

POLITECNICO DI TORINO

Laurea Magistrale
in Ingegneria Informatica

Tesi di Laurea Magistrale

**Piattaforma in realtà virtuale
per l'addestramento di operatori
in procedure CBRN**



Relatore

Prof. Fabrizio Lamberti

Candidato

Federico De Lorenzis

Anno accademico 2018/2019

Sommario

La *Realtà Virtuale* o *Virtual Reality* (VR) può essere impiegata per la creazione di strumenti utili all'addestramento di personale specializzato, superando le problematiche che sono da sempre legate alle esercitazioni nel mondo reale. Simulatori virtuali, infatti, forniscono esperienze ripetibili, a basso costo, condivisibili e che non richiedono l'impiego di mezzi specifici o ampi spazi.

Questo elaborato ha l'obiettivo di presentare uno strumento in VR pensato per l'addestramento CBRN: gli utenti sono immersi nel mondo virtuale nel ruolo di operatori CBRN (ovvero operatori addestrati alla gestione dei rischi Chimico, Biologico, Radiologico e Nucleare), in base a come affrontano un determinato scenario (nel caso specifico, l'attacco di un drone) sono valutati. La ripetizione dell'esperienza permette loro di correggere i propri errori e prendere confidenza con i movimenti e gli automatismi propri delle procedure considerate.

Nell'ultima parte dell'elaborato sono riportati un'analisi dei dati raccolti in occasione della Toxic Trip 2019 (evento in cui il simulatore è stato sperimentato da militari ed esperti del settore) ed una discussione in merito a quali potrebbero essere i possibili sviluppi futuri dello strumento considerato.

Ringraziamenti

Ringrazio in primis il mio relatore, il prof. Fabrizio Lamberti, per avermi dato fiducia all'inizio di questa tesi e per il supporto e la guida che mi ha fornito durante tutto lo sviluppo di questo progetto. Ringrazio anche Davide Calandra per i preziosi suggerimenti che mi hanno permesso di portare a termine questo lavoro.

Ringrazio quindi Stefano Cimaroli e Davide Preziusi del Terzo stormo dell'Aeronautica, che hanno condiviso con me questo viaggio, aiutandomi nello sviluppo e dandomi preziose indicazioni. Un Grazie in particolare per l'incoraggiamento fornitomi da Stefano e per l'accoglienza calorosa offertami da Davide, in occasione della trasferta a Villafranca. È stato un piacere essere colleghi, lo sarà ancora di più continuare a essere amici.

Grazie anche a Massimo Migliorini della Fondazione LINKS, per il supporto morale, lavorativo e logistico durante questi mesi, e a Mario Bonu, con cui ho collaborato e che ha dovuto sopportare mesi di battute e discorsi, in contrasto con la consueta quiete presente nel laboratorio.

Un Grazie in particolare va ai miei genitori, che mi hanno permesso di arrivare dove sono e che non hanno mai smesso di credere in me, e ai miei amici che mi hanno sostenuto e sopportato durante tutti questi anni.

Indice

1 Introduzione

1.1 La collaborazione con l'Aeronautica

1.1.1 Il Terzo Stormo

1.1.2 La Fondazione LINKS

1.1.3 La Toxic Trip

1.2 Le procedure CBRN

1.2.1 Contesto

1.2.2 I rischi

1.2.3 Il Recce Team

1.3 Lo scopo della tesi

2 Stato dell'arte

2.1 La Realtà Virtuale

2.1.1 Immersione e presenza

2.1.2 Storia

2.2 Strumenti per la Realtà Virtuale

2.2.1 Display

2.2.2 Metodi di interazione

2.2.3 Locomozione

2.3 Realtà Virtuale e addestramento

2.3.1 Vantaggi

2.3.2 Svantaggi

3 Tecnologie

3.1 Game Engine

3.1.1 Vantaggi

- 3.1.2 Unity
- 3.2 Software di supporto
 - 3.2.1 Visual Studio
 - 3.2.2 Blender
 - 3.2.3 Optitrack
- 3.3 Hardware
 - 3.3.1 Schede grafiche e rendering
 - 3.3.2 HTC Vive
 - 3.3.3 Oculus Rift

4 Progettazione

- 4.1 L'applicazione precedente
- 4.2 Lo storyboard
 - 4.2.1 Realizzazione
 - 4.2.2 Il soggetto
 - 4.2.3 Modalità
 - 4.2.4 La simulazione
 - 4.2.5 I ruoli
 - 4.2.6 La valutazione
 - 4.2.7 Il menu iniziale
- 4.3 La validazione

5 Realizzazione

- 5.1 SteamVR e la Realtà Virtuale in Unity
 - 5.1.1 La libreria
 - 5.1.2 Le implicazioni
- 5.2 Unet e la gestione di utenti multipli in Unity
 - 5.2.1 Lo strumento
 - 5.2.2 Le implicazioni

5.3 Struttura finale

5.4 Dettaglio sulle implementazioni

- 5.4.1 Le interazioni con lo scenario
- 5.4.2 L'interfaccia utente
- 5.4.3 Tool aggiuntivi
- 5.4.4 La modalità multi-utente
- 5.4.5 Gli oggetti dell'inventario
- 5.4.6 Personaggi non giocabili
- 5.4.7 Il sistema di valutazione
- 5.4.8 La registrazione dei dati

6 L'esperienza alla Toxic Trip 2019

- 6.1 La preparazione
- 6.2 La logistica
- 6.3 L'evento

7 Analisi sperimentale

7.1 Il questionario e i risultati

- 7.1.1 Gli utenti
- 7.1.2 Motion sickness
- 7.1.3 Background
- 7.1.4 Input e output
- 7.1.5 Interfaccia
- 7.1.6 Immersione e presenza
- 7.1.7 Utilizzabilità
- 7.1.8 Il questionario multi-utente
- 7.1.9 Commenti

7.2 I dati registrati

8 Conclusioni e sviluppi futuri

8.1 Conclusioni

8.2 Sviluppi futuri

Elenco delle figure

- 1.1 Simboli per i rischi CBRN utilizzati nell'applicazione
- 2.1 Sensorama, schema. 1962, Morton Heiling
- 3.1 Interfaccia di Unity 5.6.7
- 3.2 Interfaccia di Blender
- 3.3 Base station dell'HTC Vive
- 3.4 HTC Vive, visore
- 3.5 Oculus Rift - visore, sensori e controller
- 4.1 Menu iniziale, schema concettuale
- 5.1 Modello 3D delle mani
- 5.2 Inventario
- 5.3 Clean Man e Data Compiler, NPC
- 5.4 Esempio di JSON con i dati raccolti
- 6.1 Un militare che prova la simulazione in una delle postazioni
- 7.1 Genere dei partecipanti
- 7.2 Nazionalità dei partecipanti
- 7.3 Background VR
- 7.4 Background Videogame
- 7.5 Background Serious Game
- 7.6 CBRN (utenti italiani)
- 7.7 CBRN Italiana (utenti italiani)
- 7.8 CBRN (utenti non italiani)
- 7.9 CBRN Italiana (utenti non italiani)

Capitolo 1

Introduzione

1.1 La collaborazione con l'Aeronautica

Questa tesi nasce nel contesto della collaborazione tra Politecnico di Torino, Fondazione LINKS e Terzo Stormo dell'Aeronautica Militare Italiana siglata a dicembre del 2018. Scopo di questo connubio è la creazione di scenari in VR con finalità di addestramento (*training*), che siano all'avanguardia dal punto di vista tecnologico e fedeli rispetto alle procedure rappresentate. Da un lato, il Politecnico di Torino e la Fondazione LINKS mettono sul piatto le competenze, l'attività di ricerca e l'attività di tutoraggio, mentre l'Aeronautica risponde con supporto logistico, umano e strumentale.

La piattaforma sviluppata, in particolare, affonda le sue radici in un'applicazione creata in precedenza, sempre in collaborazione con l'Aeronautica. Il vecchio programma, in pratica, è da considerarsi come una versione embrionale di quello che viene presentato in questo elaborato.

1.1.1 Il Terzo Stormo

Il Terzo Stormo fa parte del Comando logistico dell'Aeronautica Italiana e ha base nell'Aeroporto di Villafranca-Verona. Si occupa principalmente di offrire supporto logistico ai reparti del Comando che operano fuori dalle proprie sedi, inviando team per la creazione e la chiusura dei campi e fornendo attrezzature e equipaggiamenti in occasione di rischieramenti campali. È inoltre responsabile della gestione di attività legate a operazioni nazionali e internazionali, in particolare dell'acquisizione di beni da impiegare fuori sede.

1.1.2 La Fondazione LINKS

La Fondazione LINKS è un'istituzione senza scopo di lucro fondata dal Politecnico di Torino e dalla Compagnia di San Paolo, nata dall'unione dell'Istituto Superiore Mario Boella (ISMB) e dell'Istituto Superiore sui Sistemi Territoriali per l'Innovazione (SiTI). Svolge attività di ricerca ed è specializzata sia nelle scienze politecniche dell'informazione e delle telecomunicazioni, sia nelle scienze territoriali e ambientali.

1.1.3 La Toxic Trip

La Toxic Trip è un'esercitazione annuale organizzata dalla NATO e rivolta ai militari dell'Aeronautica. Si concentra in particolare sulle procedure CBRN e ha lo scopo di condividere tali procedure e varie tecnologie, in modo da mettere sullo stesso livello i paesi membri della NATO per raggiungere uno standard comune.

L'evento consiste in una settimana di esercitazioni che coprono tutto lo spettro delle procedure CBRN (Chimico, Biologico, Radiologico, Nucleare) [1] e a cui partecipano operatori esperti provenienti da nazioni diverse. All'evento intervengono anche aziende che forniscono attrezzature ai vari reparti dell'Aeronautica e che, per l'occasione, organizzano esposizioni per presentare e condividere nuovi prodotti.

In particolare, alla Toxic Trip 2019, tenuta nella base militare del Terzo Stormo a Villafranca di Verona, hanno partecipato membri dell'Aeronautica, membri dell'Esercito Italiano e militari provenienti da 12 paesi della NATO e da 3 paesi partner, per un totale di più di 800 operatori.

1.2 Le procedure CBRN

Il termine CBRN indica tutte quelle procedure di difesa e contenimento che devono essere seguite quando si ha a che fare con un'arma di distruzione di massa (sebbene questa definizione possa essere fuorviante, poiché in molti casi lo scopo di un attacco che utilizza materiale chimico, radioattivo, biologico o nucleare è il terrorismo, non la distruzione). La sigla ha le sue origini nell'acronimo ABC (Atomico, Biologico e Chimico), che prima si è evoluto in NBC (Nucleare, Biologico e Chimico) e, infine, in CBRN per racchiudere dentro sé anche gli attacchi di tipo radiologico (che, a differenza di quelli nucleari, non prevedono una detonazione).

1.2.1 Contesto

La necessità di procedure specifiche è diretta conseguenza dello sviluppo tecnologico e della nascita di nuovi tipi di armamenti (si pensi alle "bombe sporche", ovvero agli ordigni che detonano con lo scopo di spargere materiale radioattivo o chimico nell'area circostante). Non bisogna però pensare che la CBRN sia una disciplina legata unicamente all'ambito militare: esercitazioni di questo tipo vengono svolte anche dal personale degli aeroporti e servono a preparare gli operatori in vista di incidenti che non hanno nulla a che fare con conflitti o schermaglie. Anche nel caso di un incidente aereo, per esempio dovuto a un malfunzionamento o all'errore umano del pilota, è necessario l'intervento di operatori specializzati e, in particolare, l'applicazione di una serie di procedure specifiche.

1.2.2 I rischi

Come si può capire dal nome, il termine CBRN fa riferimento un ampio spettro di rischi (*Figura 1.1*):

- Chimico. In questo rischio rientrano tutti gli attacchi e, in generale, le situazioni in cui si ha a che fare con una sostanza chimica dalle proprietà tossiche. Si pensi, per esempio, a dispersioni di antrace, bombe che diffondono gas dalle proprietà urticanti, ma anche incidenti in cui si libera nell'aria un gas dalle proprietà nocive per l'uomo.
- Biologico. Si parla di questo tipo di rischio quando si è in presenza di un agente microbiologico (o una tossina derivata da esso) che può portare al contagio della popolazione. In questo caso, si possono fare esempi di attacchi con lo scopo di diffondere l'agente (si pensi a dispersione aeree o alla contaminazione dei viveri) ma anche di casi in cui la presenza del contaminante è dovuta a un incidente (la rottura di un contenitore, per citarne uno).
- Radiologico. Consiste nella dispersione di materiale radioattivo con lo scopo di causare danni al territorio e alla popolazione. Si differenzia dal caso nucleare perché le cosiddette bombe sporche, rispetto a una bomba nucleare, hanno un potenziale distruttivo notevolmente minore e hanno come scopo ultimo la diffusione dell'agente (e quindi il terrorismo), non la distruzione.
- Nucleare. Rientrano in questo tipo di rischio le armi e gli incidenti che si basano sulla fissione e/o la scissione nucleare. Gli esempi più conosciuti sono le bombe atomiche sganciate su Hiroshima e Nagasaki, rispettivamente il 6 e 9 agosto del 1945.



Figura 1.1: Simboli per i rischi CBRN utilizzati nell'applicazione

La grande varietà di rischi e, conseguentemente, di situazioni che devono essere gestite ha portato alla nascita, all'interno del dominio CBRN, di una lunga serie di procedure specializzate, ognuna con un compito specifico. A seconda del rischio affrontato, inoltre, le procedure prevedono variazioni in termini di strumenti usati (sonde per rilevare particelle o altri sensori per riconoscere l'agente chimico), tipi di approcci (per esempio, avvicinarsi ad una

zona da sopravvento o sottovento), formazioni (progressione in fila indiana o in formazione triangolare), etc.

Tra le varie discipline che fanno parte del dominio CBRN si possono citare: Aircraft Decon (per la decontaminazione di velivoli), Colpro (per la decontaminazione approfondita delle persone, inclusi gli operatori), SIBCRA (con lo scopo di raccogliere campioni degli agenti contaminanti) e Recce (termine con cui si identifica il team di ricognizione, che esegue il primo intervento).

1.2.3 Il Recce Team

La cosiddetta Recce (termine alternativo per “reconnaissance”) è la procedura CBRN che si occupa del primo intervento e della ricognizione dell’area a rischio. Il *Recce Team* (un gruppo di quattro operatori che si occupa di questa operazione) entra in gioco subito dopo che l’incidente scatenante (sia esso un attacco o il frutto di un errore) si è concluso e ha il compito di controllare l’area, effettuare letture, individuare il punto con la maggior concentrazione di agente contaminate, delimitare l’area a contaminazione non nulla e tornare alla base.

Le azioni del Recce Team servono per identificare dei corridoi e delle aree sicure nella zona dell’incidente, che poi potranno essere utilizzate da altri operatori per eseguire altre letture e altri controlli.

1.3 Lo scopo della tesi

Lo scopo di questa tesi è la realizzazione di una piattaforma in VR che simuli un’esercitazione CBRN del Recce Team: in poche parole, un *serious game*, cioè un’applicazione il cui scopo non sia il puro intrattenimento, ma piuttosto l’apprendimento e l’accrescimento personale. L’applicazione è stata sviluppata negli ultimi 6 mesi in modo poter essere presentata e sperimentata alla Toxic Trip 2019.

In particolare, la piattaforma si presenta come una simulazione orientata all’operatore CBRN, o comunque a un utente che si sta addestrando nella disciplina. Particolare attenzione è stata dedicata, durante lo sviluppo, alla riproduzione dello scenario, delle procedure e della strumentazione, in modo da ricreare un’esperienza che sia complementare, o possa addirittura sostituirsi, in prospettiva, ad un’esercitazione classica. Sono presenti due modalità: una modalità utente singolo per permettere all’utente di apprendere gli automatismi e i dettagli della procedura, e una multi-utente per poter allenare anche la collaborazione e rinforzare il lavoro di squadra.

Concentrandosi sul Recce Team e ambientando l’esperienza in un piazzale dell’aeroporto di Villafranca si intende porre l’accento sui vantaggi dell’esperienza VR rispetto a quella classica: costi ridotti, nessuna necessità di chiudere parte dell’aeroporto, l’utente non viene esposto a nessun rischio effettivo senza però ridurre il realismo, etc. Le simulazioni in VR, infatti, si

prestano molto bene alla riproduzione di situazioni rischiose e di emergenza, riuscendo infatti a replicare fedelmente grandi aree, effetti di incidenti o disastri su larga scala e strumentazione specifica e costosa.

