

POLITECNICO DI TORINO

Corso di Laurea Magistrale in
Ingegneria del Cinema e dei Mezzi di Comunicazione

Tesi di Laurea Magistrale

Realtà Virtuale per l'addestramento alla gestione del rischio idrogeologico.

Creazione di uno scenario virtuale e gestione degli NPC.



Relatore

Prof. Fabrizio Lamberti

Candidato

Nicolò Massobrio

Sessione di Luglio 2020

Abstract

La Realtà Virtuale o *Virtual Reality* (VR) è una tecnologia ormai largamente disponibile nel mercato dell'intrattenimento, che favorisce un salto in avanti nelle modalità di fruizione dei contenuti, con una veste ricca di immersione, feedback sensoriali e grande coinvolgimento da parte del consumatore.

L'evoluzione dell'elettronica nel campo delle tecnologie immersive e la presa di coscienza delle potenzialità della VR nella comunicazione e nel coinvolgimento personale hanno permesso di guardare oltre il semplice consumo a scopo ludico. Infatti, l'utilizzo della tecnologia è stato allargato ad ambiti come l'educazione e l'addestramento, evidenziando una particolare efficacia nel rendere l'utente parte di un'esperienza realistica ed altamente formativa per gli aspetti pratici.

L'addestramento di volontari della Protezione Civile in previsione di emergenze di tipo idrogeologico si prefigura come obiettivo finale del progetto di tesi, concretizzandosi in una simulazione VR destinata all'addestramento sul campo degli operatori. La realizzazione di un percorso di studio e progettazione svolto con il Coordinamento di Protezione Civile della Regione Piemonte ha portato a identificare una struttura standard di fasi operative e a definire una serie di caratteristiche necessarie per restituire un'esperienza verosimile ed efficace nella formazione dei volontari.

La simulazione VR si presenta come uno scenario fittizio, in cui l'utente si ritrova ad affrontare un'emergenza idrogeologica, venendo addestrato nella messa in operatività del Modulo internazionale High-Capacity Pumping (HCP), destinato al trasporto di ingenti quantità d'acqua e alla messa in sicurezza di vaste aree naturali e urbane.

L'intero processo di progettazione e realizzazione si pone come intento quello di creare un contenuto dalla forte componente di realismo, sia nelle interazioni con le risorse a disposizione e con gli altri membri della squadra, che nel ricreare un'ambientazione verosimile in cui il Modulo potrebbe effettivamente operare. In particolare, il presente lavoro di tesi si è focalizzato, nel contesto del progetto, sull'aspetto grafico della simulazione e sulla modellazione, l'animazione e le

modalità di interazione dei Personaggi Non-Giocanti o *Non-Player Character* (NPC).

In primo luogo, il lavoro si è concentrato sulla creazione dello scenario, l'ottimizzazione grafica e la creazione dei sistemi particellari dell'acqua e degli elementi naturali che compongono l'ambientazione. Sono stati quindi realizzati e animati i modelli 3D degli avatar virtuali che svolgono il ruolo di NPC nella simulazione. La parte più corposa, infine, ha riguardato il lavoro di gestione della cinematica inversa o *Inverse Kinematics* (IK) per gli arti dei modelli e l'integrazione di tali elementi con la Macchina a Stati Finiti o *Finite State Machine* (FSM), con cui è gestita la logica degli operatori virtuali.

Indice

Abstract.....	3
Elenco delle figure.....	8
1 Introduzione	11
1.1 Il Progetto PITEM RISK.....	11
1.2 Protezione Civile	12
1.2.1 Rischio Idrogeologico	14
1.2.2 Il modulo europeo HCP	16
1.3 La Realtà Virtuale	18
1.3.1 Definizione di Realtà Virtuale	19
1.3.2 Immersione e Presenza.....	20
1.4 Obiettivi del progetto di tesi.....	21
1.5 Organizzazione delle attività	22
1.6 Struttura del documento	24
2 Stato dell'arte	26
2.1 Storia della VR.....	26
2.2 Campi di utilizzo	31
2.2.1 Serious Game	33
2.3 La VR nell'addestramento	35
2.3.1 Vantaggi e motivazioni dell'approccio.....	36
2.3.2 Svantaggi dell'approccio	39
2.3 Reale Virtuale per la gestione delle emergenze	39
2.2.1 Rischio esondazione nel parco fluviale Gesso e Stura a Cuneo	40
2.2.1 Incidente stradale presso St.Paul-sur-Ubaye.....	41

2.2.1	<i>Dispersione di sostanze chimiche sul colle della Maddalena</i>	42
2.2.1	<i>Nube tossica di sostanza radioattiva in avvicinamento a Torino</i>	43
3	Tecnologie	45
3.1	Hardware	45
3.1.1	<i>HTC Vive.....</i>	46
3.1.2	<i>Oculus Rift.....</i>	49
3.1.3	<i>Altre tecnologie sul mercato.....</i>	51
3.2	Software	52
3.2.1	<i>Game engine e Unity.....</i>	52
3.2.2	<i>Visual Studio</i>	55
3.2.3	<i>Blender</i>	56
3.2.4	<i>MakeHuman e Mixamo</i>	57
4	Progettazione.....	59
4.1	Addestramento dei volontari della Protezione Civile.....	59
4.1.1	<i>Risorse del modulo HCP.....</i>	60
4.1.2	<i>Obi Rope</i>	62
4.1.3	<i>Operazioni da svolgere per la messa operativa.....</i>	63
4.2	Creazione dello scenario	64
4.2.1	<i>Asset utilizzati</i>	66
4.3	Logica della simulazione	66
4.3.1	<i>Struttura</i>	67
4.4	SteamVR	69
4.5	Interfaccia della simulazione.....	72
4.6	La progettazione NPC	73
4.6.1	<i>Inverse Kinematics (IK) e Final IK.....</i>	75

4.6.2	<i>Macchina a Stati Finiti e NavMesh</i>	77
4.7	Locomozione.....	79
5	Realizzazione	811
5.1	Creazione dello scenario.....	85
5.1.1	<i>Cascade e la creazione del fiume</i>	85
5.1.2	<i>Particellari d'acqua</i>	87
5.1.3	<i>Ottimizzazione grafica</i>	88
5.2	Modellazione degli NPC e degli oggetti di scena	91
5.3	Gestione degli NPC	93
5.3.1	<i>Macchine a Stati Finiti</i>	94
5.3.2	<i>Navigazione nello scenario virtuale</i>	98
5.3.3	<i>Inverse Kinematics e Final IK</i>	103
5.4	Interfaccia utente e Voice Over.....	109
5.5	Altri sistemi implementati	112
5.5.1	<i>Sistema di interazione</i>	112
5.5.2	<i>Menu iniziale e il sistema di valutazione</i>	114
5.5.3	<i>Obi Rope e la gestione delle manichette</i>	115
6	Conclusione	115
	Bibliografia	122

Elenco delle figure

2.1 Sensorama, Morton Heilig, 1960.....	27
2.2 La spada di Damocle, Ivan Sutherland, 1968	27
2.3 DataGlove, 1987.....	29
2.4 SEGA VR, SEGA, 1993.....	29
2.5 Virtual Boy, Nintendo, 1995.....	30
2.6 Esperienza VR nella città antica di Ullastret	32
2.7 Interfaccia di Battlezone, 1980.	34
2.8 Prototipo di VR multimodale simulante la cabina di un AOP	37
2.9 Prova dello scenario del Parco Gesso con pedana Virtualizer	41
2.10 Dettaglio soccorsi scenario Paul-sur-Ubaye.....	42
2.11 Scenario dell'incidente sul Colle della Maddalena	43
2.12 Dettaglio dell'area di lavoro nello scenario della nube tossica.....	44
3.1 HMD del sistema VR HTC Vive	47
3.2 Controller del HTC Vive	48
3.3 HMD e controller dell'Oculus Rift	50
3.4 Interfaccia di Unity 2019.3.2	54
3.5 Interfaccia di Blender 2.82	57
4.1 Esercitazione EU Modex, tenutasi in Montenegro nel 2019	59
4.2 Confronto tra la motopompa Godwin e il modello 3D.....	61
4.3 Scenario virtuale creato per la simulazione.....	64
4.4 Il fiume e la pozza d'acqua nell'ambientazione 3D	65
4.5 Le impostazioni della componente VR in Unity 2019.3.2	70
4.6 Input Binding UI di SteamVR	71
4.7 Equipaggiamento di un volontario della PC della Regione Piemonte	74
4.8 Esempio in cui è vantaggioso l'uso della IK.....	76

4.9 Final IK, asset per la gestione dell'IK in Unity	77
4.10 Gli elementi principali della componente AI in Unity.....	79
5.1 Brushing per la creazione di terreni, all'interno del Terrain Editor	82
5.2 Terrain Editor di Unity	83
5.3 Tool per la creazione della texture nel Terrain Editor.....	84
5.4 Tool di spawning degli alberi nel Terrain Editor	85
5.5 Logica a nodi sequenziali di Cascade	86
5.6 Cascade shader per la creazione del materiale dell'acqua	87
5.7 Particellare d'acqua nella simulazione	88
5.8 Funzionamento della tecnica del LOD nella simulazione	89
5.9 Component LOD nell'Inspector di Unity	90
5.10 Confronto tra il modello 3D e il billboard di una roccia	91
5.11 Modello 3D del camion con gru presente nel Modulo HCP	91
5.12 Edit mode in Blender per la modellazione 3D	92
5.13 Modello dell'operatore in Blender	93
5.14 Metodo per il cambiamento di stato nella FSM dell'NPC	95
5.15 Schema a nodi della FSM dell'NPC	96
5.16 Pezzo del codice di gestione della FSM	98
5.17 Pannello di gestione dei modelli 3D in Unity	100
5.18 Animator tool per la gestione delle animazioni in Unity	101
5.19 Impostazioni per la creazione della NavMesh in Unity	102
5.20 Il component NavMesh Agent all'interno del prefab dell'NPC	103
5.21 Component Full Body Biped IK all'interno del prefab dell'NPC	104
5.22 Particolare del component Full Body Biped IK nell'Inspector di Unity	105
5.23 Target per la IK di una mano dell'operatore, durante la simulazione	106
5.24 Il component Interaction Object nell'Inspector di Unity	107
5.25 Il component Interaction Target in Unity	108
5.26 Metodo per l'attivazione delle interazioni utilizzando la libreria di Final IK.....	109

5.28 Rappresentazione grafica di due modalità di evidenziazione utilizzate	110
5.28 Il component AudioSource all'interno del prefab del Camera Rig	112
5.29 Modalità di selezione all'interno della simulazione per gli oggetti interagibili	114
5.30 Menu iniziale con il RayCast per la selezione delle operazioni	115
5.31 Il component Obi Rope assegnato ad un modello, in Unity	116

1 Introduzione

L'addestramento di volontari della Protezione Civile (PC) in previsione di emergenze di tipo idrogeologico si prefigura come obiettivo finale del progetto di tesi, concretizzandosi in una simulazione digitale basata sulla *Realtà Virtuale* o *Virtual Reality* (VR), una tecnologia dalle grandi potenzialità nell'ambito dell'addestramento (*training*) e dell'educazione in genere. La realizzazione di un percorso di studio e progettazione svolto con il Coordinamento di Protezione Civile della Regione Piemonte, che ha portato ad identificare una struttura standard di fasi operative e a definire una serie di caratteristiche necessarie per restituire un'esperienza verosimile e realistica, si traduce in una simulazione con l'intento di creare un contenuto didattico con caratteristiche innovative, promuovendo l'utilizzo della tecnologia VR come strumento educativo in un ambito complesso come quello della gestione delle emergenze.

1.1 Il Progetto PITEM RISK

Il Progetto PITEM RISK [1] nasce nell'ambito del Programma INTERREG Francia-Italia ALCOTRA [2] di cooperazione transfrontaliera, con lo scopo di sviluppare strumenti di gestione dei rischi naturali e industriali, nel territorio diviso tra le regioni francesi di Auvergne Rhône-Alpes e di Provenza-Alpi-Costa Azzurra e le regioni italiane di Liguria, Piemonte e Valle d'Aosta.

PITEM RISK si propone di migliorare la resilienza dei territori di fronte alle emergenze, in modo da consentire a tutti di comprendere meglio i rischi così come acquisire i giusti comportamenti in una situazione di crisi, che si tratti di civili residenti nei territori sopracitati oppure operatori della PC, volontari del

corpo Antincendi Boschivi (AIB) o altri tecnici nell'ambito della sicurezza del territorio.

Nel campo della prevenzione e dell'addestramento si sviluppa, all'interno del piano tematico PITEM RISK, un sotto-progetto denominato PITEM RISK FOR [3], il quale vede un accordo operativo siglato tra la Regione Piemonte ed il Politecnico di Torino [4] allo scopo di declinare l'aspetto formativo con la sviluppo di strumenti di VR in una rete di apprendimento condivisa a supporto delle attività della PC della Regione.

La creazione di tale rete, destinata ad inserirsi in un insieme di "centri complementari e interconnessi di realtà virtuale, che lavoreranno sulla base software condivisa, distribuiti su tutto il territorio ALCOTRA, viene considerata nell'ottica di:

- formare ciascun soggetto ad intervenire nell'ambito del ruolo che gli compete (compresa la popolazione);
- insegnare alle diverse parti interessate (servizi di emergenza, amministratori civili, ecc.) a lavorare con altri servizi.

Questo tipo di strumento pedagogico permette di definire una nuova modalità di apprendimento e consente di rivolgersi a pubblici diversi e complementari, sviluppando una piattaforma in grado di formare sulla gestione di più fenomeni, tra cui emergenze di tipo idrogeologico (alluvioni, movimenti del suolo, ecc.) ed incendi boschivi.

1.2 Protezione Civile

La "protezione civile" è l'insieme delle attività e delle politiche messe in campo sia a livello nazionale sia a livello locale per tutelare l'integrità della vita, i beni, gli insediamenti, gli animali e l'ambiente dai danni o dal pericolo di danni che

derivano dalle calamità: previsione, prevenzione e mitigazione dei rischi, gestione delle emergenze e il loro superamento.

Queste attività non sono assegnate a una singola amministrazione, ma si riuniscono in una funzione attribuita ad un sistema complesso, istituito dalla legge 24 febbraio 1992, n. 225: il Servizio Nazionale della Protezione Civile [5]. In Italia tale sistema trova il suo punto di raccordo nelle responsabilità di indirizzo e coordinamento affidate al Dipartimento della Protezione Civile [6], la struttura del governo della Repubblica Italiana, facente capo alla Presidenza del Consiglio dei ministri.

Il Dipartimento si occupa a livello nazionale della previsione, prevenzione, gestione e superamento di disastri, calamità, umane e naturali, e al soccorso e all'assistenza delle popolazioni in situazioni di emergenza, operando in stretto rapporto con le Regioni e le Province autonome.

Le attività di protezione civile individuate dalla legge n. 225/92 si profilano come obiettivo principale la tutela delle persone e la salvaguardia del territorio e vengono suddivise in:

- attività di previsione, volte a valutare gli scenari di rischio e, quando possibile, a preannunciare, monitorare, sorvegliare e vigilare gli eventi e i livelli di rischio attesi, con il concorso di soggetti scientifici e tecnici competenti;
- attività di allertamento, pianificazione, formazione, diffusione della conoscenza della protezione civile, informazione alla popolazione, esercitazioni e applicazione della normativa tecnica;
- attività di soccorso consistenti nell'insieme degli interventi di prima assistenza alle popolazioni colpite;

- attività di superamento dell'emergenza, ovvero l'insieme delle iniziative necessarie per rimuovere gli ostacoli alla ripresa delle normali condizioni di vita nei territori colpiti.

Con il D.G.R. del 12.01.2008, n. 12-8046 è stato recepito il Progetto Colonna Mobile nazionale delle Regioni [7], costituente il quadro prescrittivo di riferimento per le strutture del Sistema regionale di Protezione Civile proprie di ciascuna Regione, con l'elaborazione di una struttura di Procedure Operative Standard; tale struttura consiste nell'impiego di moduli funzionali, sulla base delle potenzialità e disponibilità in termini di mezzi, materiali e squadre operative in dotazione alla Regione stessa, oltre che alle Organizzazioni di Volontariato convenzionate con essa.

1.2.1 Rischio idrogeologico

L'Idrogeologia è la disciplina delle scienze geologiche che studia le acque sotterranee, anche in rapporto alle acque superficiali. Nell'accezione comune, il termine dissesto idrogeologico viene invece usato per definire i fenomeni e i danni reali o potenziali causati dalle acque in generale, siano esse superficiali, in forma liquida o solida, o sotterranee.

In Italia il dissesto idrogeologico è diffuso in modo capillare e rappresenta un problema di notevole importanza le cui manifestazioni più tipiche sono frane, alluvioni, erosioni costiere e valanghe.

Nel sistema di allertamento del Sistema Nazionale di Protezione Civile il rischio viene differenziato e definito come:

- rischio idrogeologico [8], se corrispondente agli effetti indotti sul territorio dal superamento dei livelli pluviometrici critici lungo i versanti, dei livelli idrometrici dei corsi d'acqua della rete idrografica minore e di smaltimento delle acque piovane;
- rischio idraulico [8], se relativo agli effetti indotti sul territorio dal superamento dei livelli idrometrici critici (possibili eventi alluvionali) lungo i corsi d'acqua principali.

La PC in ambito locale definisce un Modulo rischio idraulico, ovvero un asset di risorse strumentali tra cui autopompe e motopompe di diversa portata d'acqua, manichette e tubi di diverso diametro e lunghezza ed una serie di squadre di operatori specializzati, finalizzati a fronteggiare le conseguenze di eventi sul territorio.

Si è proceduto, successivamente, con l'organizzazione di un Modulo specialistico rischio idraulico di rilevanza nazionale, espresso dalla struttura regionale e dalle varie componenti operative del volontariato su scala regionale, per il concorso nelle emergenze di carattere nazionale. Le operazioni che caratterizzano questo Modulo consistono nell'aspirazione e scarico di ingenti quantità d'acqua allo scopo di intervenire nel mantenimento di una condizione geofisica del territorio stabile e sicura per la popolazione.

La configurazione predisposta corrisponde ad un asset idoneo ad operare in caso di eventi di tipo C¹ per il concorso alle emergenze nazionali/internazionali ed in caso di eventi di tipo B o di tipo C per le emergenze sul territorio regionale.

Il Piemonte, a causa della particolare conformazione geofisica ed idrogeologica e dell'alto grado di piovosità, si è trovato ad affrontare numerosi eventi calamitosi quali le drammatiche alluvioni del Tanaro nel 1994 e del 2000, fino ai più recenti casi di Villar Pellice nel 2008 e del 2016. Per tale ragione, si è investito in maniera significativa da parte della Regione e della PC regionale in un aumento e rafforzamento delle risorse a disposizione della Colonna Mobile regionale per il soccorso alla popolazione così come per il contenimento dei danni dovuti alle inondazioni.

1.2.2 Il Modulo europeo HCP

Il Modulo di pompaggio ad alta capacità o High-Capacity Pumping (HCP) [9] della Regione Piemonte (in seguito denominato anche ITA-HCP) è stato aggiunto alla Colonna Mobile regionale in accordo con il Dipartimento nazionale di Protezione Civile, poiché creato per lavorare nel Meccanismo di Protezione Civile dell'UE (a seguito del trattato di Lisbona - decisione del Comitato del 20/12/2007 - 2008/73 / CE, Euratom). Questo Modulo europeo consiste in un team di operatori addestrati ad una rapida capacità di dispiegamento, con il compito di svolgere attività di pompaggio dell'acqua, per supportare le autorità locali o altre organizzazioni, generalmente sotto il coordinamento del Centro di Coordinamento della Protezione Civile europea, offrendo ulteriore aiuto nella gestione delle emergenze di grande entità.

¹ Il tipo A si riferisce alle emergenze locali, gestibili su scala comunale; quelle di tipo B alle emergenze che richiedono una risposta e risorse su scala provinciale o regionale; quelle di tipo C alle emergenze di rilievo nazionale, per estensione e/o gravità.

Il Modulo HCP funziona secondo le linee guida internazionali stabilite dalle Nazioni Unite, svolgendo le attività di drenaggio delle aree allagate, monitoraggio dei livelli dell'acqua e approvvigionamento idrico per operazioni antincendio. Nell'ambito regionale il Modulo richiede le seguenti caratteristiche:

- pronto per partire entro 12 ore dall'attivazione formale del Modulo (6 ore per i componenti principali e 12 ore per le restanti strutture di supporto);
- utilizzo di motopompe autoadescanti con una capacità di pompaggio con una portata 1500 m / ora (20 m di dislivello);
- massima differenza di altezza: 40 m (portata ridotta);
- capacità di pompaggio di acqua fangosa: 5% di elementi solidi e particelle con dimensioni fino a 40 mm;
- distanza di pompaggio: 2500 m (tubi flessibili / giunti compatibili con Storz²);
- trasferimento via terra e via mare entro un raggio di 3000 km da Torino (Italia);
- autosufficienza per un minimo di 96 ore.

EUROMODEX³ [10] organizza annualmente diverse esercitazioni internazionali a cui ha partecipato anche la PC della Regione Piemonte, ad esempio le esercitazioni in Repubblica Ceca (2017) e Montenegro (2019) [11] in cui diverse squadre costituiti

² Il raccordo di Storz è un tipo di raccordo antincendio, ideato nel 1882, il quale viene collegato utilizzando ganci e flange interbloccabili, ampiamente utilizzato su tubi antincendio e nelle applicazioni antincendio.

³ Organizzazione non governativa (NGO) con sede in Romania, che si occupa a livello internazionale di prevenzione di rischi naturali e artificiali e di soccorso alla popolazione.

da volontari da tutta Europa hanno svolto operazioni simulate per la formazione in caso di emergenze di tipo idrogeologico, utilizzando risorse del Modulo HCP.

L'importanza di questo Modulo nella struttura della Colonna Mobile Regionale, dovuta alla vulnerabilità geologica e idraulica della Regione, è riconosciuta a tal punto da venir selezionata come elemento di riferimento nello sviluppo della simulazione VR nell'ambito del progetto PITEM RISK FOR, simulazione il cui scopo è proprio fornire uno strumento di formazione nel trasporto dell'attrezzatura, nella valutazione dell'area operativa e nella messa in azione della pompa appartenente a tale Modulo.

1.3 La Realtà Virtuale

L'efficacia di un addestramento mirato ad insegnare ai volontari della PC i corretti processi di messa operativa e di utilizzo delle risorse del Modulo HCP impone alla simulazione di assumere determinate caratteristiche. In particolare, sono di cruciale importanza il realismo dello scenario in cui operare, il complesso quadro delle azioni da compiere, l'utilizzo di un grande numero di strumenti di varia complessità e l'interazione con altri membri della squadra.

Tecnologie come lo schermo di un monitor, una tastiera o un semplice mouse non permetterebbero ad un utente di vivere un'esperienza simile ad una vera prova sul campo, poiché costituirebbero un limite evidente nell'interazione fisica con l'ambiente artificiale e con gli oggetti presenti.

Un utente che necessita di imparare a muoversi in un ambiente esterno in condizioni di emergenza, e a cooperare con altre persone nell'impiego di strumenti complessi, esige una tecnologia in grado di presentare uno spazio in cui potersi

muovere fisicamente ed interagire con una libertà tale da comprendere le dinamiche operative proprie di un Modulo come l'HCP.

1.3.1 Definizione di Realtà Virtuale

Gregor Burdea definisce con il termine “Realtà Virtuale” una particolare interfaccia costituita da una simulazione uomo-computer (ad alto livello) real-time di un mondo realistico, in cui sussistono infinite possibili interazioni con oggetti attraverso canali sensoriali multipli [12].

Un'esperienza in VR deve necessariamente essere multisensoriale e multimodale, riproducendo un luogo che l'utente assimila come realistico secondo le leggi fisiche, logiche e sociali del mondo reale in cui vive, illudendosi di essere stato “trasportato” in un altro luogo; quindi un ambiente virtuale immersivo in cui l'utente si può muovere, interagire con gli elementi presenti e fruire di un'esperienza più simile possibile all'analogica situazione reale.

Sherman e Craig in *Understanding Virtual Reality* [13], richiamano l'attenzione sulla forte contrapposizione in termini di significato tra il termine “virtuale”, usato per descrivere mondi immaginari ricreati attraverso le tecnologie e il termine realtà, come sinonimo di ciò che esiste, tangibile e ben focalizzato nella mente dell'utente.

Gli indizi sensoriali forniti al cervello umano dall'ambiente circostante, che siano visivi (il campo visivo a 360°, la disparità stereoscopica, la parallasse) o uditivi (il suono *surround*) sono simulati attraverso stimoli artificiali, portando l'organismo a non percepire alcun tipo di interferenza o filtro [14], mantenendo così labile il confine definito tra realtà e finzione.

Le interazioni con oggetti sintetici o avatar virtuali ed il movimento in uno spazio artificiale producono delle reazioni verosimili e realistiche, portando ad un coinvolgimento emotivo e fisico che donano all'esperienza virtuale una grande efficacia nel mantenere alta l'attenzione e nell'“immergere” l'utente nel contesto scelto.

Le applicazioni in VR trovano spazio nei campi dell'intrattenimento e dello spettacolo così come dell'educazione, dell'addestramento in ambito militare, medico, industriale e della psicologia (Virtual Reality Exposure Therapy o VRET). Con uno sviluppo della tecnologia sempre più crescente ed un'applicazione in contesti sempre più disparati, la VR si propone come un sistema totalmente immersivo, in cui tutti i sensi umani possono essere sfruttati, raggiungendo un livello massimo di immersione, seppure attualmente ad un livello ancora in parte teorico.

1.3.2 Immersione e presenza

L'importanza di realizzare una simulazione che presenti un ambiente verosimile e che sia il più fedele possibile al contesto ricercato, richiede di definire fattori che permettano di misurare e, di conseguenza, valutare l'efficacia di una sistema il cui obiettivo è proprio impedire all'utente di dissociarsi dal mondo virtuale.

Due concetti fondamentali al fine di definire quanto un utente abbia la sensazione di trovarsi a tutti gli effetti in un mondo realistico, i cui stimoli sensoriali richiamino quelli fisici e la cui struttura manifesti in esso una costante sensazione di vivere l'esperienza sono immersione e presenza.

- Il concetto di immersione si riferisce alla percezione fisica di trovarsi nel mondo virtuale. Questo parametro è funzione degli stimoli forniti dall'applicazione, siano essi video, audio, aptici o inerziali, ed aumenta con

la presenza di caratteristiche come un'elevata risoluzione grafica, tempi di latenza molto bassi ed una risposta realistica alle interazioni con oggetti e avatar virtuali. L'immersione rappresenta il metro oggettivo con cui si misura la capacità di un sistema di VR di riprodurre gli stimoli reali percepiti dai nostri sensi [15].

- Il concetto di presenza indica, invece, la sensazione di appartenenza al mondo virtuale, l'inganno della mente che crede di vivere un'esperienza reale. Essa è una misura soggettiva legata al concetto di "sospensione dell'incredulità" ed è stata anche definita come uno stato psicologico proprio di un utente che non percepisce la tecnologia quale mezzo attraverso cui viene fruito il contenuto, ma si immedesima senza filtri nella simulazione virtuale [16].

In una recente pubblicazione [17] Mel Slater ri-definisce entrambi i concetti attraverso nuove denominazioni, riferendosi alla cosiddetta *Place Illusion* (PI) come la percezione attraverso stimoli fisici di essere realmente nel mondo virtuale, sebbene la mente umana sappia di non trovarsi in quel luogo, ed indicando come *Plausibility Illusion* (PSI) l'illusione mentale di vivere l'esperienza simulata, nonostante la sensazione di un qualcosa di non reale.

1.4 Obiettivi del progetto e della tesi

Il progetto si pone come obiettivo la realizzazione di una simulazione in VR di uno scenario fittizio in cui l'utente, con il solo supporto di Personaggi non Giocanti o *Non-Player Character* (NPC), viene addestrato nella messa in operatività dell'intero Modulo HCP, apprendendo come agire nel processo di montaggio e

posizionamento delle manichette, fino alle fasi di accensione, controllo e spegnimento della pompa.

