lighting, it's easy for players to lose orientation. Adding a visual indicator that can rotate 360 degrees around the screen is therefore a highly valuable feature. It enables players to react quickly, reposition effectively, and significantly improves survival chances, even in low-visibility conditions.

4.1.7 Minimap

Another important element within the user interface is the **minimap**. This minimap is only available during the main levels of the game. In other game modes, such as the tutorial or the shooting range, the minimap is completely absent. This is because those environments are linear and limited—for example, the tutorial has only one path to follow, and the shooting range's movement area is so small that a map would be unnecessary.

In the main missions, however, the situation is different. Players are dropped into an **unfamiliar area** without prior knowledge of the environment. To prevent players from getting lost, a minimap has been added. It is located in the **upper right corner of the screen** and displays a cross-shaped icon at the center to indicate the player's position and facing direction. Additionally, the **street name** of the current location is shown—a useful feature for orientation.

The minimap also has its limitations. It **cannot display information about building interiors** or **covered spaces**. To avoid the map becoming a redundant or obstructive element in these situations, an automatic **safety system is implemented**: as soon as a player is about to enter a building or covered area, the minimap **automatically deactivates**. This frees up screen space and increases the player's field of view—something crucial in tight, dangerous environments.



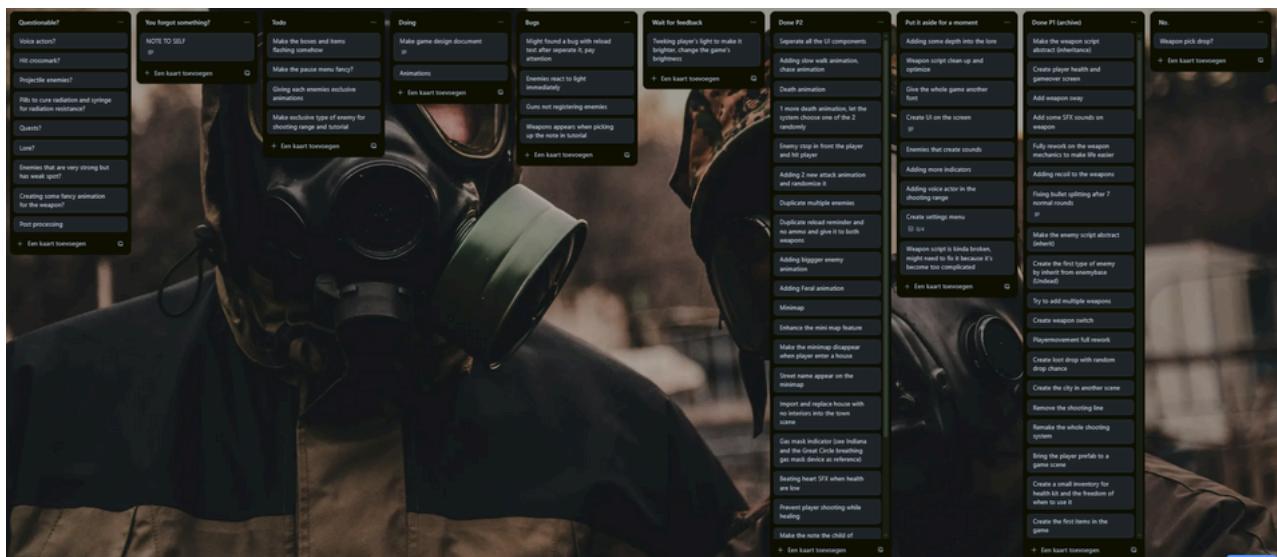
Chapter FIVE



A LOOK BACK, IN 4
PAGES

The original plan for the development of *Canberra* was about **three months**. Initially, this seemed like a realistic estimate based on my knowledge, skills, and the project scope as it was envisioned at the time. But, as is often the case in creative processes, reality proved to be far more dynamic. Due to a variety of objective circumstances—including school obligations, technical obstacles, and especially because I kept challenging myself with new and interesting ideas—the development period ultimately extended significantly.

However, I don't see this as a setback. On the contrary: it's proof that this project kept growing, just like my ambition and skills as a developer. What started as a conceptual experiment has grown into my first full-fledged game project, one I worked on with dedication until it truly began to feel like a 'real game.' The fact that I continuously refined, rewrote, and improved it may have caused it to take longer than planned—but it also made the game richer, more consistent, and more engaging.