Capitolo 2

Stato dell'arte

2.1 La Realtà Virtuale

Con questo termine si identificano tutte quelle esperienze in cui l'utente interagisce con un ambiente virtuale, verosimile e interattivo, sia esso una riproduzione di un luogo realmente esistente o frutto dell'immaginazione dello sviluppatore. Rientrano nella VR sia le applicazioni desktop (si pensi a videogiochi quali Tomb Raider o Uncharted), sia le applicazioni fruibili attraverso un visore: la differenza tra le due esperienze consiste semplicemente nel diverso grado di immersione e presenza suscitato nell'utente.

2.1.1 Immersione e presenza

Poiché la VR colloca l'utente in una riproduzione di un ambiente, è essenziale che esso sia verosimile e possa essere fruito senza filtri, ossia che esso non rappresenti un ostacolo che si pone tra l'utente e lo scopo dell'applicazione (sia esso l'intrattenimento, l'addestramento, l'apprendimento o altro). Per essere più chiari: è inutile creare un'applicazione per il training, se poi l'esperienza dell'utente viene costantemente minata dalla sensazione di essere davanti a un gioco. In una situazione simile, infatti, l'utente si dissocia dalle procedure ricreate, annullando di conseguenza gli effetti positivi che la piattaforma dovrebbe fornire.

Concetti fondamentali per la VR sono, quindi, *immersione* e *presenza* [2].

- Con immersione si indica la percezione di essere fisicamente presenti nel mondo virtuale. È funzione degli stimoli forniti dall'applicazione, siano essi video, audio, aptici o inerziali. Un maggior numero di canali stimolati da una simulazione (e una maggior cura dell'output fornito dalla stessa) corrisponde a una maggiore immersione dell'utente.
- Con presenza, invece, si indica la sensazione di appartenenza al mondo virtuale. Può essere vista come un'immersione mentale dell'utente, e si basa sul concetto di “sospensione dell'incredulità” caratteristico di qualsiasi tipo di narrazione (libri, film, videogame, racconti). L'utente, in pratica, sospende volontariamente le proprie capacità critiche, accettando la riproduzione virtuale come nuovo contesto in cui dovrà agire.

La VR non ha come scopo la riproduzione totale della realtà (non siamo in Matrix), ma si pone come obiettivo la simulazione fedele del contesto in cui la procedura riprodotta viene eseguita (nel caso della piattaforma qui presentata, la ricreazione dello scenario, della strumentazione e

delle azioni che possono essere eseguite) in modo da rendere l'esperienza il più verosimile possibile.

2.1.2 Storia

Negli ultimi anni si è assistito a un boom della VR: nuove periferiche a basso costo sono state introdotte sul mercato, varie aziende e Istituti di ricerca (compreso il Politecnico di Torino, attraverso l'iniziativa VR@POLITO) esplorano nuove tecnologie e nuovi utilizzi per le applicazioni VR e, in generale, si tende a parlare molto di simulazioni, riproduzione virtuale di ambienti e esperienze immersive (basti pensare al successo del libro Ready Player One e del relativo film).

La VR, tuttavia, è presente da molto più tempo di quanto si pensi, infatti se ne possono trovare esempi già negli anni sessanta: si pensi al *Sensorama* (Figura 2.1) [3] e alla prima versione di HMD (*Head-Mounted Display*) progettata da Sutherland nel 1968. In generale, nella seconda metà del 1900 può essere identificata una spinta verso la realtà virtuale che, con alti e bassi (dovuti principalmente a limiti tecnologici e ai costi eccessivi) è arrivata fino ad oggi. Il recente boom è quindi solo un'appendice di questa continua ricerca di nuove tecnologie, e in particolare deve la sua esplosione all'industria videoludica che, spinta dalla necessità di sorprendere l'utente con nuove esperienze, è tornata a investire su simulazioni *immersive* (con scarsi risultati dal punto di vista dell'intrattenimento, ma dal grande potenziale dal punto di vista dell'addestramento e dei serious game in generale).

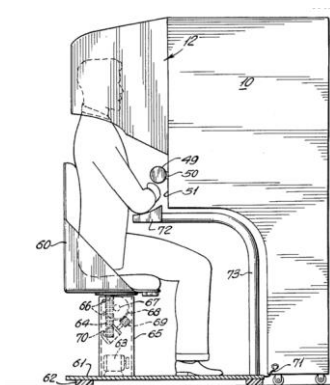


Figura 2.1: *Sensorama*, schema. 1962,
Morton Heiling,

2.2 Strumenti per la Realtà Virtuale

Le applicazioni in VR possono essere fruite utilizzando mouse, tastiera e uno schermo da scrivania, tuttavia una configurazione del genere fornisce una bassa immersione e una scarsa sensazione di presenza. Nel tempo sono state impiegate nuove tecnologie che permettono di aumentare l'immersione dell'utente: schermi avvolgenti e HMD per la componente video dell'applicazione, cuffie e sistemi di audio 3D per il suono, controller e guanti per l'interazione.

2.2.1 Display

Per quanto riguarda i sistemi di output visivo, la naturale evoluzione dei desktop consiste nell'aumento della superficie di output: l'utilizzo di molteplici schermi che circondano l'utente, di schermi curvi, *dome display* e di sistemi come il *CAVE* [4] aumenta il coinvolgimento degli utenti, ma queste soluzioni hanno, in genere, problemi di sincronizzazione, costi elevati, occupano molto spazio e limitano i movimenti degli utenti.

Un'altra evoluzione del classico schermo da scrivania consiste nell'utilizzo di HMD. I classici visori per la VR (ormai entrati nell'immaginario comune) permettono all'utente di vivere un'esperienza immersiva, fornendo contenuti 3D sfruttando la stereoscopia e imponendo meno vincoli sulla libertà di movimento dell'utente. Non sono comunque esenti da problemi: costituiscono infatti un ingombro aggiuntivo che appesantisce la testa, sono spesso collegati a un filo che intralcia i movimenti degli utenti, soffrono di problemi dovuti alla bassa risoluzione e, dovendo renderizzare due immagini contemporaneamente (per dare la sensazione di profondità) richiedono schede grafiche dalle prestazioni elevate.

Nella piattaforma VR presentata in questo elaborato si è deciso di concentrarsi sull'esperienza fornita dagli HMD.

2.2.2 Metodi di interazione

I metodi di interazione utilizzabili in VR sono molteplici. Utilizzando schermi su scrivania è possibile interagire con l'applicazione attraverso i classici mouse e tastiera, ma lo sviluppo di tecnologie più immersive ha richiesto lo sviluppo di nuove interfacce che permettano un'interazione più immediata e naturale con la simulazione.

I sistemi di input utilizzati nascono con lo scopo di emulare elementi conosciuti dagli utenti, per favorire così l'immersione: si pensi alla maggior parte dei controller con sensori di movimento (come il Wii Mote della Nintendo o i controller dell'HTC Vive), che ricalcano la forma e la funzionalità di un telecomando per la televisione e sfruttano la familiarità dell'utente con esso, ma anche ai guanti o alle telecamere che permettono di tracciare i movimenti delle

dita della mano, offrendo la possibilità di interagire con il mondo virtuale in modo simile a come si interagisce con il mondo reale.

È necessario notare che, per quanto un sistema di input possa essere funzionale, non permetterà mai un'interazione completamente realistica con l'ambiente virtuale: saranno comunque necessarie delle astrazioni e delle interfacce che permettano all'utente di eseguire determinati tipi di interazione con gli oggetti virtuali.

Nella piattaforma presentata, anche in virtù della scelta di utilizzare gli HMD, si è deciso di concentrarsi sull'utilizzo di controller che ricalcano le funzionalità del telecomando.

2.2.3 Locomozione

Uno degli elementi chiave di un'applicazione VR è costituito dalla modalità di locomozione, cioè da come l'utente riesce a muoversi e a esplorare l'ambiente riprodotto. Chiaramente, si potrebbero utilizzare delle soluzioni classiche, come l'utilizzo della tastiera o delle levette analogiche presenti su alcuni tipi di controller, ma l'utilizzo degli HMD apre ad altre possibilità.

Innanzitutto, essendo gli HMD tracciati nello spazio (almeno quelli utilizzati per la piattaforma qui descritta), uno spostamento eseguito nel mondo reale, indossando il visore, viene tradotto in un movimento corrispondente e scalato nel mondo virtuale. Se fossero disponibili molti sensori, si potrebbe teoricamente tracciare un'intera stanza, ad esempio un capannone o un hangar, e dare la possibilità all'utente di muoversi fisicamente. Nel caso di questa piattaforma, tuttavia, i sensori a disposizione sono due per ogni visore, quindi il movimento reale tradotto in movimento virtuale ha una portata limitata. È quindi necessario trovare delle astrazioni, dei metodi che permettano di simulare la camminata.

In generale, sono disponibili dispositivi aggiuntivi che permettono di superare questo problema, come pedane e sensori da agganciare alle gambe, ma volendo limitare il più possibile la strumentazione da utilizzare (anche per problemi di trasporto, in vista della Toxic Trip 2019) si è deciso di limitarsi ai semplici controlli manuali. In una situazione del genere, le opzioni offerte dai controller sono due: il teletrasporto, escluso perché poco compatibile con la procedura CBRN riprodotta, e l'Arm Swinging, un meccanismo che permette di muoversi nel mondo virtuale tenendo premuti due tasti dei controller e simulando con le braccia i movimenti che si eseguono durante una camminata o una corsa. Nella piattaforma presentata si è deciso di utilizzare quest'ultimo.

2.3 Realtà Virtuale e addestramento

Sebbene lo sviluppo della VR sia strettamente legato al mondo dei videogiochi, i suoi possibili impieghi all'esterno del mondo dell'intrattenimento sono presto saltati agli occhi di tutti. Basti pensare ai mondi virtuali impiegati per il superamento di fobie [5] o per la riabilitazione, alle

riproduzioni virtuali di luoghi o oggetti per il turismo o per la conservazione dei beni culturali, o ancora alle applicazioni che permettono agli utenti di affinare le proprie capacità comunicative, ponendoli di fronte a una folla virtuale e dando loro il compito di recitare un discorso.

Molti di questi esempi rientrano nel campo dell'addestramento virtuale, ossia di quelle simulazioni che trasportano l'utente in un mondo ricreato al computer, con lo scopo di fargli affrontare una determinata situazione e di valutare le sue azioni. La ripetizione dell'esperienza virtuale, similmente a quella di esercitazioni pratiche nel mondo reale, permette quindi all'utente di notare e correggere i propri errori, di apprendere gli automatismi e le corrette procedure legate all'esperienza che sta affrontando.

Strumenti di addestramento virtuale fanno parte dell'immaginario comune, ma sono sempre più utilizzati per addestrare sia operatori sia utenti senza esperienza: basti pensare, nel primo caso, ai simulatori di volo impiegati dall'Aeronautica (per rimanere nel contesto in cui è nato lo strumento descritto da questa tesi) o, nel secondo caso, alle simulazioni di situazioni d'emergenza rivolte al personale civile (per permettere all'utente comune di trovare velocemente le uscite di emergenza in caso d'incendio, per esempio).

2.3.1 Vantaggi

L'addestramento in un mondo virtuale porta con sé una serie di vantaggi che lo rendono una soluzione migliore rispetto alle esercitazioni reali, in particolare sotto il punto di vista economico, organizzativo, e soprattutto per quanto riguarda la sicurezza degli utenti e dei mezzi impiegati negli addestramenti.

Nel dettaglio, una simulazione virtuale è:

- meno costosa della controparte reale. Le esercitazioni, in particolare quelle che riguardano compiti complessi, come ad esempio l'addestramento al volo o un addestramento CBRN, richiedono l'impiego di mezzi e attrezzature specifiche, costose (gli aerei e gli altri mezzi di trasporto) e fragili. Richiedono inoltre la mobilitazione di un gran numero di personale, sia attivo sia di supporto.
- Più completa della controparte reale. Le esercitazioni pongono l'utente in situazioni fittizie e, senza l'ausilio di un mondo virtuale, richiedono dei compromessi e delle astrazioni. Un attacco aereo, un incendio, la presenza di un materiale radioattivo non sono dettagli secondari, ma in un'esercitazione normale possono essere replicati solo su scala ridotta o tramite dei simulacri o dei segnaposto, che minano il realismo dell'esperienza complessiva.
- Più semplice da organizzare. Un'esercitazione classica deve essere preparata con largo anticipo, specialmente se coinvolge molte persone e richiede molto materiale. Oltre questo, un'esercitazione classica ha bisogno di strutture specifiche per essere eseguita (un piazzale della base di Villafranca nel caso CBRN, un edificio o un tunnel per la risposta agli incendi). D'altro canto, una simulazione virtuale permette di immergere

l'utente nelle stesse situazioni con poco preavviso, richiedendo semplicemente un computer e un visore per ogni utente.

- Ripetibile. Come conseguenza di tutti gli altri vantaggi, ne risulta che una simulazione virtuale è molto più facile da eseguire e da ripetere rispetto a un'esercitazione classica, ed è proprio ripetendo più volte l'esperienza che l'utente riesce ad assimilare le procedure e i gesti tipici della disciplina in cui deve addestrarsi.

2.3.2 Svantaggi

I principali svantaggi delle simulazioni virtuali sono fondamentalmente legati all'aspetto tecnologico, in particolare a tutti gli aspetti che non possono ancora essere replicati nel mondo virtuale e a tutti i dispositivi e alle astrazioni che sono ancora necessari per ricreare le esperienze reali.

Per quanto riguarda il primo punto, non è ancora possibile integrare odori, feedback tattili e forze d'inerzia nella simulazione virtuale senza richiedere costi elevati o ingombri eccessivi. Per il secondo punto, invece, è necessario considerare che occhialini, visori e controller sono astrazioni, elementi in più che aggiungono un livello di complessità alla simulazione (sebbene facilmente superabile) e che si frappongono tra l'utente e l'esperienza simulata.

Con il costante sviluppo della tecnologia questi limiti verranno progressivamente ridotti e superati, ma attualmente sono presenti e da tenere in considerazione.

Capitolo 3

Tecnologie

3.1 Game Engine

Un'applicazione VR è un sistema complesso, con il compito di gestire molteplici elementi contemporaneamente e in tempo reale. A supporto dello sviluppatore, esistono numerose librerie che permettono di avere il controllo sui vari aspetti del prodotto finale:

- OpenGL e DirectX, tra le tante, per il rendering Video;
- OpenAL, DirectSound per il rendering audio;
- OpenHaptics, CHAI3D, H3DApi per il rendering aptico;
- Open Dynamics Engine per la gestione della fisica;
- SmartBody per la gestione degli avatar.

La grande varietà di librerie disponibili porta, tuttavia, ad avere problemi di integrazione: non tutte supportano lo stesso linguaggio, non tutte supportano gli stessi formati e, in generale, non sono forniti gli elementi necessari per far “comunicare” tra loro le varie parti. Una possibile soluzione a questo problema consiste nello scrivere una macro applicazione in c++ che integri i vari componenti: questa scelta, tuttavia, non è ottimale perché richiede molto tempo, molte linee di codice, è prona agli errori e, in generale, dirotta verso il basso livello (cioè verso la comunicazione tra le varie librerie) tutte le risorse che, in teoria, sarebbe meglio dedicare alle logiche dell'applicazione.

La soluzione migliore per uno sviluppatore, quindi, consiste nell'utilizzare un programma che si posizioni tra lui e la componente hardware, gestita direttamente dalle librerie. Questo programma, questo framework gestisce e maschera gran parte della complessità dei livelli più bassi (la comunicazione tra librerie, la gestione del networking, etc.) e fornisce allo sviluppatore un'interfaccia semplificata, più gestibile e semplice da usare. Nel caso specifico dello sviluppo di videogame e serious game, questo programma prende il nome di Game Engine.