L'intero processo di progettazione e realizzazione si pone come intento quello di creare un contenuto dalla forte componente di realismo sia nelle interazioni con le risorse a disposizione e con gli altri membri della squadra, che nel ricreare un'ambientazione verosimile in cui il Modulo potrebbe effettivamente operare.

In particolare, il presente lavoro di tesi si è focalizzato sull'aspetto grafico del progetto sopra descritto e sugli aspetti legati alla modellazione, animazione ed interazione degli NPC.

In primo luogo, il lavoro si è concentrato sulla creazione dello scenario, l'ottimizzazione grafica e la creazione dei sistemi particellari dell'acqua e degli elementi naturali, quali il fiume e la pozza d'acqua. In secondo ambito, gli sforzi si sono concentrati sulla realizzazione dei modelli 3D, compresi quelli degli avatar virtuali che svolgono il ruolo di NPC nella simulazione. La parte più corposa, infine, ha riguardato il lavoro di animazione degli NPC, la gestione della cinematica inversa o *Inverse Kinematics* (IK) per gli arti dei modelli e l'integrazione di tali elementi con la Macchina a Stati Finiti o *Finite State Machine* (FSM), con cui è gestita la logica degli operatori NPC.

1.5 Organizzazione delle attività

La prima parte delle attività ha riguardato l'impostazione del progetto, che ha richiesto una serie di riunioni presso il Dipartimento ENERGIA (DENERG) del Politecnico di Torino e presso la sede della PC di Torino, in presenza del professor

Fabrizio Lamberti, responsabile del progetto per il Dipartimento di Automatica e Informatica (DAUIN) del Politecnico di Torino, e dei rappresentanti del Dipartimento della Protezione Civile Regione Piemonte.

In primo luogo, si sono stabiliti gli obiettivi principali, l'ambito in cui si sarebbe progettata la piattaforma (rischio idrogeologico) e si sono ricercate le prime fonti di informazione audiovisive e testuali. Nello stesso periodo, si è tenuto un raduno di volontari presso San Mauro Torinese (TO) durante la Settimana nazionale della Protezione Civile (13-19 ottobre 2019), in cui è stata raccolta ulteriore documentazione sulle modalità di formazione degli operatori. Successivamente si è deciso, come detto, di focalizzarsi sulla realizzazione di un contenuto didattico per l'addestramento dei volontari con il Modulo europeo HCP.

A seguito della scelta di tale Modulo come oggetto della simulazione virtuale, si sono svolti alcuni sopralluoghi presso le sedi distaccate della PCe a Druento e Vercelli, allo scopo di reperire materiale fotografico e video, ricostruendo inoltre con alcuni volontari e membri della PC l'intero quadro delle operazioni da svolgere per la messa operativa delle risorse proprie del modulo.

Contemporaneamente, si è svolta una prima parte della progettazione della piattaforma, realizzando un documento di design che rappresentasse l'evolversi del progetto di tesi, una prima bozza dello storyboard e la realizzazione dei modelli 3D.

Il gruppo di lavoro ha assistito ad un'esercitazione svolta presso il Nodo Idraulico di Ivrea e sono stati successivamente creati lo scenario ed una prima versione prototipale, con le interazioni base tra gli elementi principali della simulazione (motopompa, tubi, manichette e NPC), mantenendo nello stesso tempo un costante canale di comunicazione con la PC per aggiornamenti e feedback.

Successivamente, sono stati presentati una serie di video riguardanti l'evolversi della simulazione a partire da una versione del voice-over ancora acerba, fino ad arrivare a due versioni definitive della simulazione relative alle cosiddette modalità "guidata" e "valutata". Le versioni definitive delle due simulazioni hanno, inoltre, subito una serie di modifiche, a seguito dei commenti e delle richieste riportate dalla PC durante le riunioni che si sono svolte in via telematica, durante i mesi finali di lavoro.

1.6. Struttura del documento

Il presente documento è suddiviso nei seguenti capitoli:

- Capitolo 1: Introduzione

Vengono illustrati il caso di studio su cui si basa il lavoro di tesi, la PC con le proprie funzioni, il Modulo europeo HCP oggetto della simulazione ed infine la VR. Inoltre, vengono riportate le attività svolte ai fini della realizzazione del progetto e gli obiettivi della tesi.

- Capitolo 2: Stato dell'arte

Aprire una breve storia della VR, con un approfondimento agli utilizzi della tecnologia nell'ambito dell'addestramento e dell'educazione con esempi inerenti a tali campi di sviluppo.

- Capitolo 3: Tecnologie

Vengono riportate le tecnologie hardware e software utilizzate nella realizzazione del progetto di tesi.

- Capitolo 4: Progettazione

Le fasi di progettazione della simulazione sono riportate a partire dallo studio del materiale raccolto nei vari incontri con la PC e durante le esercitazioni a cui il gruppo di lavoro ha partecipato, fino alla scrittura dell'esperienza di addestramento, il voice-over, riportando gli asset utilizzati per la realizzazione dei contenuti.

- Capitolo 5: Realizzazione

Sono riportate le varie fasi di sviluppo, illustrando in modo specifico le scelte che hanno influito sulla forma definitiva della simulazione dal punto di vista della grafica, dell'ottimizzazione, della logica dell'applicazione e della gestione degli NPC.

Capitolo 6: Conclusioni

Vengono riportate alcune considerazioni finali sul lavoro svolto, fornendo una sintesi di limiti e mancanze riscontrati, ed una breve analisi sui possibili sviluppi futuri.

2 Stato dell'arte

Il capitolo tratta della VR dal punto di vista storico e di utilizzo nei campi dell'addestramento e della formazione, esponendo i vantaggi e gli svantaggi della tecnologia in tali ambiti. Inoltre, vengono presentati esempi di simulazioni VR destinate all'addestramento di operatori, militari o personale medico ed, in particolare, applicazioni realizzate per la gestione delle emergenze.

2.1 Storia della VR

Le tecnologie di simulazione VR che si sono diffuse nell'ultimo decennio hanno proposto un vasto numero di modalità di fruizione, variando dal più comune computer ad apposite interfacce indossabili (visori, tute, guanti), fino a comprendere metodologie per la fruizione di un'esperienza virtuale incredibilmente immersiva in alcuni loro aspetti (le pedane per la camminata a 360° oppure la VR Dome, un ambiente di grande coinvolgimento sensoriale).

L'evoluzione di tali tecniche di immersione in una realtà artificiale è frutto di una graduale ricerca, fin dall'inizio del XX secolo, nel campo delle tecnologie sensoriali, finalizzate al coinvolgimento dell'utente ad un livello superiore rispetto ai dispositivi comuni, quali un monitor oppure una cassa acustica.

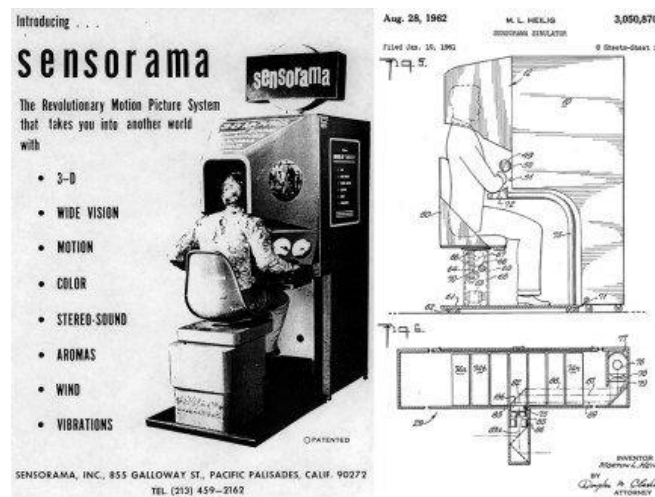


Figura 2.1: Sensorama, Morton Heilig, 1960.

Nel 1960, Morton Heilig, videomaker statunitense, brevettò un display, che successivamente chiamò *Sensorama* (Figura 2.1) [18]. Il dispositivo venne montato su un sistema di movimento, il quale permetteva di riprodurre di un film immersivo con la presenza di un vasto campo visivo, di suono stereoscopico, un seggiolino mobile, vibrazioni e odori, in un'unica sola esperienza che chiamò *Experience Theater*. La prima versione del Sensorama distribuita al pubblico permetteva di vivere una simulazione di un viaggio in moto tra le strade di Brooklyn.



Figura 2.2: La spada di Damocle, Ivan Sutherland, 1968.

Alcuni anni più tardi, Ivan Sutherland, presso l'Università dello Utah, inventò quello che ora viene considerato il primo visore di tipo *Head-Mounted Display* (HMD). Il dispositivo era costituito da due tubi catodici che permettevano di proiettare immagini tridimensionali, realizzando una tecnologia di sovrapposizione di contenuti virtuali a ciò che la persona vede attraverso le lenti, antesignana della moderna realtà aumentata. Il nome attribuito alla tecnologia dal proprio creatore fu "Spada di Damocle" (Figura 2.2) [19]; a causa del suo peso, infatti, la tecnologia necessitava di una struttura meccanica per sostenerla al soffitto.

Negli anni '70, sono stati sviluppati i primi dispositivi per esperienza immersive navigabili come gli HMD di tipo *LEEP* (*Large Expanse, Extra Perspective*), che aumentano gradualmente il campo visivo dell'utente. Tuttavia, il termine "Realtà Virtuale" venne coniato da Jaron Lanier, solamente nel 1982. Lanier, informatico pioniere della tecnologia immersiva e multisensoriale, lavorava all'epoca presso l'*Atari Research Lab*, fondato dall'omonima azienda di videogiochi, nel tentativo di integrare la nascente tecnologia al mondo videoludico. Il laboratorio diede lavoro ad altri pionieri della VR come Thomas Zimmerman, inventore dei *DataGlove*, nel 1982.

Il dispositivo fu costruito utilizzando sensori a fibre ottiche, i quali riuscivano a rilevare ben 256 posizioni delle dita. Successivamente, la tecnologia venne acquistata dalla Nintendo nel 1987, la quale decise di progettare e distribuire al pubblico il *PowerGlove*, dispositivo di consumo la cui struttura hardware era stata costruita sulla base del *DataGlove* (Figura 2.3).

Il prodotto fu un flop in quanto, oltre ad avere pochi videogiochi compatibili, il guanto era impreciso e di difficile utilizzo.

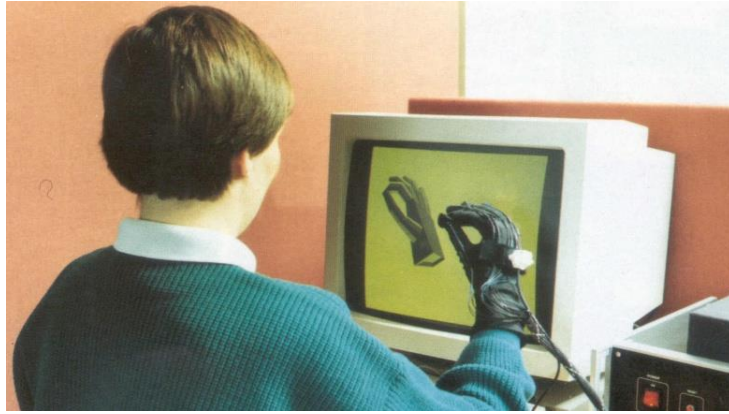


Figura 2.3: DataGlove, 1987.

Negli anni '90, a seguito del sempre più crescente importanza del mercato dell'intrattenimento, le aziende del settore si concentrarono sul rendere disponibili al pubblico le tecnologie basate su questa nuova tecnologia, imponendo una riduzione dei costi di sviluppo e di distribuzione.

La SEGA annunciò il suo rivoluzionario HMD, chiamato *SEGA VR* (Figura 2.4) [20], nel 1993, con 4 giochi abbinati. Il visore era provvisto di schermi LCD, cuffie stereo ed un supporto per il tracciamento dei movimenti della testa. Il visore non venne mai distribuito nel mercato a causa dei forti mal di testa e cinetosi che procurava all'utente.



Figura 2.4: SEGA VR, SEGA, 1993.



Figura 2.5: *Virtual Boy*, Nintendo, 1995.

Nel 1995, a Nintendo lanciò sul mercato il *Virtual Boy* (Figura 2.5) [20]. L'azienda annunciò il proprio prodotto come la prima console portatile, a prezzo contenuto, con una grafica 3D nel campo dell'intrattenimento. Il dispositivo si presentava come una sorta di elmetto da appoggiare su un tavolo collegato ad un *controller* e, nonostante tutto, presentava grosse limitazioni: un effetto 3D non impressionante, un display monocromatico, un prezzo elevato e effetti di nausea e mal di testa nel consumatore.

Dopo questi primi decenni di grandi aspettative per questa promettente tecnologia, gli anni 2000 furono caratterizzati da un parziale calo dell'interesse, a causa delle difficoltà tecniche e di potenza di calcolo, che si ritenevano un ostacolo troppo grande per l'epoca.

Un nuovo sviluppo si ebbe solo nel 2012, con l'uscita di *Oculus Rift*. Il nuovo visore risolveva alcuni dei grandi problemi che si erano riscontrati nelle tecnologie precedenti e divenne un trampolino di lancio per le interfacce che lo seguirono, tra cui l'*HTC Vive* e il *Samsung Gear VR*. Al giorno d'oggi, l'obiettivo ultimo della

VR, ovvero il raggiungere una forma di immersione e presenza assoluta non è però ancora stato raggiunto. Nonostante i nuovi ingenti investimenti nel settore valutato ormai come un mercato in crescita, da oltre un miliardo di euro, la tecnologia presenta ancora molti margini di sviluppo per gli anni a venire.

2.2 Campi di utilizzo

La tecnologia VR ha trovato spazio in un vasto numero di campi, proponendo un nuovo modo di fruire i contenuti, proponendoli con una veste nuova, più vicina, immediata e altamente coinvolgente. Queste caratteristiche hanno permesso alla VR di avere una notevole diffusione principalmente nel campo dell'intrattenimento.

I video 3D sono sfruttati nella diffusione di clip musicali, nei parchi a tema, nel cinema e negli eventi sportivi; nel campo videoludico, come visto nel paragrafo precedente, lo sviluppo di nuove interfacce a costo contenuto ha permesso la vendita di piattaforme VR ad uso personale. Esempi di dispositivi per la visione di video 3D a 360° sono il *Google Cardboard* e il *Samsung Gear VR*.

Nel campo della cultura la VR presenta molti campi d'utilizzo. La tecnologia ha portato a nuove modalità di fruizione per musei e siti archeologici, con un enorme potenziale nel campo della formazione culturale e nella digitalizzazione dell'eredità culturale. Si può utilizzare la VR per ricreare esperienze impossibili da vivere attraverso un'esperienza fisica, come la visita a luoghi inaccessibili (ad esempio le *Grotte di Pleito* in California, vietate perché sacre agli autoctoni [21]), per permettere l'osservazione di un'opera da punti di vista irraggiungibili o per ricreare i siti ormai perduti (ad esempio *Ullastret* città Iberica di 2200 anni fa [22]).



Figura 2.6: Esperienza VR nella città antica di Ullastret.

Nella campo delle psicologia, la VR viene comunemente utilizzata per ricreare situazioni particolari in un ambiente controllato, solitamente in occasione di particolari terapie. In questo modo il terapeuta può studiare le reazioni del paziente aiutandosi in una diagnosi oppure utilizzarla nella cura di alcuni disturbi, come la sindrome da stress post-traumatico e il morbo di Alzheimer. Ad esempio, nella cura delle fobie è possibile ricreare scene in cui il paziente deve affrontare le proprie paure (terapia dell'esposizione) [23]. Casi di utilizzo della VR per aiutare i pazienti a controllare le proprie fobie sono state terapie per curare l'aracnofobia [24] e la paura di parlare in pubblico [25].

Con scarsi risultati si è anche tentato di applicare la VR alla riabilitazione fisica, risultando per lo più inefficace rispetto ai metodi tradizionali. Si continua invece ad usarla per il trattamento del dolore agli arti fantasma con la creazione di stanze 3D in cui il paziente può controllare, tramite impulsi mioelettrici, gli arti mancanti [26].

Nell'ingegneria e nell'architettura, la VR si è rivelata utile nella progettazione di edifici e costruzioni senza gli ingenti costi di costruzione per la fase di verifica e collaudo. È possibile portare progetti su carta in VR, in modo da potervi interagire prima della loro attuazione, o ricreare eventi particolari, come terremoti, per osservare il comportamento di un edificio. Inoltre, si sta tentando ultimamente

di integrare sistemi VR per il mercato dell'elettronica, ma è un campo che richiederà ancora molti anni prima di risultare efficace dovendo anche combattere la reticenza dei consumatori.

Grazie alla sua capacità di creare scenari in un ambiente sicuro, la VR viene anche utilizzata per scopi educativi e di formazione. Gli utenti, agendo in un ambiente fittizio, hanno la possibilità di sviluppare le proprie capacità senza la paura di poter sbagliare e delle conseguenze dei loro eventuali errori. Gli ambiti sono innumerevoli: istruzione scolastica, medicina, forze armate, aeronautica, sicurezza sul lavoro, addestramento, ecc.

2.2.1 *Serious Game*

Il termine *Serious Game* (SG) nasce nel 1970 riferito a qualsiasi tipologia di giochi che utilizzi la componente ludica di un contenuto, l'aspetto legato al divertimento, allo scopo di educare e formare studenti, personale militare, medici e altre figure professionali nel campo della salute, dell'istruzione, della comunicazione strategica e della formazione militare.

Non esiste una singola definizione di SG, benché in genere ormai si colleghi il termine ai giochi realizzati per il mercato del digitale, si può riferire a giochi di qualsiasi genere, utilizzanti qualsiasi tecnologia e per qualsiasi piattaforma.

Il primo SG è considerato *Battlezone*, sviluppato da Atari, nel 1980. Il gioco era pensato per l'addestramento militare. Tuttavia, è solo negli anni '90 che gli studiosi iniziarono a pensare

seriamente ad andare oltre il mercato dell'intrattenimento, portando gli sviluppatori a trovare diversi utilizzi per i videogiochi.

Oggi, invece, c'è un notevole interesse per i SG sia per quanto riguarda il Governo e l'esercito, che per la formazione professionale, l'istruzione, la pubblicità e l'assistenza sanitaria.

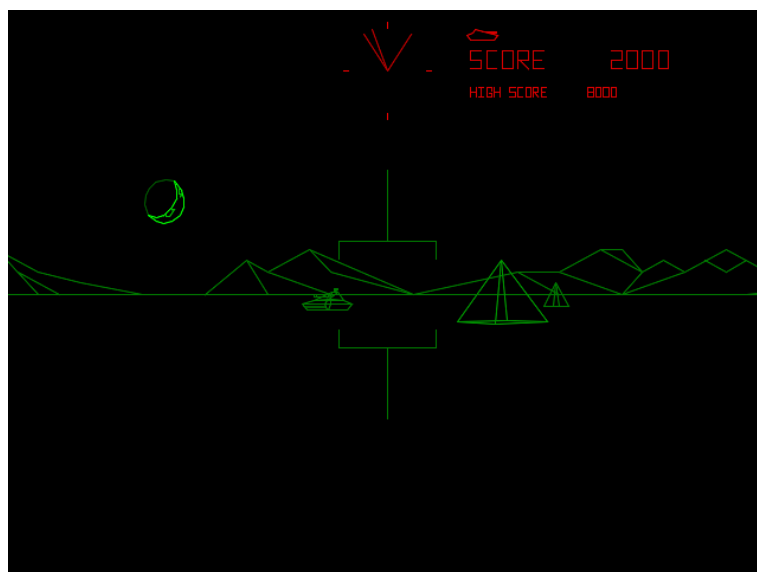


Figura 2.7: Interfaccia di Battlezone, 1980.

I vantaggi dei SG sono molteplici. In questi giochi è possibile la ricostruzione parziale o totale del mondo reale, nei suoi particolari artificiali e naturali: strumentazione, veicoli, macchinari, edifici, località e scenari naturali, tutto ad una spesa irrisoria in confronto alle effettive esercitazioni e sperimentazioni in luoghi reali e con i medesimi oggetti fisici. In questo modo è possibile renderli accessibili all'utente, senza che questi debba realmente fare uso di una strumentazione costosa e ingombrante oppure vivendo un'esperienza in situazioni critiche o di pericolo. Inoltre, è anche possibile ricreare qualsiasi scenario reale o immaginario, come battaglie o terremoti, altrimenti impossibili o altamente complessi da portare in atto. La costruzione e l'uso di simulatori tradizionali è infatti molto alto in confronto allo sviluppo di un videogioco.

Numerosi studi scientifici hanno inoltre dimostrato l'utilità dei SG nell'apprendimento, osservando un aumento della concentrazione e della velocità nell'imparare un contenuto o una procedura da parte degli studenti che utilizzavano SG, rispetto quelli che studiavano con un metodo tradizionale [27]. Ciò è dovuto per lo più al fatto che i SG sono in grado di promuovere l'interesse, l'impegno e il pensiero ad un livello di coinvolgimento superiore per gli studenti,

in aggiunta all'assenza della "paura del fallimento" e delle conseguenze che ne derivano.

2.3 La VR nell'addestramento

L'immersività, caratteristica prima di una simulazione VR, unendosi agli aspetti educativi e formativi di un SG può aiutare ad apprendere più efficacemente concetti e procedure che si vogliono comunicare agli operatori, inserendosi in un contesto verosimile alla mansione svolta. L'interesse all'utilizzo della tecnologia accomuna enti governativi, istituzioni accademiche e privati legati al settore tecnico, i quali condividono l'intenzione di adottare metodi alternativi e/o complementari a quelli tradizionali, e la VR sembra poter colmare alcune lacune presenti in questi ultimi.

Un addestramento efficace è caratterizzato, senza dubbio, dalla qualità dell'insegnamento, dalla frequenza con cui vengono effettuati test per la valutazione delle competenze insegnate e la capacità di esporre le giuste procedure in caso si verifichino situazioni particolari ed imprevisti che possono sorgere nell'esecuzione del lavoro assegnato. Tuttavia, le principali modalità di addestramento su cui puntano le aziende sono costituite da lezioni frontali e videolezioni oppure da esercitazioni sul campo che simulano le situazioni d'intervento in ambienti controllati. Questi protocolli, pur essendo ampiamente consolidati, presentano una serie di svantaggi: la formazione in un aula o online non aiuta ad incrementare l'attenzione e la concentrazione necessarie ad imparare procedure specifiche in breve tempo e con efficacia; d'altra parte un'esercitazione dal vivo richiede un impiego di risorse e tempo notevoli a causa dell'organizzazione, del trasporto degli operatori e dell'attrezzatura e dell'esecuzione, con l'ulteriore ostacolo di dover presentare una situazione verosimile in cui il personale si troverà ad operare. Infine, lo sforzo nell'organizzazione delle prove sul campo impone un ulteriore dispendio di

risorse nel garantire la sicurezza degli studenti durante la formazione e, spesso, gli eventi da fronteggiare, quali emergenze di tipo idrogeologico, incendi o altri fenomeni naturali, richiedono una particolare attenzione nel preservare l'incolumità ed una notevole difficoltà nel ricreare le giuste condizioni di intervento.

Perciò, La VR si presenta come il potenziale innovativo strumento di addestramento e, sebbene la diffusione della tecnologia in forma massiccia non sia ancora così ampia al giorno d'oggi (con rare eccezioni nell'ambito medico e aerospaziale, già agli inizi degli anni 2000 le possibilità di utilizzo di questi sistemi veniva sempre più indagata, in molteplici settori, sia nel settore pubblico che in quello privato. Risulta, quindi, opportuno presentare quali siano le motivazioni che potrebbero portare ad implementare una soluzione del genere, e i relativi punti di forza e criticità.

2.2.1 Vantaggi e motivazioni dell'approccio

Uno dei principali motivi per cui l'implementazione un sistema VR per l'addestramento è senza dubbio il dispendio economico dovuto all'organizzazione delle esercitazioni dal vivo. Sicuramente, la preparazione teorica in aula non permette di preparare gli operatori all'utilizzo della strumentazione sul campo e all'approccio richiesto in fase di intervento, tuttavia le esercitazioni pratiche comportano la presenza di gravi rischi per la salute se affrontate con leggerezza e disorganizzazione. Inoltre, qualora l'addestramento preveda l'occupazione di una infrastruttura pubblica o l'utilizzo di macchinario privato, un potenziale errore da parte degli operatori o del personale in fase di addestramento, potrebbe causare un disservizio o rovinare l'attrezzatura, portando ad ulteriori costi oltre quelli già stanziati.

Allo scopo di evitare queste situazioni, si valutano le abilità dell'individuo tendenzialmente su un insieme ristretto di operazioni e macchinari, fornendo,

quindi, una preparazione parziale che potrebbe portare a momenti di indecisione sul campo [28]. La VR risulta, perciò, uno strumento utile nell'offrire un intervallo più esteso di casistiche e la piena funzionalità delle apparecchiature in una simulazione virtuale, fornendo un addestramento completo e, al tempo stesso, evitando di danneggiare le attrezzature reali (Figura 2.7).



Figura 2.8: Prototipo di VR multimodale simulante la cabina di un AOP (Air Observation Post).

Allo stesso modo, il personale preparato all'intervento in caso di eventi catastrofici e naturali deve confrontarsi con un elemento particolarmente impattante sulle prestazioni sul campo: lo stress psicologico e fisico a cui ogni operatore, sebbene in misura differente, viene sottoposto. La familiarità che l'operatore ha con una situazione reale di intervento può fare la differenza sulla qualità dell'operato, ma molte situazioni di emergenza non possono essere facilmente replicate in una semplice sessione di addestramento anche per evitare di mettere a repentaglio l'incolumità degli operatori e la possibilità di incorrere in una certa genere di situazioni critiche può rivelarsi estremamente bassa, tanto da non permettere un'esplorazione adeguata dell'effetto che tali condizioni di intervento hanno sulle abilità decisionali, comunicative e di coordinamento tra i membri di una squadra di primo intervento [29]. Anche in questo caso

un'esperienza in VR può aiutare la formazione, permettendo non solo di presentare questi casi limite, ma anche elementi che possono modellare la sessione virtuale in modo da renderla più verosimile, come la presenza di ostacoli visivi, suoni e rumori realistici, folle di civili da evacuare, ecc. senza incorrere nei costi e nella pericolosità di ricreare la situazione d'emergenza dal vivo.

Oltre a garantire questa particolare condizione di emergenza “simulata”, si può introdurre la misurazione dello stato emotivo e psicologico degli utenti durante la simulazione, verificando che lo stress percepito e la risposta dell'individuo ad esso siano comparabili con la situazione reale e le esercitazioni dal vivo [30]. Un altro elemento positivo che si è riscontrato in diversi esperimenti di formazione con questa tecnologia è la somiglianza con l'efficacia dell'insegnamento fornito dai metodi tradizionali, sia per aspetti nozionistici [31] che di procedure da applicare sul campo [32].

Infatti, è possibile fornire un feedback immediato alle azioni degli utenti, permettendo loro di assimilare con più facilità le procedure corrette e recepire eventuali errori commessi. La tecnologia VR permette, inoltre, di poter scalare il livello di difficoltà in base alle competenze del personale addestrato, e dimensionarlo in base alle esigenze richieste in fase di addestramento.

Il costo relativamente basso di dispositivi per la VR e la facilità d'utilizzo che migliorano assieme a un costante progresso grafico e di prestazioni della tecnologia consentono di adottare questa soluzione sempre più di frequente e con maggiore facilità rispetto alle esercitazioni su media e larga scala, dal momento che essa rende la pianificazione di contenuti più comoda, economica e scalabile. I dispositivi in questione hanno la possibilità di offrire elevati livelli di qualità nel mostrare anche i minimi dettagli della strumentazione, come per le cabine di pilotaggio degli aerei o per gli astronauti, in casi in cui è richiesto un alto livello di dettaglio e fedeltà [33]. Una strumentazione a basso costo consente, inoltre, di superare le limitazioni geografiche, permettendo al personale di addestrarsi anche in remoto e, utilizzando piattaforme di training consistenti e sincronizzate, si ha la possibilità di sviluppare capacità di comunicazione anche con gli altri

membri della squadra operativa, specialmente per eventi che richiedono una risposta distribuita geograficamente.

