Virtual Environments Final Project
- HELLRUSH -

Leonardo Salpini
leonardo25@ru.is
leonardo.salpini@studenti.unicam.it
Reykjavik University
Reykjavik, Iceland,
University of Camerino
Camerino, Marche, Italy

Figure 1: HELLRUSH - In game screenshot

Abstract

This project is a fast-paced, Doom-inspired first-person shooter where players battle demons. Its gameplay is inspired by DOOM I/II, while the soundtrack draws influence from DOOM Eternal. The game includes a brief narrative to provide context for its world and environments. To enhance immersion, it features a comprehensive sound system covering weapons, footsteps, in-game objects, music, and camera effects such as bobbing and tilting. A key feature is the integration of 2D enemies within 3D environments and weapons, blending retro visuals with modern gameplay. Additionally, a shader graph applies dithering and color quantization, reinforcing the retro aesthetic and unifying the visual elements.

ACM Reference Format:
Leonardo Salpini. 2018. Virtual Environments Final Project - HELLRUSH -. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 4 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction & Motivation

The goal of this project was to develop a fast-paced, first-person shooter inspired by the great classics of the late 1990s, such as

Quake, DOOM, and Wolfenstein 3D. These titles are considered the foundation of modern FPS games, thanks to their relatively simple yet highly engaging gameplay loops. My objective was to capture the essence of those early shooters, while incorporating modern design techniques and technical implementations to give the game a refreshed, modern feel.

The motivation behind this project comes from my passion for FPS games and my curiosity about how a seemingly simple (nowadays we can say that, but for the period, they were not so simple) gameplay structure can still maintain such a high level of engagement and adrenaline, even today. I wanted to understand how these games were able to deliver such intense and fast-paced action through a combination of simple but powerful design choices, such as audio and visual cues, camera movement and shake, and other immersive systems that, when combined, significantly enhance the overall experience.

Rather than simply recreating or copying these classic titles, my goal was to study and internalize their core mechanics and reinterpret them through modern game development practices, using the Unity engine.

One of the design approaches I decided to explore was the mixing of 2D and 3D elements. While this technique is sometimes used in modern games, mostly in cutscenes, to blend 2D and 3D cinematics seamlessly, I wanted to apply it directly to gameplay. Specifically, I experimented with having 2D enemies within a 3D environment, aiming to find a visually coherent way to merge these two styles and create a distinctive style that bridges retro and modern visuals.

2 Related Work

When talking about fast-paced first-person shooters, which can be considered the pillars of the genre, several iconic titles can be identified.

Wolfenstein 3D (1992) is often regarded as the first true FPS. It introduced the core formula of navigating maze-like levels, collecting keys, finding secrets, and defeating enemies in real-time 3D environments. Its fast action and simple yet effective design laid the groundwork for all shooters that followed.

Quake (1996) took the genre further by introducing fully 3D environments, true 3D enemies, and real-time multiplayer, setting a new technical and gameplay standard. Its dark, gothic atmosphere and high-speed movement defined the style of arena shooters for years.

Finally, my main references were DOOM (1993) and DOOM II (1994). These games perfected the formula introduced by Wolfenstein 3D, featuring faster movement, more complex level design, and a wider variety of weapons and enemies. Their intense pacing, heavy metal soundtrack, and demonic setting created a unique experience that became the defining blueprint for fast-paced FPS games and a major source of inspiration for my project.

Newer entries in the series, especially DOOM Eternal (2020), also served as reference points. Its soundtrack, atmosphere, and environment are incredible, and I used them as inspiration for the audiovisual direction of this project.

3 Approach

To reach my goal to recreate a fast paced shooter with a retro style but more moder audiovisual system, the approach was divided in several key concepts. Weapon System, Camera Movement, Audio Design, Environment, Brief story, 2D + 3D integration and shaders. The weapon system is designed to be ideally fully modular and easily extendable, allowing new weapons to be introduced without modifying existing code or modifying a bit if necessary, for example at the moment the only firing type is full auto, to introduce like semi-auto or burt it needs to be adjusted. This is achieved through the use of a ScriptableObject called *WeaponDefinition*, which store all relevant data for each weapon, such as the prefab, audio effects, position offsets, fire rate, magazine size, and damage values and much more things. The *PlayerLoadoutController* script manages the player's equipped weapons and overall inventory. For each equipped weapon, it creates a runtime instance of its *WeaponDefinition*, enabling parameter adjustments during gameplay without altering the original asset. It also handles weapon prefab instantiation, weapon addition to inventory, and switching. The *Weapon* script, on the other hand, manages the behavior of the currently active weapon, handling firing logic, reloading, audio playback, and interactions with other game systems. When the player fires, the system uses a raycast to detect hits; if the raycast collides with an object implementing the *IDamageable* interface, it applies the weapon's specific damage to that target. When the player switches the active weapon, the *PlayerLoadoutController* assigns the corresponding runtime *WeaponDefinition* to the *Weapon* script, allowing a single script to handle all weapons instead of requiring a separate script for each one. A really simplified graphical representation can be seen in the figure below.