3.1.1 Vantaggi

L'utilizzo di un Game Engine, quindi, porta una serie notevoli di vantaggi:

- riduce i costi, sia dal punto di vista monetario (lo sviluppatore ha bisogno, sostanzialmente, di un unico prodotto per poter creare un'applicazione) sia in termini di tempo (poiché esula lo sviluppatore dalla gestione di basso livello);
- permette di programmare tramite script, riducendo quindi (almeno in teoria) le competenze di programmazione necessarie per ottenere risultati soddisfacenti;
- mette a disposizione un'interfaccia semplificata, da cui il programmatore può gestire i vari elementi dell'applicazione tramite meccanismi intuitivi e facili da usare (per esempio, per aggiungere una componente audio o una luce basta utilizzare un meccanismo drag & drop);
- fornisce meccanismi di alto livello per la gestione di shader, online, effetti particellari, forze, etc.
- trattandosi di un software robusto e testato continuamente, risulta essere meno soggetto ad errori rispetto ad una applicazione scritta ad hoc per lo scopo prefissato.

3.1.2 Unity

Nel caso particolare di questa applicazione si è scelto di utilizzare Unity [6], nella versione 5.6.7 (perché stabile e già utilizzata al Politecnico di Torino per produrre strumenti VR simili alla piattaforma qui descritta).

Unity è un Game Engine gratuito sviluppato da Unity Technologies. Inizialmente pensato per lo sviluppo di applicazioni per Mac, nel tempo si è evoluto in un motore di gioco capace di creare applicazioni per piattaforme diverse (PC, sistemi Android, Playstation, Xbox, etc.). Permette di realizzare prodotti sia in 2D che in 3D, sia in VR che in realtà aumentata, e di integrare all'interno dello stesso prodotto, in maniera semplice, audio, video, modelli 3D, immagini e script (scritti sia in c++ che in Javascript). È dotato di un'interfaccia personalizzabile e facile da usare (*Figura 3.1*), che permette di gestire la scena di gioco attraverso un meccanismo drag and drop.

Sebbene nato in ambito videoludico, viene spesso utilizzato in altri contesti come: cinema, sviluppo di progetti ingegneristici, architettura, design, etc.

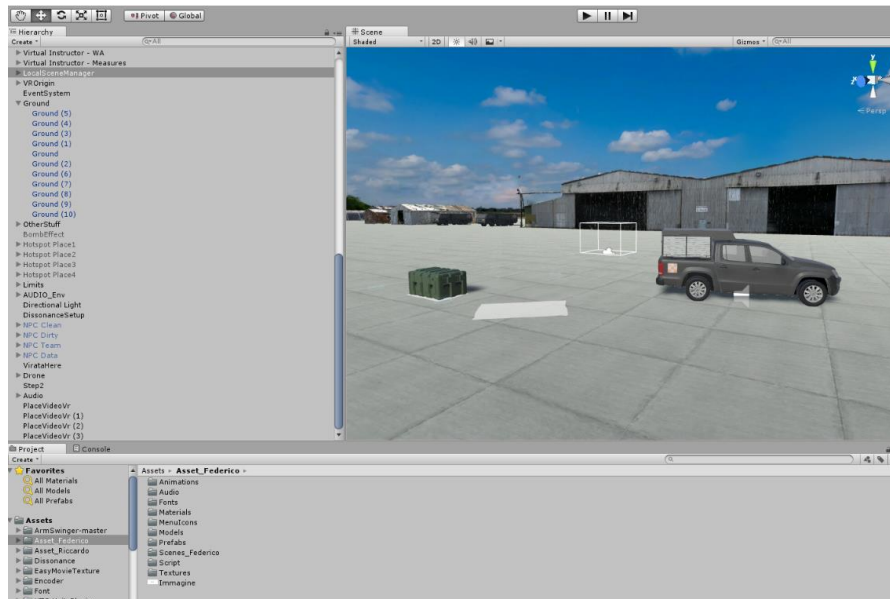


Figura 3.1: Interfaccia di Unity 5.6.7

3.2 Software di supporto

Unity offre la possibilità di interfacciarsi facilmente e in maniera ottimizzata con altri programmi: in particolare, Unity non è un software di modellazione 3D, quindi si appoggia pesantemente su programmi dedicati, e non offre nemmeno un IDE (Integrated Development Environment) molto sviluppato, quindi permette di utilizzare altri programmi per la scrittura degli script e il debug.

Nella piattaforma descritta in questo elaborato, in particolare, si è scelto di utilizzare Blender e Maya per la creazione di modelli 3D, Visual Studio per la scrittura degli script e per il debug, e Optitrack per il *motion capture* e la registrazione delle animazioni.

3.2.1 Visual Studio

Visual Studio [7] è un ambiente di sviluppo integrato sviluppato dalla Microsoft. Permette di programmare in vari linguaggi (c#, Visual Basic, c++) e di sviluppare applicazioni per piattaforme diverse (PC, mobile, console). Offre inoltre una serie di add-on che facilitano la comunicazione con altri programmi: l'integrazione di Unity, in particolare, dà la possibilità di utilizzare Visual Studio come IDE di scripting predefinito e di utilizzare le librerie specifiche di Unity, oltre a permettere il debug run-time delle applicazioni sviluppate con il motore di gioco.

Nella piattaforma descritta, è stato utilizzato per la creazione di tutti gli script e per l'organizzazione generale del progetto.

3.2.2 Blender

Blender [8] è un software gratuito e open source per la modellazione 3D. Permette all'utente di creare e scolpire modelli, creare animazioni 3D, gestire materiali e texture, montare video e, nelle versioni meno recenti, anche di programmare applicazioni interattive grazie a un semplice game engine. Come nel caso di Unity, l'interfaccia (*Figura 3.2*) è personalizzabile ed è composta da pannelli, ognuno dei quali ha uno scopo specifico (gestione delle animazioni, gestione delle proprietà di un oggetto, mappatura delle texture, etc.).

Nel contesto della realizzazione della piattaforma per l'addestramento CBRN, è utilizzato assieme a Maya per la riproduzione di alcuni oggetti di gioco (la PSDS, i triangoli, le mani e la testa degli utenti, l'orologio, etc.) e per la mappatura di alcune texture (la cartina e i segni dell'esplosione del drone). Si è scelto di utilizzare la versione 2.79 perché, all'inizio del progetto, era la più stabile.



Figura 3.2: Interfaccia di Blender

3.2.3 Optitrack

Optitrack [9] è un'azienda fornitrice di sistemi per il *motion tracking*. Il programma omonimo permette, attraverso i sensori e l'attrezzatura proprietari, di registrare e ripulire animazioni tracciando una persona che indossa una tuta con dei marker. Il Politecnico di Torino possiede un sistema completo di motion tracking che è stato utilizzato per registrare gesti e posizioni specifiche per la procedura CBRN considerata (il cosiddetto motion capture). Le animazioni generate in questo modo sono state ripulite e inserite in Unity per rendere più realistici gli NPC (Non-Player Character).

3.3 Hardware

3.3.1 Schede grafiche e rendering

La piattaforma è stata sviluppata e pensata per essere utilizzata su pc VR-Ready e con visori (in particolare, HTC Vive e Oculus Rift). L'utilizzo della VR richiede pc performanti, con schede grafiche di buon livello: dei quattro PC a disposizione durante lo sviluppo dell'applicazione, quello dotato di scheda grafica Nvidia GeForce GTX 970 ha dimostrato di reggere a fatica l'Oculus. Le altre tre postazioni disponibili, dotate invece di scheda Nvidia 1080 TI, hanno fornito ottime prestazioni sia con l'Oculus Rift che con l'HTC Vive.

3.3.2 HTC Vive

L'HTC Vive [10], annunciato nel 2014 e commercializzato nel 2016, è un visore wired (dotato di filo) utilizzabile grazie a due sensori (base station, *Figura 3.3*) che possono essere posizionati all'interno di una stanza per tracciare i movimenti del dispositivo. Questi due sensori sono ottici e possono essere soggetti a interferenze (in particolare, l'utilizzo di una terza base station può creare dei conflitti, con conseguente comportamento imprevedibile del sistema).



Figura 3.3: Base station dell'HTC Vive

Il visore (*Figura 3.4*) si collega al PC attraverso una presa USB 3.0 e una presa HDMI, e necessita inoltre di essere collegato a una presa di corrente (da qui l'utilizzo del cavo). Modelli più recenti permettono di utilizzare un adattatore wireless per non limitare i movimenti dell'utente. Le base station devono essere collegate a una presa di corrente e offrono la possibilità di essere collegate tra loro per aumentare la sincronizzazione.



Figura 3.4: HTC Vive, visore

Il display del visore è costituito da due schermi da 1080x1200. Il *field of view* pubblicizzato è di circa 110 gradi in orizzontale e 110 gradi in verticale, e il field of view effettivo rientra in questi valori [11]. Il visore possiede inoltre un giroscopio, un accelerometro, sensori infrarossi che servono per il tracking, più una telecamera che, assieme alle base station, permette di tenere traccia dell'area sicura in cui l'utente può muoversi liberamente.

L'HTC Vive è dotato di due controller, anch'essi tracciati, che sono visibili all'interno del mondo virtuale (di base, viene visualizzato un modello che li riproduce fedelmente, ma all'interno di altre applicazioni possono essere sostituiti da mani o altri oggetti). I due controlli hanno una batteria interna e devono essere caricati regolarmente. Il visore può anche essere utilizzato assieme a dei sensori generici, i Vive Tracker, che possono essere utilizzati per tracciare vari oggetti o parti del corpo.

3.3.3 Oculus Rift

L'Oculus Rift [12] (*Figura 3.5*) è un HMD prodotto dalla Oculus VR e di proprietà della Facebook Inc. È un visore cablatto che richiede di essere collegato al PC tramite una presa USB 3.0 e una presa HDMI. È dotato di due sensori che, a differenza di quelli del Vive, sono pensati per essere posizionati sulla scrivania. Questa soluzione aumenta la praticità, ma comporta una perdita in termini di prestazioni: il posizionamento dei sensori sullo stesso livello implica una perdita di segnale vicino al terreno e/o in posizioni troppo elevate. Il segnale, inoltre, tende a indebolirsi nei casi in cui l'utente si ritrovi a dare la schiena ai sensori, coprendo i controller.



Figura 3.5: Oculus Rift - visore, sensori e controller

Il display del visore è costituito da due schermi OLED (Organic Light Emitting Diode) da 1080x1200. Il field of view pubblicizzato è di 110 gradi in orizzontale e 110 gradi in verticale, ma risulta essere effettivamente minore. In generale, la qualità grafica è minore rispetto a quella garantita dal Vive. L'Oculus, a differenza del Vive, è fornito di due cuffie integrate per l'audio 3D.

I controlli dell'Oculus sono simili a quelli del Vive, ma la forma è stata pensata per essere più ergonomica e per mappare meglio la posizione delle dita della mano. I controller, inoltre, forniscono più tasti rispetto a quelli del Vive, anche se si possono trovare delle caratteristiche comuni (tasi come il Grip e il Trigger sono presenti in entrambi, mentre la levetta analogica dell'Oculus corrisponde al touchpad del Vive). I controller dell'Oculus hanno bisogno di una batteria AA ciascuno per funzionare.

Capitolo 4

Progettazione

La realizzazione di una piattaforma come quella descritta, soprattutto considerando la portata dell'evento in cui è stata presentata, non poteva non seguire un'attenta pianificazione. Scrivere direttamente centinaia di righe di codice senza una direzione è il primo passo che porta a errori e alla riscrittura da zero di intere logiche, con conseguente spreco di tempo e di risorse. In situazioni come queste è quindi evidente come sia necessaria un'attenta progettazione, che può richiedere anche parecchio tempo: tale processo è da vedere come un investimento in ottica futura, che permette di spendere risorse in un primo momento per evitare contrattempi e passi indietro a lavoro avviato.

4.1 L'applicazione precedente

Il punto di partenza per la piattaforma presentata in questo elaborato è un'applicazione sviluppata dal SiTI (prima di convergere, insieme all'Istituto Superiore Mario Boella, nell'attuale Fondazione LINKS). Similmente al nuovo strumento, la vecchia applicazione aveva l'obiettivo di replicare un'esercitazione CBRN per addestrare i membri del Recce Team.

Il primo passo per lo sviluppo di questa tesi è stato quindi l'analisi e la decomposizione del materiale già disponibile, per comprendere quali elementi potevano essere utilizzati e quali, invece, dovevano essere scartati. A supporto di questa fase erano presenti dei documenti che riassumevano i feedback e le migliorie proposte dai militari del Terzo Stormo. Alla fine di questo processo, si è giunti a un risultato molto chiaro: la vecchia applicazione non era fedele alle procedure che si volevano replicare. Per quanto riguarda il lato grafico, i modelli 3D e le texture erano ben realizzati e facilmente riutilizzabili, sebbene necessitassero di alcuni aggiornamenti. Dal punto di vista delle logiche e del codice, invece, il materiale doveva essere riscritto da zero, sia per ragioni tecniche (non si prestava all'integrazione della componente multi-utente), sia per ragioni di fedeltà alle procedure (mancavano completamente delle fasi, erano presenti degli elementi che non riguardavano la procedura presa in considerazione, gli utenti avevano poca libertà di azione, e così via).

Conclusa questa fase, è stato quindi evidente che non si poteva procedere senza prima avere un documento che mettesse su carta, senza ambiguità, tutti i requisiti e le caratteristiche della procedura CBRN che doveva essere replicata.

4.2 Lo storyboard

Il documento necessario a questa applicazione era uno *storyboard*, cioè un elaborato che racchiudesse in sé tutte le caratteristiche della piattaforma finale. Il documento in questione doveva essere onnicomprensivo, quindi doveva contenere informazioni sulla procedura, sull'aspetto grafico, sull'audio, sulle logiche e sul sistema di valutazione.

4.2.1 Realizzazione

Lo storyboard è stato realizzato presso i locali della Fondazione LINKS, ed è il frutto della collaborazione con un team di persone provenienti sia dalla Fondazione Links che dal Terzo Stormo. La validazione tecnica è stata possibile, inoltre, grazie ai feedback di docenti e ricercatori del Politecnico di Torino, mentre la correttezza delle procedure è stata garantita grazie a una continua comunicazione con il personale CBRN dell'aeroporto di Villafranca.

Il processo di creazione dello storyboard è durato più di un mese, anche per via della grande quantità di opzioni da considerare e del fatto che, comunemente, si volevano evitare gli errori commessi nell'applicazione precedente. In particolare, è stata dedicata una grande attenzione ai seguenti aspetti.

- Soggetto. Per comprendere se convenisse concentrarsi sul Recce Team, o se fosse necessario aggiungere anche altre procedure al prodotto finale.
- Fedeltà. Per evitare gli stessi errori che avevano caratterizzato la vecchia applicazione, per esempio l'utilizzo di mezzi sbagliati, l'assenza di elementi fondamentali della procedura o la presenza di oggetti e fasi che non dovevano essere presenti.
- Modelli. Per sfruttare al meglio il team di modellatori si è dedicato parecchio tempo alla divisione e all'ottimizzazione dei compiti.
- Logiche. Per ottenere un giusto compromesso tra la complessità del codice e il risultato ottenuto su schermo, si è svolta un'analisi basata sulla priorità per capire quali elementi meritassero maggior attenzione e quali, invece, potessero essere semplificati.

Lo storyboard finale, prodotto grazie allo sforzo congiunto citato in precedenza, descrive interamente la piattaforma, andando a considerare anche la grafica, l'audio, l'aspetto dei menu, le fasi della simulazione, il video iniziale e le logiche (descritte ad alto livello, lasciando l'effettiva realizzazione allo sviluppatore).

4.2.2 Il soggetto

Si è deciso in primo luogo di mantenere il soggetto precedente, e quindi di simulare un'esercitazione del solo Recce Team, modificando però il contesto dell'operazione: se prima

l'utente doveva rispondere a un'emergenza causata da un incidente aereo, adesso lo scenario iniziale prevede un intervento in seguito all'abbattimento di un drone che vola senza permesso sull'aeroporto di Villafranca. Il cambiamento è stato dettato dal fatto che, nel contesto dell'Aeronautica, i cosiddetti "drone strike" sono un argomento più attuale e verso cui è richiesta una maggiore attenzione [13]. Per ampliare la scelta offerta all'utente, è stata aggiunta la possibilità di affrontare rischi CBRN diversi, come riportato di seguito.