2.2.1 Svantaggi dell'approccio

Sebbene le premesse siano positive, è necessario presentare anche gli svantaggi portati dalla tecnologia nel suo stato attuale.

L'utilizzo della VR si sta espandendo sempre di più in alcuni settori, ma la sua adozione per molti altri risulta ancora allo stadio embrionale, dal momento che la capacità degli operatori di utilizzare i dispositivi in VR richiede molto spesso un allenamento preliminare allo scopo di diventare competenti ed efficaci nel loro utilizzo, e, spesso, ciò potrebbe precludere l'implementazione in forma estensiva, oltre al costo iniziale per la creazione degli scenari e la progettazione e realizzazione delle logiche di simulazione.

Un altro punto a sfavore della tecnologia è la mancanza di un feedback fisico paragonabile a quello reale, a meno di un utilizzo di dispositivi all'avanguardia come alcuni dei moderni guanti e tute aptiche; in generale con *Head Mounted Display* (HMD) ed i classici controller, la simulazione può trarre vantaggio solo da feedback sonori e visivi, con una limitazione degli stimoli tattili. Questa motivazione potrebbe portare a sottostimare gli sforzi richiesti nell'esercitazione sul campo [34].

2.3 Reale Virtuale per la gestione delle emergenze

Esempi interessanti di impiego della VR per la gestione delle emergenze, incluse quelle connesse al rischio idrogeologico, si sono osservate, tra gli altri, nel progetto PRODIGE. Il progetto nasce dal programma di coordinazione

transfrontaliera ALCOTRA noto con il nome di INTERREG destinato a “rafforzare la collaborazione tra le Autorità italiane e francesi preposte alla gestione delle emergenze, incrementando la capacità di eseguire efficaci operazioni di risposta congiunta ai rischi naturali e antropici attraverso la definizione di protocolli innovativi di training degli operatori basati sulla realtà virtuale” [35].

Questo progetto rappresenta un precursore al progetto PITEM RISK, nato in risposta ai cambiamenti climatici e i relativi rischi ambientali che possono portare a disastri come frane, alluvioni, dissesti idrogeologici ed altri disastri naturali. [36]. In particolare, le zone transfrontaliere risultano essere particolarmente vulnerabili a fenomeni di questo genere e per garantire la miglior cooperazione possibile tra autorità locali, nazionali ed internazionali e portare ad una pronta risposta operativa nell’assistere la popolazione e nel mettere in sicurezza le località colpite dal disastro.

A fronte di differenze in ambito linguistico, gestionale e organizzativo, si sono sviluppati strumenti in VR, tuttora a disposizione della PC piemontese e d’oltralpe per agevolare la transizione verso un nuovo approccio d’addestramento basato sulla combinazione di formazione, organizzazione e tecnologie, destinato ad “addetti ai lavori”, alle autorità e alla popolazione civile.

Le quattro applicazioni presentate sono state fonti d’ispirazione per la simulazione oggetto del progetto di tesi, la quale, come detto, si è andata a sviluppare all’interno del progetto PITEM RISK, focalizzandosi sull’intervento della PC per il rischio idrogeologico, con attenzione nello specifico sul Modulo HCP.

2.3.1 Rischio esondazione nel parco fluviale Gesso e Stura a Cuneo

Questo scenario simula un situazione di rischio alluvionale per una possibile esondazione nell’area del Parco fluviale, a Cuneo. In una situazione di allerta meteo di tipo 2 (criticità moderata) la pista ciclabile che costeggia il torrente

Gesso rimane aperta al pubblico. A seguito di un improvviso allarme meteo vengono attivate le procedure di PC per la chiusura al pubblico e l'evacuazione dell'area. L'operatore arriva sul posto e procede all'evacuazione delle persone presenti e alla chiusura degli accessi. La simulazione non offre un ampio livello di realismo dal punto di vista grafico, caratterizzata da uno scenario minimale e spoglio e non presenta nemmeno un alto grado di immersione nell'interazione con i civili.

Le tecniche introdotte da questa applicazione sono la navigazione mediante pedana omnidirezionale (Figura 2.8) e un sistema di valutazione automatica degli errori commessi.



Figura 2.9: Prova dello scenario del Parco Gesso e Stura con pedana Virtualizer della Cyberith.

2.3.2 Incidente stradale presso St. Paul-sur-Ubaye

Lo scenario presenta il recupero del conducente di un veicolo, trascinato contro un ponte a seguito di un'esondazione del fiume Riou Sec, nei pressi di St. Paul-sur-Ubaye. Le pareti del ponte impediscono al mezzo di venire trascinato via dalla

corrente mentre l'autista è bloccato all'interno della vettura. Vengono allertati i soccorsi ed un equipaggio di servizio dotato di imbracatura e corda si occupa di agganciare il veicolo, mentre il conducente viene messo in salvo da un elicottero della PC (Figura 2.9).



Figura 2.10: Dettaglio soccorsi scenario Paul-sur-Ubaye.

2.3.3 Dispersione di sostanze chimiche sul Colle della Maddalena

Lo scenario creare un incidente di un veicolo pesante che trasporta sostanze chimiche altamente inquinanti in prossimità del Colle della Maddalena, in territorio francese. Il veicolo esce dalla strada e cadendo nel fiume all'ingresso di un paese, riversa il prodotto chimico nell'acqua (Figura 2.10). Il conducente rimane all'interno della cabina, privo di sensi, finché viene messo in salvo dai Vigili del Fuoco. La sostanza chimica, si diffonde attraverso il fiume, causando un'emergenza inquinamento: vengono quindi attivate le procedure per la delimitazione dell'area colpita e per il contenimento della perdita.

Sia questo che lo scenario precedente offrono un modello georeferenziato, con la possibilità di vivere un'esperienza multiutente, la simulazione di guida di un veicolo, ampie librerie di oggetti 3D legati all'equipaggiamento degli operatori

della PC e una “cabina di regia” dove poter modificare lo scenario VR durante la simulazione.



Figura 2.11: Scenario dell'incidente sul Colle della Maddalena.

2.3.4 Nube tossica da sostanza radioattiva in avvicinamento a Torino

Il quarto scenario è costituito da due momenti consequenziali. La prima parte è presentata come un errore di manovra di un elicottero carico di torio 232, che causa un incendio, dopo essersi scontrato con una cisterna di rifornimento. Gli operatori CBRN (preposti alla gestione del rischio Chimico, Biologico, Radiologico e Nucleare) intervengono sull'incendio e delimitano la zona radioattiva, dalla quale nel frattempo si sta diffondendo una nube tossica (Figura 2.11). Il secondo momento prevede che la nube tossica investa il Parco Dora, nella città di Torino. Gli operatori di Polizia Municipale della Città di Torino, presenti sul posto per un evento che raduna centinaia di persone, vengono allertati dalla Centrale Operativa dell'arrivo della nube tossica.

Questa simulazione si contraddistingue per la grafica immersiva, la possibilità di operare scegliendo un ruolo definito, l'integrazione delle procedure di

comunicazione con i cittadini e di un software realizzato per gestire la simulazione dell'incendio.

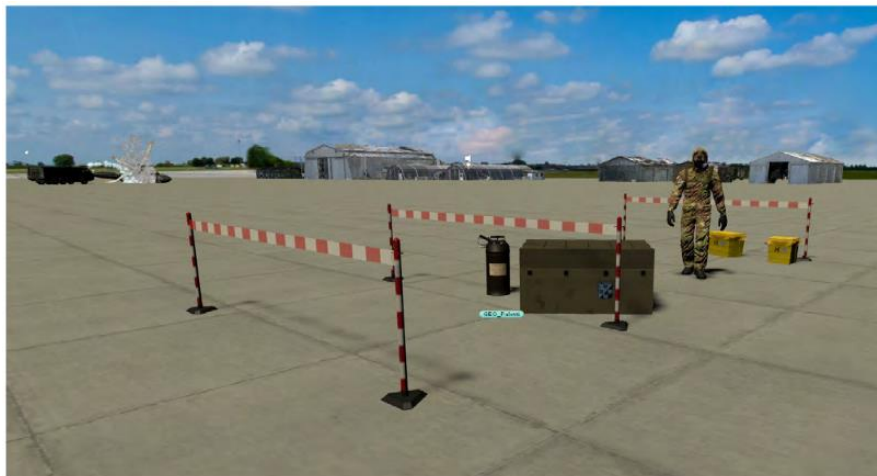


Figura 2.12: Dettaglio dell'area di lavoro nello scenario della nube tossica.

3 Tecnologie

3.1 Hardware

Il numero delle tecnologie che offre il mercato nel campo della VR è esponenzialmente aumentato negli ultimi anni, conseguenza del crescente interesse da parte delle aziende per un settore ormai emergente; ciò ha portato necessariamente ad una categorizzazione dei diversi dispositivi, in base alla tipologia di visore, ai tracker utilizzati, agli strumenti di input sensoriali, alla piattaforma di destinazione e alla complessità del sistema utilizzato.

La tecnologia VR necessita essenzialmente di un visore, uno schermo indossabile in testa definito HMD [37], di dispositivi di input per le interazioni con lo scenario, i quali possono essere indossabili come fossero guanti, dei controller tenuti in mano oppure dei sensori applicabili a parti del corpo o ad oggetti e di sensori di *motion tracking*, ovvero dispositivi per il tracciamento in tempo reale dei movimenti della testa e degli arti (denominati *base station*). Il tracciamento real-time del corpo dell'utente permette a quest'ultimo di sentirsi immerso nel mondo virtuale, percependo i vari feedback derivati dalle interazioni e godendo di una visione a 360° dell'ambiente fittizio.

I visori, in particolare, risultano essere le risorse più importanti di un sistema VR, classificabili in base alla presenza o meno di processori di calcolo complessi all'interno del dispositivo. Gli HMD che non presentano una struttura hardware complessa sono definiti *mobile HMD*, mentre i visori più evoluti dal punto di vista elettronico sono chiamati *wired HMD*; questi ultimi, essendo in genere cablati, sono caratterizzati da sensori, display in alta risoluzione e richiedono un processore di calcolo esterno a cui sono collegati (es. un computer VR-ready) diversamente rispetto ai *mobile HMD* che si appoggiano direttamente ai sensori presenti nei dispositivi mobili, quali i moderni smartphone.

I dispositivi VR più diffusi sono provvisti di controller [37], che forniscono i principali feedback aptici e garantiscono anche l'interazione con il mondo virtuale, sebbene siano presenti sul mercato strumenti di interazione più elaborati come i guanti VR, in grado di aumentare il realismo delle interazioni e l'intensità dei feedback sensoriali.

I due dispositivi VR che sono stati utilizzati nella realizzazione della simulazione oggetto del lavoro di tesi sono l'HTC Vive e l'Oculus Rift.

3.1.1 HTC Vive

L'HTC Vive [38] è un dispositivo VR wired realizzato da Valve in collaborazione con HTC e commercializzato nel 2016. Il kit disponibile sul mercato è costituito da un visore, due controller e due sensori (base station) basati su tecnologia "Lighthouse", utilizzati per tracciare la posizione e il movimento dell'utente all'interno dello spazio fisico e riprodurli nel mondo virtuale. Le base station generano nello spazio fisico un reticolo ad infrarossi, chiamato "room-scale", il quale interagendo con i 70 sensori disposti sull'HMD e sui controller, consente all'utente un movimento libero e naturale in un'area di 5 m x 5 m. I sensori presenti sull'HMD e sui controller sono di tipo ottico, con un rischio di interferenze in presenza di altri dispositivi elettronici che sfruttano tali sensori o di una terza base station.

L'HMD (Figura 3.1) 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 tramite il cavo di cui è provvisto. Le base station, allo stesso modo, richiedono di essere collegate alla corrente, offrendo inoltre un possibile collegamento tra loro con un

aumento della sincronizzazione. Infine, i controller sono caratterizzati da una tecnologia wireless e dotati di semplici batterie interne.



Figura 3.1: HMD del sistema VR HTC Vive.

Il display del visore [39] è costituito da due schermi AMOLED (*Active Matrix Organic Light Emitting Diode*) da 1080 x 1200 pixel ed è caratterizzato da una frequenza di aggiornamento (*refresh rate*) di 90 Hz, da un angolo di visuale (*field of view*) di 110° e da un paio di lenti *Frenel*⁴ non completamente circolari, allo scopo di regolare la distanza interpupillare dell'utente (*Interpupillary Distance* o *IPD*). L' HMD possiede inoltre un giroscopio, un accelerometro, sensori infrarossi per il tracking a 360° e una telecamera che, assieme alle base station, utilizza il sistema di sicurezza "Chaperone", il quale garantisce all'utente di visualizzare l'area sicura in cui può muoversi liberamente, disegnando un muro virtuale a griglie all'interno dell'ambiente virtuale.

L'HTC Vive è dotato di due controller, anch'essi tracciati e dotati di 24 sensori sull'anello superiore, che permettono le interazioni con l'ambiente virtuale;

⁴ Frenel indica una tipologia di lenti utilizzati nei campi della fotografia e di prodotti per la riproduzione video, caratterizzate da grande dimensione e piccola lunghezza focale senza, tuttavia, l'ingombro, lo spessore ed il peso del materiale necessario per costruire una lente sferica convenzionale di equivalente potere diottrico.

caratterizzati da 6 gradi di libertà (*Degree of Freedom* o DoF), 3 tasti e un touchpad cliccabile, dispongono anche di un sistema di *force feedback* che garantisce all'utente una leggera sensazione di tipo aptico. Durante la simulazione, i controller possono essere visualizzati tramite modelli che li riproducono fedelmente, oppure possono essere sostituiti da mani virtuali o oggetti 3D (Figura 3.2).

Inoltre, la tecnologia Vive fornisce separatamente i *Vive tracker*, dispositivi di tracciamento, anch'essi dotati di sensori ottici, finalizzati al tracciamento di oggetti fisici o parti del corpo, dotati di un foro a vite da 1/4" per essere agganciati su supporti fotografici.



Figura 3.2: Controller del HTC Vive.

Le base station permettono un tracking a 360° grazie all'emissione di luce ad infrarossi e 2 emettitori laser. I led ad infrarossi lampeggiano con una frequenza di 120Hz mentre i fasci laser a 60 Hz. Basandosi sulla posizione dei singoli sensori dell'HMD e dei controller, sul numero di volte che un flash ad infrarossi ha colpito il suddetto sensore e su quando questo è stato irradiato, è possibile tracciare con precisione la posizione e l'orientamento di ogni dispositivo Vive nell'area coperta dalle due base station.

OpenVR è l'SDK⁵ di riferimento rilasciato da Valve per il supporto dei propri dispositivi (la serie *Vive*, *Cosmos*, *Pro* e *Pro Eye*) e di altre tecnologie VR; *OpenVR* viene anche utilizzato come API⁶ per *SteamVR*, software distribuito insieme alla piattaforma *Steam*, la quale costituisce un'interfaccia per la gestione e la sincronizzazione di dispositivi VR (HMD, controller, base station). *Unity* e altri motori grafici permettono di includere *OpenVR* come SDK predefinito nello sviluppo di applicazioni in VR e forniscono appositi asset di risorse (*SteamVR asset*) per facilitare la creazione di una prima semplice esperienza VR da parte dello sviluppatore.

3.1.2 Oculus Rift

L'Oculus Rift (Figura 3.3) [40] è un dispositivo per la VR distribuito nel 2016 e prodotto dalla *Oculus VR*, azienda acquisita da *Facebook Inc.* nel 2014. L'Oculus consiste in un visore cablato che richiede di essere collegato al PC tramite una presa USB 3.0 e una presa HDMI, dotato di due sensori per il motion tracking che, a differenza delle base station del *Vive*, sono pensati per essere posizionati su una superficie piana (es. la scrivania) e collegati direttamente al PC tramite connessione USB.

Questa soluzione aumenta la comodità e la facilità d'utilizzo, ma comporta una limitazione in termini di tracciamento (diversamente dal *Vive*): il posizionamento dei sensori sullo stesso livello e con la presenza di una zona di occlusione implica una perdita di segnale sia vicino al terreno e in posizioni troppo elevate sia nel caso in cui l'utente copra i sensori posti sull'HMD o sui controller, dando la schiena ai sensori.

⁵ Software Development Kit, un pacchetto di strumenti per la programmazione e la documentazione di software.

⁶ Application Programming Interface, indica ogni insieme di procedure disponibili al programmatore (un SDK solitamente include le API per lo sviluppo di applicazioni su uno specifico software).



Figura 3.3: HMD e controller dell'Oculus Rift.

Il display dell'HMD [41] è costituito da due schermi OLED (*Organic Light Emitting Diode*) da 1080 x 1200 pixel ciascuno, un field of view di 110° sia in verticale sia in orizzontale, una frequenza di 90 Hz, un accelerometro e un giroscopio (caratteristiche che lo accomunano all'HTC Vive). Il peso del visore risulta essere effettivamente minore rispetto all'altro dispositivo, con 470 grammi rispetto ai 555 grammi del Vive, che forniscono all'HMD dell'Oculus Rift una maggiore comodità nelle fasi di utilizzo prolungato.

I due controller dell'Oculus sono simili a quelli dell'HTC Vive, ma la particolare forma a mezza luna dei due dispositivi (chiamati appunto “Half Moon”) è stata pensata per essere più ergonomica e per mappare meglio la posizione delle dita della mano sui diversi pulsanti. Inoltre, i due controller presentano i medesimi input presenti su altri controller per dispositivi VR (tra cui il Grip e il Trigger, mentre la levetta analogica risulta corrispondere al touchpad del Vive), sfruttati, ad esempio, nelle interazioni con i menu e in alcune modalità di navigazione come il teletrasporto presente nell'asset di SteamVR.

Infine, il dispositivo dell'Oculus VR si caratterizza per la presenza di un sistema audio integrato appositamente destinato alla fruizione di un suono direzionale 3D,

provvisto di un paio di cuffie smontabili e sostituibili (diversamente dai semplici auricolari disponibili con la tecnologia Vive).

3.1.3 Altre tecnologie sul mercato

Il mercato delle tecnologie VR presenta diversi dispositivi che, tuttavia, riportano limitazioni e non presentano tutte le caratteristiche necessarie alla realizzazione della simulazione.

Tra i dispositivi più diffusi possono essere citati, ad esempio, *PlayStation VR*, destinato a contenuti di intrattenimento disponibili sull'omonima console e i principali mobile HMD, che sfruttano gli smartphone per la fruizione delle esperienze VR, tra cui Google Cardboard e Samsung GearVR. Inoltre, sono stati distribuiti recentemente i nuovi modelli del sistema HTC Vive, chiamato HTC Vive Pro, e del Oculus Rift, l'*Oculus Rift S*.

Questi ultimi due sistemi VR, seppure presentino alcuni miglioramenti rispetto alle rispettive controparti d'azienda, non fornivano una documentazione così estesa come quella a disposizione riguardante HTC Vive e Oculus Rift, i quali, inoltre, sono due dispositivi VR disponibili presso il nodo DAUIN del laboratorio VR@POLITO, del Politecnico di Torino e maggiormente sfruttati dai dottoranti e dai tesisti del Dipartimento.

3.2 Software

3.2.1 Game engine e Unity

Nello sviluppo di una applicazione VR le componenti in gioco richiedono un'intensa organicità, dal momento che ogni fase del processo realizzativo viene accompagnata dalla gestione simultanea di più elementi, anche molto differenti, in tempo reale. In aiuto allo sviluppatore sono state create nel tempo una grande varietà di librerie software che hanno supportato la creazione di innumerevoli contenuti negli aspetti grafici (*OpenGL* e *Direct X*), di gestione della fisica (*Open Dynamics Engine*), di gestione dei contenuti audio (*DirectSound*), ecc.

Tuttavia, l'ingente costo economico e di tempo nella realizzazione di un processo di integrazione tra queste librerie, spesso molto diverse tra loro ha, portato in genere gli sviluppatori ad utilizzare uno strumento che costituisca il nucleo della componente software dell'applicazione, denominato motore grafico o *game engine*. Questo framework fornisce tutte le tecnologie di base, semplifica lo sviluppo e spesso permette al gioco o alla simulazione di funzionare su piattaforme differenti come i vari sistemi operativi, mobile o per PC, le console e i dispositivi per la VR e la Realtà Aumentata o *Augmented Reality* (AR).

Diversamente lo sviluppatore sarebbe obbligato alla realizzazione di un'applicazione in un linguaggio di programmazione (solitamente C++) che permetta di integrare i diversi componenti, con una netta diminuzione dell'ottimizzazione ed un alto rischio di errori durante la programmazione.

Inoltre, l'utilizzo di un game engine permette non solo di ridurre i costi, le tempistiche e i possibili errori, ma anche di avere a disposizione un'interfaccia su cui poter gestire i contenuti visivi, le sorgenti audio, i sistemi particellari, i VFX⁷

⁷ *Visual Effects* (VFX) è un termine definisce i vari processi con cui un'immagine è creata e/o modificata, fuori dal contesto di una ripresa live action, generata attraverso la *Computer-Generated Imagery* (CGI).

e altri elementi fondamentali come gli avatar virtuali, limitando potenzialmente la componente di programmazione al solo sviluppo logico dell'applicazione.

Unity [42] è il motore grafico sviluppato da *Unity Technologies* e distribuito a partire dal 2005, scelto per la realizzazione della piattaforma oggetto del lavoro di tesi. Inizialmente pensato per lo sviluppo di applicazioni esclusivamente su macOS, successivamente è stato distribuito come un game engine multipiattaforma in grado di realizzare prodotti in 2D, 2.5D e 3D e contenuti in VR ed AR. Unity [43] permette di integrare contenuti audio, video, elementi 3D e script in C# nello stesso prodotto e, inoltre, essendo caratterizzato da un'interfaccia personalizzabile ed intuitiva, facilita la gestione delle risorse attraverso l'uso del *drag & drop* e di specifici pannelli di controllo.

Sebbene nato in ambito videoludico, Unity è spesso utilizzato come motore grafico nella realizzazione di applicazioni nell'ambito dell'animazione, del cinema, dell'architettura e del design industriale e di interni. Il software supporta numerose piattaforme tra cui Android, Windows, iOS, macOS, PlayStation 4, Xbox One, Oculus Rift, HTC Vive, Gear VR, Microsoft HoloLens, Apple TV e Samsung Smart TV.

Le sezioni principali che compongono l'interfaccia del programma (Figura 3.4) sono:

1. il pannello *Project*, che permette di organizzare i file (asset) importati nel progetto;
2. la *Hierarchy*, ovvero l'elenco degli asset presenti nella scena corrente del progetto;
3. l'*Inspector*, il quale permette di visualizzare e modificare gli attributi di un elemento della scena; tali attributi spaziano dalle componenti fisiche come la posizione, la rotazione e la scala, alle collisioni real-time, materiali,

texture e illuminazione, garantendo anche l'assegnazione di attributi *custom*, sviluppati dal programmatore (script in C#);

4. la *Scene*, la quale rappresenta l'aspetto visivo e grafico degli oggetti 3D o 2D che sono indicati nella *Hierarchy*; l'utente può modificare le caratteristiche fisiche di ognuno di essi, muoversi nello spazio 3D attraverso le modalità di navigazione comuni ai principali programmi di grafica computerizzata e VFX come Maya, Blender, Nuke e Houdini; quando viene caricata una scena, tutti gli oggetti attivi nella *Hierarchy* sono allocati in memoria e vengono caricati nella *Scene*, sostituendo gli oggetti presenti nella scena precedente;
5. il pannello *Game*, che è rappresentativo del prodotto finale realizzato dallo sviluppatore, permette a quest'ultimo di testare velocemente la propria applicazione, direttamente all'interno del motore grafico.

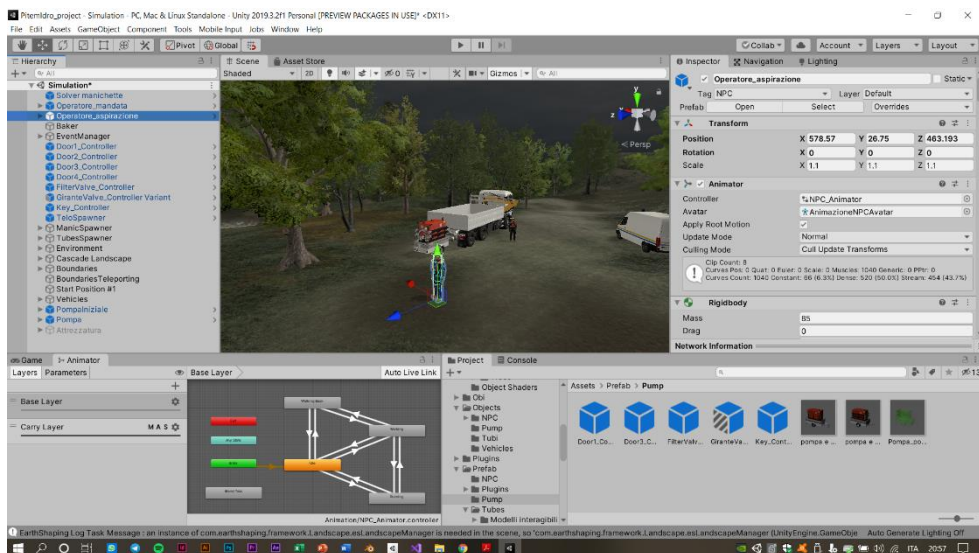


Figura 3.4: Interfaccia di Unity 2019.3.2.

Il software è caratterizzato da quattro paradigmi su cui si basa l'intera logica di sviluppo: *GameObject*, *component*, *prefab* e *script*. Il primo concetto definisce

ogni elemento presente in una scena, che sia un avatar, un elemento dello scenario, un effetto particellare, una luce o una camera. Ogni GameObject viene creato direttamente in scena e modificato progressivamente tramite l'aggiunta di component, attributi assegnati a ciascun oggetto, i quali possono essere build-in di Unity oppure creati dall'utente tramite script. Successivamente, ogni GameObject può essere salvato in un prefab, file di progetto salvato e disponibile nelle cartelle del pannello Project per essere nuovamente reimportato in una qualsiasi scena.

Gli script assegnati all'interno di Unity vengono scritti in C#, linguaggio di programmazione che, insieme a C++, costituisce lo standard de facto dell'industria videoludica, con una grande versatilità ed una documentazione particolarmente accurata (lo stesso sito di Unity presenta una sezione in cui è riportata la documentazione delle proprie librerie di codice per lo sviluppo di applicazioni sul proprio game engine).

Inoltre, Unity dispone di un proprio store chiamato *Asset Store* [44], che consiste in un ampio database di modelli, prefab e script sia gratuiti sia a pagamento, in grado di supportare enormemente gli sviluppatori nella realizzazione dei progetti più complessi. Dall'*Asset Store* sono stati reperiti gli asset di risorse utilizzati per la creazione dello scenario e la gestione della IK e della fisica dei tubi.

3.2.2 Visual Studio

Unity si appoggia ad altri programmi esterni sia per la produzione di contenuti grafici (modellazione, *texturing*, *rigging* e animazione 3D) sia per la scrittura del codice e la componente di debugging.

Durante la fase di realizzazione della simulazione l'aspetto relativo al coding è stato sviluppato attraverso *Visual Studio* (nella versione 2019) [45], un ambiente di sviluppo integrato o *Integrated Development Environment* (IDE) rilasciato da Microsoft, il quale permette di programmare in vari linguaggi di

programmazione, tra cui C++, C#, Visual Basic.NET, JavaScript, XML, HTML 5 e CSS.

Il software è caratterizzato da funzionalità complete per la scrittura di codice, il debug, il test e la distribuzione su qualsiasi piattaforma che sia PC, console o mobile, offrendo inoltre estensioni per la scrittura in altri linguaggi di programmazione tra cui Python e Node.js e *add-on* che facilitano il collegamento con altri programmi esterni, tra cui lo stesso Unity.

Questo motore grafico, infatti, permette di impostare Visual Studio come IDE predefinito e di semplificare la fase di coding attraverso librerie specifiche di Unity e un debug run-time. Inoltre, la compatibilità con Git per la condivisione del progetto ha facilitato anche l'integrazione dei contenuti durante le varie fasi di realizzazione.