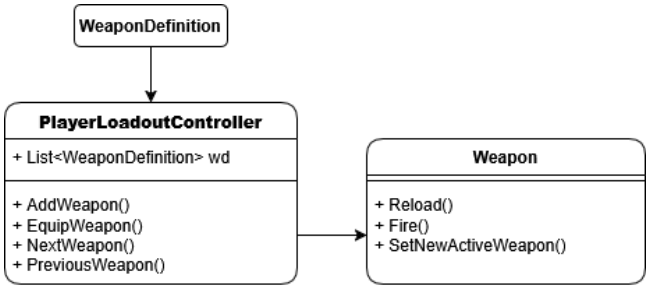

Figure 2: Weapon System

Another aspect is camera movement and visual effects, which play a key role in player immersion. Classic DOOM games demonstrated how camera motion can enhance gameplay, so I aimed to recreate that effect with head bobbing. I achieved this by moving the camera slightly up and down while the player is moving. Additionally, I implemented a camera tilt when moving sideways to reinforce the sense of motion and realism further. Since the game includes a running mechanic, I wanted to provide stronger visual feedback to emphasize the feeling of speed and fast-paced action. To accomplish this, I added a field of view (FOV) change while running to help show acceleration and intensity. Although this feature is conceptually simple, it was somewhat challenging: I had to adjust the main camera's FOV without affecting the weapon models in

view. To solve this problem, I introduced a second overlay camera dedicated to rendering only the weapons. This allowed the main camera's FOV to change dynamically while keeping the weapon view stable.

When it comes to audio design, I put a great deal of effort into it, as I believe a strong audio system can completely transform the player's perception of the game. My goal was to make the world feel more immersive by adding sound to every element that felt important, going from the ambient background noise of lava and footsteps to weapon sounds, such as firing, reloading and switching. I also included distinct sound effects for portals, altars, and item pickups, ensuring that every interaction had a satisfying auditory response. Of course, no demon-slaying experience would be complete without an intense soundtrack. For this reason, I incorporated music directly taken from DOOM Eternal, which perfectly captures the fast-paced, adrenaline-filled atmosphere and keeps players fully engaged in the action.

As mentioned earlier, the environment design is heavily inspired by DOOM Eternal. Its characteristic reddish-orange color palette perfectly captures the feeling of being in an infernal, hellish world, and I tried to capture that same atmosphere. To achieve this, I spent time selecting a suitable environment asset pack that matched this visual direction. The level layout consists of three small islands, separated by a vast sea of magma. Although the playable area is relatively compact, this large, empty space helps create the illusion of a much larger world. I surrounded the entire map with rock formations to clearly define its boundaries and reinforce the impression that the player is exploring a specific sector within a much greater "Hell". To enhance depth, I added a fog effect, which desaturates distant objects and makes them appear farther away. As a final touch, I adjusted the directional light's color and temperature to a warmer tone and increased its emission intensity, emphasizing the dominant red and orange colors typical of an infernal environment.

I added a brief story through a few dialogues in the game to guide the player and provide better context about the world and its environment. I also included an image on the left to show who is speaking, giving the impression that the player is a special soldier sent to defeat the Demon King. The game features two possible endings: a good one, where the player kills the Demon King and saves the world, and a bad one, where the player dies and the demon army destroys everything.

One approach I wanted to explore was mixing 2D and 3D elements to recreate the visual style of classic, retro games, while combining it with modern, 3D environments. I focused primarily on making the enemies 2D, using sprite-based characters, implementing them was relatively easy, as Unity's Animator Controller works well with 2D sprites, allowing me to easily manage animations such as idle, movement, and attack.

The main challenge, however, was determining how a 2D enemy could interact with and damage a 3D player in a physically consistent way. To solve this, I designed hybrid attack systems that use 3D elements as the damage source. For example, enemies can launch 3D projectiles, such as fireballs, or trigger 3D area-of-effect attacks that interact naturally with the 3D environment and the player's collision system. In this way I was able to mantain the retro inspired 2D visuals while maintaining coherent gameplay mechanics within the 3D space.

Lastly, I wanted to add a retro visual effect to the game by applying a shader to the final image. This also helped blend the 2D and 3D elements more naturally. To achieve this, I first created a Render Texture and set it as the output target of the main camera. This means everything the camera captures is rendered onto this texture instead of directly to the screen. The render texture uses a lower resolution than the main camera to emulate the pixelated look of older games and enhance the retro vibe.