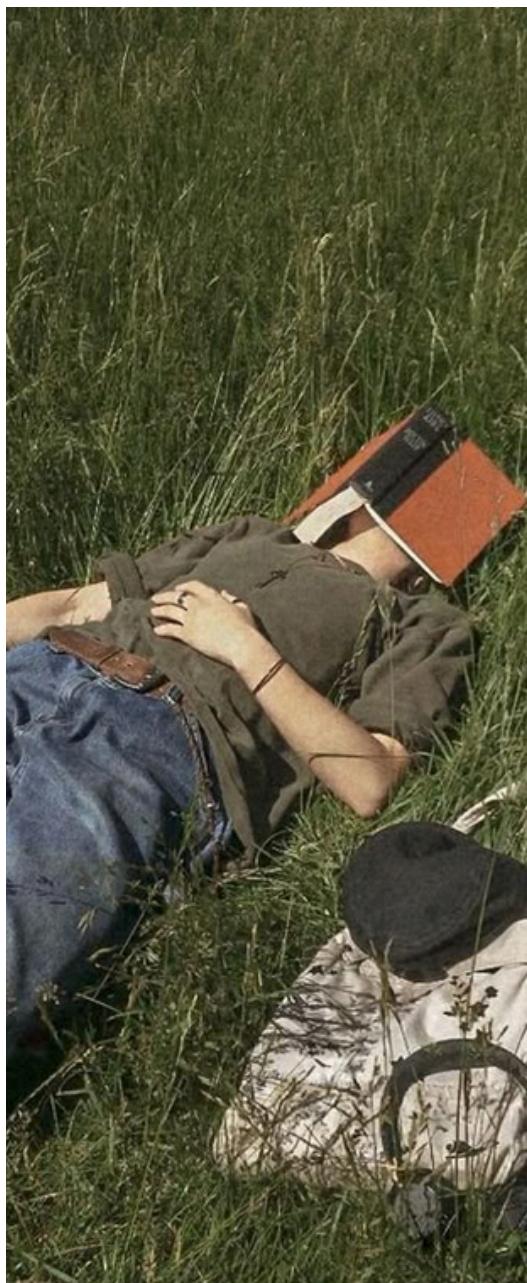


Throughout this journey, I not only invested a lot of time and energy into building systems, testing gameplay, and designing levels, but I also gained valuable insights into planning, prioritization, and what it truly means to 'finish' a game. These are lessons I will definitely carry forward into future projects.

This project was developed entirely by me, from start to finish. The image above shows my *scrum board* used during the development period. The details on the cards might not be clearly readable, but that doesn't really matter. What matters is that it reflects the scale, complexity, and perseverance behind *Canberra*.

Only while writing this *Game Design Document* did I truly realize how much this project has grown. What began as a handful of loose, sometimes even bizarre ideas, has evolved into a game world with its own rules, systems, and atmosphere. And although it may not be perfect yet, it feels like a milestone—not just technically, but personally as well.

THE POSITIVE ASPECT



The most positive aspect of the Canberra project was undoubtedly the **motivation**. Immediately after finishing my internship, I began developing ideas for this game. Building on the lessons I had learned during that internship, I decided to push my boundaries and tackle bold concepts and development techniques that I had previously avoided. Although my motivation wasn't always consistent, I stubbornly stuck to completing small goals as a way to strengthen my discipline. With the motto "**a little every day**," I gradually built larger systems on a solid foundation.

Another key point was how I drew **inspiration from existing sources and transformed it into something unique**. Most of the projects I've developed so far have been influenced, to varying degrees, by existing games – and *Canberra* is no exception. Players will undoubtedly notice parallels with popular titles like *Rainbow Six: Siege*. But this is not about copying; it's about carefully selecting, adapting, and combining existing elements to create a unique experience.

Of course, this process was not flawless. Certain mechanisms I tried to replicate didn't work as intended, leading to bugs. Instead of simply removing them, I learned to **think creatively**: I tried to fix these issues or even turn them into unique features of the game.

THE NEGATIVE ASPECT

Of course, alongside all the positive aspects, I also made several mistakes and learned valuable lessons during the three months of development.

One of the biggest pitfalls was that the project became **too ambitious**. What was originally intended as a modest shooter quickly grew into an ever-expanding whole. I kept adding new ideas while many existing components weren't even finished yet. The result? The project outgrew my control, leading to a scope and complexity that I could hardly keep track of myself. This caused various bugs to linger in the game – some well known, others difficult to trace or fix within the available time.

Additionally, I regularly made things unnecessarily difficult for myself when writing the instructions that determine how parts of the game should behave. **I linked too many of these instructions directly to each other**, causing everything to become dependent on everything else. Every time I built a new scene, I had to manually configure dozens of elements. The structure behind the scenes quickly turned into a tangled mess: changing one part often led to errors in several other places. Instead of spending time developing new gameplay elements, I was mainly occupied with solving technical problems I had caused myself through that complex setup.