3.2.3 Blender

La scelta di un software per lo sviluppo dei modelli e delle animazioni è ricaduta su *Blender* (versione 2.82) [46], software multipiattaforma open source per la modellazione, il texturing, il rigging, l'animazione 3D e 2D ed il *rendering*.

Il software è caratterizzato da un'interfaccia personalizzabile e divisa in pannelli (Figura 3.5), similmente a Unity e ad ogni altro software di grafica 3D, tra cui il pannello di navigazione nella scena o *viewport*, l'interfaccia a nodi per la gestione dei materiali (*shading*), il pannello con le funzionalità di *UV mapping* e texturing, la *timeline* per l'animazione tramite *keyframe* e il pannello di rendering. Blender non costituisce solamente uno strumento per la creazione e l'animazione di elementi 3D ma supporta l'intera *pipeline* di realizzazione di contenuti in grafica computerizzata, comprese le fasi di *compositing*, simulazione di sistemi particellari e VFX, motion tracking, montaggio video e, inoltre, comprende una *pipeline* di illustrazione e animazione 2D.

Nel contesto della realizzazione della simulazione oggetto di tesi, Blender è stato utilizzato per la creazione dell'attrezzatura del modulo HCP, in particolare la motopompa Godwin, il modello dell'operatore NPC, le animazioni principali di quest'ultimo, la fase di UV mapping e di *retopology*⁸ dei modelli. Si è scelto di utilizzare la versione 2.82 perché si tratta dell'ultima versione stabile rilasciata dal sito ufficiale.

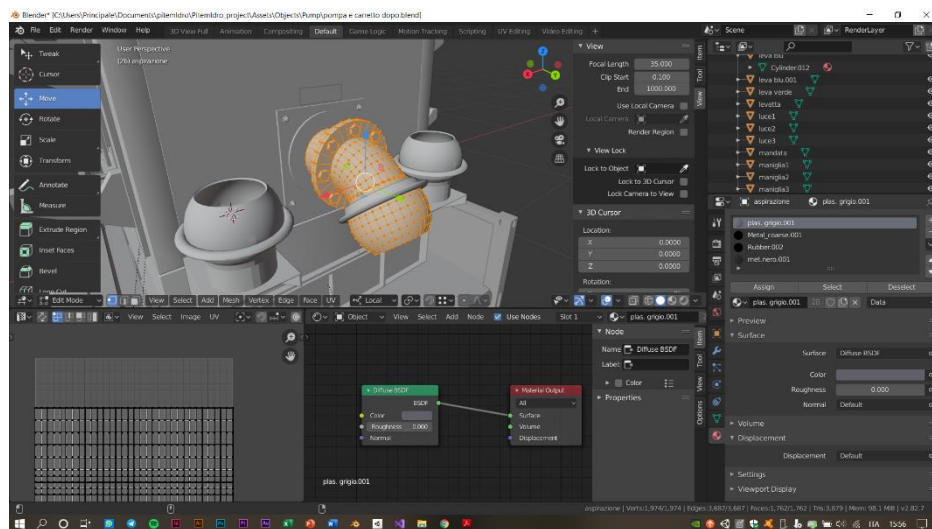


Figura 3.5: Interfaccia di Blender 2.82.

3.2.4 MakeHuman e Mixamo

MakeHuman [47] è un'applicazione open source per la realizzazione di prototipi di umanoidi in computer grafica 3D, che è stato sfruttato durante il lavoro di tesi per la creazione dei modelli di operatore con il ruolo di NPC. Il software è caratterizzato dall'utilizzo del *morphing* come tecnologia per la trasformazione di un modello umano standard in una gran varietà di personaggi, ottenuti attraverso interpolazione lineare.

⁸ Lavoro di trasformazione di modelli 3D ad alto numero di poligoni (high-poly) in modelli con un minor numero (low-poly) allo scopo di migliorare le prestazioni oppure per essere utilizzati nell'animazione.

Mixamo [48] consiste in un servizio online di animazioni 3D sviluppato dall'azienda omonima, attiva nel campo della grafica computerizzata ed acquisita da Adobe System nel 2015. La piattaforma web fornisce un vasto database di animazioni 3D, scaricabili ed importabili in altri programmi esterni come Autodesk Maya, Blender, 3ds Max e altri; il sito offre anche la possibilità di importare al suo interno un modello umanoide realizzato dal modellatore e di creare uno *rig*, ovvero uno “scheletro” associato al modello a cui verranno automaticamente applicate run-time le varie animazioni disponibili.

4 Progettazione

Il capitolo presenta le fasi di progettazione della simulazione a partire dallo studio del materiale raccolto nei vari incontri con la PC e durante le esercitazioni a cui il gruppo di lavoro ha partecipato, fino alla scrittura dell'esperienza di addestramento, il voice-over, riportando gli asset utilizzati per la realizzazione dei contenuti.

4.1 Addestramento dei volontari della Protezione Civile

I volontari vengono formati nella messa in operatività del Modulo HCP svolgendo esercitazioni pratiche, coordinate da volontari più esperti oppure da membri del Dipartimento di Protezione Civile regionale, anch'essi distinti da un'esperienza nella gestione delle emergenze e nell'utilizzo delle risorse del modulo.



Figura 4.1: Esercitazione EU Modex, tenutasi in Montenegro nel 2019.

Queste esercitazioni sono svolte, in alcuni casi, nelle zone vicine ai presidi della PC, presso un fiume di medie dimensioni (come il fiume Dora Baltea in Ivrea,

dove è stata documentata una prova sul campo) oppure in luoghi adibiti alle esercitazioni di volontari in cui essi vengono formati sulle varie attività pratiche, tra cui il montaggio delle manichette, la gestione del pannello di controllo della pompa, il posizionamento dei teli di protezione, ecc. In altre occasioni, i Dipartimenti nazionali di Protezione Civile degli stati membri dell'UE organizzano esercitazioni internazionali [11] in cui diverse squadre partecipano a giornate di formazione sulle procedure HCP.

Un'ulteriore occasione per istruire i volontari risultano essere eventi particolari, mensili o annuali (come la Settimana nazionale della Protezione Civile), che coinvolgono non solo operatori addestrati ad operare all'interno del Modulo HCP, ma anche volontari destinati ad altri Moduli, come il soccorso alla popolazione, AIB (Anti Incendi Boschivi) e la gestione delle risorse della Colonna Mobile (tende, officine mobili, ecc.).

4.1.1 Risorse del Modulo HCP

L'attrezzatura disponibile per gli operatori HCP è costituita, in parte, da strumenti che si trovano anche in altri Moduli della PC per l'intervento idraulico, in parte da risorse specifiche destinate ad operazioni in cui è necessario un ingente lavoro di pompaggio (nei casi di allagamenti di ampie aree urbane, piene di grandi fiumi, ecc.).

La motopompa scelta per la simulazione è la Godwin 150m (Figura 4.2), poiché utilizzata più spesso di altre motopompe come ad esempio la Albatross, la quale è caratterizzata da una portata d'acqua superiore, ma anche da una maggiore difficoltà nelle fasi di trasporto e scarico, ciò dovuto alla sua dimensione e al peso.



Figura 4.2: Confronto tra la motopompa Godwin e il modello 3D, realizzato per la simulazione.

Nell'elenco seguente sono riportati i principali componenti del modulo, scelti per essere introdotti nella simulazione:

- motopompa Godwin CD 150m, con diametro delle aperture di 150 mm e rimorchio (come dispositivo per il pompaggio dell'acqua);
- camion con gru (per il trasporto delle manichette appiattibili e dei tubi lunghi);
- officina mobile (per il trasporto dell'attrezzatura minore);
- manichette appiattibili e semirigide (per le operazioni di trasporto e scarico dell'acqua);

- tubi rigidi curvi e dritti (per il pompaggio dell'acqua in presenza di ostacoli, dossi e curve strette);
- strumento di supporto, come un tubo ricurvo o una chiave inglese (per aiutare gli operatori nella fase di aggancio delle manichette);
- filtro pescante (per l'aspirazione dell'acqua nella zona allagata);
- telo blu e picchetti (per la protezione della zona di mandata dal getto d'acqua erogato dalla motopompa);
- paletti e nastro (per la delimitazione dell'area operativa).

4.1.2 *Obi Rope*

La realizzazione dei modelli 3D delle manichette e dei tubi hanno comportato un notevole lavoro nella calibrazione delle caratteristiche fisiche, che avrebbero condizionato largamente l'interazione con esse da parte degli utenti, durante la simulazione. Nella fase di progettazione, si è deciso di utilizzare un asset apposito per la gestione della fisica dei tubi, chiamato *Obi Rope* [49], il quale permette di controllare effetti fisici quali collisioni, l'elasticità e il grado di distorsione di ciascun componente, facilitando enormemente le operazioni di calcolo durante la simulazione.

4.1.3 Operazioni da svolgere per la messa operativa

Le esercitazioni organizzate dalla PC e le visite ai presidi di Druento e Vercelli hanno permesso di definire un quadro completo delle procedure da svolgere per la messa in operatività del modulo. La simulazione è stata progettata a partire dalla raccolta di informazioni essenziali [7] per definire le singole operazioni da compiere per portare a termine ciascuna delle attività sotto riportate:

- posizionamento della motopompa e dell'attrezzatura nell'area operativa;
- delimitazione dell'area con nastro segnaletico per impedire l'ingresso ai civili;
- montaggio delle manichette e dei tubi di aspirazione e di mandata;
- posizionamento del filtro e del capo finale della mandata nelle rispettive zone d'intervento;
- controllo delle valvole della girante e del filtro e di altri parametri della motopompa;
- avviamento pompa e spegnimento della motopompa;
- controllo manichette e tubi di aspirazione e di mandata durante il pompaggio, per individuare eventuali perdite o per rimediare ad errori commessi;
- apertura dei tubi e delle manichette per la fase di smontaggio dell'attrezzatura.

4.2 Creazione dello scenario



Figura 4.3: Scenario virtuale creato per la simulazione.

Inizialmente, lo scenario pensato come l'ambientazione della simulazione consisteva nel Nodo Idraulico di Ivrea, dove, appunto, vengono svolte le esercitazioni dei volontari addestrati all'uso di motopompe e alle risorse del Modulo idraulico e del Modulo HCP.

Successivamente, si è ritenuto più opportuno procedere con la creazione di uno scenario ad hoc, che non rappresenta alcun luogo reale ma che costituisce un'ambientazione generica e scalabile, allo scopo di fornire un'applicazione non destinata ad un unico nucleo di PC regionale o nazionale, ma con una potenziale estensione anche ad altri fruitori. Lo scenario è costituito da una radura in mezzo alle montagne, la quale costituisce l'area operativa dei membri del Modulo, i cui confini sono costruiti a partire da un fiume di medie dimensioni e da un torrente sul versante opposto (Figura 4.3).



Figura 4.4: Il fiume e la pozza d'acqua nell'ambientazione 3D.

Il fiume viene presentato come in condizioni di piena, con gli argini scavati in un punto del terreno, tanto da creare una pozza d'acqua, la quale potenzialmente rischia di erodere ed allargare ulteriormente i suddetti argini. Il torrente, invece, rappresenta il punto di mandata o scarico, dove l'acqua della pozza (la quale consiste nel punto di aspirazione della motopompa) verrà rigettata in modo da scongiurare un danneggiamento dal punto di vista idrogeologico del territorio circostante.

L'area di operazione dei volontari non risulta caratterizzata da ostacoli o zone con pendenze o buche, poiché lo scopo è quello di riprodurre un luogo idoneo ad una prima esercitazione sul campo. Questa scelta permette di focalizzare l'apprendimento sulle procedure operative per l'utilizzo delle risorse in condizioni di sicurezza, allo scopo di comprendere, in modo completo, l'intero processo e le norme richieste per l'utilizzo corretto dell'attrezzatura e della motopompa.

4.2.1 Asset utilizzati

Per la realizzazione dello scenario si è ricorso ad un *package* (ovvero un pacchetto di risorse e tool con svariate funzioni) chiamato *Terrain* [50], un set build-in disponibile sul motore grafico con cui è stata sviluppata la simulazione, e ad un asset denominato *Cascade* [51], scaricato dallo store ufficiale di Unity, con cui sono stati realizzati graficamente il fiume e il torrente di scarico. In un secondo momento, si è passato alla creazione dei dettagli in grado di aumentare il realismo dell'ambiente virtuale, come gli alberi, le rocce e le texture, successivamente raccolti ed aggiunti attraverso il Terrain Editor offerto da Unity o manualmente nella scena 3D; alcuni di questi modelli sono stati realizzati attraverso il software Blender, mentre altri sono stati raccolti da asset gratuiti disponibili sull'Asset Store.

4.3 Logica della simulazione

Il processo di creazione della simulazione, dal punto di vista della struttura e degli obiettivi finali, è stato accompagnato dalla realizzazione di un documento di design, finalizzato all'interfacciarsi con la PC, presentando i gradual progressi raggiunti, gli scopi finali dell'applicazione e le fasi che caratterizzano l'esperienza. Inoltre, è stato creato uno storyboard per rappresentare visivamente le operazioni da svolgere, e l'ordine con cui le varie procedure vengono apprese e sperimentate durante la simulazione.

Il singolo utente impersona un membro della squadra HCP, il quale, interagendo con altri operatori, è chiamato ad intervenire presso un argine di un fiume in piena in uno scenario boschivo, con l'obiettivo di mettere in sicurezza l'area.

Le operazioni di maggior rilievo nell'intero processo di messa in operatività del modulo sono, senza dubbio, le operazioni di montaggio delle manichette per la creazione di due catene colleganti il punto di aspirazione, la motopompa utilizzata e il punto di mandata. La procedura completa risulta in una sequenza di azioni ripetute, con la necessità da parte del volontario di allenare l'attenzione, la prontezza e la coordinazione nel trasporto e nell'interazione con i componenti dei tubi, piuttosto che soffermarsi sulla tecnica e la memoria. Infatti, le pratiche da tenere presente per ogni volontario novizio sono semplici e molto meccaniche, non richiedendo uno sforzo mentale particolarmente elevato nella memorizzazione e nella precisione, ma piuttosto una grande prontezza e velocità di esecuzione.

4.3.1 Struttura

Le operazioni da svolgere durante l'intervento si possono dividere in quattro macro-fasi, che comprendono:

1. Il sopralluogo e la fase in cui gli operatori indossano i rispettivi DPI;
2. le azioni di posizionamento delle attrezzature, della pompa e delle risorse richieste per l'intervento;
3. il sopraccitato montaggio delle manichette e dei tubi di aspirazione e di mandata;
4. le operazioni di accensione della motopompa, di controllo della fase di pompaggio, di intervento in caso di pericoli, danneggiamenti alle risorse o perdite d'acqua e, infine, lo smontaggio e sgombrò dell'area da parte della squadra.

Questa suddivisione in macro-fasi non rappresenta una suddivisione per i volontari della PC dei compiti e delle operazioni da imparare per poter prendere parte agli interventi, poiché essi devono essere formati su ciascuna delle procedure mostrate. Tuttavia, è stato preso in considerazione l'utilizzo di tale suddivisione allo scopo di decidere a priori, in fase di inizio della simulazione, i compiti da assegnare all'utente, allo scopo di diminuire la durata dell'esperienza VR e di focalizzare l'attenzione sull'apprendimento di una mirata procedura. La scelta delle azioni da compiere è assolutamente libera, affinché l'utente possa decidere di occuparsi del montaggio delle manichette di aspirazione o di mandata, della scelta della posizione per la motopompa oppure del controllo delle perdite durante la fase di pompaggio, senza doversi preoccupare delle restanti procedure.

La simulazione comprende due versioni distinte:

- Una versione guidata, caratterizzata da una voce fuoricampo, rappresentata da un membro esterno della PC che impartisce all'utente gli ordini da eseguire per rendere operativa la motopompa. Svolgendo l'intera sequenza di azioni richieste, l'utente viene formato su tutte le procedure e le corrette norme da seguire, con una spiegazione passo-passo dell'utilizzo dell'attrezzatura e la gestione delle risorse in campo;
- Una versione valutata, che consiste in un'esperienza libera di interazione all'interno di uno scenario identico alla precedente versione, ma senza una voce fuoricampo che illustra le azioni da svolgere. Le indicazioni fuoricampo sono sostituite da quelle impartite dal caposquadra del modulo HCP. Il caposquadra, ruolo ricoperto da un NPC, permette all'utente di immergersi in un'esperienza molto simile ad una vera esercitazione sul campo, in cui ogni volontario opera in collaborazione con altri operatori, in una situazione di completa autonomia e libertà di azione, senza vincoli e indicazioni esterne su come agire.

La differenza sostanziale nella versione valutata consiste proprio nella assenza di segnalazioni di errori durante l'operato dell'utente, il quale verrà valutato al termine della simulazione in base a specifiche norme e parametri decisi in accordo con i membri della PC stessa.

4.4 SteamVR

La componente di VR è stata progettata a partire da un pacchetto di risorse disponibile direttamente sullo store di Unity: Steam VR [52]. Come detto, l'asset è stato rilasciato dalla Valve (azienda produttrice dell'HTC Vive) come supporto tecnico alla realizzazione di applicazioni VR, senza limitazioni ad alcun dispositivo, allo scopo di fornire script, prefab e altre risorse utili per le interazioni, la locomozione e la componente grafica dell'applicazione.

Valve distribuisce gratuitamente questo plugin su Unity per interfacciarsi con il software omonimo, anch'esso proprietà della Valve, il quale permette a sviluppatori e giocatori di fruire dei contenuti VR con un'interfaccia semplificata e un maggior controllo dei propri dispositivi hardware. Ciascun sviluppatore può scegliere come target l'API di Steam VR (OpenVR), a cui si collegano i propri dispositivi, offrendo una maggiore intuitività e facilità d'uso nella gestione di tre elementi principali di una simulazione VR: caricare modelli 3D per i controller VR, gestire gli input e stimare le caratteristiche fisiche e spaziali delle mani dell'utente, durante l'utilizzo di tali controller. Oltre alla gestione di tali elementi, SteamVR fornisce scene di esempio, presentando le varie possibilità di interazione pratica con il mondo virtuale di cui questo asset dispone.

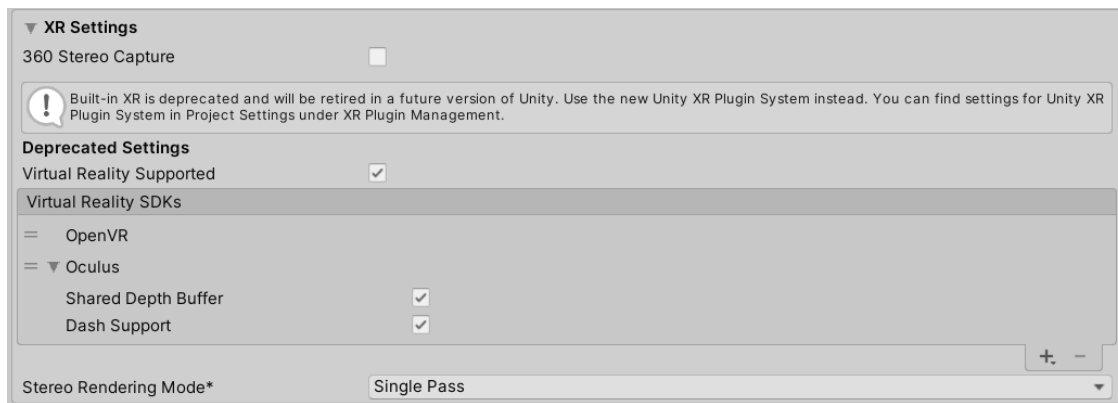


Figura 4.5: Le impostazioni della componente VR in Unity 2019.3.2.

L'interfaccia di Unity consente di impostare OpenVR come libreria di default nella gestione della componente VR di un progetto, fornendo, inoltre, una finestra di collegamento diretto per la mappatura dei tasti di ogni dispositivo. Infatti, partire dalla versione 2.0, l'API definisce una nuovo tool di controllo delle interazioni con i controller, gli Action Sets [53], i quali garantiscono agli sviluppatori una libertà assoluta nella personalizzazione delle proprie impostazioni per le interazioni nel mondo virtuale; il programmatore si può limitare a salvare il set di azioni preimpostate oppure accedere ad un'interfaccia online apposita (*Input Binding UI*, Figura 4.6) dove, con semplici comandi intuitivi, può associare ogni tasto del controller ad un azione particolare.

Utilizzare SteamVR è stata una scelta portata principalmente da necessità logistiche, dal momento che la documentazione riguardante questo specifico asset risulta molto consistente ed è presente una folta schiera di sviluppatori in grado di fornire consigli e supporto tecnico a riguardo. In secondo luogo, le applicazioni di riferimento realizzare in passato da altri tesisti e dottorandi del laboratorio sono realizzate sfruttando questo asset, anziché altri disponibili, come Oculus VR⁹.

⁹ OculusVR è l'API di riferimento per lo sviluppo di applicazioni VR su Unity, utilizzando esclusivamente tecnologia Oculus.



Figura 4.6: Input Binding UI di SteamVR.

La tecnologia VR ha definito le modalità di interazione fisica che l'utente ha con le risorse del modulo HCP durante la simulazione, oltre a permettere la navigazione all'interno della scena. In particolare:

- la locomozione utilizza il tasto *grip*¹⁰;
- il sollevamento di oggetti, la rotazione ed il rilascio a terra utilizzano il tasto *trigger*;
- le interazione con leve, porte, valvole e chiavi sfruttano il tasto *trigger*;
- le interazione con il menù, all'inizio e al termine della simulazione sono compiute attraverso l'uso dei Raycast¹¹ e del tasto *trigger*;

¹⁰ I tasti riportati sono propri della tecnologia HTC Vive. La tecnologia dell'Oculus Rift utilizza una mappatura dei tasti corrispondente, con nomi differenti.

¹¹ Ray casting corrisponde a una tecnica di intersezione raggio-superficie sfruttata nella computer grafica 3D per il calcolo delle collisioni, il rendering volumetrico, il calcolo delle occlusioni e altri scopi.

4.5 Interfaccia della simulazione

Il menù iniziale permette all'utente di selezionare la versione della simulazione in cui potersi esercitare, scegliendo tra la versione guidata e la versione valutata, con una possibile scelta tra le operazioni che si vogliono svolgere all'interno dell'esperienza immersiva, garantendo una buona divisione dei compiti, che il caposquadra assegnerà formalmente durante la simulazione.

L'interfaccia progettata per la simulazione si basa prevalentemente sui feedback di tipo uditivo, che costituiscono, nella realtà, il principale strumento di comunicazione da un membro all'altro della squadra di volontari, così come verso membri esterni, come, ad esempio, la sede della PC o i membri di un altro Modulo della Colonna Mobile.

Perciò, l'utente comunica in due modalità differenti, ma che richiedono sempre l'uso di dispositivi audio come cuffie, auricolari o casse. Nella versione guidata della simulazione, l'utente si interfaccia, attraverso un walkie talkie, con un operatore esterno (lo ascolta), il quale gli fornisce le indicazioni dettagliate su come agire nelle varie procedure di messa in operatività della motopompa e del resto del modulo. Questa versione è caratterizzata da una comunicazione il cui scopo è insegnare al volontario come muoversi nella scena, come trasportare e maneggiare gli oggetti presenti nello scenario, facilitando l'apprendimento dell'utente attraverso segnalatori 3D presenti nell'ambientazione per il posizionamento dell'attrezzatura e le zone dove spostarsi. I potenziali errori commessi da parte dell'utente durante l'esperienza sono immediatamente rilevati dal sistema e prontamente sottolineati, sempre attraverso clip audio trasmesse dall'applicazione, allo scopo di richiamare l'utente a correggere il proprio comportamento e a formarlo su possibili situazioni pericolose.

Nella versione valutata, invece, il volontario che fruisce l'esperienza viene "trasportato" in una situazione realistica in cui il caposquadra del modulo HCP impartisce una serie di istruzioni preliminari sulle operazioni da svolgere,

suddividendo i compiti tra i membri della squadra e lasciando, successivamente, libertà assoluta agli utenti nel decidere su come operare durante varie fasi di intervento. Al termine della simulazione valutata, verrà riportato il resoconto delle operazioni svolte, con particolare attenzione al rispetto delle procedure da seguire, agli errori che sono stati commessi e alle dimenticanze avvenute durante l'intervento.

4.6 Progettazione NPC

Lo sviluppo di una simulazione ad utente singolo ha portato alla realizzazione di una struttura logica in grado di gestire la presenza di operatori controllati dal sistema (NPC), i quali sono destinati a ricoprire i ruoli degli altri operatori che partecipano all'intervento insieme all'utente.

Per tali operatori si sono creati dei modelli 3D, rappresentanti nell'aspetto i volontari della PC. Questi modelli, realizzati attraverso Blender, sono stati sviluppati a partire da un modello umano precostruito, dal momento che non era richiesta una fisionomia precisa e l'utilizzo di modelli senza tratti fisici particolari avrebbe reso scalabile il prodotto di partenza nell'ottica di altre simulazioni simili e in diversi contesti di sviluppo.

Il modello è stato creato a partire dalla documentazione che è stata raccolta durante le visite alle sedi della PC della Regione Piemonte. In particolare, sono state prese come riferimento le numerose fotografie scattate ai volontari che hanno partecipato alle esercitazioni e ai manichini d'esposizione (Figura 4.8), il cui scopo è proprio quello di illustrare l'equipaggiamento personale che ogni operatore deve necessariamente possedere al momento della partenza per un qualsiasi intervento sul campo. Tali strumenti destinati alla salvaguardia e alla sicurezza della persona sono definiti Dispositivi di Protezione Individuale (DPI).

L'insieme dei dispositivi che caratterizzano il DPI di un operatore di un squadra della PC comprende:

- caschetto con logo;
- giubbotti di salvataggio;
- moschettone, per fissarsi a corde di sicurezza;
- scarponi;
- tuta con strisce catarifrangenti;
- radio;
- torcia da casco a LED.



Figura 4.7: Equipaggiamento di un volontario della Protezione Civile, Regione Piemonte.

All'interno della simulazione, l'NPC che impersona un operatore deve interagire con gli elementi dello scenario in maniera verosimile rispetto a come agirebbe in una situazione reale. Perciò, le infinite posizioni in cui il modello 3D solleva e trasporta un oggetto nello scenario virtuale non permettono la gestione della cinematica delle mani, delle braccia e delle gambe attraverso animazioni 3D

preimpostate, dal momento che richiederebbero un enorme lavoro nella disposizione del corpo e degli arti in base alle varie tipologie di elementi con cui il modello andrebbe ad interagire. Inoltre, l'orientamento nello spazio 3D del modello non sempre permette di sfruttare un calcolo diretto delle giunzioni del corpo per definire la posizione e la rotazione delle varie parti di esso (arti, busto e testa), soprattutto se si considera che le azioni da svolgere sono calcolate in tempo reale.

4.6.1 *Inverse Kinematics (IK) e Final IK*

Su queste basi, si è deciso di controllare le animazioni dei modelli attraverso la IK, che sfrutta i punti terminali di una catena, come può essere una mano rispetto ad un braccio, per raggiungere la posa desiderata. Nell'animazione tridimensionale si può calcolare il movimento di tutti i rami di una struttura articolata, in base alla posizione iniziale (posa iniziale) e finale (posa desiderata) di un'estremità dell'oggetto. La Cinematica Diretta costituisce, invece, un metodo opposto, dal momento che richiede la pianificazione di ogni movimento per ogni parte che compone un oggetto.

La Figura 4.9 riporta un esempio perfetto su come si può sfruttare la IK per il controllo di un braccio composto da 6 giunti e 5 segmenti di lunghezza differente. Il calcolo della posizione del giunto finale (*End-Effector*), con lo scopo di raggiungere il *Target*, viene ottenuto identificando un angolo per ogni giunto e adottando soluzioni numeriche per le posizioni dei punti intermedi; il calcolo spesso riporta più configurazioni a parità di *End-Effector*. L'uso della IK per le animazioni di mani o piedi, perciò, rivela una difficoltà nell'ottenere una posizione ricercata evitando soluzioni non realistiche e problematiche nell'individuare una situazione in cui risulta impossibile trovare una soluzione.