- **Rischio radiologico.** Il rischio presente anche nella vecchia simulazione. Il drone abbattuto porta un oggetto sospetto e il Recce Team deve controllare se è presente materiale radiologico e, eventualmente, delimitare l'area contaminata. La scelta del rischio radiologico comporta l'utilizzo di strumenti che erano già presenti nella vecchia simulazione.
- **Rischio chimico.** Rischio aggiunto da zero nella nuova simulazione. La situazione è simile a quella del rischio radiologico, con una sostanziale differenza: l'agente chimico è volatile e la sua dispersione dipende dalla direzione del vento. L'area contaminata sarà quindi allungata e dalla forma triangolare. La scelta di questo rischio comporta l'utilizzo di strumenti non presenti nella vecchia simulazione e realizzati per l'occasione.
- **Rischio *random* (casuale).** Questa terza opzione consiste nella scelta casuale tra uno dei rischi appena citati e comporta la possibilità di essere valutati anche sul tipo di strumentazione utilizzato: l'utente si ritrova, infatti, a dover scegliere tra le sonde tipiche del caso radiologico e quelle tipiche del caso chimico, basandosi su indizi di contorno (audio e letture degli strumenti). A seconda della scelta, può completare la simulazione utilizzando solamente gli strumenti non adatti al rischio affrontato e, quindi, ricevere un giudizio negativo.

4.2.3 Modalità

Per quanto riguarda la componente online, si è deciso di aggiungere, oltre alla modalità utente singolo (utile soprattutto per apprendere gli automatismi e prendere confidenza con le sonde), anche una modalità a quattro utenti. Inizialmente si è pensato di realizzare anche delle configurazioni intermedie, ma sono state scartate perché potenzialmente fuorvianti, visto che il Recce Team è formato da quattro persone e la cooperazione deve essere valutata tra tutti e quattro gli operatori.

Si è scelto, inoltre, di inserire una modalità chiamata Virtual Instructor (VI), in cui una serie di pannelli e indizi a schermo guidano l'utente, spiegando ogni passo della procedura utente singolo. Il VI inizialmente era nato come un semplice menu da richiamare esplicitamente attraverso un tasto, ma si è poi evoluto in questa forma finale perché ritenuta più utile per gli utenti che non hanno dimestichezza con la piattaforma o con le procedure CBRN.

4.2.4 La simulazione

Per definire la struttura della simulazione, sono state identificate sette fasi principali in cui dividere l'esperienza, seguendo un'ossatura che traspare naturalmente dall'effettiva procedura CBRN che è stata presa in considerazione.

- Fase 0: è costituita dal video iniziale, in cui si vede un drone che vola sopra all'aeroporto di Villafranca e viene abbattuto. Il drone porta con sé un oggetto sospetto, che al contatto col terreno esplode e contamina la zona circostante. Questo avvenimento richiede l'intervento del Recce Team.
- Fase 1: il punto di vista si sposta all'interno del veicolo del Recce Team. Poiché agenti chimici e particelle volatili sono facilmente trasportati dal vento, il Team deve avvicinarsi alla zona sopravento. In base a degli indizi inclusi nella scena (in particolare, una scritta sul GPS dell'auto) l'utente deve scegliere la direzione da cui avvicinarsi al luogo dell'incidente. La scelta avviene selezionando il punto cardinale desiderato su una mappa dell'area che si trova davanti all'utente. Una diversa direzione del vento comporta una diversa posizione iniziale della simulazione.
- Fase 2: il Dirty Man scende dal veicolo e controlla se la zona in cui il Team si è fermato è sicura o meno. Questa fase è presente in modalità singolo utente solo se l'utente ha scelto di interpretare il Dirty Man, mentre è sempre presente nella versione multi-utente.
- Fase 3: il Team scende dal veicolo e monta la Working Area (WA), cioè la zona operativa a cui ogni operatore dovrà tornare, alla fine della procedura. Questa fase è sempre presente nella modalità multi-utente, poiché è interessante valutare la collaborazione tra gli utenti, ma è eseguibile in modalità utente singolo solo se si interpreta il Clean Man.
- Fase 4: una volta che la WA è completa, tutti i membri del Team tranne il Clean Man si avvicinano alla zona dove è stato abbattuto il drone per cercare l'*hotspot*, cioè il punto con il livello più alto di contaminazione (sia essa radiologica o chimica). Trovato il punto, questo va segnalato con un paletto e con un apposito cartello che dipende dal rischio affrontato.
- Fase 5: i membri del Team (Clean Man escluso) effettuano delle misurazioni attorno all'*hotspot* e delimitano con paletti, cartelli e nastro la zona in cui le sonde mostrano un livello di contaminazione non nullo.
- Fase 6: gli operatori che sono stati esposti all'agente contaminante (chimico o radiologico) devono essere decontaminati. Se l'utente interpreta il Clean Man, ha il compito di decontaminare gli altri operatori (siano essi gestiti dal calcolatore o gestiti da altri utenti). Se l'utente interpreta uno degli altri operatori, ha il compito di farsi decontaminare.

4.2.5 I ruoli

Dal punto di vista dei ruoli interpretabili, si è scelto di dare la possibilità di selezionare uno tra tre ruoli nella modalità singola (in particolare, Dirty Man, Clean Man e Data Compiler) e di aggiungere il Team Leader solo nella simulazione a quattro utenti, poiché il suo compito è soprattutto gestionale e acquista senso solo e ha dei personaggi attivi con cui interagire.

I quattro ruoli (con i rispettivi compiti) sono, appunto, quelli elencati di seguito.

- Dirty Man. È l'operatore che scende per primo dal veicolo e controlla la zona per assicurarsi che sia sicura. Ha il compito di individuare l'hotspot e di delimitare l'area contaminata. Nel caso radiologico, utilizza una "sonda alfa" per rilevare la presenza di particelle alfa, nel caso chimico utilizza il CAM e una cartina tornasole per trovare l'agente chimico.
- Data Compiler. È l'operatore che, assieme al Dirty Man, ha il compito di individuare l'hotspot e di delimitare l'area contaminata. Nel caso radiologico utilizza una sonda beta/gamma per rilevare la presenza di particelle beta, nel caso chimico utilizza la stessa attrezzatura del Dirty Man.
- Clean Man. È l'operatore che non viene esposto all'agente contaminante e ha il compito di costruire la WA. Rimane nella zona sicura individuata in precedenza dal Dirty Man e attende che gli altri operatori abbiano finito per decontaminarli e assicurarsi che siano puliti.
- Team Leader. Segue sul campo il Dirty Man e il Data Compiler e ha il compito di coordinare i loro sforzi e gestire la squadra.

4.2.6 La valutazione

Alla fine della simulazione, viene compilato un rapporto sulle azioni compiute dall'utente. I risultati mostrati dipendono dal ruolo interpretato e dal tipo di rischio affrontato. Il fatto che la procedura CBRN del Recce Team sia, entro certi margini, ben definita ha permesso di identificare una lunga serie di punti critici da tenere in considerazione. Tra i principali:

- scelta della direzione da cui si approccia l'area, data la direzione del vento;
- uso delle sonde corrette a seconda del rischio;
- posizionamento degli oggetti all'interno della WA;
- correttezza della procedura di misurazione;
- posizione del paletto per segnalare l'hotspot;
- posizione dei paletti per delimitare l'area;
- scelta del cartello adeguato a seconda del rischio affrontato;

- correttezza delle informazioni scritte sui paletti;
- posizionamento del nastro per delimitare l'area;
- completezza della procedura di contaminazione, sia se si interpreta il Clean Man sia se si interpreta un altro operatore.

4.2.7 Il menu iniziale

Particolare attenzione è stata dedicata anche al menu iniziale, essendo esso il biglietto da visita dell'applicazione. Si è scelto un menu ad albero (*Figura 4.1*) e si è deciso di inserire anche i pulsanti relativi alle procedure non riprodotte ma considerate (non sono selezionabili, ma l'idea di base era di mostrare agli utenti i possibili sviluppi dell'applicazione). La grafica dei pulsanti e delle icone è stata realizzata da zero con il contributo del Terzo Stormo.

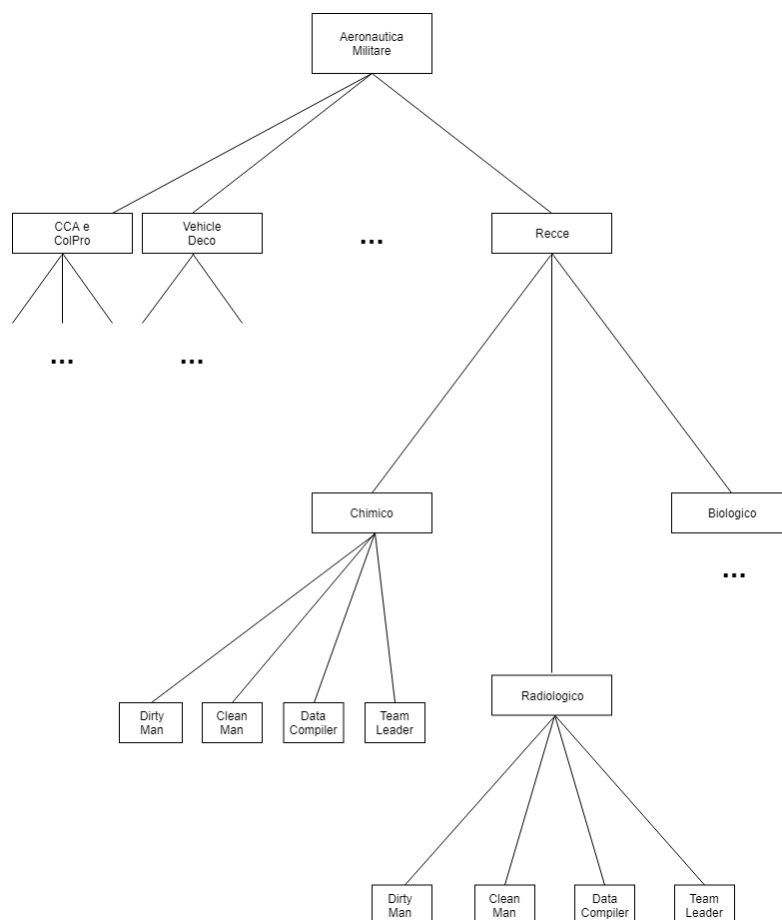


Figura 4.1: Menu iniziale, schema concettuale

4.3 La validazione

Al pari della progettazione, è importante che ogni scelta fatta e ogni informazione inserita nello storyboard siano approvate dall'utente finale, dai portatori della cultura di dominio (in questo caso, dai responsabili CBRN di Villafranca). Ogni aspetto della simulazione (il soggetto, le modalità, le fasi, i ruoli e la suddivisione dei compiti, i criteri di valutazione) è stato aggiunto alla piattaforma solo dopo aver ottenuto l'approvazione diretta del Terzo Stormo. Il feedback continuo con gli operatori CBRN è stato possibile soprattutto grazie alla mediazione continua dei militari del Terzo Stormo presenti a Torino.

Capitolo 5

Realizzazione

La realizzazione della piattaforma ha occupato la maggior parte dei 6 mesi dedicati alla tesi. Come detto, l'applicazione di partenza era da riscrivere da zero, sia per poter garantire una maggiore fedeltà alla procedura rappresentata, sia per poter integrare la componente multi-utente. Lo sviluppo è stato agevolato da asset e librerie pre-esistenti, sempre comunque da riadattare, che hanno permesso una gestione semplificata sia della componente di VR immersiva (cioè l'utilizzo di visore e controller) sia della componente online.

5.1 SteamVR e la Realtà Virtuale in Unity

SteamVR [14] è uno strumento software che permette di utilizzare la VR immersiva su un qualsiasi tipo di hardware che sia VR-Ready. È prodotto dalla Valve e consiste in un'interfaccia software in grado di supportare HTC Vive, Oculus Rift, Valve Index, Windows Mixed Reality e altri visori.

5.1.1 La libreria

Sull'asset store di Unity è presente un plugin [15] che permette al Game Engine di interfacciarsi con SteamVR. Una volta aggiunto al progetto su cui si sta lavorando, questo asset consiste in una cartella in cui si possono trovare: *prefab* di Unity da poter utilizzare immediatamente, script già scritti per la gestione della componente di VR immersiva, scene di prova e documentazione varia.

5.1.2 Le implicazioni

L'utilizzo del plugin ha permesso di semplificare notevolmente le fasi iniziali dello sviluppo della piattaforma qui descritta. Invece di dover studiare da zero un modo per gestire controller e visore, in particolare la gestione del sistema di tracking, è stato possibile utilizzare i *prefab* di SteamVR e dedicarsi fin da subito alla realizzazione della simulazione vera e propria, cioè alle logiche che hanno permesso di replicare la procedura CBRN.

5.2 UNet e la gestione di utenti multipli in Unity

Per la gestione della componente online, Unity mette a disposizione una componente integrata di default (almeno fino alla versione 2018.4) nel Game Engine: UNet [16].

5.2.1 Lo strumento

La menzionata componente di Unity permette di integrare funzionalità online nel progetto che si sta sviluppando, lasciando allo sviluppatore la possibilità di decidere se lavorare a basso livello (ricreando da zero la struttura di rete della propria applicazione, in modo da ottenere una gestione più dettagliata e funzionalità più specifiche) o ad alto livello (sfruttando GameObjects e componenti consolidati e ampliandoli all'occorrenza). Per la piattaforma sviluppata si è scelta la seconda opzione.

L'API d'alto livello, in particolare, fornisce gli strumenti necessari per avere variabili costantemente aggiornate su tutti i pc connessi alla stessa simulazione, in modo da evitare discrepanze nei dati di gioco, oltre a un sistema di funzioni che permette ai client di far eseguire operazioni al server e, viceversa, al server di far eseguire operazioni al client. Attraverso queste funzioni è possibile creare architetture in cui il server gestisce ogni cosa, ma anche strutture peer-to-peer o ibride.

5.2.2 Le implicazioni

A differenza del plugin per SteamVR, che può essere utilizzato fin da subito, potenzialmente senza richiedere alcun lavoro da parte dello sviluppatore, UNet deve essere integrato manualmente nell'applicazione e pone numerosi vincoli sulla struttura della stessa. Ciò che è a disposizione dello sviluppatore, infatti, consiste in semplici parti di una struttura più ampia che, però, non è già costruita.

Esiste un componente principale, il NetworkManager, che deve essere inserito nella scena per controllare tutti gli altri. La versione fornita, tuttavia, è basilare e deve essere riscritta o, almeno, ampliata per poter garantire delle funzionalità accettabili. In particolare, se si vogliono vincoli sul numero di connessioni, se si ha necessità di filtrare gli utenti che provano a connettersi, se è richiesta la condivisione di informazioni particolari sulla rete o se è preferibile avere una gestione particolare delle disconnessioni, è obbligatorio aggiungere tutte queste funzionalità tramite script.

Il maggiore vincolo imposto da UNet, che è anche uno dei motivi per cui è stato necessario riscrivere l'applicazione da zero, è tuttavia quello che va a toccare la struttura gerarchica del progetto. Unity permette di organizzare in gerarchie gli oggetti presenti nelle varie scene, garantendo così una gestione intuitiva di tutti gli elementi dell'applicazione. Un oggetto "auto",

per esempio, può essere padre di un oggetto “porta”, che a sua volta può avere come figlio un oggetto “finestrino” o un oggetto “maniglia”. Anche Blender, programma da cui si possono prendere i modelli 3D che poi sono utilizzati in Unity, permette un’organizzazione simile (anzi, spesso la richiede per preservare la sanità mentale del modellatore). Ebbene, UNet fornisce, per ogni oggetto di gioco che deve interagire sulla rete, un componente chiamato `NetworkIdentity`, che gestisce la presenza online, fornisce un ID univoco e, in generale, permette di aggiungere all’oggetto una serie di script che sfruttino le funzionalità di rete. Il problema, o almeno la limitazione di questo componente è che esso può essere agganciato solo al padre di una gerarchia. Tornando all’esempio di prima, solo l’oggetto “auto” può avere una `NetworkIdentity` e può sfruttare appieno la rete, mentre per gli oggetti “porta” o “finestrino” la gestione deve essere delegata al padre. Nel semplice scenario in cui un utente apre una porta della macchina su un PC1 e vuole che questa azione sia vista anche da un utente su un PC2, l’oggetto “porta” non può dire direttamente a “porta” sul PC2 “sono stata aperta”, ma deve prima mandare questo messaggio all’oggetto “auto” del PC1, che lo comunicherà all’oggetto “auto” del PC2 e che, finalmente, lo inoltrerà all’oggetto “porta” del PC2.