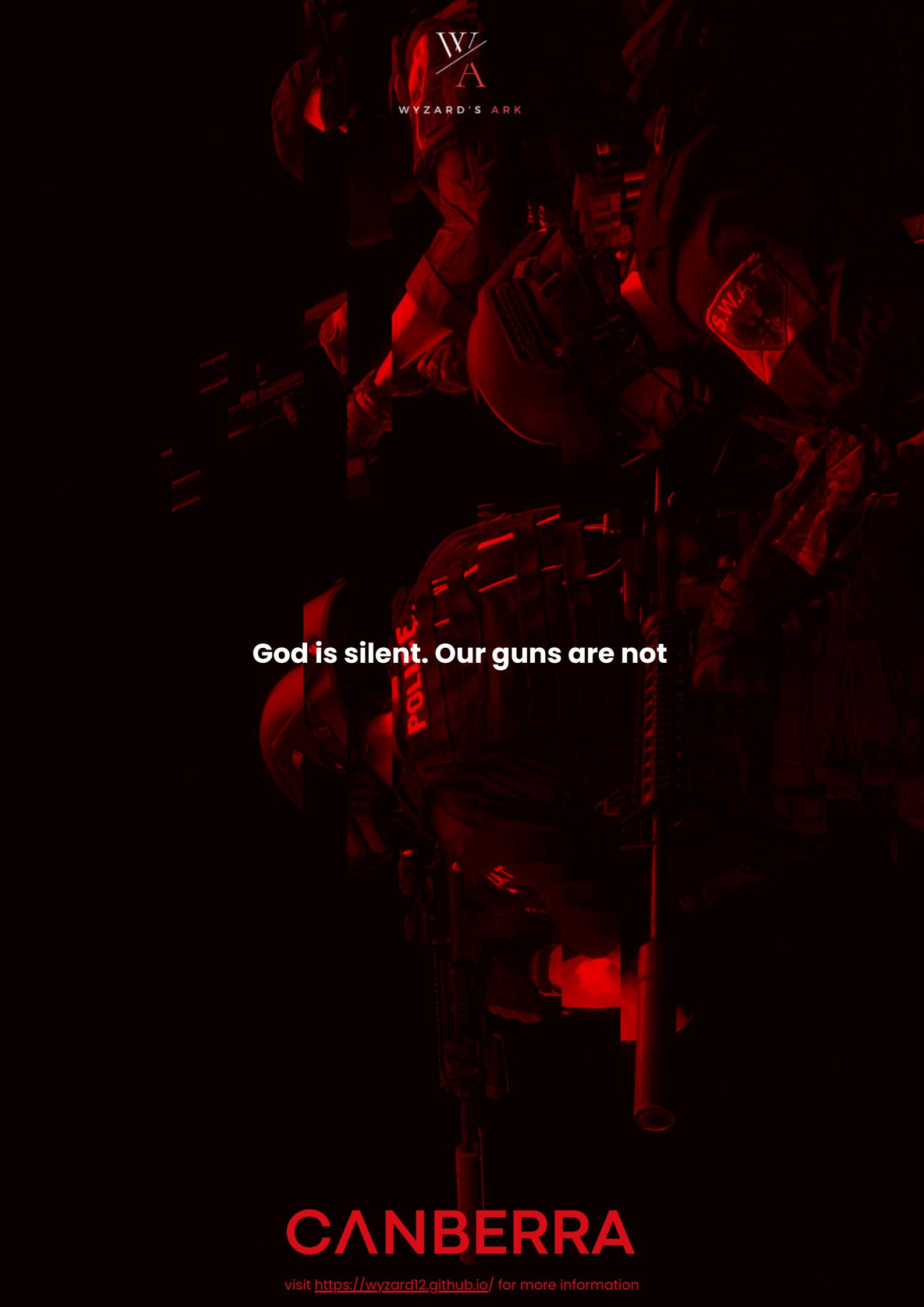


Although my work habits have improved significantly since my first internship, I no longer jump straight into working but instead take the time to put ideas on paper – gathering both big and small concepts, refining them, and making them my own. Where I once worked mostly intuitively, picking up things I enjoyed and proceeding without clear structure, I now use a scrum board, work methodically, and carefully track my progress to avoid losing direction.

These behavioral changes are certainly positive, but I also recognize they still need further refinement. As the project progressed, I noticed old habits – like working chaotically or randomly picking up tasks – resurfacing. This led to a noticeable drop in my productivity. I regard this as a valuable lesson and am determined to avoid these pitfalls in future projects and further professionalize my working methods.

This concludes the **Game Design Document** for the *Canberra* project.

Thank you very much for your time and interest in my project.



W
A

WYZARD'S ARK

God is silent. Our guns are not

CANBERRA

visit <https://wyzard12.github.io/> for more information