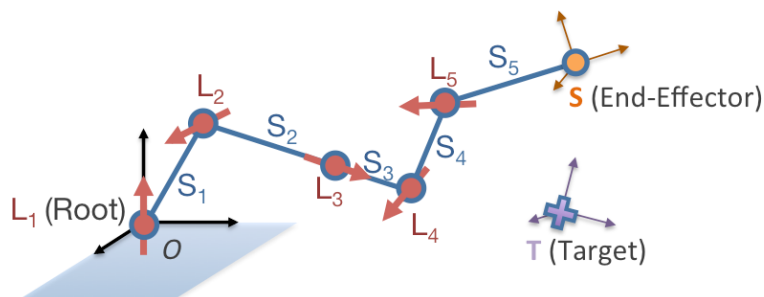


Figura 4.8: Esempio in cui è vantaggioso l'uso della IK.

Una soluzione procedurale per ottenere un risultato finale appropriato sfrutta una considerazione di base: gli angoli più lontani dall'End Effector (ovvero dalla mano) siano quelli che maggiormente ne influenzano la posizione, mentre giunti più vicino ad esso sono usati per impartire “regolazioni fini”. l'effetto di ogni grado di libertà è calcolato mediante il prodotto vettoriale tra l'asse di rotazione del giunto e il vettore che congiunge il giunto stesso all'End-Effector. Si introduce, inoltre, una regola empirica che prevede che lo sviluppatore specifichi un piano sul quale far giacere il giunto della spalla e del gomito.

Nel caso specifico degli NPC presenti nel progetto di tesi, il piano viene calcolato automaticamente a partire da un rig¹², generato, a sua volta, a partire dalla mesh¹³ del modello importato, facendo uso di una apposita libreria di script.

Unity fornisce numerosi asset sul proprio store online per la gestione della IK, tra cui *Final IK* (Figura 4.10) [54], una libreria di scripts per il controllo della IK prodotta dalla *RootMotion* [55] e molto diffusa tra gli sviluppatori che utilizzano questo sistema grafico. *Final IK* è stato utilizzato per il controllo delle mani del modello, in modo da rendere più realistico il posizionamento delle dita sull'oggetto preso e di semplificare il lavoro, limitando la creazione delle animazioni ai movimenti di corsa, camminata e *idle* (stato di riposo dell'NPC).

¹² Il rig o skeleton è un insieme gerarchico di ossa interconnesse, impiegato nell'animazione digitale per controllare in modo più intuitivo il movimento nello spazio di un modello 3D.

¹³ La mesh o skin è la rappresentazione superficiale utilizzata nella Computer Grafica per disegnare un personaggio o un oggetto.

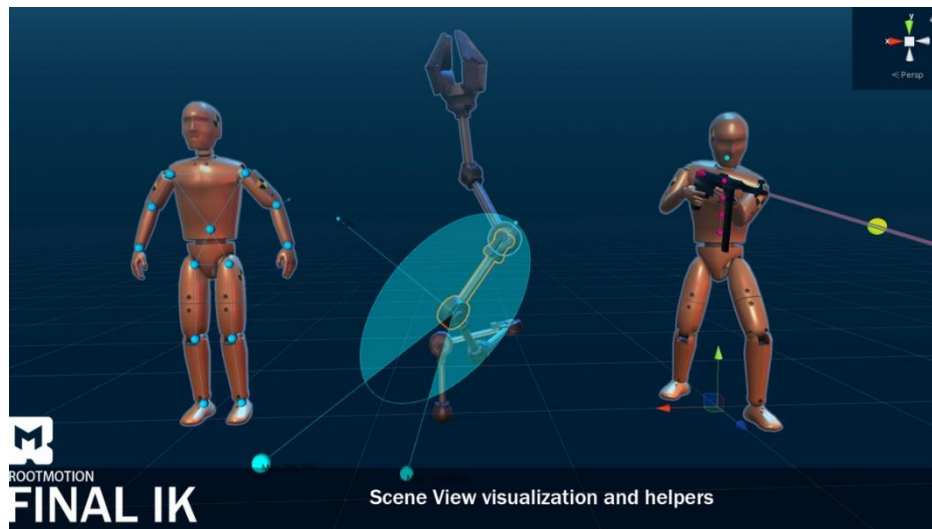


Figura 4.9: Final IK, asset per la gestione dell'IK in Unity.

4.6.2 Macchine a Stati Finiti e NavMesh

La simulazione ha richiesto un lavoro coordinato tra diversi operatori in contemporanea, dove l'utente ha bisogno del supporto di un'altra persona per il trasporto dell'attrezzatura e il montaggio dei tubi. Perciò, i modelli 3D sono stati utilizzati, attraverso il sistema di Intelligenza Artificiale o *Artificial Intelligence* (AI) che fornisce Unity, nella creazione di personaggi automatizzati, le cui azioni sono dipendenti da come l'utente interagisce con l'ambiente della simulazione

Al fine di sviluppare una logica realistica di azioni che un NPC può compiere in maniera conforme a quanto svolto dall'utente, sono state sviluppate delle FSM che gestiscono lo "stato" temporaneo in cui un NPC si trova prima che uno stimolo o *trigger* cambi questo stato. Un esempio di trigger può appunto essere un'azione dell'utente, come il sollevamento di un capo di una manichetta, che richiede all'NPC di muoversi dall'altro capo per sollevarlo e aiutare nel trasporto.

L'FSM è un tipo di automa¹⁴, molto diffuso nell'ingegneria e nelle scienze, soprattutto nell'informatica e nella ricerca operativa, il quale permette di descrivere con ampia precisione e in maniera formale il comportamento di molti sistemi fisici reali. Il modello di FSM è costituito, come per ogni altra macchina a stati da un insieme X di variabili di ingresso (controllabili), un insieme Y di variabili di uscita (osservabili) e un insieme S di variabili di stato, che non possono essere osservate o misurate direttamente. La differenza peculiare delle FSM consiste nella presenza di un insieme di variabili d'ingresso e da stati caratterizzati da una cardinalità finita e da una dinamicità nel passaggio da uno stato ad un altro.

Il sistema ideato per il progetto di tesi comprende una variazione delle FSM in base allo scopo per cui è presente l'NPC nella simulazione, il quale può consistere nel supporto all'utente durante le operazioni di messa in operatività, oppure nello svolgimento in autonomia di operazioni che l'utente non ha il compito di svolgere durante l'esperienza virtuale. Le FSM sono state progettate partendo tutte da una logica di base che comprende una serie di classi e di funzioni comuni a tutti gli NPC, le quali rivelano una struttura primaria condivisa.

Ogni NPC, inoltre, sfrutta la logica di *navigation* o navigazione [56], propria di Unity, la quale permette all'utente di generare una mappa (*NavMesh*) che consente all'NPC (anche denominato *NavMesh Agent*) di muoversi autonomamente nello scenario modellato in precedenza, rilevando ostacoli e luoghi dove non è possibile camminare (sulla superficie del fiume, sulle montagne, oltre i confini creati con l'ambientazione) e modificando il percorso per raggiungere una destinazione in tempo reale (Figura 4.11).

¹⁴ Nella teoria dei sistemi, un automa è un sistema dinamico discreto, nella scansione del tempo e nella descrizione del suo stato, e tempo-invariante, ovvero se il sistema si comporta alla stessa maniera indipendentemente dall'istante di tempo in cui agisce.

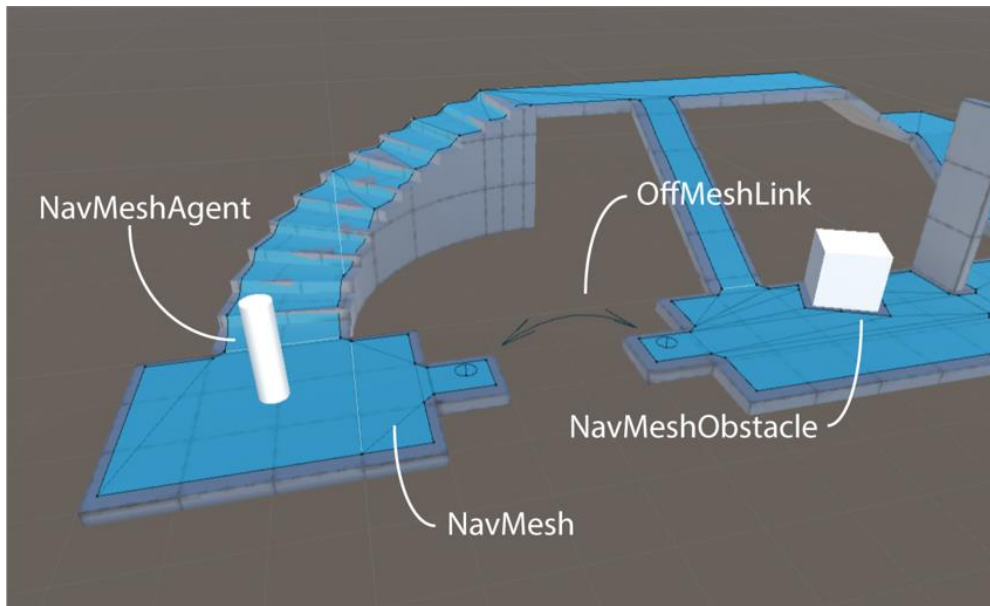


Figura 4.10: Gli elementi principali della componente AI in Unity.

Gli script di IK, così come per quanto riguarda la navigation, sono compresi nella logica delle FSM, senza alcuna variazione dovuta al diverso ruolo assunto dall’NPC, poiché i modelli non variano la loro modalità di interazione con gli oggetti presenti nella simulazione, che essi siano manichette, pannello di controllo della pompa o altri.

4.7 Locomozione

Il sistema di *CameraRig* di SteamVR presenta uno spazio in cui l’utente può muoversi liberamente, definendo, tuttavia, un’area piuttosto ristretta che non consente di spostarsi oltre i confini definiti. D'altronde, La necessità di percorrere uno spazio particolarmente esteso, come l’ambiente pensato per la simulazione VR, richiede un metodo di locomozione che consenta di muoversi ne modo più naturale possibile e senza un eccessivo dispendio di energie fisiche. Negli anni più

recenti, sono state progettate alcune tecniche che permettono all'utente il movimento nello spazio virtuale oltre l'area di tracciamento delle base station; tali metodi variano per ergonomia, facilità d'uso e realismo nella locomozione e, a seguito di una breve ponderazione dei vantaggi e degli svantaggi, una singola tecnica è stata scelta per essere utilizzata all'interno della simulazione.

Il primo approccio valutato è stato il room scale, metodo di locomozione basato sul teletrasporto da un punto all'altro dello scenario virtuale. La documentazione e gli script che permettono l'implementazione di questa tecnica sono presenti nel SDK distribuito dalla Valve sullo store di Unity e forniscono un valido supporto per le applicazioni che non richiedono un realismo nella navigazione della scena virtuale. Un'altra tecnica consiste nello sfruttare il movimento dei dispositivi VR come i controller e i tracker allo scopo di muovere la stanza virtuale nello scenario, senza che l'utente debba muoversi nello spazio fisico. Il metodo chiamato "foot-swingering" consiste nel simulare una corsa sul posto, agganciando i tracker ai polpacci dell'utente in modo da rilevare il movimento delle gambe e muovere simultaneamente la stanza virtuale. Il metodo denominato "arm-swingering" consiste, invece, in un movimento delle braccia, simulando la loro rotazione durante una corsa, in modo da permettere alle base station di tracciare il movimento rotatorio dei controller tenuti in mano dall'utente e mapparli sul suo movimento nello spazio 3D della stanza virtuale.

L'ultimo metodo descritto è stato scelto per il progetto di tesi poiché permette uno spostamento nella scena decisamente più realistico e intuitivo rispetto al room scale, il quale non risulta idoneo alla locomozione in una scena estesa come l'ambientazione della simulazione realizzata. Inoltre, il metodo arm-swingering garantisce anche una maggiore efficienza nel tracciamento della posizione dei sensori posti sui controller da parte delle base station rispetto al metodo foot-swingering, limitando anche lo sforzo fisico dell'utente durante la simulazione al solo movimento delle braccia.

Capitolo 5: Realizzazione

In questo capitolo sono riportate le varie fasi di sviluppo dell'applicazione, illustrando le varie scelte che hanno influito sulla forma definitiva della simulazione dal punto di vista della grafica, dell'ottimizzazione, della logica dell'applicazione e della gestione degli NPC.

5.1 Creazione dello scenario

Lo scenario virtuale in cui è ambientata la simulazione è stato realizzato a partire da un modello creato appositamente per l'applicazione, con l'intenzione di rendere l'esperienza visiva più realistica possibile e definendo, allo stesso tempo, una serie di requisiti spaziali e di confini fisici entro cui l'utente è obbligato a muoversi. L'ambientazione, infatti, doveva comprendere un fiume di medie dimensioni, che rappresentasse al meglio un contenuto appropriato per l'intervento del modulo HCP. Contemporaneamente, la squadra di volontari doveva trovarsi di fronte ad una pozza d'acqua vicina al fiume, la quale richiedeva una procedura di pompaggio, di trasporto e di scarico da parte della motopompa in un punto di scarico distante, per evitare che tale pozza danneggiasse ulteriormente gli argini del fiume in piena.

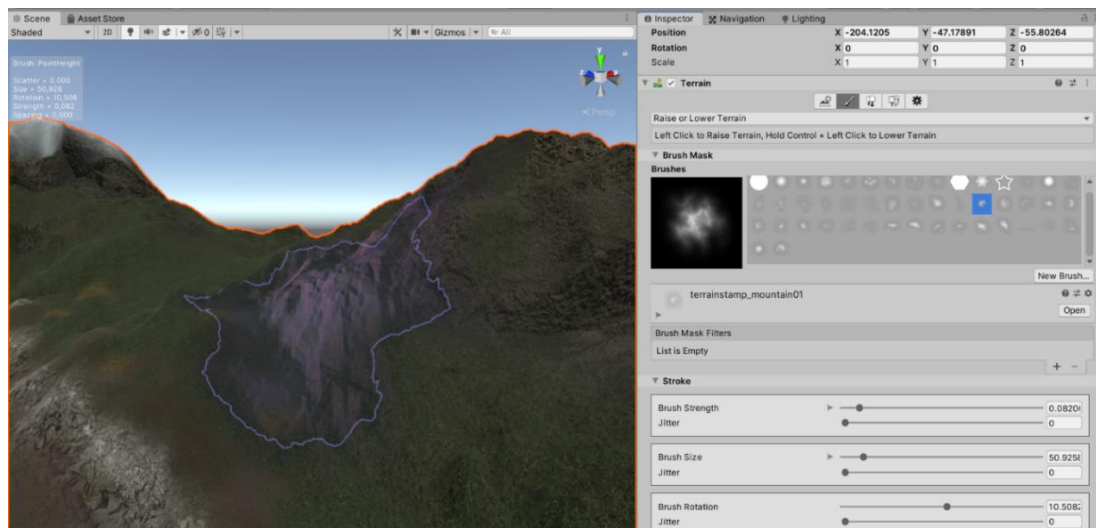


Figura 5.1: Brushing per la creazione di terreni, all'interno del Terrain Editor.

Il terreno è stato realizzato utilizzando il tool Terrain, incorporato in Unity, il quale permette di creare o modificare uno scenario naturale, offrendo, inoltre, la possibilità di modificare run-time le caratteristiche tecniche del modello e di ottimizzare il rendering in base alle esigenze dell'applicazione.

Terrain consiste in un component modificabile nell'Inspector oppure direttamente nella scena 3D in grado di modellare un piano tridimensionale, praticando una vera e propria operazione di *sculpting* sulla mesh (Figura 5.1).

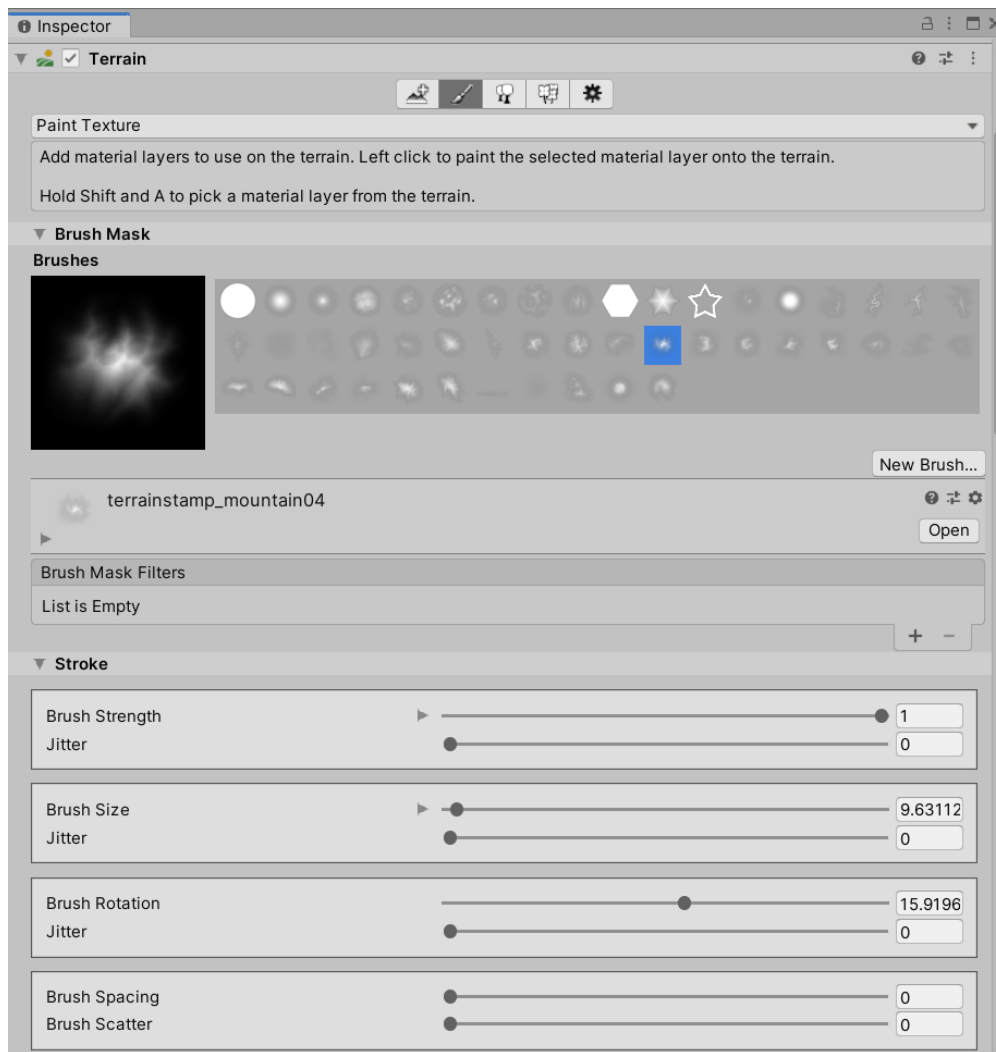


Figura 5.2: Terrain Editor di Unity.

L'interfaccia offre la possibilità di definire il tipo di “pennello” o *brush mask* (Figura 5.2) per modificare il terreno e riprodurre su di esso la forma scelta. Inoltre, lo sviluppatore può scegliere di modificare la dimensione, l'intensità, la rotazione e la spaziatura del brush, con un'ampia libertà creativa.

Il tool permette non solo di modellare il terreno ma anche di assegnare alla mesh una materiale personalizzato, realizzato combinando automaticamente una serie di texture disponibili nel package di Terrain oppure importabili direttamente dall'utente. In Terrain, la creazione di una texture per lo scenario segue una logica a livelli o *layers* (figura 5.3), la quale permette di utilizzare nuovamente il brush per posizionare uno layer di texture scelto su una parte della mesh e

sovrapponendo layer differenti al fine di creare un unico materiale finale (ad esempio, uno strato di texture raffigurante delle foglie su uno strato di texture rocciosa).

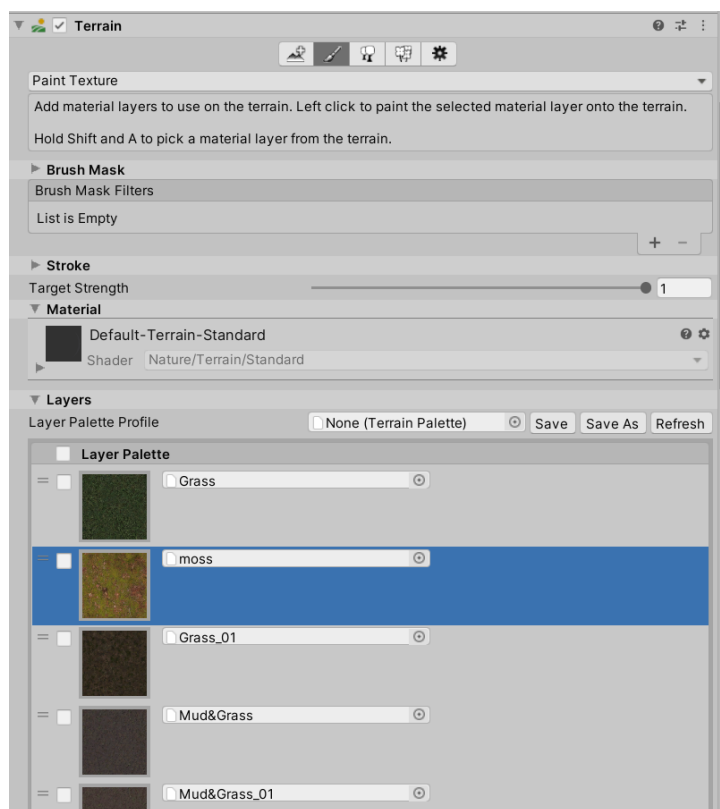


Figura 5.3: Tool per la creazione della texture nel Terrain Editor.

Allo scopo di rendere l'esperienza virtuale realistica e immersiva, lo scenario è stato caratterizzato da alberi, rocce, rami e cespugli, i quali rappresentano elementi di contorni puramente estetici, senza un'effettiva finalità all'interno della simulazione. Questi modelli tridimensionali sono stati aggiunti, in parte, utilizzando Terrain (Figura 5.4), il quale sfrutta un puntatore che permette di definire la posizione sul terreno in cui viene posizionato il modello di albero o dell'elemento scelto.

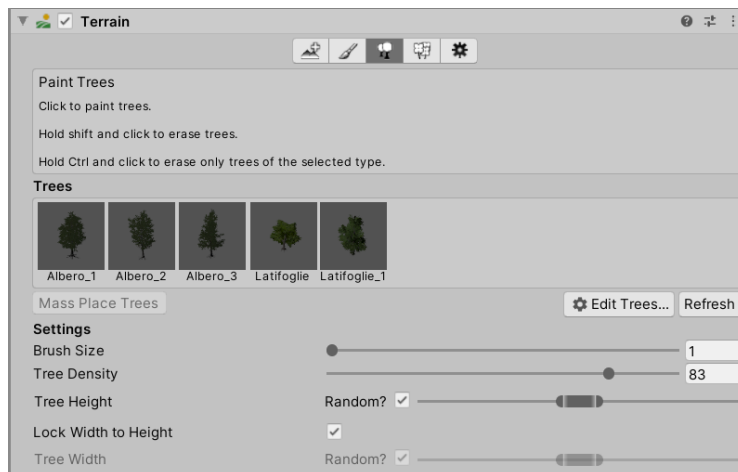


Figura 5.4: Tool di spawning degli alberi nel Terrain Editor.

5.1.1 Cascade e la creazione del fiume

Cascade è stato selezionato come l'asset per la creazione del fiume e della pozza d'acqua, gli unici elementi naturali che hanno richiesto uno shader specifico, in quanto si è cercato di ricreare l'effetto della corrente e le rifrazioni dell'acqua, allo scopo di rendere più realistica l'ambientazione. L'asset Cascade fornisce un tool di design che permette allo sviluppatore di costruire un fiume o un lago, creando una struttura virtuale a nodi orientabili, generata a partire da un'intuitiva interfaccia di controllo (Figura 5.5); i nodi costituiscono dei veri e propri puntatori che permettono di modificare le caratteristiche fisiche del piano su cui viene successivamente applicato uno shader, riproducendo la componente visiva naturale.

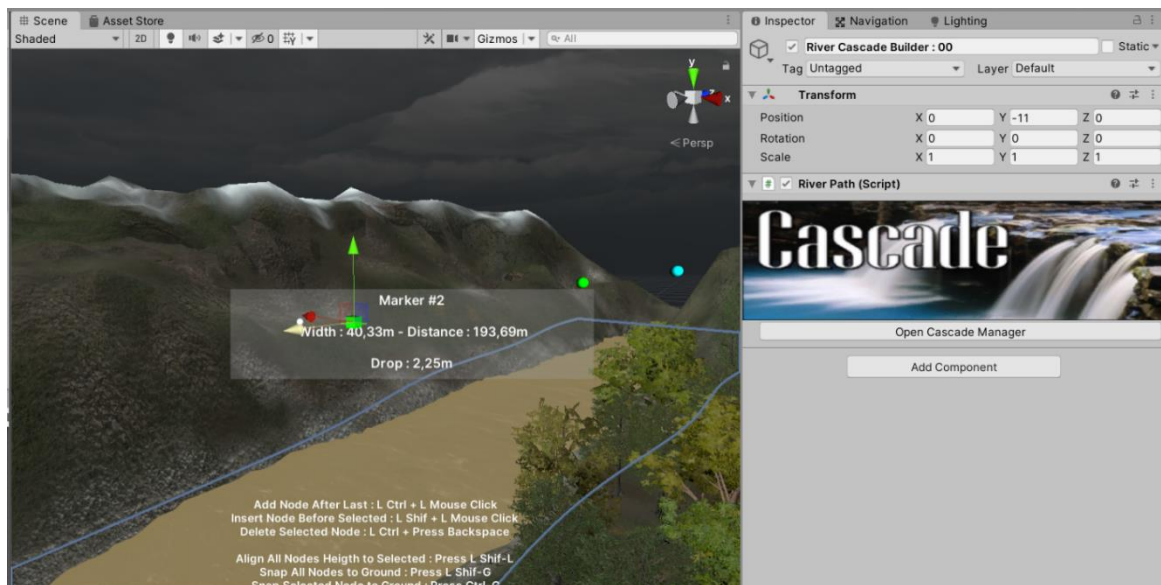


Figura 5.5: Logica a nodi sequenziali di Cascade.

Cascade offre anche uno shader specifico (Figura 5.6), per la simulazione procedurale dell'acqua, modificabile tramite Inspector, il quale permette di gestire differenti tipologie di sistemi (fiumi, cascate, laghi, ecc.) con un compilazione dinamica. Lo shader utilizzato ha permesso di settare i parametri di generazione delle onde e delle caratteristiche di rifrazione e di luminosità delle onde, in modo da rendere l'idea di un fiume in piena e di un possibile rischio idrogeologico.

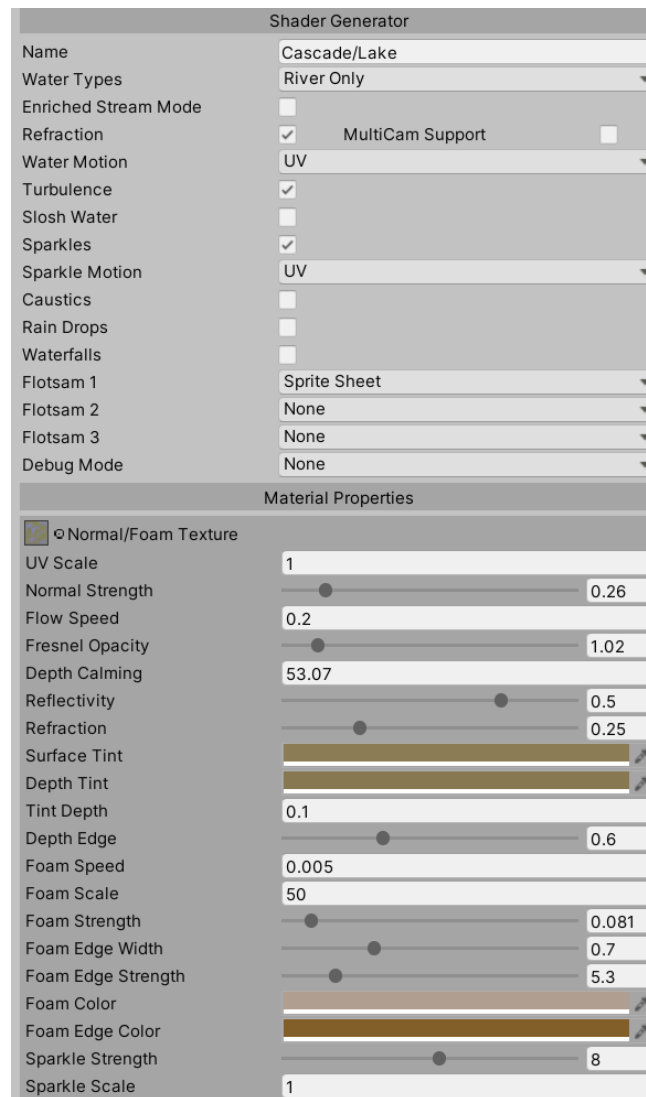


Figura 5.6: Cascade shader per la creazione del materiale dell'acqua.