5.3 Struttura finale

Tenendo in considerazione i vincoli imposti da UNet e le funzionalità da integrare nella piattaforma (in particolare la VR immersiva, la necessità di conservare alcuni dati della simulazione anche dopo il termine della stessa, etc.) si è giunti a una struttura finale suddivisa in tre scene:

- Nella prima scena, della durata di pochi secondi, viene creato il `GameObject` responsabile della configurazione della simulazione, della gestione della componente online e della scrittura su file dei risultati della simulazione: il `GlobalController`. Questo oggetto è l’unico che vive e muore assieme all’applicazione: viene creato quando essa viene aperta e viene distrutto quando essa viene chiusa. Il `GlobalController` tiene traccia, di volta in volta, della configurazione con cui si affronta la simulazione: se si sceglie di affrontare il rischio chimico, in modalità utente singolo e impersonando il Dirty Man, tutte queste informazioni verranno salvate da questo oggetto e potranno essere recuperate, facendo riferimento a esso, in qualsiasi momento. Per assicurarsi che il `GlobalController` sia sempre unico, si è scelto di adottare il pattern Singleton.
- La seconda scena ospita il menu iniziale, in cui si scelgono i parametri (salvati nel `GlobalController`) che caratterizzano la simulazione. Il menu è accessibile solo dal desktop e non dal visore: questa scelta è stata fatta per semplificare la gestione nel contesto della Toxic Trip, soprattutto nel caso di una simulazione a quattro utenti. L’utente che indossa il visore, durante questa scena, si ritrova nello stesso scenario della simulazione e può seguire un breve tutorial per prendere confidenza con i controller. L’accesso a questa scena avviene in due casi: quando viene lanciata l’applicazione (subito dopo la creazione del `GlobalController`) e al termine di una simulazione (quando l’utente arriva alla fine o decide di interrompere l’esperienza manualmente, premendo

il tasto Esc). Nel secondo caso, invece dell'interfaccia base del menu, apparirà prima una schermata aggiuntiva che mostra i risultati ottenuti dall'utente.

- La terza scena è costituita dall'effettiva simulazione, divisa nelle sette fasi descritte nel capitolo dello storyboard. Il GameObject principale è il LocalSceneManager, che gestisce le informazioni specifiche della simulazione in corso: la progressione (fasi, passaggi di fasi e sincronizzazione online), lo status degli oggetti, la generazione e la distribuzione dei valori casuali caratteristici della simulazione corrente, gli NPC (in caso utente singolo), i risultati, la registrazione dei dati e alcuni script legati a SteamVR. Questo GameObject, come tutti gli oggetti appartenenti a questa scena, nasce e muore con la singola simulazione. Per quanto riguarda la gestione della gerarchia imposta dall'utilizzo di Unet, si è cercato dove possibile di semplificare la struttura ad albero, in modo da avere degli oggetti che potessero lavorare direttamente online. Gli oggetti appartenenti all'inventario dell'utente, per esempio, non dipendono da un oggetto "zaino", ma possiedono una NetworkIdentity. Dove questo tipo di gestione non è stato possibile (nel caso del GameObject associato all'utente, per esempio) si è optato per una gestione centralizzata, in cui l'oggetto padre ha conoscenza diretta degli oggetti figli e può comunicare con loro attraverso un sistema di eventi.

5.4 Dettaglio sulle implementazioni

Avendo ricreato la piattaforma da zero, le logiche associate ai vari oggetti di gioco sono per la maggior parte nuove e presenti solo in questa particolare applicazione. Di seguito, si descrivono nel dettaglio alcune di esse.

5.4.1 Le interazioni con lo scenario

Uno degli elementi fondamentali di una simulazione VR è costituito dalle modalità con cui l'utente può interagire con il mondo e gli oggetti di gioco. Avendo a disposizione i controller, le possibilità erano sostanzialmente due: un sistema di Ray-Casting, in cui il controller viene utilizzato come un puntatore laser, e un sistema più simile all'esperienza reale, in cui l'utente può muoversi nello scenario e toccare virtualmente gli oggetti con cui interagire. Si è scelta la seconda opzione, utilizzando l'Arm Swinger per simulare la camminata e il tracking dei controlli per permettere all'utente di raggiungere gli oggetti. I modelli standard dei controlli, inoltre, sono stati sostituiti da modelli 3D di un paio di guanti (*Figura 5.1*) per aumentare l'immersione e la familiarità dell'utente.



Figura 5.1: Modello 3D delle mani

Le interazioni presenti nella piattaforma sono molto simili tra loro, e questo ha permesso di uniformare l'interfaccia per ridurre la complessità della simulazione. In sostanza, l'utente deve prendere e spostare oggetti, oppure utilizzare oggetti specifici per eseguire un particolare compito (per esempio, una sonda per ottenere una misurazione, un pennarello per scrivere, un nastro per collegare tra loro i paletti, un cartellino triangolare per contrassegnare un paletto). Questo ha permesso di creare un'interfaccia comune che si basa sullo spostamento del controller nella giusta posizione e sull'utilizzo del Trigger Button (presente sia sul Vive che sull'Oculus). Per prendere un oggetto l'utente avvicina la mano virtuale e preme il pulsante, per rilasciarlo preme nuovamente lo stesso tasto, per posizionare un cartello è necessario avvicinarlo a un paletto e premere il trigger, per collegare il nastro a un paletto si fa la stessa azione. Per interagire con le interfacce presenti nella simulazione, infine, basta avvicinare la mano al pannello (per selezionare un tasto, ad esempio) e cliccare il trigger.

Il meccanismo che permette di riconoscere gli oggetti con cui l'utente vuole interagire è basato su un sistema di collider e di flag, in aggiunta all'utilizzo di colori che permettono di identificare l'oggetto o l'interfaccia che si sta cercando di selezionare. In questo modo si evita che gli oggetti eseguano dei movimenti strani, passando di mano senza cliccare nulla per esempio, e che un utente prenda per sbaglio un GameObject che non aveva intenzione di utilizzare.

5.4.2 L'interfaccia utente

Un altro degli elementi fondamentali di un'applicazione VR è la gestione dell'interfaccia utente. Nelle applicazioni desktop, le interfacce sono spesso fisse e posizionate sui bordi dello schermo; una soluzione del genere è impensabile in VR, sia perché scritte che seguono il visore limitano eccessivamente il field of view, sia perché delle icone sempre presenti davanti agli

occhi dell'utente possono causare distrazioni, fastidio e, in generale, possono diminuire l'immersione e il senso di presenza. Una soluzione consiste nell'inserire delle interfacce che siano fisse nel mondo di gioco o che seguano le mani invece della testa.

In linea di massima, interfacce aggiuntive che rischiano di minare l'intera esperienza devono essere ridotte al minimo e utilizzate solo quando è necessario. Nella piattaforma sviluppata, il tipo di interfaccia che si è scelto di includere consiste in un pannello che presenta una serie di opzioni tra cui l'utente può scegliere. È utilizzata, in particolare, per l'inventario e per simulare la scrittura.

5.4.3 Tool aggiuntivi

Per via delle caratteristiche della procedura, è stato necessario aggiungere alla piattaforma dei tool aggiuntivi che simulano, invece di replicare, alcune azioni e alcuni concetti presenti nel mondo reale. In particolare, sono stati utilizzati due strumenti che approssimano l'inventario e la scrittura su paletto:

- il tool che gestisce l'inventario (*Figura 5.2*) è necessario per un semplice fatto: l'operatore CBRN porta con sé i vari oggetti che può utilizzare e li colloca attorno alla vita o sulla schiena. Una configurazione del genere, per un utente che indossa un visore, può essere problematica, così si è deciso di aggiungere uno strumento che simuli quest'esperienza. L'inventario è gestito tramite un pannello, apribile utilizzando il touchpad del controller del Vive o la levetta analogica del controller dell'Oculus, che presenta all'utente una serie di strumenti che può prendere come qualsiasi altro oggetto della simulazione: avvicinando la mano e utilizzando il trigger. Una prima versione di questo tool era priva di interfaccia grafica e veniva gestita premendo ripetutamente il Touchpad (o la levetta), facendo così scorrere i vari oggetti fino ad arrivare a quello desiderato. Questa soluzione è stata scartata perché l'assenza di un supporto grafico non permetteva all'utente di avere una piena consapevolezza degli strumenti a sua disposizione.
- Per quanto riguarda la scrittura, la scelta di non replicarla fedelmente è frutto di una considerazione: ciò che viene valutato nella procedura CBRN sono i contenuti, non la calligrafia; era quindi superfluo aggiungere degli algoritmi di Computer Vision per il riconoscimento della scrittura, soprattutto considerando il tempo limitato che era disponibile prima della Toxic Trip 2019. Il tool scelto per simulare la scrittura consiste in un pannello che presenta all'utente una serie di opzioni tra cui scegliere, alcune delle quali sbagliate. L'utente, utilizzando un pennarello virtuale presente nell'inventario, può selezionare queste opzioni e trasportarle sul cartello (utilizzando, come sempre, il Trigger del controller).

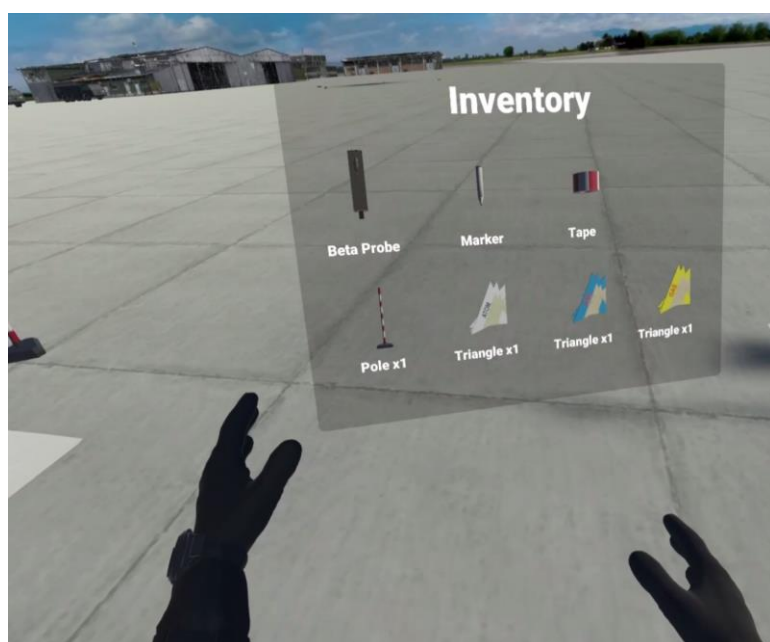


Figura 5.2: Inventario

5.4.4 La modalità multi-utente

Per la modalità multi-utente si è scelto di creare una struttura in cui il server (nel caso particolare, un host) gestisce le autorizzazioni e la sincronizzazione della maggior parte dei dati, mentre il client ha la possibilità di effettuare dei calcoli in locale per ridurre il traffico trasmesso in rete. Per fare un esempio, un utente sul client che vuole sollevare un oggetto ha bisogno dell'autorizzazione del server. Questa autorizzazione è necessaria per impedire che, per un problema di rete, ci si ritrovi in situazioni corrotte in cui un oggetto risulta essere preso da un utente su un certo client, e da un altro su un client diverso. È tuttavia inutile provare a richiedere l'autorizzazione, mandando dati sulla rete, se già in locale non ci sono le condizioni adatte perché essa venga garantita: una serie di condizioni vengono verificate in locale per scartare a priori le richieste che sicuramente non andranno a buon fine.

La struttura client-server adottata fa uso di soli quattro pc, infatti non viene utilizzato un server dedicato, ma uno degli utenti della simulazione multi-utente avrà il ruolo sia di server che di client (sarà quindi un host). Questa scelta è stata fatta per uniformare i due tipi di esperienza, almeno dal punto di vista tecnico: chi affronta la simulazione da solo, infatti, non è altro che un host senza altri client attivi.

Per quanto riguarda la gestione delle connessioni e delle disconnessioni, ogni utente che prova a connettersi a un host invia un messaggio con delle informazioni prese dal proprio GlobalController (rischio affrontato e ruolo scelto). Se queste informazioni sono coerenti con quelle scelte dall'host, quindi se il rischio scelto da entrambe le parti corrisponde, e se il ruolo scelto è ancora disponibile, allora la connessione viene accettata. In caso contrario, l'utente viene rifiutato e torna alla schermata del menu. Per quanto riguarda le disconnessioni, invece,

quando un utente si disconnette (volontariamente o meno) un meccanismo gestisce in locale l'uscita dell'applicazione, garantendo così che i risultati siano calcolati e che i dati raccolti durante la simulazione siano salvati. Gli altri client ancora online, poiché la configurazione a quattro utenti è, per ora, l'unica prevista, in corrispondenza della disconnessione faranno partire a loro volta il meccanismo che permette di calcolare i risultati e salvare i dati, tornando così al menu.

Per finire, data la somiglianza tra i vari ruoli, sia in termini di aspetto che in termini di funzionalità generali, si è deciso di realizzare un solo tipo di GameObject da aggiungere alla scena nel momento della connessione di un utente: questo oggetto racchiude in sé le logiche che permettono di sollevare oggetti, di interagire con le interfacce, di registrare dati, di decontaminare o essere decontaminato e di utilizzare l'inventario. Il ruolo specifico dell'utente viene poi impostato cambiando il valore di alcuni parametri interni a questi script, mentre le funzionalità specifiche che variano da ruolo a ruolo sono delegate agli oggetti presenti nell'inventario.

5.4.5 Gli oggetti dell'inventario

Particolare cura è stata dedicata alla realizzazione degli strumenti presenti nell'inventario, data la loro influenza sulla percezione generale della simulazione. Nell'applicazione precedente la sonda utilizzabile non funzionava in modo realistico, e ciò aveva provocato dei feedback negativi. Gli oggetti presenti nell'inventario sono diversi e cambiano a seconda del ruolo scelto.

- La sonda alfa è utilizzata per rilevare la presenza di particelle alfa. Mostra una lettura solo se molto vicina alle particelle in questione e dà un output nullo altrimenti. Dei controlli di distanza permettono di valutare se mostrare o meno una lettura positiva, mentre il valore letto dipende dalla distanza tra la sonda e l'hotspot (punto con il maggiore livello di contaminazione). L'utilizzo di valori casuali permette di variare, tra simulazioni diverse, il valore minimo letto e la distanza a cui si inizia ad avere un output non nullo.
- La sonda beta è utilizzata per rilevare le particelle beta, lo script che la controlla è una variante di quello utilizzato per la sonda alfa. Cambiano i valori delle distanze, il range di valori letti e l'unità di misura.
- Il CAM è utilizzato per rilevare la presenza di agente chimico. Anche in questo caso lo script è una variante di quello dell'alfa, ma le modifiche sono più evidenti. L'output non è più continuo, ma discreto (in accordo con il funzionamento reale della modalità implementata) e può mostrare quattro possibili valori che vanno da zero al valore massimo. Questi valori, inoltre, non dipendono più dalla distanza dal singolo hotspot, ma tengono in considerazione altri due punti che sono orientati secondo la direzione del vento (poiché l'agente chimico è volatile e viene facilmente trasportato dalle correnti d'aria): in base alla posizione dello strumento rispetto all'insieme di questi tre punti si otterrà un certo output.

- La cartina tornasole è utilizzata insieme al CAM. Anch'essa utilizza dei controlli basati sulla distanza e, in prossimità dell'hotspot, cambia colore passando dal bianco al rosso mattone. L'effetto è ottenuto interpolando due materiali diversi in base al tempo in cui la cartina si trova in prossimità della zona contaminata.
- Il pennarello, come già detto, serve per prelevare un'informazione e posizionarla su un cartello. Ha un meccanismo simile a quello utilizzato sui modelli delle mani per permettere di sollevare gli oggetti.
- Il nastro è utilizzato per delimitare l'area contaminata. Serve anche per confermare in maniera definitiva la posizione dei paletti posizionati attorno all'hotspot. Il meccanismo di base è simile a quello utilizzato per sollevare gli oggetti: avvicinando il nastro a un paletto e usando il trigger si richiede l'autorizzazione ad agire su di esso. Se si ottiene, viene creata una coppia di oggetti che servono come punti di partenza per il nastro (che viene disegnato grazie a una coppia di Line Renderer). Premendo per qualche secondo il trigger, con il nastro a mezz'aria, si annulla l'azione e si distruggono gli oggetti creati. Premendo il trigger con il nastro vicino a un altro paletto, se si ottiene l'autorizzazione dal server, si chiude il segmento di nastro.
- La PSDS 1.5 è una pompa portatile utilizzata dal Clean Man per decontaminare gli altri operatori. Utilizzando il trigger si fa partire un particellare che simula l'emissione del liquido nebulizzato. Un controllo sul tempo evita che, tenendo premuto il trigger, esca un getto continuo e obbliga l'utente a premere più volte il tasto (per simulare il comportamento reale). La decontaminazione attraverso questo strumento (quindi nel caso utente singolo se si interpreta il Clean Man, o nel caso multi-utente) avviene grazie a degli script posizionati sugli oggetti da decontaminare (le mani virtuali degli utenti, per esempio) che contano le collisioni con le particelle. In modalità multi-utente, questo calcolo viene effettuato solo sul server per evitare incongruenze, e il risultato della decontaminazione viene inviato in un secondo momento ai client.