Next, I added a Raw Image Component to a Canvas and set its rendering order to appear behind all other UI elements. I then expanded the Raw Image to cover the entire screen and assigned the Render Texture to it, allowing the gameplay view to be displayed through this UI element instead of the camera's default output.

After that, I created a Shader Graph to apply the visual effects. The shader includes two main operations: a dithering effect, which simulates the pixel patterns used in retro rendering, and color quantization, which reduces the number of visible colors to create an old-school look. I then created a Material using this shader and applied it to the Raw Image to complete the effect.

One issue I encountered was that the shader was also being applied to the UI elements, which I didn't want. To solve this, I created a child camera of the main camera dedicated solely to rendering the UI. This way, the post-processing retro effect only affects the main game view, while the UI remains crisp and unaffected.

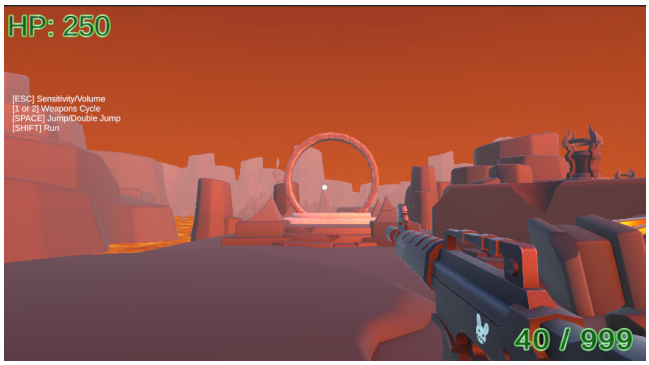

Figure 3: Game without shader

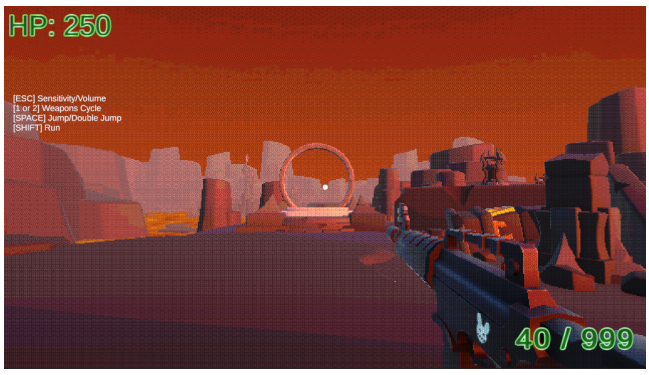

Figure 4: Game with shader

4 Results

In conclusion, I'm quite satisfied with the overall results of the project. The environment turned out well; even though the three islands are identical, the atmosphere still feels immersive and visually appealing. I'm also happy with the sound design; while there are a few elements that could be expanded or refined, the overall audio experience feels engaging, especially during fighting moments, thanks to the really amazing DOOM Eternal soundtracks. The shooting mechanics also feel enjoyable and responsive, but of course, there's a lot of room for improvements to make the combat feel even more rewarding. The one aspect I'm still unsure about is the enemies. While I think the combination of 2D and 3D works better than expected, I'm not entirely convinced by the enemy attack behaviors, since they don't feel like a real threat, especially during the final boss fight, where it is simply standing still, shooting fireballs, without giving any sense of danger and tension. Overall, I can say that I believe the foundation is solid and provides a great starting point for further refinement.

5 Future Work

VThere are many features I would have liked to include, but due to limited time, I wasn't able to implement them. Starting with the weapon system, one planned addition was temporary or permanent power-ups that could enhance weapon attributes, such as damage, fire rate, or other stats. I had initially started implementing this functionality in the Weapon Script, but ultimately decided to focus on higher-priority features. Another improvement I wanted to make was introducing a bullet spread mechanic so that shots would slightly deviate from the crosshair instead of always traveling in a perfectly straight line, making the shooting experience feel more natural and satisfying.

I also intended to add more weapons, including projectile-based ones. While the raycast approach works well for fast weapons, like the assault rifle, it's less suitable for slower ones, such as a grenade launcher. At the moment, the grenade launcher still uses a raycast that instantly triggers an explosion at the hit point, but using a visible projectile with travel time and physics would provide a much more authentic and engaging experience.

Regarding the environment, the current layout features three islands that are mostly identical. In the future, I would like to make each island more distinct to add visual variety and create a stronger impression on the player.

Another potential improvement would be to introduce additional enemy types with unique attack patterns, especially for bosses, to make encounters more dynamic and challenging. Finally, several smaller enhancements could improve the overall polish of the game—such as adding sound effects for enemies while running or attacking, player feedback when taking damage, and a more refined UI for a smoother and more immersive gameplay experience.

Received 09 November 2025