5.1.2 Particellari d'acqua

La creazione del particellare per il getto d'acqua è stata facilitata dalla presenza di un componente appositamente realizzato per la gestione dei particellari in Unity, il quale permette di controllare le principali caratteristiche in un sistema complesso di particelle per la generazione di effetti visivi (fumo, fuoco, liquidi, nuvole, ecc.). Il sistema built-in del motore grafico utilizzato risulta essere molto complesso e ricco di parametri da modificare per la generazione del particellare,

perciò si è deciso di utilizzare solamente una parte dei parametri del component (Figura 5.7).

Durante lo sviluppo, le sole impostazioni offerte dal component considerate rilevanti sono state quelle relative alla durata di generazione delle particelle, alla velocità e alla direzione di queste ultime nello spazio 3D e alla componente grafica del sistema, con l'obiettivo di riprodurre un effetto di schizzo d'acqua.

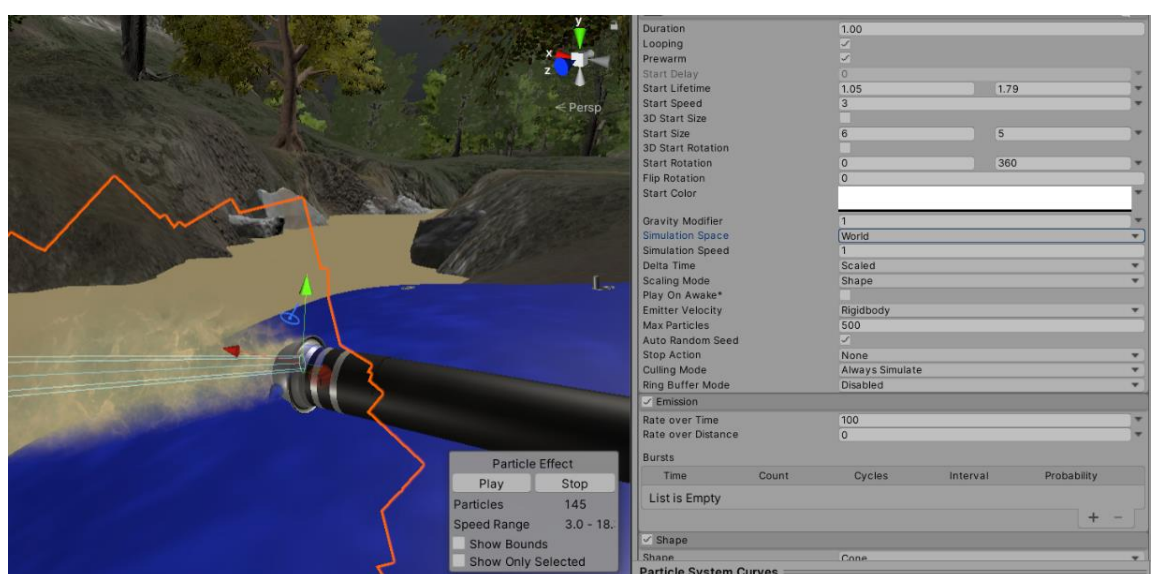


Figura 5.7: Particellare d'acqua nella simulazione.

5.1.3 Ottimizzazione grafica

L'implementazione della componente grafica, così come lo sviluppo della corposa parte di coding, ha reso necessario un lavoro di ottimizzazione allo scopo di migliorare le prestazioni in termini di vertici e di poligoni renderizzati e di frame rate stabile a 90 fps su un computer di fascia media, dotato di GPU. La realizzazione di uno scenario naturale realistico caratterizzato da un vasto numero di alberi, rocce e altri elementi organici ha portato a ricercare elementi

3D che arricchissero lo scenario dal punto di vista grafico senza tuttavia pesare troppo in termini di rendering real-time.

Perciò, la simulazione è stata sviluppata facendo largo uso di una tecnica molto sfruttata per ridurre le operazioni di calcolo della GPU, richieste da Unity per eseguire il rendering di mesh distanti: il Livello di Dettaglio o *Level of Detail* (LOD).

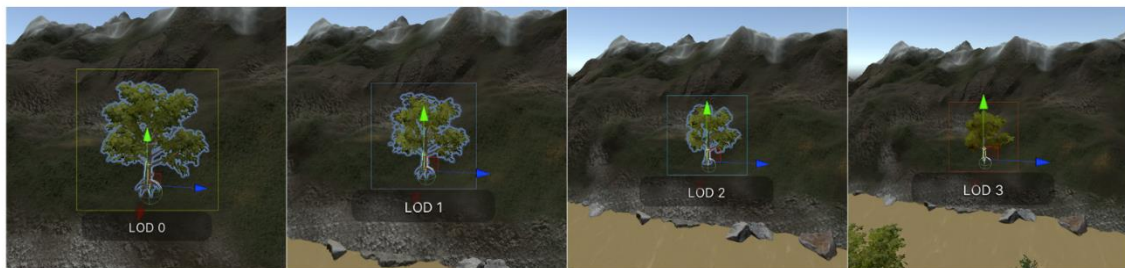


Figura 5.8: Funzionamento della tecnica del LOD nella simulazione.

Il metodo di LOD è caratterizzato dal rappresentare oggetti 3D complessi con una topologia più semplice se sono più lontani dalla camera (Figura 5.8). Poiché l'oggetto è più lontano, è spesso sufficiente abbassare la risoluzione del modello, senza la necessità di presentare tutti i dettagli; questo è dovuto al fatto che le dimensioni sullo schermo e la bassa probabilità di essere un punto focale nella visuale della camera impediscono all'utente di percepire con precisione le caratteristiche fisiche di un modello.

Unity comprende un component assegnabile a ciascun GameObject presente nella scena, in cui è possibile scegliere diverse mesh da associare allo stesso elemento 3D, che vengono sostituite dinamicamente in base alla distanza tra il modello e la camera.

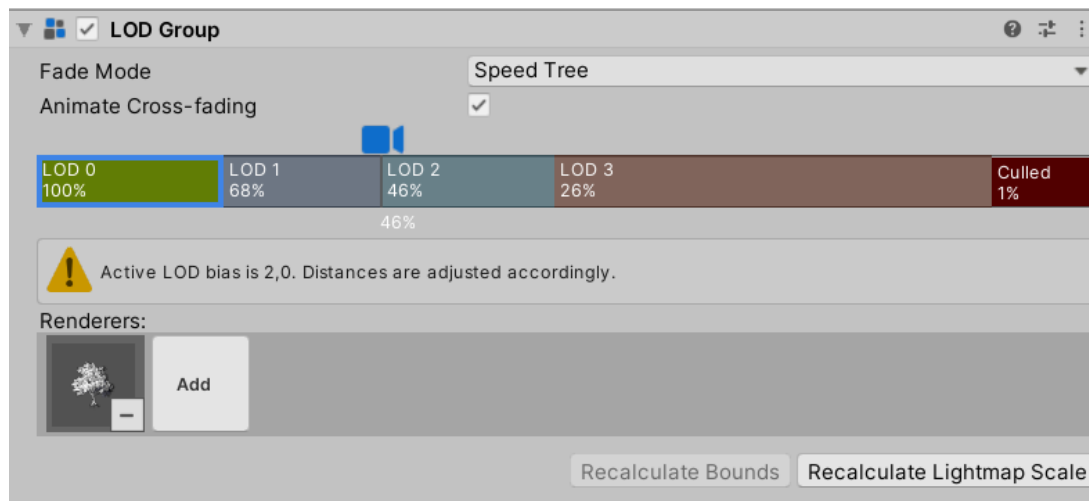


Figura 5.9: Component LOD nell'Inspector di Unity.

L'oggetto può, in alcuni casi, essere semplicemente pre-renderizzato a una trama e questa trama viene utilizzata su una mesh di geometria piatta molto semplice rivolta verso la telecamera (spesso semplicemente un quadrilatero) nota come *billboard*. Questa mesh bidimensionale, caratterizzata da un bassissimo numero di vertici, costituisce una tecnica per ottimizzare un'applicazione dal punto di vista grafico, poiché a grandi distanze un oggetto non cambia in modo significativo il suo orientamento rispetto alla telecamera, quindi una billboard assomiglia molto all'oggetto che rappresenta da un fotogramma all'altro, senza dover essere ridisegnato a partire dal modello 3D. La classe *BillboardAsset* di Unity consente la creazione di billboard che vengono renderizzati da diverse direzioni, portando a rappresentare in modo efficiente un oggetto con un livello di dettaglio basso da qualsiasi punto di vista approssimativamente orizzontale.

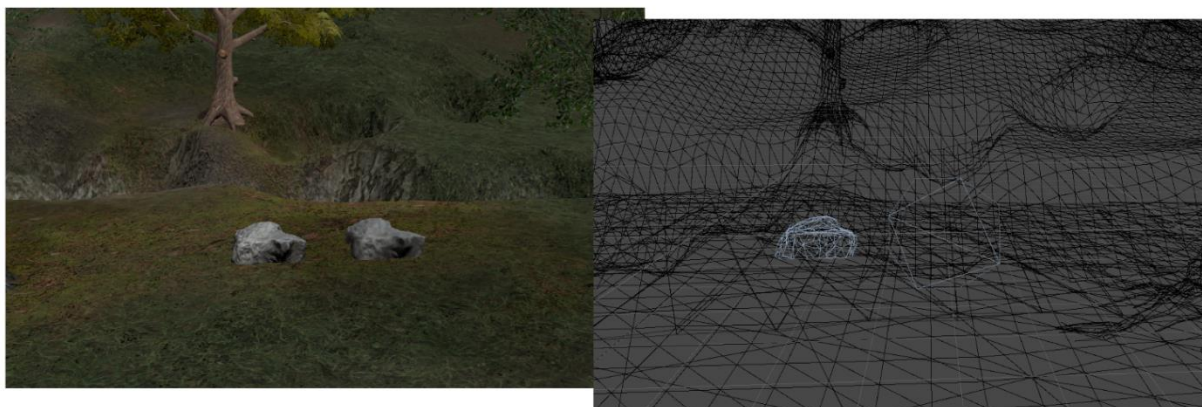


Figura 5.10: Confronto tra il modello 3D e il billboard di una roccia.

5.2 Modellazione degli NPC e degli oggetti in scena



Figura 5.11: Modello 3D del camion con gru presente nel Modulo HCP.

Durante le riunioni presso la sede della PC di Torino, sono stati raccolti dei modelli 3D dei veicoli principali che costituiscono la Colonna Mobile regionale, allo scopo di aumentare la componente di realismo durante l'esperienza virtuale.

I modelli utilizzati sono quelli dell'officina mobile per il trasporto dell'attrezzatura più leggera e il camion provvisto di gru per il trasporto delle manichette e dei tubi e per il traino della motopompa (Figura 5.11).

I modelli della motopompa e dell'attrezzatura, invece, sono stati creati da zero a partire dai riferimenti raccolti durante le esercitazioni a cui il gruppo di lavoro ha partecipato, in particolare la giornata di addestramento sul campo svolta presso il nodo idraulico di Ivrea. I modelli riproducono fedelmente gli oggetti reali nelle loro dimensioni e proporzioni rispetto all'utente durante la simulazione virtuale.

Durante la realizzazione dell'applicazione è stato necessario trasformare i modelli imponendo una diminuzione dei vertici per ciascun oggetto 3D (Figura 5.12), allo scopo di migliorare ulteriormente le prestazioni e aumentare il frame rate della simulazione, effettuando un intenso lavoro di retopology.

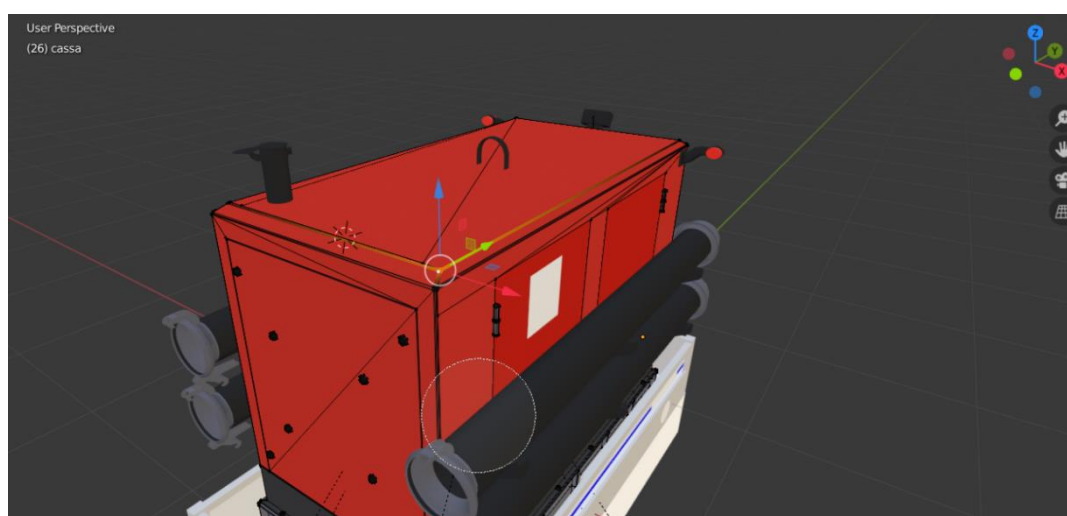


Figura 5.22: Edit mode in Blender per la modellazione 3D.

La realizzazione del modello dell'operatore (Figura 5.13) ha richiesto un lavoro maggiore per aumentarne il realismo dal momento che rappresenta un elemento organico con un alto livello di dettaglio, essendo, inoltre, l'elemento della scena con il maggior numero di vertici, insieme allo scenario virtuale. In aggiunta, il modello ha richiesto l'applicazione di un determinato numero di animazioni

realistiche di camminata, corsa, oltre a movimenti compositi gestiti in tempo reale in base agli obiettivi dell'azione da compiere.

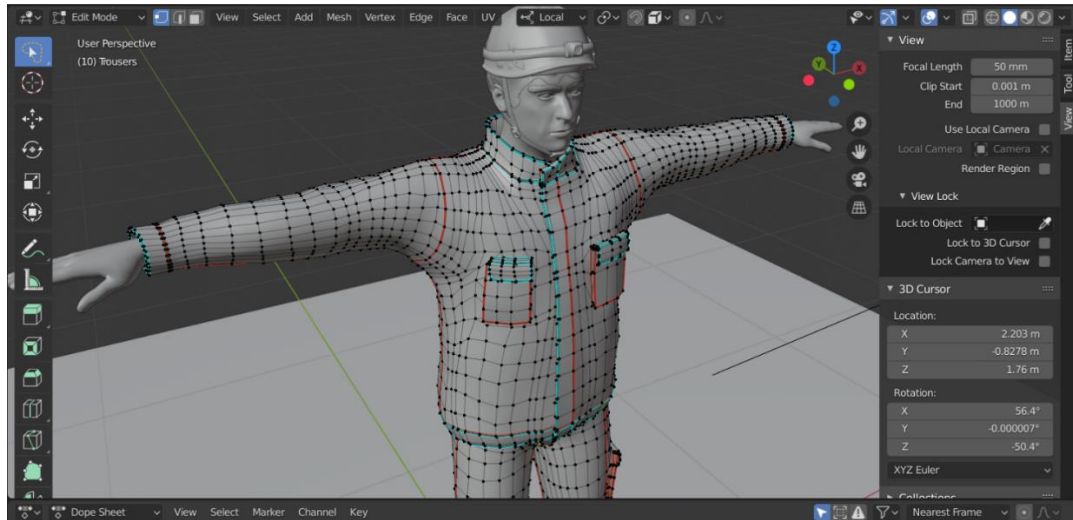


Figura 5.13: Modello dell'operatore in Blender.

Il modello è stato creato a partire da un prototipo di umanoide creato da MakeHuman, il quale permette ha permesso di generare la mesh del corpo, dei capelli e del viso dell'essere umano, al quale sono stati aggiunti i modelli dell'uniforme della PC e i dispositivi di sicurezza personale.

5.3 Gestione degli NPC

La fase di sviluppo degli NPC ha costituito una parte essenziale nella fase di implementazione di ciascuna procedura svolta dall'utente, andando a ricoprire i ruoli propri degli altri operatori del modulo e garantendo il supporto pratico essenziale nelle azioni di messa operativa.

La componente di realismo richiesta da un'applicazione di addestramento, inoltre, ha richiesto un notevole sforzo di progettazione allo scopo di implementare una logica di controllo, movimento e interazione dei modelli 3D, in grado di sostenere le azioni svolte da un operatore reale.

Gli NPC presenti nella simulazione sono ulteriormente suddivisi in due categorie: la prima è costituita dagli NPC che svolgono la funzione di supporto all'utente durante le fasi di lavoro nel montaggio dei tubi, aiutando a sollevare le manichette e supportando le azioni di aggancio e di trasporto; la seconda categoria risulta essere quella composta dagli NPC automatizzati che svolgono in maniera autonoma l'intero processo di montaggio dei tubi, senza dipendere dalle azioni dell'utente. Entrambe le tipologie di NPC condividono la medesima struttura di base, con qualche variazione necessaria, dovuta allo scopo che ciascuno di essi ha durante la simulazione.

5.3.1 Macchine a Stati Finiti

La cura richiesta per le animazioni e lo sviluppo di un'automazione in grado di sostenere una lunga serie di procedure consecutive in coordinazione con quelle svolte dall'utente hanno portato alla realizzazione di un insieme di FSM, il cui scopo risulta essere la risposta in tempo reale delle azioni compiute dall'utente.

La costruzione delle FSM è partita dall'individuazione degli stati di azione fondamentali in cui l'NPC si sarebbe trovato a operare, a partire dallo stato di riposo fino alle fasi di sollevamento dei tubi e di accensione della pompa. Le fasi essenziali che costituiscono la macchina a stati dell'NPC di supporto sono:

- *Idle* (stato di riposo dell'NPC);
- *Follow_player* (stato in cui l'NPC segue l'utente);
- *Reach_target* (stato in cui l'NPC raggiunge un punto specifico nella scena);

- *Grab_tube* (stato in cui l’NPC deve raggiungere il capo del tubo da sollevare);
- *Carry_tube* (stato in cui l’NPC deve trasportare il tubo insieme all’utente);
- *Drop_tube* (stato in cui l’NPC deve posizionare il capo a terra prima di sostenere l’altro capo per la fase di aggancio);
- *Hold_tube* (stato in cui l’NPC deve sostenere il capo del tubo durante la fase di aggancio);

```

18 riferimenti
public void changeState(States s) //Funzione che modifica il valore della mia SyncVar "stato corrente"
{
    _current_state = s;

    if (s == States.idle)
        NPC_Dest_script.IdleState();
    else if (s == States.hold_tube && NPC_capoPlayer_script != null)
    {
        NPC_IK_script.StopInt();
        NPC_hold_target = NPC_capoPlayer_script.Capo_target_hold_NPC;
        NPC_capoNPC_script.SetRelative(null, null, false, false);
        ChangeRigidbody(false);
    }
    else if (s == States.drop_tube & NPC_capoPlayer_script != null)
    {
        NPC_distance = 0.2f;
        NPC_drop_target = NPC_capoPlayer_script.Capo_target_drop_NPC;
    }
    else if (s == States.carry_tube)
        NPC_distance = 0.4f;
    else if (s == States.follow_player && montaggio == false)
        _current_state = States.reach_target;
    else if (s == States.reach_target && montaggio == true)
        _current_state = States.follow_player;

    NPC_Dest_script.SetStopDistance(_current_state);

    CmdChangeState(s);
}

```

Figura 5.14: Metodo per il cambiamento di stato nella FSM dell’NPC.

Ogni passaggio di fase è impostato su un trigger, ovvero un valore stabilito che, se modificato, fa avanzare l’esecuzione della FSM da uno stato a quello successivo (questo valore può essere incapsulato in un booleano o in una variabile), oppure avviene chiamando direttamente una funzione che cambia lo stato corrente della FSM. Ogni funzione che richiede un passaggio di stato deve chiamare necessariamente il metodo *ChangeState()* (Figura 5.14), il quale permette di aggiornare lo stato corrente.

Unity mette a disposizione una sezione apposita chiamata *Animator*, presente sul motore grafico per la gestione delle FSM, la quale permette di gestire i diversi stati

ed il passaggio tra di essi attraverso un'interfaccia logica a nodi, in maniera simile al metodo standard utilizzato per rappresentare le FSM in campo teorico. Questo metodo consiste nel mostrare ogni stato come un nodo, collegato agli altri stati da un segmento che rappresenta il trigger che ha portato al cambiamento di stato.

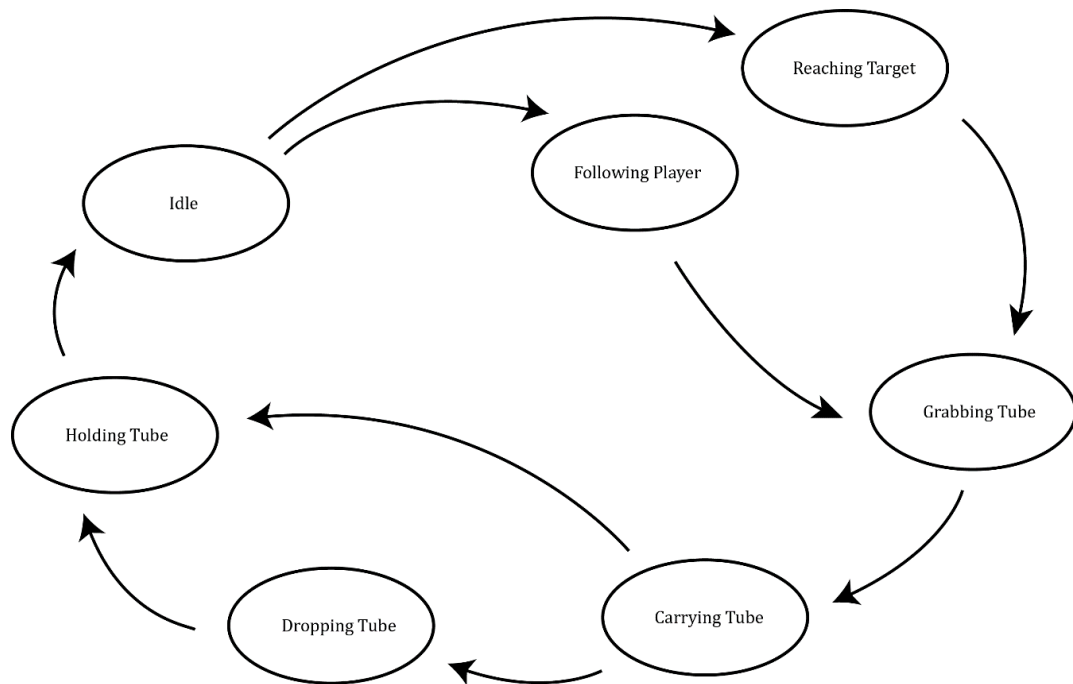


Figura 5.15: Schema a nodi della FSM dell'NPC.

Inoltre, Unity permette di sfruttare anche un modello predefinito di script per implementare ogni stato della FSM, definito *StateMachineBehaviour*. Questo script ha alcuni messaggi predefiniti: *OnStateEnter*, *OnStateExit*, *OnStateIK*, *OnStateMove*, *OnStateUpdate*. Ciascuno di questi messaggi garantisce un notevole snellimento del codice in caso di grande complessità, suddividendo in comandi di ingresso e di uscita e aggiungendo chiamate per la gestione della navigazione e dell'IK dell'NPC.

Tuttavia, il modello presentato da Unity risulta essere eccessivamente legato all'Animator nelle controllo di passaggio tra i diversi stati, potendo gestire i trigger esclusivamente dall'interfaccia proposta ed inoltre non offre la possibilità di integrare in maniera semplice ed intuitiva componenti di IK differenti da quelle

predefinite offerte dal motore grafico. Oltre a ciò, il modello di FSM (Figura 5.16) progettato per gli NPC del progetto di tesi non richiedeva un numero di stati particolarmente elevato, ma per lo più un'attenzione rivolta verso le animazioni (dirette e con l'IK) e le fasi di navigazione all'interno dello scenario.

L'Animator di Unity è stato ugualmente sfruttato, ma esclusivamente per le animazioni di movimento ed il passaggio tra lo stato di riposo, di camminata e di corsa, gestiti sempre attraverso un apposito componente chiamato *MovementController*.

```

1 riferimento
private void UnitFSM() //FSM del NPC
{
    switch (_current_state)
    {
        case States.idle:
            break;

        case States.reach_target:
            if (montaggio == false) ...
            else ...
            break;

        case States.follow_player:
            if (montaggio == true) ...
            else
                changeState(States.reach_target);
            break;

        case States.grab_tube:
            if (NPC_capo_player != null) ...
            else
            {
                Debug.LogWarning("L'npc non rileva alcun tubo");
            }
            break;

        case States.carry_tube:
            SetRotation(NPC_capo_player);

            if (NPC_capo_npc_script.grabbato == true)
                return;

            //Setto la destinazione da raggiungere
            Vector3 dest_carry = new Vector3(NPC_carry_target.transform.position.x,
                this.transform.position.y, NPC_carry_target.transform.position.z);
            //Setto la distanza momentanea tra il capo che l'NPC ha in mano e la distanza
            float distance = Vector3.Distance(NPC_capo_player.transform.position, NPC_previousEndPosition);

            NPC_Dest_script.FollowTarget(dest_carry, NPC_distance, distance, NPC_capoPlayer_script.Capo_carry_position);

            NPC_previousEndPosition = NPC_capo_npc.transform.position;
            break;

        case States.hold_tube:
            Vector3 dest_holding = new Vector3(NPC_hold_target.transform.position.x,
                this.transform.position.y, NPC_hold_target.transform.position.z);
    }
}

```

Figura 5.16: Pezzo del codice di gestione della FSM.

5.3.2 Navigazione nello scenario virtuale

La realizzazione delle animazioni dell'NPC per il movimento nello spazio 3D è avvenuta in due fasi separate: la prima ha riguardato la creazione delle animazioni da applicare al modello dell'operatore e da importare ed applicare direttamente su Unity, mentre la seconda parte è stata caratterizzata dall'impostazione di una

struttura di navigazione, la NavMesh, generata a partire da alcuni componenti specifici presenti sul motore grafico.

Dopo aver realizzato il modello 3D, è stata creato il *rig*, importando il file FBX sul programma Mixamo, che ha permesso di associare direttamente ciascun osso alla parte di mesh corrispondente per poi applicarvi sopra una serie di animazioni predefinite.

Successivamente, il modello con il rig e le animazioni scelte per l'esportazione sono stati convertiti in un nuovo file FBX, che ha permesso di importare il lavoro su Blender, ripulire sia lo scheletro, sia raffinare leggermente le animazioni, in modo da ottenere una serie di animazioni a 60 fps, pronte per l'esportazione. Il file finale del modello dell'operatore comprende un'animazione in idle, due animazioni di camminata (in avanti e all'indietro) e un'animazione di corsa.

Questa serie di animazioni è stata raccolta in un unico file in formato *blend* importato direttamente in Unity, in cui è stato generato un ulteriore rig a partire da quello utilizzato per le animazioni, configurando il modello come *humanoid* e generando una serie di clip in loop, corrispondenti alle quattro animazioni create in precedenza (Figura 5.17).