5.4.6 Personaggi non giocabili

Gli NPC (*Figura 5.3*) della simulazione sono presenti solo nella modalità a un solo utente e vengono attivati in base al ruolo interpretato. Se l'utente interpreta il Clean Man, per esempio, saranno attivi gli NPC del Team Leader, del Data Compiler e del Dirty Man. Sono, in sostanza, delle macchine a stati (o una coppia di macchine a stati, in alcuni casi) che reagiscono agli input dell'utente. Sono quindi completamente programmati.

- Il Team Leader è guidato da uno script che gli impone di seguire un GameObject presente nella scena. A seconda del ruolo scelto, il GameObject può essere l'utente o un altro NPC.
- Il Data Compiler, in modo simile al Team Leader, si limita a seguire un GameObject nella scena, ma a intervalli regolari si ferma per effettuare una misurazione (leggasi, si ferma per far partire un'animazione). A seconda del ruolo può seguire l'utente o un altro NPC.

- Il Dirty Man è caratterizzato da due modalità differenti: se l'utente interpreta il Data Compiler, questo NPC si comporta come quelli descritti finora, seguendo però un punto posizionato davanti al Team Leader (e, di conseguenza, davanti all'utente). Se invece si sta interpretando il ruolo del Clean Man, il Dirty Man è controllato da una macchina a stati che, una volta completata la Working Area, lo fa andare prima verso l'hotspot (effettuando misurazioni a intervalli regolari), poi verso la Working Area per recuperare altri paletti e altri cartelli, e infine verso dei punti prestabiliti che corrispondono ai paletti che dovranno delimitare l'area contaminata.
- Il Clean Man è caratterizzato da una macchina a stati che prima lo guida fino al limite della Working Area per offrire i paletti e i cartelli all'utente, poi lo fa mettere in posizione per decontaminare l'utente. La fase di decontaminazione (passiva, in questo caso, perché l'utente si fa decontaminare) è gestita attraverso dei collider e un sistema a eventi. Se l'utente posiziona la mano davanti alla PSDS del Clean Man (NPC), un evento dice alla macchina a stati di far partire l'animazione di decontaminazione. A seconda della mano posizionata (destra o sinistra) dell'orientamento della stessa (si offre il palmo o il dorso) e dell'ordine in cui l'utente posiziona le mani, inoltre, vengono modificati i valori di alcuni flag per poter calcolare, in seguito, i risultati.

Se l'utente interpreta il Clean Man, infine, sugli NPC presenti nella scena entra in gioco un altro script, cioè una macchina a stati che gestisce la decontaminazione (attiva, perché l'utente la esegue) degli altri operatori. Questa macchina fa avvicinare gli NPC alla Working Area (uno per volta) e attiva in sequenza delle animazioni: “porgere il dorso della mano sinistra”, “porgere il dorso della mano destra” e così via. Utilizzando la PSDS, l'utente fa collidere delle particelle con le varie parti dell'NPC, attivando così altre animazioni e progredendo nella procedura.



Figura 5.3: Clean Man e Data Compiler, NPC

5.4.7 Il sistema di valutazione

Una procedura dettagliata come quella simulata in questa piattaforma può essere valutata sotto molti aspetti. A seconda del ruolo interpretato, inoltre, i punti da considerare variano, facendo aumentare di conseguenza tutti i dettagli che devono essere tracciati all'interno della simulazione.

Il sistema di valutazione integrato nella simulazione tiene traccia delle azioni dell'utente utilizzando dei *flag* (per gli elementi che devono essere valutati alla fine della simulazione e il cui status può cambiare rapidamente) e degli eventi (per quelle azioni che vengono valutate istantaneamente e a cui non si può rimediare). L'unione di tutti questi parametri ha dato vita allo script più massiccio della simulazione, che ha il compito di lanciare una funzione di valutazione una volta che l'esperienza si è conclusa.

Tra gli elementi considerati per la valutazione rientrano:

- direzione di approccio iniziale (si guarda se l'utente ha provato a posizionare la Working Area nella posizione sbagliata);
- misurazione attorno al veicolo (si valuta se il Dirty Man ha controllato in modo adeguato il veicolo e la zona circostante tramite un sistema di collider);
- posizionamento degli oggetti nella Working Area (sono considerati errori sia il posizionamento errato dei paletti che identificano l'Entry Point e l'Exit Point, valutato tramite un'etichetta associata al palletto, sia il posizionamento a diretto contatto con il terreno degli altri oggetti della Working Area, valutato tramite collider);
- correttezza della misurazione (si contano le volte in cui le sonde sono state utilizzate in modo corretto e si confronta il valore ottenuto con una soglia);
- posizionamento dei paletti (si controllano le posizioni di tutti i paletti utilizzati per delimitare l'area contaminata e si guardano le distanze rispetto all'hotspot);
- correttezza delle informazioni scritte sui cartelli (si effettua una elaborazione del contenuto delle scritte posizionate sui cartelli e si controlla se sono presenti tutte le informazioni attese);
- correttezza della procedura di decontaminazione (si controlla se sono stati decontaminati tutti i GameObject previsti e se è stato seguito l'ordine atteso);
- confini della Working Area (si valuta se il Clean Man è rimasto nella Working Area, o se gli altri operatori sono entrati nella Working Area senza essere decontaminati).

5.4.8 La registrazione dei dati

La registrazione dei dati è possibile grazie all'utilizzo dell'asset Record and Play [17] presente nell'asset store di Unity. Questo strumento mette a disposizione una serie di GameObject e

script che permettono di registrare posizioni e rotazioni di qualsiasi oggetto, più eventi personalizzati che devono essere tracciati manualmente dal programmatore.

L'oggetto base fornito dall'asset è il Recorder: un contenitore a cui si possono associare degli oggetti e che racchiude dentro sé i dati raccolti durante la simulazione. Per poter salvare effettivamente i dati, è essenziale chiudere il Recorder e trasferire tutto su file, altrimenti le informazioni raccolte vengono buttate via. L'asset fornisce solo questi elementi base e, in modo simile a quanto successo con Unet, è stato necessario un lavoro di integrazione e ampliamento per poter ottenere i risultati desiderati. In particolare, sono stati creati:

- uno script che possa tenere traccia dei gruppi di oggetti associati a un singolo Recorder (la simulazione ne utilizza molteplici, per raggruppare oggetti simili e dividerli dagli altri);
- uno script che si appoggia al primo e permette di registrare gli oggetti creati durante la simulazione (quindi non presenti all'inizio dell'esperienza);
- una serie di script specifici per i diversi oggetti (paletti, oggetti dell'inventario, mani virtuali dei giocatori, etc.) che salvano eventi creati per l'occasione e che permetteranno in un secondo momento di riprodurre l'esperienza.

I dati raccolti in questo modo sono, quindi, salvati in cartelle diverse (una per Recorder) in un doppio formato: un JSON (*Figura 5.4*), per favorirne la lettura e l'analisi, e un file che può essere utilizzato dall'asset per una riproduzione veloce all'interno di Unity.

```
1
2
3 { "Name": "", "Duration": 958.748, "Metadata": {}, "CustomEvents": [], "Subjects": [{ "ID": -4888
4 { "Time": 2009.271, "X": -83.96344, "Y": 0.5675499, "Z": 5.001966 }, { "Time": 2009.381, "X": -83
5 { "Time": 2007.248, "X": 0.005266227, "Y": 292.4236, "Z": -0.001547431 }, { "Time": 2007.348, "X"
6 { "Time": 2055.495, "X": -86.45055, "Y": 0.5276768, "Z": 0.03842707 }, { "Time": 2055.605, "X": -
7 { "Time": 2108.353, "X": 277.2457, "Y": 114.4246, "Z": 194.7536 }, { "Time": 2108.464, "X": 277.1
8 { "Time": 2045.594, "X": -84.37198, "Y": 0.5214441, "Z": 5.139784 }, { "Time": 2045.705, "X": -84
9 { "Time": 2085.118, "X": 0.003412312, "Y": 277.3323, "Z": -0.002494955 }, { "Time": 2085.229, "X"
10 { "Time": 2490.646, "X": 0.003235842, "Y": 277.3314, "Z": -0.002448155 }, { "Time": 2490.858, "X"
11 { "Time": 2126.353, "X": -115.1517, "Y": 0.761527, "Z": 2.116285 }, { "Time": 2126.464, "X": -115
12 { "Time": 2127.731, "X": -115.1519, "Y": 0.4729514, "Z": 2.117001 }, { "Time": 2190.478, "X": -11
13 { "ID": -6068, "Name": "SpawnedPalettoGas (Clone) 2", "Metadata": {}, "Duration": 566.0979, "Lif
14 { "Time": 2166.854, "X": 0.003452288, "Y": 309.6757, "Z": -0.002808683 }, { "Time": 2166.954, "X"
15 { "Time": 2183.6, "X": 0.003436219, "Y": 309.6694, "Z": -0.002784301 }, { "Time": 2186.022, "X":
16 { "Time": 2250.202, "X": 0.003563613, "Y": 59.69339, "Z": -0.002735914 }, { "Time": 2250.313, "X"
17 "CustomEvents": [{ "Time": 2549.593, "Name": "Grabbed", "Contents": "GrabbedBy DataCompiler Righ
18 "Rotations": [{ "Time": 2190.536, "X": 0.05250121, "Y": 94.44659, "Z": 359.9587 }, { "Time": 2190
19 { "Time": 2332.205, "X": 0.003298009, "Y": 94.4072, "Z": -0.002496496 }, { "Time": 2383.597, "X":
20 { "Time": 2237.19, "X": -101.2561, "Y": 0.4731225, "Z": 21.38182 }, { "Time": 2246.756, "X": -101
21
```

Figura 5.4: Esempio di JSON con i dati raccolti

La struttura del JSON è organizzata nel modo seguente:

1. Recorder
 1. Name
 2. Duration
 3. Metadata
 4. CustomEvents
 5. Subjects
 1. ID
 2. Name
 3. Metadata
 4. Duration
 5. LifeCycleEvents
 1. Time
 2. Type
 6. CustomEvents
 1. Time
 2. Name
 3. Contents
 7. Positions
 1. Time
 2. X
 3. Y
 4. Z
 8. Rotations
 1. Time
 2. X
 3. Y
 4. Z

Per ogni Recorder viene salvato il nome, la durata, i metadati e i CustomEvents. Questi campi, fatta eccezione per la durata, non sono stati utilizzati nella simulazione perché si è preferito spostare l'analisi sui singoli oggetti registrati. Per ogni Recorder viene quindi salvata una lista di Subject, cioè gli effettivi attori della registrazione. Ognuno di questi Subject è caratterizzato da un ID, da un nome (corrispondente al nome con cui l'oggetto è conosciuto in Unity), da alcuni metadati, dalla durata del ciclo vitale dell'oggetto e da quattro liste.

- La prima tiene traccia degli eventi che caratterizzano la presenza dell'oggetto nella scena. Questi eventi sono standard e rappresentano alcuni momenti chiave della vita del Subject, quali la creazione, la distruzione, l'attivazione e la disattivazione. Ognuno di questi eventi è salvato grazie a un campo Time, che indica il momento in cui è avvenuto, e Type, che ne distingue il tipo.
- La seconda lista rappresenta un elenco degli eventi personalizzati associati al Subject. Questi eventi sono stati creati per quest'applicazione e rappresentano le interazioni dell'oggetto con l'utente (per esempio: quando un oggetto viene sollevato o fatto cadere, quando il pennarello viene usato per scrivere un'informazione, etc.) Ogni elemento della lista è registrato tramite un campo Time, che rappresenta anche qui il momento in cui l'evento è avvenuto, un campo Name e un campo Contents, che contiene informazioni aggiuntive sull'interazione salvata.
- La terza e la quarta lista rappresentano, rispettivamente, la sequenza di posizioni e la sequenza di rotazioni assunte dal Subject nel tempo. Per entrambe le liste, ogni singolo elemento è caratterizzato da Time, cioè il tempo di campionamento, e dai valori assunti sugli assi X, Y e Z.

Capitolo 6

L'esperienza alla Toxic Trip 2019

Come già anticipato, lo scopo principale della tesi consiste nel produrre una piattaforma VR per l'addestramento CBRN, e la Toxic Trip 2019 ha offerto l'opportunità di sperimentarla sul campo [18].

6.1 La preparazione

Fin dalle fasi iniziali della tesi si è deciso di organizzare uno stand in cui far provare la piattaforma VR. Una delle prime questioni da decidere era il tipo di esperienza da portare a Villafranca: organizzare prove multi-utente, infatti, poteva risultare problematico in un contesto simile a quello di una fiera, poiché molte delle persone presenti alla base avrebbero avuto poco tempo a disposizione (in particolare, i militari potevano essere richiamati per un'esercitazione reale da un momento all'altro). Riuscire a riunire ogni volta un team di quattro persone, inoltre, poteva essere difficile, considerando il numero limitato di persone di supporto a nostra disposizione. Organizzare solo prove in modalità utente singolo, invece, rischiava di non rendere giustizia al lavoro svolto. Alla fine si è deciso di adottare una soluzione intermedia: organizzare prove dell'ultima tipologia, con la volontà di allestire almeno una dimostrazione multi-utente con gli utenti finali della piattaforma, gli operatori CBRN del Recce Team, documentando tutta l'esperienza con filmati e catture della simulazione.

Questa scelta ha portato dei vincoli dal punto di vista tecnico: una prova multi-utente comportava, stando allo storyboard, la presenza di quattro postazioni VR-Ready collegate localmente su una LAN Ethernet, ed era quindi necessario trovare almeno quattro PC e quattro visori. Attingendo dalle risorse delle tre parti coinvolte nella collaborazione, si è arrivati alla lista finale:

- due PC, due Vive e un Oculus forniti dalla Fondazione LINKS;
- due PC, un Oculus, due controller del Vive di backup e una telecamera forniti dal Politecnico di Torino;
- un PC, un Oculus, cinque schermi, quattro tastiere e quattro mouse forniti dall'Aeronautica.

La lista finale includeva, quindi, anche un visore e una postazione di backup, per evitare possibili problemi.

6.2 La logistica

Dal punto di vista logistico, è stata messa a disposizione una tenda composta da quattro “stanze” destinate a ospitare i quattro PC. Questa tenda, soprannominata per comodità “VR Lab”, è stata collocata al centro della zona attiva dell’evento, a metà strada tra le esposizioni delle aziende e i luoghi di ristoro, per garantire una buona affluenza. Per quanto riguarda il personale, era necessario avere, per tutta la durata della Toxic Trip, quattro persone sempre presenti nello stand, in modo da poter gestire quattro simulazioni in contemporanea in ogni momento, più (eventualmente) qualcuno all’esterno che si occupasse dell’accoglienza. Alla fine è stato possibile garantire la presenza costante di almeno quattro persone per tutti e cinque i giorni dell’evento: sfruttando il contributo di membri della Fondazione LINKS e dell’Aeronautica, infatti, tutte e quattro le postazioni sono rimaste sempre attive e utilizzabili.

6.3 L’evento

La Toxic Trip 2019 è durata cinque giorni.