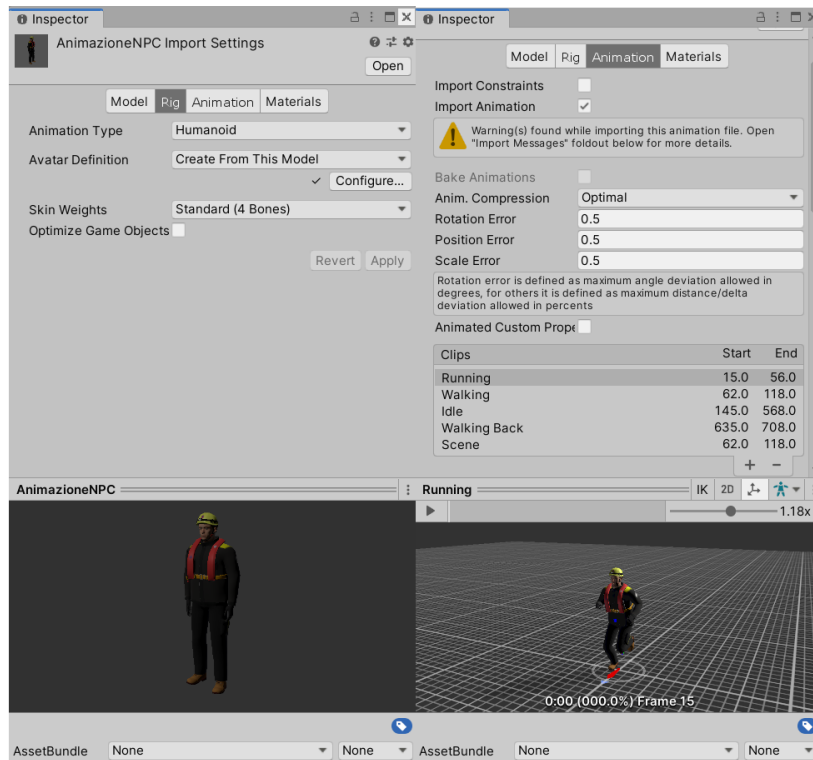


Figura 5.17: Pannello di gestione dei modelli 3D in Unity.

Il controllo dello stato in cui si trova ciascun modello e la conseguente applicazione di una specifica animazione è stato gestito sfruttando il *Mecanim Animation System* di Unity (Figura 5.18), con cui è stata generata una struttura a nodi e ad archi, con ciascun arco definito a partire da un cambiamento di una serie di booleani. Ciascun cambiamento di questi booleani definisce il trigger che cambia lo stato corrente della FSM che è stata generata, impostando una serie di condizioni controllabili da codice attraverso un semplice metodo `animator.SetBool("name")`.

Il controllo dello stato in cui si trova ciascun modello e la conseguente applicazione di una specifica animazione è stato gestito dal pannello Animator di Unity, con cui è stata generata una struttura a nodi e ad archi, con ciascun arco definito a partire da un cambiamento di una serie di booleani. Ciascun cambiamento di questi booleani definisce il trigger che cambia lo stato corrente

della FSM che è stata generata, impostando una serie di condizioni controllabili da codice attraverso un semplice metodo *animator.SetBool("name")*.

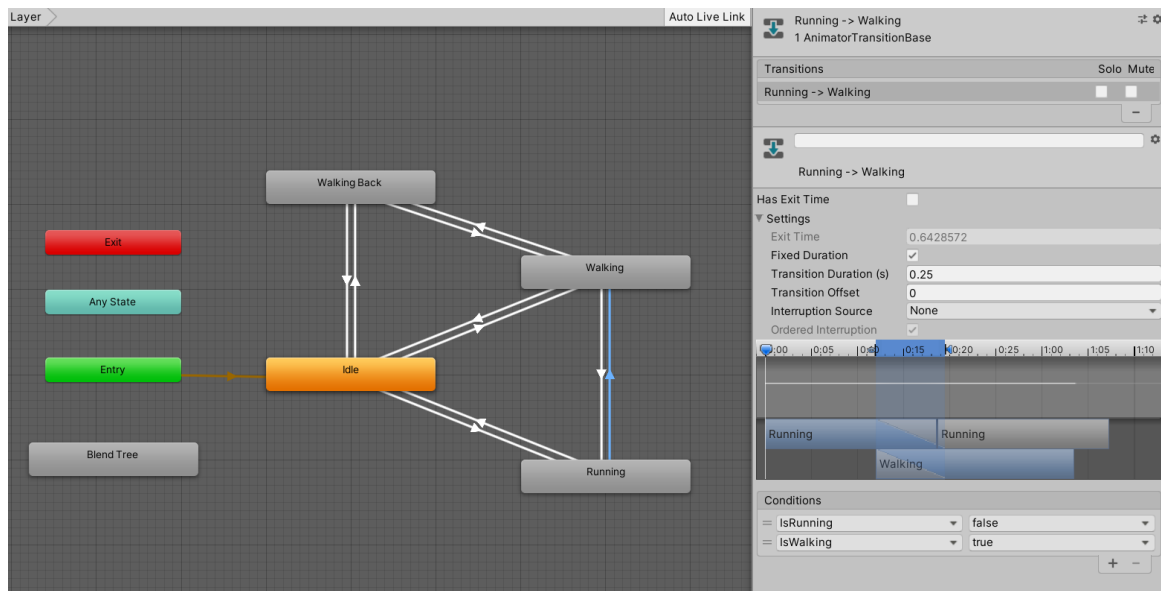


Figura 5.18: Animator tool per la gestione delle animazioni in Unity.

Le animazioni realizzate hanno permesso di rendere maggiormente realistica la navigazione degli NPC nello scenario 3D, la quale è stata gestita diversamente rispetto agli altri elementi presenti in scena. Infatti, la complessità maggiore che si ha ricercato nella programmazione degli operatori automatizzati ha richiesto l'utilizzo del componente di Unity in grado di gestire l'AI all'interno di una scena di gioco: il già citati componente NavMesh Agent.

Come detto, Unity fornisce una struttura base di controllo per il movimento di NPC nella scena virtuale, che richiede di aggiungere questo componente al GameObject designato e generare una mappa virtuale dell'intero scenario che la logica di gioco acquisisce per poter controllare lo spostamento del NPC nell'ambiente 3D. Questo processo di generazione dell'area navigabile dal modello (la NavMesh) avviene dapprima scegliendo le caratteristiche fisiche da assegnare all'NPC, quali l'altezza, il raggio e i parametri di salto del cilindro di navigazione, insieme alle caratteristiche proprie della NavMesh come la risoluzione e l'accuratezza nella generazione (Figura 5.19). Gli elementi della

scena che sono stati considerati percorribili dall’NPC sono stati considerati area *Walkable*, mentre gli altri elementi statici, quali i veicoli, sono stati aggiunti come elementi *Non Walkable*.

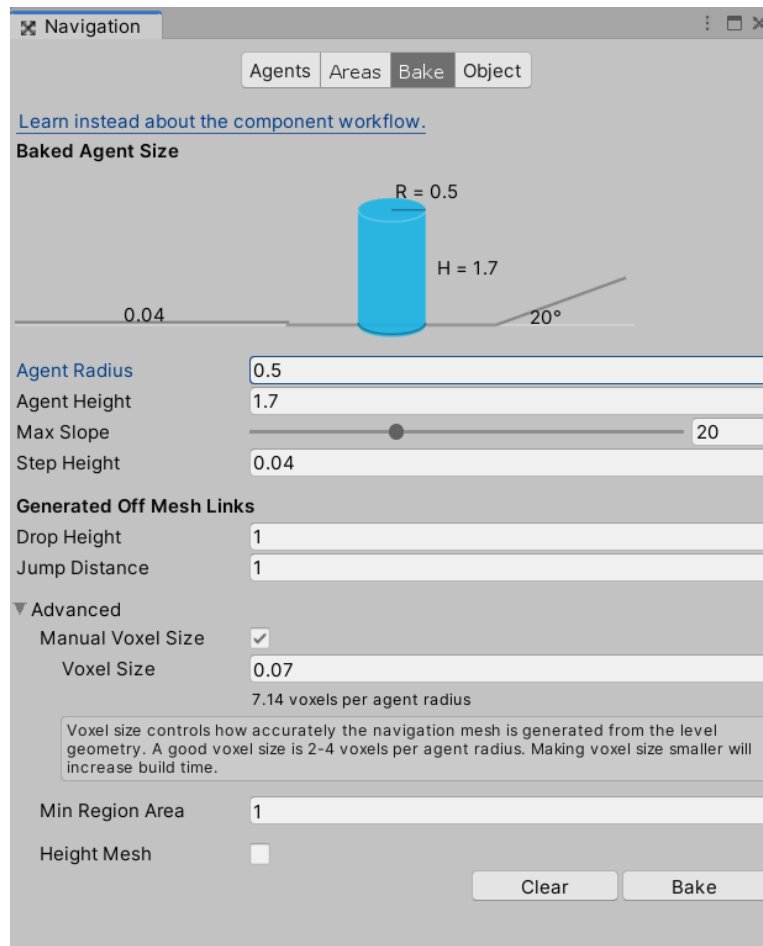


Figura 5.19: Impostazioni per la creazione della NavMesh in Unity.

Successivamente, è stato fatto il baking della NavMesh ed è stato aggiunto il component NavMesh Agent (Figura 5.20) direttamente sul prefab del modello in modo da associare l’NPC direttamente all’area di navigazione creata; diventa così possibile controllare lo spostamento tramite script, assegnando una destinazione particolare oppure un altro GameObject da seguire, lasciando il calcolo del percorso da compiere al sistema. Il component presenta alcuni parametri da impostare molto intuitivi tra cui la velocità, l’accelerazione dell’NPC e la distanza da mantenere rispetto al punto d’arrivo.

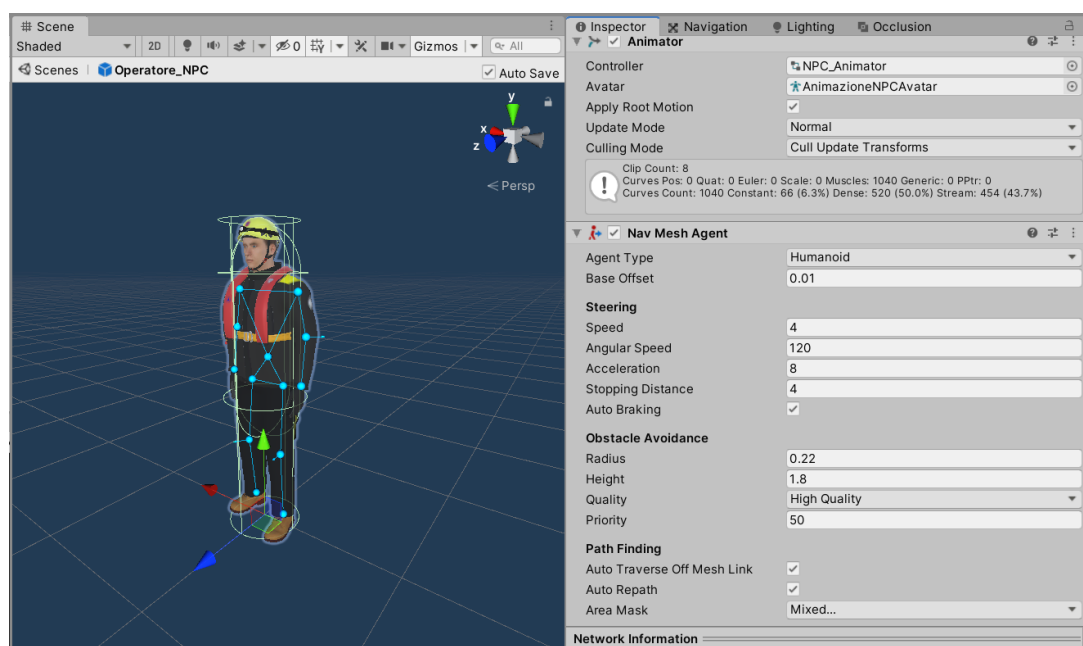


Figura 5.20: Il component NavMesh Agent all'interno del prefab dell'NPC.

Un'ulteriore component utilizzato per definire i confini e gli ostacoli all'interno della simulazione è il *NavMesh Obstacle*, il quale costituisce una particolare versione di Collider in grado di deviare il percorso calcolato da Unity per arrivare alla destinazione oppure direttamente di bloccare la navigazione di un NPC verso una particolare direzione. Questo tipo di component è stato assegnato agli elementi dinamici della scena che, potendosi muovere non potevano essere considerati nel processo di baking della NavMesh, ma richiedevano di essere considerati come un ostacolo, come, ad esempio, la pompa e gli altri NPC.

5.3.3 Inverse Kinematics e Final IK

Il processo di implementazione della IK per animare l'NPC durante la simulazione ha costituito il passaggio più importante della tesi, consentendo un diverso approccio nel tentare di far coesistere tre diversi sistemi di animazione nella

medesima struttura. Come mostrato in precedenza, la NavMesh e le animazioni sono state gestite internamente al motore grafico, mentre le animazioni IK in tempo reale sono state realizzate attraverso l'asset Final IK, il quale ha permesso di implementare le animazioni via script, offrendo una libreria di funzioni in grado di bilanciare ogni componente fisica dello scheletro dell'NPC.

Il prefab dell'operatore, a cui sono stati precedentemente associati i componenti dell'Animator per il controllo delle animazioni tradizionali e il NavMesh Agent per la navigazione, è stato importato nella scena con il rig realizzato su Blender e importato su Unity. FinalIK mette a disposizione un component principale, denominato *Full Body Biped IK* (vedi Figura 5.21), che genera un'armatura a bassa risoluzione (al fine di aumentare l'ottimizzazione delle prestazioni) ed estremamente flessibile, destinata esclusivamente al controllo delle animazioni in IK da applicare durante la simulazione.

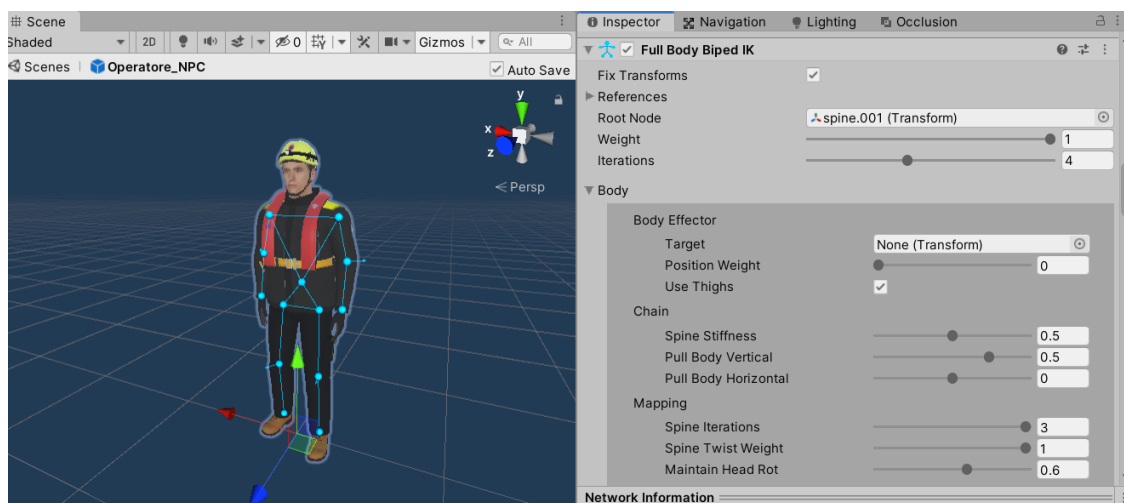


Figura 5.21: Component *Full Body Biped IK* all'interno del prefab dell'NPC.

Questo componente offre un'interfaccia a slide che gestisce i parametri di animazione dell'armatura, controllando gli *effectors*, che costituiscono i giunti della catena di cinematica inversa. Full Body Biped IK (Figura 5.22) ha tre tipi di effectors: *end-effector* (mani e piedi), *mid-body effector* (spalle e cosce) e *multi-effector* (il corpo). La modifica della rotazione di un effector della catena, come

una mano, modifica anche la direzione di flessione dell'arto (mid-body), con la possibilità di controllare separatamente la rotazione dei giunti intermedi della spalla e del gomito attraverso funzioni di bending calcolate automaticamente da Final IK. L'effector del corpo è di tipo multi-effector, il che significa che la sua animazione propaga gli effetti lungo entrambi gli effectors della coscia e delle spalle (con una semplificazione del modello dell'armatura).

Le impostazioni presenti permettono di assegnare un target per ciascun effector e di modificare il *weight*, ovvero il peso da attribuire alle caratteristiche fisiche di posizione, rotazione e ad altre proprietà specifiche che consentono di manipolare i giunti finali della catena scelta.

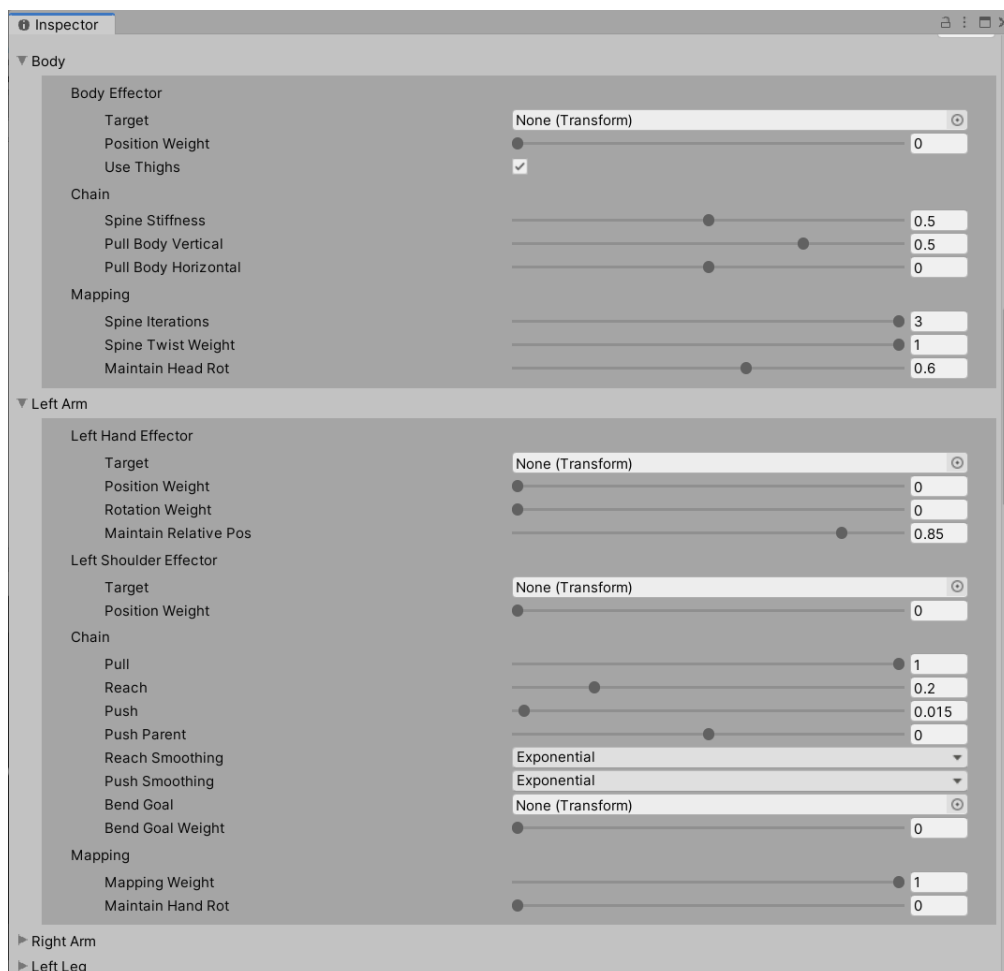


Figura 5.22: Particolare del component Full Body Biped IK nell'Inspector di Unity.

Full Body Biped IK mappa qualsiasi personaggio bipede su un rig IK multi-effettor a bassa risoluzione e ad alto frame rate, aggiornando le modifiche e mappando il risultato al personaggio nello stato *LateUpdate*, dopo che *Mecanim* ha terminato l'animazione, quindi risulta completamente indipendente dal sistema di animazione.

Il sistema introdotto con Final IK necessita, inoltre, di assegnare a ciascun elemento un target dove posizionare il giunto della catena e, nel caso delle mani, si è ricercato il naturale effetto di *grabbing*, ovvero l'azione di afferrare con le dita l'oggetto da prendere e sollevare. L'asset utilizzato permette, a tale scopo, di generare dei target direttamente dall'armatura generata per il modello 3D, assegnando un componente chiamato *Interaction Target* e assegnandogli l'effettor di riferimento (*Left Hand Effector*, *Right Hand Effector*, ecc.). Il GameObject ottenuto (Figura 5.23) risulta essere un elemento con rig privo di mesh, invisibile durante la simulazione, e associabile via script alla mano dell'utente, attraverso il componente *Interaction Target*.

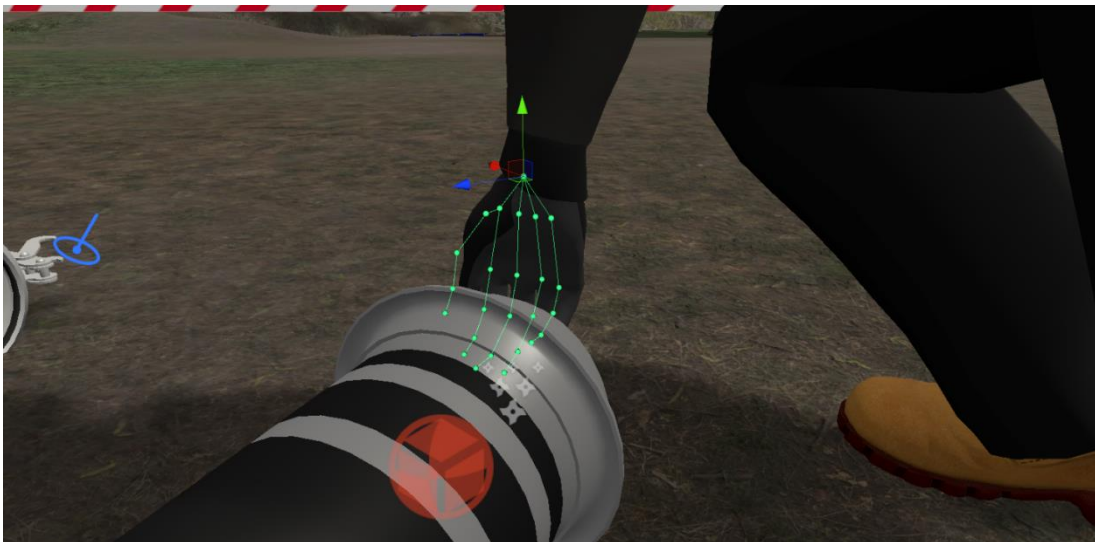


Figura 5.23: Target per la IK di una mano dell'operatore, durante la simulazione.

Ognuno di questi target ha richiesto di essere inserito all'interno di un ulteriore elemento in scena, in una vera e propria scala gerarchica di associazioni, allo scopo di mantenere un riferimento del target all'interno dello spazio 3D. Ciascun target, infatti, è stato imparentato ad un oggetto interagibile presente nella scena (attrezzatura, DPI, valvole della motopompa, ecc) e a tale oggetto è stato assegnato il componente *Interaction Object* (Figura 5.24).

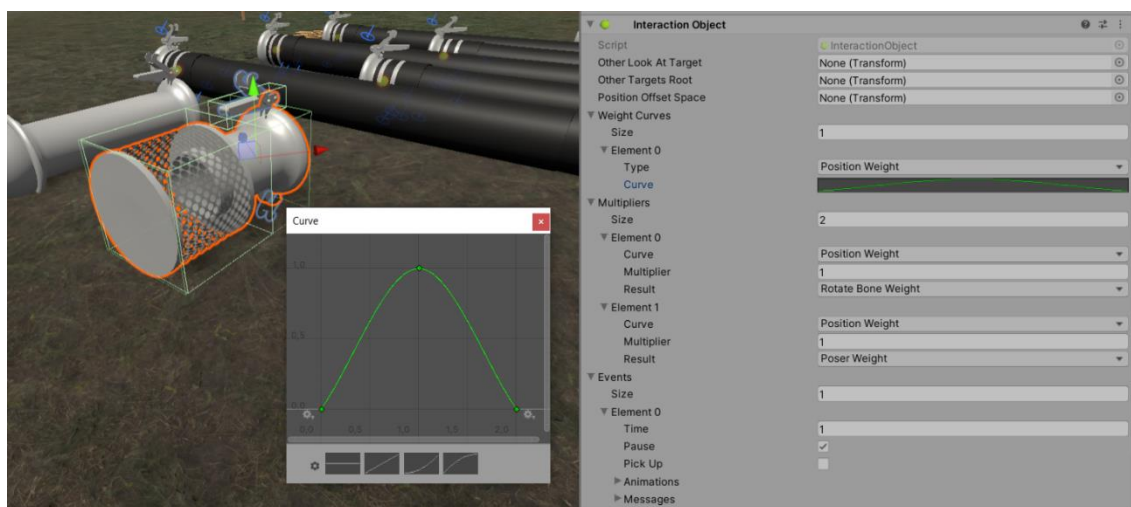


Figura 5.24: Il component *Interaction Object* nell'Inspector di Unity.

Questo componente deve essere aggiunto agli oggetti di gioco con cui il modello deve interagire. Contiene la maggior parte delle impostazioni da settare per la gestione delle interazioni. Non specifica quali parti del corpo verranno utilizzate e i parametri associati (i loro valori vengono controllati dal Full Body Biped IK), ma piuttosto l'aspetto e la logica dell'interazione. In questo modo le caratteristiche come il posizionamento dell'effector, la velocità dell'interazioni e altre interpolazioni necessarie a migliorare l'estetica dell'animazione sono definite dall'oggetto e possono essere condivise tra più effettori. Ad esempio, un pulsante verrà premuto nello stesso modo, indipendentemente dall'effettore utilizzato per esso.

Tuttavia, per animazioni presenti nella simulazione si è deciso di associare a ciascun *Interaction Object* un *Interaction Target* (figura 5.25), con un rapporto

uno a uno tra elementi interagibili ed effector da associare, decidendo di impostare diversi Interaction Object per uno stesso oggetto, qualora esso richieda di interagirvi con più effector (ad esempio, sollevando un capo di un tubo con due mani).

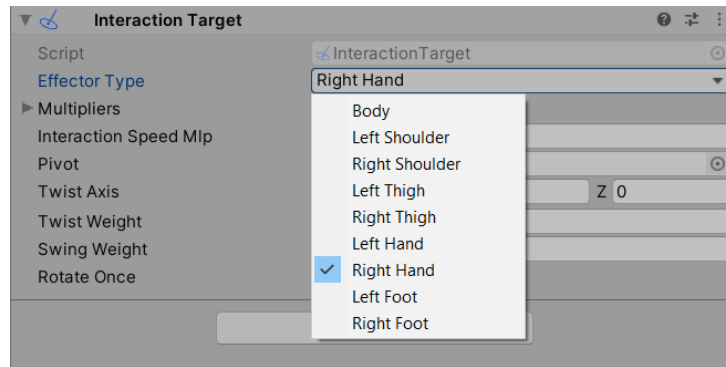


Figura 5.25: Il component *Interaction Target* in Unity.