- Sabato 28 settembre si è aperto con un briefing generale a cui hanno presenziato anche il prof. Fabrizio Lamberti del politecnico di Torino e l’ing. Massimo Migliorini della Fondazione LINKS. Per l’occasione, è stata preparata una presentazione per illustrare a tutti i partecipanti la piattaforma VR realizzata. La seconda metà della giornata è stata dedicata principalmente al montaggio dello stand, ma non sono mancate anche delle prove dell’applicazione.
- Domenica 29 e lunedì 30 settembre sono stati dedicati unicamente ai militari: operatori italiani e di altri paesi (Belgio, Francia, Danimarca, Germania, Paesi Bassi, Nuova Zelanda, Turchia, Stati Uniti) sono passati dallo stand per provare l’applicazione. Durante il primo dei due giorni l’affluenza è stata molto buona, anche se leggermente incostante: in alcuni momenti era presente un solo militare, in altri un gruppo formato da più di quattro persone; si è deciso quindi di definire degli slot di mezz’ora in cui fino a quattro persone avrebbero potuto prenotarsi. Grazie a questa soluzione, i giorni successivi sono risultati più gestibili.
- Martedì 1 ottobre è stato caratterizzato dal Vip e Media Day, cioè un’occasione dedicata ad alte cariche dell’Aeronautica e al personale della stampa. C’è stata comunque una buona affluenza di militari intenzionati a provare l’applicazione, ma uno dei quattro PC disponibili è stato tenuto costantemente occupato in vista di possibili “ospiti d’onore”. A metà giornata, un gruppo che includeva alcune di queste alte cariche ha fatto visita allo stand, seguito dalla stampa, e ha assistito a una dimostrazione dell’applicazione.
- Mercoledì 2 ottobre, l’ultimo giorno dell’evento, è stato infine dedicato ai militari (*Figura 6.1*) e a rappresentanti delle aziende che espongono nell’area dedicata alle industrie. Sono state eseguite varie prove ed è stato possibile organizzare

un'esercitazione virtuale multi-utente con i membri di un Recce Team dell'Aeronautica.



Figura 6.1 Un militare che prova la simulazione in una delle postazioni

Capitolo 7

Analisi sperimentale

7.1 Il questionario e i risultati

Ad una parte degli utenti che hanno provato l'applicazione (30 persone in tutto) è stato fatto compilare un questionario (Appendice A) per ottenere una valutazione soggettiva della piattaforma sviluppata. Il questionario è diviso in sei parti:

- la prima è costituita da gruppo di domande, a carattere generale, riguardanti l'utente ("age", "gender", "nationality");
- la seconda racchiude alcune informazioni tecniche sulla simulazione appena effettuata ed è costituita dal tipo di visore utilizzato, dallo status della simulazione (se è stata completata o meno) e da un ID che permette di associare il questionario ai dati registrati dall'applicazione;
- la terza ha lo scopo di tenere traccia dell'eventuale *motion sickness* causata dall'utilizzo della piattaforma;
- la quarta serve per raccogliere alcune informazioni sul background dell'utente (esperienza con la VR, con i videogiochi e con i serious game, conoscenza delle procedure CBRN e, in particolare, di quelle italiane);
- la quinta è volta a ottenere feedback sull'interfaccia dell'applicazione, ed è costituita dalle domande generali presenti nello strumento VRUSE [19];
- la sesta è composta da tre domande aperte in cui si dà all'utente la possibilità di elencare gli aspetti positivi e negativi della piattaforma e di aggiungere eventuali commenti o suggerimenti.

Dei 30 questionari, 26 sono stati sottoposti a utenti che hanno provato la simulazione in modalità utente singolo, interpretando il Dirty Man o il Data Compiler. Gli altri 4 questionari riguardano invece una prova multi-utente (eseguita da 4 operatori CBRN membri del Recce Team).

7.1.1 Gli utenti

Dei 30 utenti che hanno risposto al questionario, 26 sono maschi e 4 sono femmine (*Figura 7.1*). Per quanto riguarda la provenienza (*Figura 7.2*), 23 utenti sono italiani e 7 stranieri (6 dalla Francia, 1 dai Paesi Bassi). Due terzi degli utenti, infine, hanno affermato di avere una

buona conoscenza delle procedure CBRN. L'intervallo di età va dai 20 anni dell'utente più giovane ai 65 di quello più anziano.

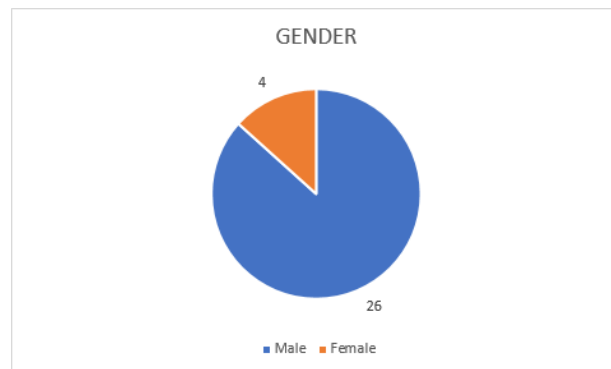


Figura 7.1: Genere dei partecipanti

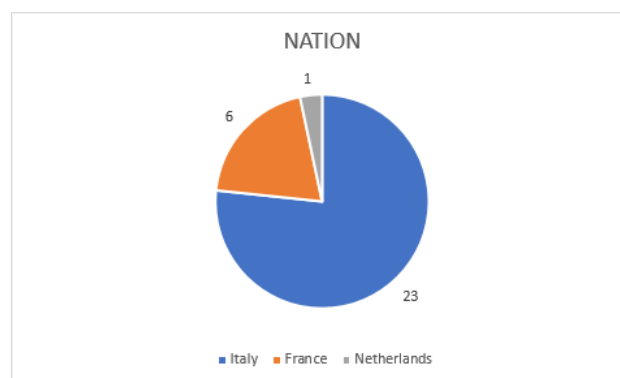


Figura 7.2: Nazionalità dei partecipanti

7.1.2 Motion sickness

Un utente che interagisce con un ambiente virtuale riceve una serie di stimoli (video, audio, tattili, etc.) che servono a simulare la realtà e ad aumentare il senso di presenza. Se questi stimoli non sono accurati o, nel caso estremo, entrano in contrasto con quello che l'utente si aspetta di provare, rischiano di causare una serie di sintomi fisici (quali nausea, affaticamento, disorientamento, fastidio agli occhi) che possono essere più o meno accentuati a seconda dell'utente.

Questi sintomi, che rientrano nel termine generale di motion sickness, sono spesso causati da una cattiva progettazione e dalle numerose astrazioni che sono necessarie per poter inserire l'utente all'interno della simulazione. Interfacce che seguono la testa dell'utilizzatore possono distrarre l'utente, la risoluzione del visore può stancare gli occhi, la resa della profondità nel mondo virtuale può causare nausea, il metodo di locomozione può causare cinetosi, e così via.

Il passaggio mondo reale – mondo virtuale – mondo reale, in pratica, espone l’utente a possibili malesseri [20].

Nel questionario utilizzato alla Toxic Trip è presente una domanda (la numero 1) in cui si chiede all’utente di valutare, su una scala che va da 1 (nessun sintomo) a 5 (elevata motion sickness), la propria condizione fisica prima e dopo l’utilizzo della piattaforma. I risultati finali sono incoraggianti: su 30 utenti (che prima della simulazione non manifestavano alcun sintomo), 3 hanno manifestato un leggero affaticamento, mentre solo 1 ha interrotto la simulazione. In quest’ultimo caso, l’utente non è riuscito a indossare gli occhiali da vista assieme al visore e ha manifestato, di conseguenza, affaticamento agli occhi (legata in particolare alla visione periferica). Su 30 utenti, il livello di motion sickness dopo la simulazione ha raggiunto una media di 1.183 (su una scala da 1 a 5), con deviazione standard $\sigma = 0.4997$.

7.1.3 Background

Le domande riguardanti il background (da 2 a 6) prevedono l’assegnazione di un valore da 1 (“not familiar”) a 5 (“extremely familiar”) a diversi argomenti (VR, videogame, serious game, procedure CBRN, procedure CBRN italiane). Per quanto riguarda la VR, solo 2 utenti hanno affermato di avere esperienza nell’ambito (assegnando 4, “moderately familiar”, alla voce corrispondente), mentre gli altri 28 non hanno familiarità o ne hanno poca (*Figura 7.3*).

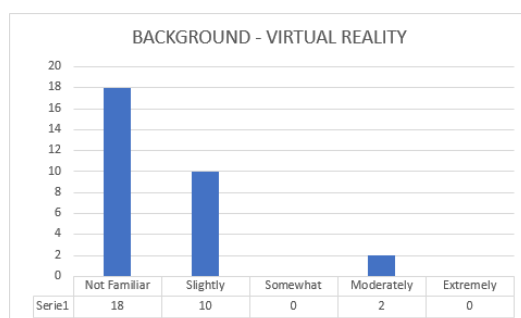


Figura 7.3: Background VR

Per quanto riguarda altri tipi di applicazione, come atteso, la familiarità nei confronti dei videogiochi (*Figura 7.4*) con scopo di intrattenimento (più diffusi commercialmente) è generalmente più elevata rispetto a quella nei confronti dei serious games (*Figura 7.5*). In questo secondo caso ci sono alcuni picchi (valutazione 4 o 5) legati all’utilizzo di simulatori di volo o di guida.

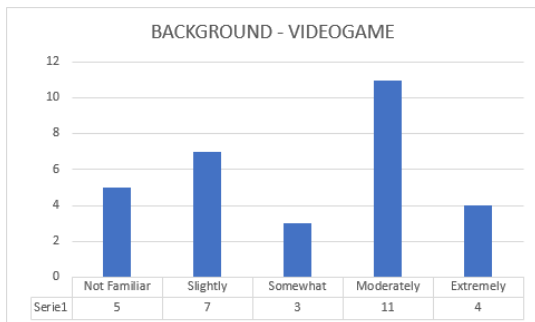


Figura 7.4: Background Videogame

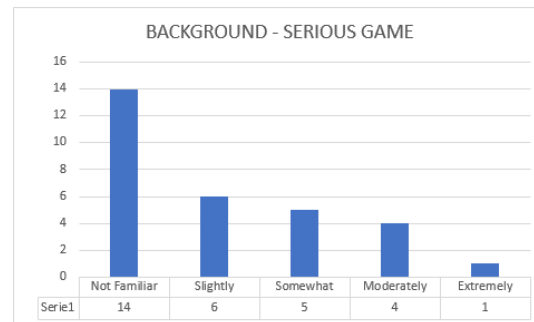


Figura 7.5: Background Serious Game

Venendo alle procedure CBRN, si è distinta la conoscenza delle procedure italiane da quella delle procedure CBRN in generale (quindi lo standard NATO). Gli utenti che non hanno conoscenze in questo campo hanno assegnato un valore basso a entrambi i campi. Gli utenti italiani del settore (che hanno dato almeno 4 alla domanda sulla CBRN italiana) hanno assegnato lo stesso valore o si sono tenuti leggermente più bassi sulla domanda generale (Figure 7.6 e 7.7), segno di una possibile differenza tra alcuni aspetti delle due categorie di procedure (nel dettaglio, le operazioni CBRN seguono le direttive NATO, ma la strumentazione e alcuni passaggi sono leggermente diverse da paese a paese). Questo andamento è molto più marcato negli utenti stranieri, che hanno dato un valore più alto alla domanda generale e si sono tenuti molto bassi su quella specifica per le procedure italiane (Figure 7.8 e 7.9).

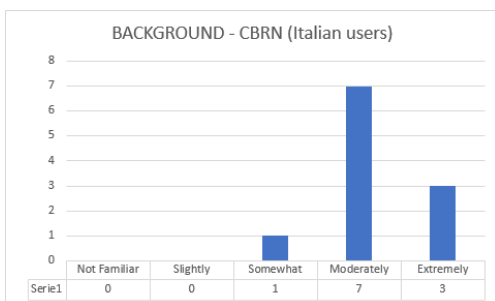


Figura 7.6: CBRN (utenti italiani)

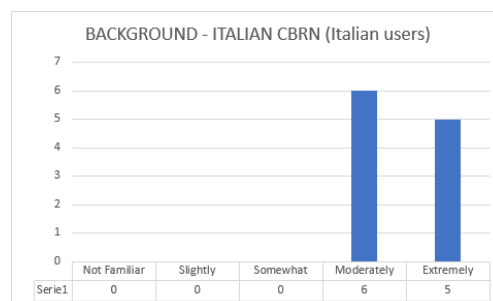


Figura 7.7: CBRN Italiana (utenti italiani)

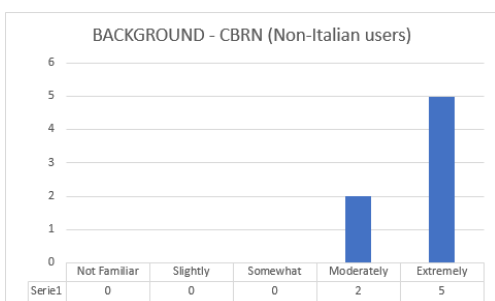


Figura 7.8: CBRN (utenti non italiani)

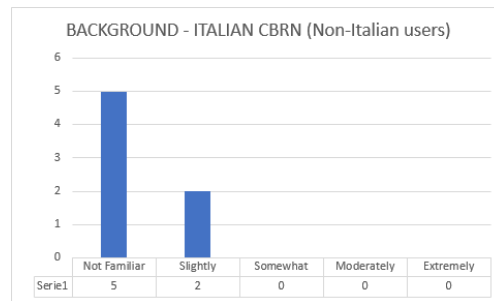


Figura 7.9: CBRN Italiana (utenti non italiani)

Andando ad analizzare la domanda sulla fedeltà della simulazione (domanda 12) per gli utenti italiani del settore, si ottiene un valore medio di 4.181 con deviazione standard $\sigma = 0.7507$, segno che in generale la fedeltà della simulazione, al netto delle semplificazioni effettuate per l'occasione, è più che soddisfacente e quindi attendibile. Per gli utenti stranieri si nota che il valore medio è alto (4.714) con deviazione standard $\sigma = 0.4879$, a conferma che, nonostante i dubbi iniziali (espressi per lo più dagli utenti di altri paesi) le procedure italiane sono conformi agli standard.

7.1.4 Input e output

I sistemi di input (domanda 8) hanno ottenuto generalmente dei punteggi sopra al soddisfacente, indipendentemente dal ruolo interpretato (in particolare: media di 4.7 e deviazione standard $\sigma = 0.4830$ per gli utenti che hanno interpretato il ruolo del Dirty Man e media di 4.437 e deviazione standard $\sigma = 0.5123$ per i Data Compiler). Il risultato era atteso, poiché non ci sono differenze nei dispositivi di input tra i due ruoli.

Per quanto riguarda l'output (domanda 9), quindi il giudizio sulla qualità grafica generale e sulla risoluzione del visore, i punteggi sono leggermente inferiori ma comunque sopra al 4 (soddisfacente). La media per il Dirty Man è di 4.2 ($\sigma = 0.7888$) e di 4.4375 ($\sigma = 0.7274$) per il Data Compiler. Anche qui, non essendoci differenze grafiche tra i due ruoli i due risultati sono comparabili.

7.1.5 Interfaccia

Le domande sull'interfaccia della simulazione (7, 10, 11 e 13) servono tutte a dare un giudizio sull'interfaccia del sistema. In particolare:

- “functionality” misura il livello di controllo offerto all'utente, in relazione al compito da svolgere. In particolare, serve anche a stabilire se l'interfaccia è intuitiva o ambigua; per il Dirty Man il valore medio è di 4.7 ($\sigma = 0.4830$), per il Data Compiler è di 4.437 ($\sigma = 0.5123$);
- “consistency” indica la coerenza dell'interfaccia, e quindi il fatto che essa si comporti come stabilito; la media dei valori forniti dagli utenti che hanno interpretato il Dirty Man è di 4.6 ($\sigma = 0.5163$), mentre per il Data Compiler è di 4.437 ($\sigma = 0.6291$);
- “flexibility” rappresenta la libertà che l'interfaccia fornisce all'utente; vincoli troppo stringenti sull'ordine delle azioni da eseguire, per esempio, comportano un punteggio minore in questo campo; il valore medio ottenuto per il Dirty Man è di 4.7 ($\sigma = 0.4830$), quello per il Data Compiler è invece 4.687 ($\sigma = 0.4787$).
- “error correction/handling and robustness” si riferisce alla gestione degli errori e al fatto che essi siano vincolanti o meno; la media per il ruolo del Dirty Man è 4.5 ($\sigma = 0.7071$), per il Data Compiler è 4.125 ($\sigma = 0.7187$).

In generale i valori sono tutti sopra al soddisfacente e sono accomunabili. I due ruoli, dopotutto, sono simili, quindi era lecito non aspettarsi sostanziali differenze. È comunque presente un leggero scarto a favore del Dirty Man, probabilmente dovuto alla presenza di una fase in più nella simulazione di quest'ultimo: gli utenti hanno qualche minuto in più per fare pratica, e alla fine della simulazione hanno acquisito una confidenza maggiore con l'interfaccia.

7.1.6 Immersione e presenza

I valori ottenuti per immersione e presenza sono anch'essi al di sopra del soddisfacente, con un solo caso in cui un utente non si è sbilanciato, dando 3 ("neutral") come valore. Per il Dirty Man si ha un valore medio di 4.6, con una variazione standard di $\sigma = 0.5164$, per il Data Compiler si ha invece una media di 4.75, con $\sigma = 0.5773$. In generale, l'immersione e la presenza sono stati tra gli elementi ritenuti più positivi dall'utenza, anche da chi non ha compilato il questionario.

7.1.7 Utilizzabilità

L'utilizzabilità, quindi la possibilità di impiegare la piattaforma come strumento effettivo di training, ha ottenuto il valore medio più alto del questionario, confermando quindi l'effettiva validità dello strumento. Per gli utenti che hanno interpretato il Dirty Man il valore medio assegnato a questo campo arriva a 4.9 ($\sigma = 0.3162$), per il Data Compiler è comunque alto ma si ferma a 4.75 ($\sigma = 0.4472$). Oltre all'utilizzo dello strumento attuale è stato apprezzato anche il potenziale della piattaforma, cioè il fatto che si presti ad essere ampliata aggiungendo scenari, ruoli e nuovi strumenti.

7.1.8 Il questionario multi-utente

Merita una menzione il fatto che, dai questionari posti agli operatori CBRN che hanno effettuato la prova multi-utente, è emersa una discrepanza tra quanto stabilito in fase di storyboard e quanto si aspettassero gli utenti: durante la progettazione è confermato che Data Compiler e Dirty Man costituiscono ruoli distinti, il primo con il compito di compilare i cartellini segnalatori e il secondo con il compito di posizionare i paletti. È stato dato, inoltre, il via libera a suddividere i compiti in modalità multi-utente, in modo da far gestire la scrittura al solo Data Compiler e il posizionamento dei paletti al Dirty Man (va notato, comunque, che qualsiasi personaggio può spostare un paletto, una volta posizionato la prima volta). In contrasto con queste informazioni, nei questionari è emersa come possibile aggiunta l'uniformazione dei due ruoli.

7.1.9 Commenti

Per quanto riguarda gli aspetti più negativi, vanno segnalate le attese lamentate rivolte alla presenza del cavo e al tracking dei sensori dell'Oculus. Nonostante alcune soluzioni empiriche adottate sul campo (appendendo il cavo al soffitto o aggiustando la posizione dei sensori) i due problemi si sono presentati ripetutamente in maniera più o meno intensa. Sono comunque aspetti legati all'hardware che sono stati risolti dalle versioni più recenti dei sensori e non hanno un diretto legame con la piattaforma sviluppata.

Un altro elemento che è stato indicato da molti come aspetto negativo è l'utilizzo dell'Arm Swinging, sia in termini di precisione dei movimenti, sia in termini di velocità. Per quanto riguarda il primo aspetto, si potrebbe aggiungere il corpo al modello dell'utente, in modo da aumentare ulteriormente la consapevolezza di sé all'interno del mondo virtuale; per la velocità dei movimenti, invece, si può lavorare sui parametri dello script dell'Arm Swinging. È comunque necessario notare che in pochi hanno mosso questa seconda lamentela (tre persone su 30 tra i questionari), quindi eventuali cambiamenti sotto questo aspetto potrebbero portare a un risultato peggiore di quello attuale.

Tra gli aspetti più positivi, invece, rientrano come già detto l'immersione e la presenza, oltre al realismo della simulazione, all'immediatezza dell'interfaccia e alla possibilità concreta di integrare la piattaforma nell'effettivo addestramento degli operatori. È inoltre interessante notare che un utente ha segnalato la mobilità offerta come aspetto positivo, in contrasto con le lamentate mosse all'Arm Swinging.

7.2 I dati registrati

I dati raccolti automaticamente durante la Toxic Trip (posizioni, rotazioni, eventi, valori casuali) avranno principalmente un duplice impiego. In primo luogo, potranno essere utilizzati per riprodurre la simulazione, rendendo così possibile un effettivo replay delle azioni di gioco e dei movimenti. Sfruttando le potenzialità dell'asset utilizzato, inoltre, durante queste riproduzioni sarà possibile utilizzare nuovi script per registrare nuovi dati e nuove statistiche, effettuando così un'analisi a posteriori.

Il secondo impiego di queste registrazioni consisterà nell'utilizzarle per "addestrare" elementi dell'applicazione, rendendo così l'esperienza più realistica. Per fare un esempio, si potranno utilizzare i movimenti e le azioni degli utenti per migliorare i comportamenti e i gesti degli NPC, in modo da rendere più coinvolgente l'esperienza in modalità utente singolo. Un altro esempio potrebbe consistere nell'utilizzare le posizioni degli oggetti (come i paletti utilizzati per delimitare l'area) per affinare il sistema di valutazione facendo riferimento alle azioni di operatori esperti.

Capitolo 8

Conclusioni e sviluppi futuri

8.1 Conclusioni

La tesi ha portato allo sviluppo di una piattaforma che potrebbe essere utilizzata effettivamente come strumento di training. Le prove sul campo hanno confermato la validità dell'interfaccia utilizzata, che infatti risulta essere molto intuitiva, facile da apprendere e poco invasiva. Dopo i primi minuti della simulazione, gli utenti hanno dimostrato dimestichezza con i comandi e con strumenti quali l'inventario e il sistema di scrittura, che quindi non hanno fatto da filtro e non hanno minato l'esperienza complessiva. Questo fatto è molto importante, perché conferma che l'applicazione potrebbe essere utilizzata continuativamente senza rappresentare una difficoltà aggiuntiva per l'utente e senza distogliere l'attenzione dall'effettiva procedura simulata.

La validità dell'applicazione è stata confermata dagli operatori CBRN che l'hanno provata direttamente in occasione della Toxic Trip, elogiandone in particolare l'immersione, la sensazione di presenza e il realismo: molti degli utilizzatori hanno apprezzato la fedeltà della simulazione, rimanendo colpiti in particolare dalla riproduzione fedele delle sonde e degli altri strumenti. Sotto questo aspetto sono comunque presenti margini di miglioramento, infatti si potrebbe lavorare ulteriormente sulle prime fasi della simulazione, permettendo all'utente di posizionare liberamente tutti gli oggetti della Working Area.

Infine, oltre ad aver evidenziato la possibilità di utilizzare la piattaforma, allo stato attuale, come strumento di training, molti militari ne hanno sottolineato il potenziale: l'applicazione di presta infatti a possibili estensioni e potrebbe essere facilmente estesa alle altre procedure del dominio CBRN, sia Italiane che straniere.

8.2 Sviluppi futuri

Tra i possibili sviluppi futuri, quello più interessante consisterebbe nell'utilizzare la piattaforma per una serie di esercitazioni nella base del Terzo Stormo. Questo non solo permetterebbe di raccogliere altri dati utili al training degli NPC, ma darebbe anche la possibilità di effettuare dei test sulla lunga distanza: mettendo a confronto, in un'esercitazione reale, un gruppo di utenti che si è addestrato continuativamente con il VR e un altro gruppo che si è allenato con metodi tradizionali, si potrebbe verificare ulteriormente la validità della piattaforma VR.

Un altro possibile sviluppo potrebbe vedere delle sessioni multi-utente in cui operatori esperti devono collaborare con del personale inesperto, allenato solo grazie alla piattaforma VR. Andando poi a raccogliere i dati e i feedback delle varie sessioni, si potrebbe dare anche in questo caso un ulteriore giudizio sulle qualità della piattaforma.

Per quanto riguarda gli elementi da aggiungere alla piattaforma, infine, le possibilità sono molteplici, anche per via della grande varietà di procedure e di situazioni che rientrano nel campo della CBRN. Rimanendo sulla procedura del Recce Team, si potrebbe:

- aggiungere il rischio biologico;
- aggiungere la strumentazione utilizzata da altre nazioni, per rendere più flessibile la simulazione;
- realizzare la modalità utente singolo per il Team Leader, creando un'interfaccia di tipo gestionale in cui l'utente deve gestire gli NPC;
- aggiungere delle variabili allo scenario attuale, inserendo fattori come il cambiamento run-time della direzione del vento, il ciclo notte-giorno, la pioggia e altre condizioni ambientali;
- creare nuovi scenari, cercando in particolare di non limitarsi a spazi aperti, ma modellando ambienti indoor, strade di una città (considerando, magari, anche la simulazione delle folle), ecc.

Volendo espandere la piattaforma oltre al Recce Team, invece, il passo successivo consisterebbe nell'implementare altre procedure. La decontaminazione di velivoli e veicoli sarebbe la progressione più naturale, sebbene anche la meno interessante (si tratterebbe semplicemente di riadattare le logiche di decontaminazione già presenti, indirizzandole verso altri obiettivi). Procedure come la Colpro, invece, sarebbero simili all'esperienza attuale, ma aggiungerebbero comunque nuovi spunti (maggiore focus sull'interazione con gli NPC, simulazione di movimenti e procedure che richiedono maggiore precisione, simulazione di tessuti in tempo reale accompagnata da un necessario lavoro di ottimizzazione, ecc.).

Appendice A

Questionario

Age:

Gender:

Nationality:

Simulation status:

Simulation folder:

Device:

- HTC Vive
- Oculus Rift

1. **Motion sickness:** rate symptoms like general discomfort, nausea, eye fatigues on a scale from 1 to 5.

Before the simulation:

After the simulation:

Background:

2. Virtual reality

Not familiar	Slightly	Somewhat	Moderately	Extremely

3. Video games

Not familiar	Slightly	Somewhat	Moderately	Extremely

4. Serious games

Not familiar	Slightly	Somewhat	Moderately	Extremely

5. CBRN procedures

Not familiar	Slightly	Somewhat	Moderately	Extremely

6. Italian CBRN procedures

Not familiar	Slightly	Somewhat	Moderately	Extremely

Usability Factor 1 Functionality: *The interface should be able to provide the level of functionality (control) the user expects in order to perform a given task. The interface should not be ambiguous, it should be easy to learn, remember and access all the functions available, as well as to understand their meaning.*

7. I would rate the VR system in terms of functionality as:

Very unsatisfactory	Unsatisfactory	Neutral	Satisfactory	Very satisfactory

Usability Factor 2 User Input: The input device (the hand controllers, in this case) used to access the system should be easy to use. Responsiveness and sensitivity should be appropriate for the task, and ideal for interacting with the virtual environment (for moving in the virtual space, for interacting with objects, etc.).

8. I would rate the user input as:

Very unsatisfactory	Unsatisfactory	Neutral	Satisfactory	Very satisfactory

Usability Factor 3 System Output: Providing users with visual feedback in a VR system is extremely important. Video output should have the appropriate quality (e.g., in terms of resolution, lack of glitches, etc.), objects should appear as three-dimensional (sense of depth), and information displayed should be unambiguous and deliver all the necessary information to complete the task.

9. Overall I would rate the output system as:

Very unsatisfactory	Unsatisfactory	Neutral	Satisfactory	Very satisfactory

Usability Factor 4 Consistency: The VR system should behave in the manner the user actually expects. It should be easy to understand the operation of the interface, which should behave in a consistent way.

10. Overall I would rate the consistency of the system as:

Very unsatisfactory	Unsatisfactory	Neutral	Satisfactory	Very satisfactory

Usability Factor 5 Flexibility: The flexibility of the interface is a function of its intuitiveness and learnability. It should be easy for the user to perform the task in the order chosen, without interfering with the way he or she wanted to interact with the system.

11. I would rate the flexibility provided by the system as:

Very unsatisfactory	Unsatisfactory	Neutral	Satisfactory	Very satisfactory

Usability Factor 6 Simulation Fidelity: The simulation underlying the VR experience should have the level of accuracy and quality required to be effective/to be of help. It should behave in the expected manner, with content recreated with proper fidelity so that the user does not feel disoriented during the experience.

12. I would rate the fidelity of the simulation as:

Very unsatisfactory	Unsatisfactory	Neutral	Satisfactory	Very satisfactory

Usability Factor 7 Error Correction/Handling and Robustness: Interface should be able to prevent the user from making mistakes while interacting with the system, and should let him or her recover from them.

All computer systems should provide error correction and recovery before a permanent change is made:

13. I would rate the reliability of the system as:

Very unsatisfactory	Unsatisfactory	Neutral	Satisfactory	Very satisfactory

Usability Factor 8 Sense of Immersion/Presence: An important characteristic of a VR system is the sense of immersion (thanks to the sensory stimulation provided by the hardware used) and the sense of presence (that is, the impression of “being there”, of “feeling part of something that is really happening”) one gets.

14. I would rate my sense of immersion and presence as:

Very unsatisfactory	Unsatisfactory	Neutral	Satisfactory	Very satisfactory

9 Overall System Usability: This section investigates a participant’s overall opinion of the system from the perspective of its usability, i.e., its intuitiveness and ease of use.

15. I would rate the system usability as:

Very unsatisfactory	Unsatisfactory	Neutral	Satisfactory	Very satisfactory

Mentions from one to three aspects that you found as <u>more negative</u> in the experience	Mentions from one to three aspects that you found as <u>more positive</u> in the experience

Comments/Suggestions:

Bibliografia

- [1] NATO – AJP-3.8, Allied Joint Doctrine for Chemical, Biological, Radiological, and Nuclear Defence.
- [2] Daniel R. Mestre – “*Immersion and Presence*”.
URL:http://www.ism.univmed.fr/mestre/projects/virtual%20reality/Pres_2005.pdf
- [3] Sensorama.
URL:<https://en.wikipedia.org/wiki/Sensorama>
- [4] CAVE Display.
URL:https://en.wikipedia.org/wiki/Cave_automatic_virtual_environment
- [5] A.Garcia-Palacios, H. Hoffman, A.Carlin, T.A. Furness III, C. Botella. “*Virtual Reality in the treatment of spider phobia: a controlled study*”. 2002.
URL:https://www.researchgate.net/publication/11125947_Virtual_reality_in_the_treatment_of_spider_phobia_A_controlled_study
- [6] Unity.
URL:<https://unity.com/>
- [7] Visual Studio.
URL:<https://visualstudio.microsoft.com/it/?rr=https%3A%2F%2Fwww.google.com%2F>
- [8] Blender.
URL:<https://www.blender.org/>
- [9] Optitrack.
URL:<https://optitrack.com/>
- [10] HTC Vive.
URL:<https://www.vive.com/us/>
- [11] Field of View: Evaluation of Oculus Rift and HTC Vive.
URL:https://www.ist.ucf.edu/Portals/10/files/RAPIDposters/RAPIDFOV_Oculus_Vive_Poster.pdf?ver=2019-06-04-095455-067
- [12] Oculus Rift.
URL:<https://www.oculus.com/>

- [13] Drone strike
URL:https://en.wikipedia.org/wiki/Drone_strike
- [14] SteamVR.
URL:<https://store.steampowered.com/steamvr?l=italian>
- [15] SteamVR Plugin.
URL:<https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647>
- [16] Unet.
URL:<https://docs.unity3d.com/560/Documentation/Manual/UNet.html>
- [17] Record And Play.
URL:<https://assetstore.unity.com/packages/tools/animation/record-and-play-148294>
- [18] La Stampa - *Il Poli “addestra” i militari della Nato con un simulatore di realtà virtuale.*
URL:<https://www.lastampa.it/torino/2019/10/02/news/il-poli-addestra-i-militari-della-nato-con-un-simulatore-di-realta-virtuale-1.37624940>
- [19] Roy S. Kalawsky. “*VRUSE—a computerised diagnostic tool: for usability evaluation of virtual/synthetic environment systems*”.
URL:https://www.researchgate.net/publication/13192025_VRUSE_-_A_computerised_diagnostic_tool_For_usability_evaluation_of_virtualsynthetic_environment_systems
- [20] Kay Stanney et al. “*Motion sickness and proprioceptive aftereffects following virtual environment exposure*”.
URL:<https://www.ncbi.nlm.nih.gov/pubmed/10098814>