Per l'assegnazione in tempo reale dell'oggetto con cui interagire e la gestione dei parametri di interazione si è deciso di creare uno script apposito (figura 5.26) contenente i metodi necessari per attivare e disattivare le animazioni, in base alle azioni dell'utente o allo stato corrente, in cui si trova la FSM in quel preciso istante di tempo. Ogni modello NPC ha un ulteriore component chiamato *Interaction System*, che permette di controllare l'attivazione delle interazioni e che viene chiamato via codice qualora sia necessario che l'NPC compia una determinata azione con l'utilizzo della IK.

```

9 riferimento
public void StartInt(InteractionObject obj, Transform left, Transform right, stato_IK ik) //Attivazione della IK
{
    interactionObject = obj;
    left_hand_target = left;
    right_hand_target = right;

    switch (ik)
    {
        case stato_IK.grab:
            interactionSystem.StartInteraction(effectors[0], interactionObject, true);
            IK.solver.righthandEffector.rotationWeight = 0.1f;
            break;
        case stato_IK.grabl:
            interactionSystem.StartInteraction(effectors[1], interactionObject, true);
            IK.solver.righthandEffector.rotationWeight = 0.1f;
            break;
        case stato_IK.hold:
            interactionSystem.StartInteraction(effectors[0], interactionObject, true);
            interactionSystem.StartInteraction(effectors[1], interactionObject, true);
            /*IK.solver.leftFootEffector.rotationWeight = 0.5f;
            IK.solver.rightFootEffector.rotationWeight = 0.5f;*/
            break;
        case stato_IK.hold_pump:
            interactionSystem.StartInteraction(effectors[0], interactionObject, true);
            interactionSystem.StartInteraction(effectors[1], interactionObject, true);
            if (LeftSide)
            {
                IK.solver.leftHandEffector.rotationWeight = 0;
                IK.solver.righthandEffector.rotationWeight = 0;
            }
            else
            {
                IK.solver.leftHandEffector.rotationWeight = 0.5f;
                IK.solver.righthandEffector.rotationWeight = 0.5f;
            }
    }
}

```

Figura 5.26: Metodo per l'attivazione delle interazioni utilizzando la libreria di Final IK.

5.4 Interfaccia utente e voice-over

Lo sviluppo di un'applicazione immersiva con l'obiettivo di donare un'alternativa realistica alle esercitazioni sul campo ha richiesto di equilibrare l'utilizzo di molteplici modalità di comunicazione presenti in un qualunque programma, ad una ricerca di verosimiglianza nell'interazione con altre figure all'interno del mondo VR. L'Interfaccia Utente o *User Interface* (UI) standard, diffusa nella maggior parte delle applicazioni da desktop e basata sul testo, immagini e video sarebbe risultata sconveniente, se introdotta nelle fasi di spiegazione e di feedback durante le azioni dell'utente nello scenario, poiché sarebbe risultata pesante e poco immersiva.

Perciò, un'interfaccia utente, caratterizzata da testo e immagini, disposta su un piano 3D è stata mantenuta esclusivamente per il menu offline, nella scelta delle fasi e della modalità di simulazione; durante l'esperienza in VR, invece, l'utente è

sottoposto a tre tipologie di stimolo sensoriale, le quali accompagnano il sistema di interazione nella maniera più intuitiva possibile:

- *outline shader*, contorni luminosi in grado di evidenziare un oggetto nella scena, se selezionato;
- icone e *highliner* colorati, che permettono di focalizzare l'attenzione dell'utente nelle fasi di apprendimento e di esercitazione (Figura 5.27);
- suoni 3D in grado di immergere l'utente nell'ambientazione virtuale (la corrente del fiume, il motore della motopompa, ecc.);
- clip audio che simulano una chiamata radio con un altro operatore della PC.



Figura 5.27: Rappresentazione grafica delle due modalità di evidenziazione utilizzate nella simulazione.

Dal momento che gli operatori della PC operano scambiandosi informazioni a voce o utilizzando un walkie talkie con altri gruppi di volontari, si è deciso di utilizzare un voice-over, per poter rendere più immediata la comunicazione tra utente e sistema e per mantenere un grado di realismo nella simulazione. Il voice-

over è stato progettato a partire da un testo appositamente scritto per la simulazione costituito da una serie di frasi raccordate tra di loro, le quali sono state generate da una programma di sintesi vocale online. Ciascuna frase riporta una serie di informazioni oppure un indicazione su come procedere durante le operazioni ed è stata convertita in clip audio trasmessa attraverso una sorgente sonora disposta nella scena virtuale.

Unity dispone di un proprio sistema di gestione della componente audio, attraverso una sintesi binaurale, che permette appunto di immergere l'utente in un'esperienza uditiva spazializzata, con una diversa ricezione del suono in base alla distanza dalla sorgente e all'orientamento della testa nell'ambiente 3D.

I componenti di riferimento con cui è stato generato il voice-over sono essenzialmente due: l'*AudioSource* (Figura 5.28), il quale consiste in una sorgente sonora in grado di riprodurre clip audio preimpostati, con un ampio asset di parametri modificabili tramite Inspector e l'*AudioListener*, che consente di acquisire l'audio 3D senza possibilità di modifiche, basandosi su HTRF¹⁵ standard.

Le clip audio aggiornate e caricate sull'*AudioSource* per la riproduzione sono raccolte e gestite da un unico script, il quale tiene conto della clip corrente appena trasmessa e permette all'utente di avanzare nelle operazioni da svolgere tra le varie fasi della simulazione.

¹⁵ Head-Related Transfer Function (HRTF) costituiscono una risposta che contraddistingue le orecchie umane ad un suono spazializzato che giunge da un punto dello spazio, definendo appunto due funzioni di trasferimento che caratterizzano ciascun orecchio nell'acquisizione di un suono.

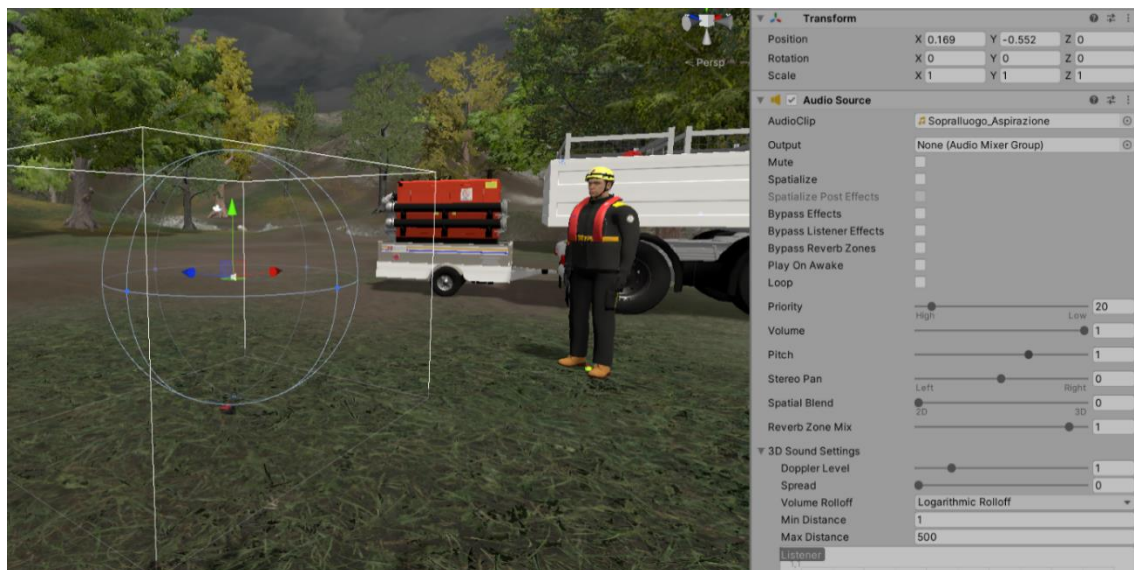


Figura 5.28: Il component AudioSource all'interno del prefab del Camera Rig.

5.5 Altri sistemi implementati

5.5.1 Sistema di interazione

Le interazioni che l'utente può compiere durante l'esperienza virtuale sono state implementate a partire dalla libreria SteamVR, in grado di supportare ogni tipologia di visore sul piano hardware e ricco di componenti quali script, prefab e scene d'esempio per la progettazione di una applicazione VR.

Per la generazione di un modello che comprendesse una camera virtuale e due oggetti mano è stato utilizzato il prefab Camera Rig, il quale permette di impostare correttamente il tracciamento del visore e dei controller mediante una serie di script predefiniti e un'area di interazione tridimensionale, generata a partire dalla calibrazione del dispositivo.

I controller permettono all'utente di muoversi nello scenario virtuale utilizzando la tecnica del arm-swinging, che traduce le informazioni di posizione e di rotazione dei controller nello spazio fisico in un movimento del Camera Rig all'interno della scena virtuale, spostamento definito in base ad un sistema di riferimento locale con origine nella camera 3D. L'utente, oscillando le braccia con una posa simile ad movimento di corsa e tenendo premuto un tasto apposito sul controller, controlla la camminata, con la possibilità di gestire la direzione verso cui muoversi e di variare la velocità in base alla forza con cui si compiono le oscillazioni.

Le interazioni sono state progettate in base alla tipologia di azione che l'utente deve compiere e con quale elemento deve interagire. Le modalità di interazione principali sfruttate durante la simulazione sono riportate di seguito.

- Raycast che consiste nel generare un puntatore che, interagendo con un oggetto provvisto di *Collider*¹⁶, permette di selezionarlo o di definire la punto nello spazio 3D. I due casi principali in cui l'utente utilizza questa modalità sono le interazioni con il menù offline e la scelta del punto dove posizionare la motopompa sul terreno, nella fase di posizionamento dell'attrezzatura;
- Interazione con gli elementi tramite Collider e un tasto prescelto per la selezione. Questa modalità è la più diffusa nelle applicazioni VR interattive, poiché permette di selezionare, sollevare, rilasciare oggetti attraverso la semplice penetrazione tra essi e la mano virtuale. L'interazione avviene tramite un'interazione tra i collider posti sui modelli 3D che rappresentano i controller e quelli dell'oggetto e la pressione di un tasto apposito che permette di sollevare e controllare l'elemento 3D.

¹⁶ Il component Collider definiscono la forma di un GameObject ai fini della gestione delle collisioni fisiche. Un collider non costituisce un elemento grafico, essendo invisibile, e non deve necessariamente avere la stessa forma della mesh dell'oggetto, per una questione di efficienza in fase di gameplay.

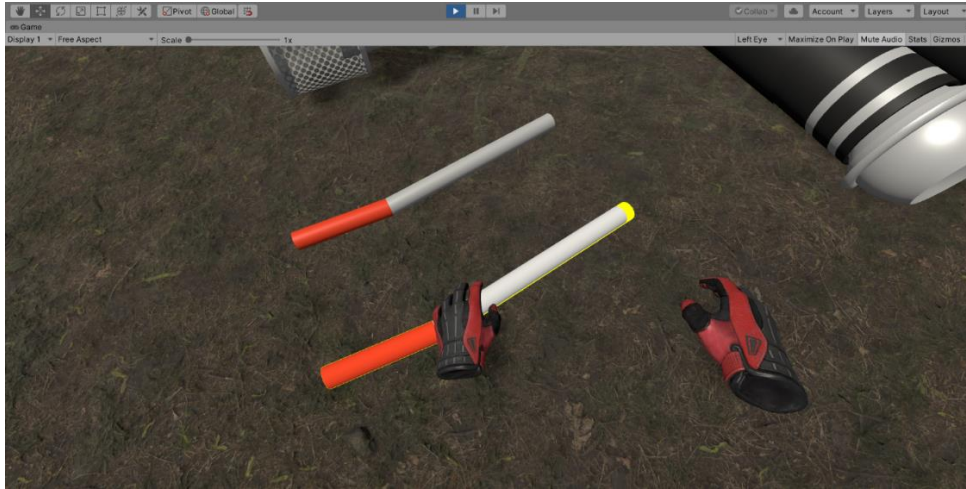


Figura 5.29: Modalità di selezione all'interno della simulazione per gli oggetti interagibili.

5.5.2 Menu iniziale e il sistema di valutazione

Il sistema di selezione delle procedure scelte per ogni esperienza di addestramento VR con il modulo HCP si basa su una scena in VR nella quale l'utente può scegliere selezionando su uno schermo nello spazio 3D le possibili scelte, spuntando dei *toggle button* che presentano le varie possibilità per ogni fase della messa operativa.

La simulazione comprende un *GameObject* impostato come un *Singleton*, ovvero caratterizzato dall'essere l'unico elemento del sistema a non venire mai eliminato quando l'applicazione è accesa e ad avere il componente chiamato *GameManager*, lo script assegnato per la gestione delle varie fasi giocabili e il controllo dei parametri di valutazione. Il *GameManager*, infatti, mantiene le modifiche compiute da menu durante la selezione dei compiti da svolgere durante l'addestramento e si interfaccia automaticamente con lo script che gestisce il voice over per controllare l'esperienza in base alle scelte prese dall'utente.

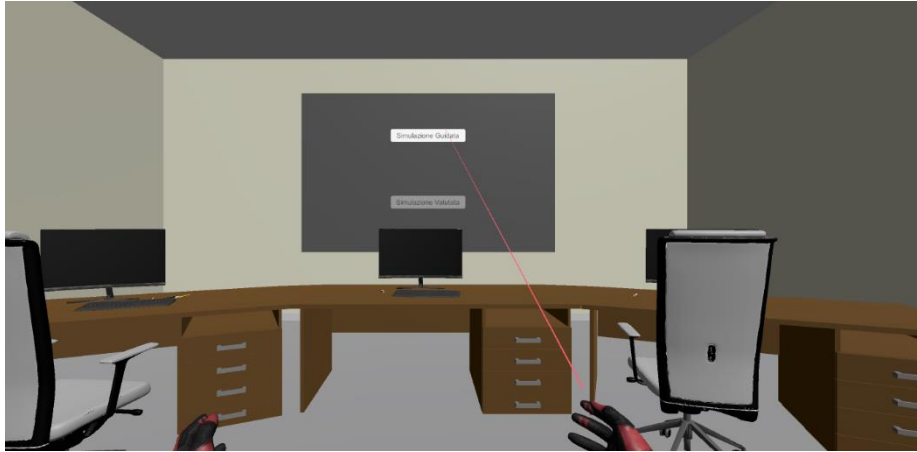


Figura 5.30: Menu iniziale con il RayCast per la selezione delle operazioni.

Nella scena di gioco è presente, inoltre, uno script apposito per il controllo dell'operato dell'utente, che si avvia solamente nella versione valutata della simulazione che tiene traccia delle azioni svolte e attraverso un sistema di check real-time, definisce una rapporto finale degli errori e delle dimenticanze dell'utente nello svolgimento dell'esperienza VR.

Uno script calcola i parametri di valutazione durante l'operato dell'utente, impostando nelle varie fasi dei trigger che permettono di definire se l'operazione sia stata svolta nella maniera corretta, oppure no. Il contenuto dello script di valutazione viene raccolto dal GameManager, il quale, essendo nello stato di DontDestroyOnLoad (elemento non eliminabile in fase di caricamento da una scena all'altra) permette di mantenere i risultati raccolti durante la sequenza di operazioni e di presentarla sotto forma di testo nel menu iniziale, a cui l'applicazione riporta al termine dell'esperienza di addestramento.

5.5.3 Obi Rope e gestione delle manichette

All'interno del progetto di tesi si è deciso di utilizzare l'asset Obi Rope, (versione 4.3) per la gestione della fisica delle manichette flessibili e semirigide necessarie

a costruire le tubazioni di aspirazione e mandata. Per implementarne le funzionalità, è stato creato un gameObject che ha assunto il ruolo di *Obi Solver* (assegnandogli lo script omonimo), caratterizzato da una serie di parametri, tra cui il massimo numero di particelle simulate per ogni tubo, il tempo di aggiornamento delle particelle nello spazio, la loro gravità e la soglia di energia cinetica sotto la quale ignorare eventuali modifiche al loro stato corrente.

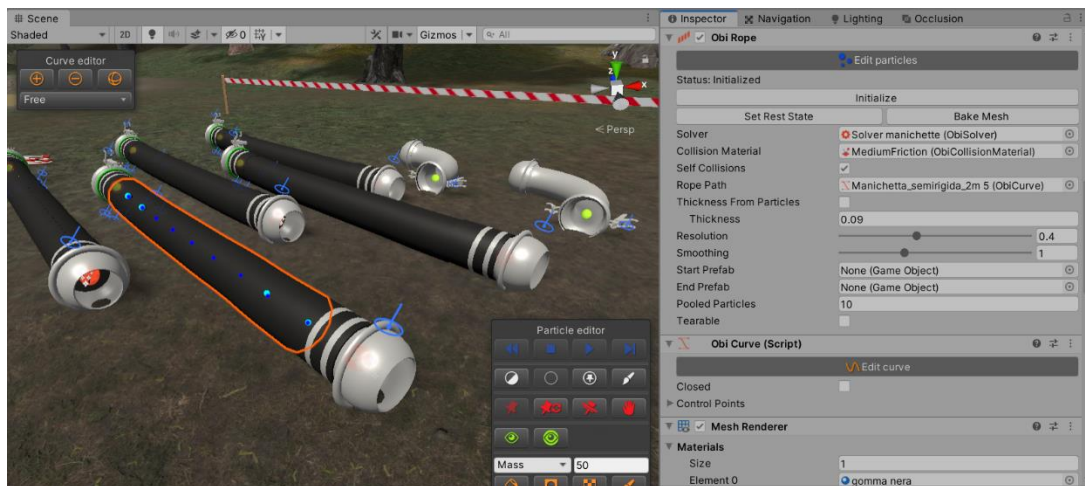


Figura 5.31: Il component *Obi Rope* assegnato ad un modello, in Unity.

Come *Obi Actor* si è invece utilizzato il component *Obi Rope*, che costruisce una catena di particelle sul modello del tubo, secondo una curva *spline* chiamata *Obi Curve*, caratterizzata da punti di controllo manipolabili da editor. Una volta generato il profilo della manichetta, si istanzia l'Actor, il quale crea le particelle e le dispone nell'ambiente secondo il grado di risoluzione ricercato. Ogni Actor è stato opportunamente modificato nei parametri relativi alle singole particelle come raggio, massa, gestione di vincoli di distanza, vincoli di piegamento e vincoli di fissaggio. Sono stati realizzati due modelli 3D in Blender che corrispondono alle estremità del tubo reale, combinati con la parte gestita da *Obi Rope*, attraverso i componenti *Obi Collider* e *Obi Rigidbody* (in grado di controllare l'interazione fisica tra elemento di *Obi Rope* e altri oggetti fisici della scena).

Durante la realizzazione della logica di interazione, sono stati creati degli script appositi, presenti sui modelli rappresentanti le due estremità. Questi script sono pensati per controllare l'aggancio dell'estremità femmina con l'estremità del maschio, attraverso un determinato procedimento di aggancio mediante leva. Le manichette che vengono agganciate, in ordine, tra loro formano le tubazioni di mandata e aspirazione, gestite attraverso un apposito manager.

Capitolo 6: Conclusione

La simulazione finale in VR, frutto dell'intero progetto, si presenta come un prodotto multimediale innovativo, destinato alla formazione dei volontari della PC della Regione Piemonte sul Modulo HCP. L'applicazione realizzata è stata progettata per sostenere un addestramento completo, in grado di preparare volontari inesperti così come offrire un valido mezzo di esercitazione per gli operatori già competenti, in quanto costituisce un programma educativo e di formazione o per l'intervento della squadra HCP in caso di rischio idrogeologico.

Purtroppo, non sono stati realizzati test sperimentali, allo scopo di raccogliere dati sull'efficacia educativa e sulle debolezze del progetto (a causa della pandemia da *Covid 19* scoppiata ad inizio 2020). Tuttavia, si sono tenuti i contatti in via telematica con i rappresentanti della PC piemontese, raccogliendo commenti e richieste di modifiche durante l'ultima fase di implementazione; ciò ha permesso di includere ogni procedura ritenuta importante per la formazione dei volontari, senza il rischio di escludere operazioni e azioni potenzialmente vitali in un addestramento pratico come quello a cui è destinata la simulazione.

Di seguito, sono riportati i principali punti critici e le problematiche riscontrate durante le fasi di realizzazione dell'applicazione che hanno rappresentato un ostacolo durante la fase di lavoro oppure che, allo stato attuale, non è stato possibile superare:

- Durata e complessità della simulazione: la prima implementazione della simulazione ha presentato al potenziale utente un'esperienza VR troppo lunga per essere svolta in un'unica sessione. Questo ha portato alla decisione di definire una serie di fasi e di procedure separate, in grado di permettere all'utente di scegliere quali compiti svolgere e quali lasciare al sistema. Ogni fase non eseguita dall'operatore è stata completata programmata in modo da venire svolta in modalità automatizzata. Il

sistema d'automazione progettato permette di eseguire in autonomia ogni fase che l'utente non desidera apprendere durante la simulazione, attraverso l'uso di espedienti;

- Ottimizzazione: uno dei problemi principali con cui si ha avuto a che fare durante il lavoro di implementazione è quello legato alla frequenza di riproduzione dei fotogrammi. Un'applicazione in VR necessita di un numero di fotogrammi al secondo al di sopra di una certa soglia (solitamente 90 fps), per evitare che l'utente fruisca di un'esperienza "scattosa" e molto disturbante dal punto di vista degli stimoli sensoriali. La simulazione in questione contiene numerosi modelli 3D da renderizzare, alcuni particellari necessari per aumentare il realismo e i componenti gestiti dall'asset Obi Rope per il controllo delle manichette; tali elementi hanno portato un elevato costo computazione sia da parte della CPU nella gestione della fisica del sistema, sia da parte della GPU nella fase di rendering della scena virtuale.

La necessità di portare l'applicazione a mantenere un frame rate stabile ha richiesto, come detto, la realizzazione di LOD e billboard per gli elementi scenografici (alberi e rocce) e ad una fase di retopology dei modelli 3D allo scopo di limitare il numero di draw call richieste per il rendering della scena; è stato opportuno, inoltre, ottimizzare gli script nelle funzioni chiamate in loop da Unity (*Update*, *FixedUpdate* e *LateUpdate*), cercando di aumentare la presenza di caching¹⁷ e di limitare la chiamata a metodi eccessivamente dispendiosi dal punto di vista computazionale;

- Gestione delle manichette: l'impegno nella gestione delle manichette, affrontata con l'intento di ricreare nella maniera più realistica possibile la fisica reale e l'interazione con esse da parte dell'utente, ha mostrato numerose limitazioni nell'utilizzo di Obi Rope, tra cui, ad esempio,

l'impossibilità di far compiere alle manichette degli spostamenti troppo bruschi. Infatti, l'asset utilizzato è progettato per la gestione di oggetti con una mesh molto più semplice di quella utilizzata per le manichette, mentre la simulazione oggetto del progetto necessita di un lavoro di implementazione decisamente differente e molto più complesso rispetto all'utilizzo comune per la quale il tool è stato progettato. Per aggirare queste limitazioni sono state applicate soluzioni che hanno favorito la maggiore efficienza e controllo della fisica, a scapito del minore realismo.

L'evoluzione della simulazione in ottica di approfondire la formazione degli operatori e di migliorare l'esperienza VR potrebbe comportare l'aggiunta di alcuni elementi tra cui quelle riportate di seguito:

- Modalità multiutente: inizialmente progettata ma non implementata nello stato attuale, che permetterebbe di migliorare l'esperienza di cooperazione con altri operatori durante le procedure richieste dall'intervento del Modulo, come il trasporto e il montaggio delle manichette.
- Nuove ambientazioni: la simulazione è stata realizzata per essere svolta in uno scenario generico boschivo, tuttavia si potrebbe diversificare la scelta iniziale portando l'utente a operare in altri contesti come l'allagamento di un sottopassaggio o di un'area urbana.
- Nuovi casi d'uso: l'esperienza VR presenta un addestramento sul processo di messa operativa standard per l'utilizzo di una motopompa autoadescante; i casi particolari, come la presenza di ostacoli o l'insorgere di un guasto, permetterebbero di ampliare la formazione degli operatori e di coprire così tutti i possibili problemi che potrebbero insorgere, garantendo un addestramento completo per ogni possibile emergenza.

- Nuove procedure: la simulazione presenta le operazioni richieste dal Modulo HCP per l'utilizzo di una motopompa Godwin, ma la squadra di operatori può essere attrezzata anche all'utilizzo di diverse tipologie di pompa (ad esempio, la pompa Albatros o ad immersione) o dover procedere con una serie di ulteriori operazioni determinate dalla situazione in cui ci si trova a dover intervenire (la costruzione di argini temporanei con barriere apposite, l'utilizzo di vasche per il trasporto dell'acqua o l'evacuazione della popolazione).

Bibliografia

[1] PITEM RISK – Progetto RISK - FOR:

<https://www.regione.piemonte.it/web/temi/fondi-progetti-europei/programmi-progetti-europei/cooperazione-territoriale-europea-piemonte/pitem-risk-progetto-risk-for>

[2] Interreg-ALCOTRA, Progetto PITEM RISK: [http://www.interreg-](http://www.interreg-alcotra.eu/it/decouvrir-alcotra/les-projets-finances/risk-resilienza-informazione-sensibilizzazione-e)

[alcotra.eu/it/decouvrir-alcotra/les-projets-finances/risk-resilienza-informazione-sensibilizzazione-e](http://www.interreg-alcotra.eu/it/decouvrir-alcotra/les-projets-finances/risk-resilienza-informazione-sensibilizzazione-e)

[3] Città Metropolitana di Torino, PITEM RISK:

<http://www.cittametropolitana.torino.it/cms/protezione-civile/speciale-pitem-risk/speciale-pitem-risk-conferenza-stampa>

[4] Link Regione Piemonte, bollettino ufficiale dall'archivio 2019 in data 24/10:

http://www.regione.piemonte.it/governo/bollettino/abbonati/2019/43/attach/dai810003358_1040.pdf

[5] Sistema nazionale della Protezione Civile:

<http://www.protezionecivile.gov.it/servizio-nazionale>

[6] Dipartimento di Protezione Civile:

<http://www.protezionecivile.gov.it/dipartimento>

[7] Protezione Civile, Colonna Mobile regionale:

<https://www.regione.piemonte.it/web/temi/protezione-civile-difesa-suolo-opere-pubbliche/protezione-civile/logistica/colonna-mobile-regionale>

[8] Protezione Civile, descrizione rischio idrogeologico:

<http://www.protezionecivile.gov.it/attivita-rischi/meteo-idro/descrizione>

[9] <https://www.laprotezionecivile.com/quaderni/quaderno8.pdf>

[10] APELL-EUROMODEX: <http://www.apell-euromodex.eu/>

[11] Esercitazione Internazionale H.C.P. in Montenegro: http://protezionecivile-coordinamentovc.it/e107_plugins/content/content.php?content.96

- [12] Burdea, G. and Coiffet, P., 2003, Virtual Reality Technology, Wiley, New Jersey.
- [13] Sherman, W. R. and Craig, A., 2002. Understanding Virtual Reality: Interface, Application and Design, Morgan Kaufmann.
- [14] Laval, S. M., 2015. Virtual reality, Cambridge University Press.
- [15] Mestre, D. and Vercher, J., 2011. Immersion and presence. Virtual Reality.
- [16] ISPR. International Society for Presence Research, 2000. The Concept of Presence: Explication Statement.
- [17] Slater, Mel, 2018. Immersion and the illusion of presence in virtual reality. British Journal of Psychology, 109(3), pp.431-433.
- [18] Jerald, J., 2015, The VR Book: Human-Centered Design for Virtual Reality, Association for Computing Machinery and Morgan & Claypool.
- [19] Sutherland, I., 1968. A head-mounted three dimensional display. Proceedings of the December 9-11, 1968, fall joint computer conference, part I on - AFIPS '68 (Fall, part I).
- [20] Martirosov, S. and Kopecek, P., 2017. Virtual Reality and its Influence on Training and Education - Literature Review. DAAAM Proceedings, pp.0708-0717.
- [21] Cassidy, B., Sim, G., Robinson, D. and Gandy, D., 2018. A Virtual Reality Platform for Analyzing Remote Archaeological Sites.
- [22] Virtual reality and archaeological reconstruction: be there, back then. MW17: MW 2017. Pubblicato a febbraio, 2017. Consultato a maggio, 2020: <https://mw17.mwconf.org/paper/virtual-reality-and-archaeological-reconstruction-be-there-be-back-then-ullastret3d-and-vr-experience-in-htc-vive-and-immersive-room/>
- [23] Gonçalves, R., Pedrozo, A., Coutinho, E., Figueira, I. and Ventura, P., 2012. Efficacy of Virtual Reality Exposure Therapy in the Treatment of PTSD: A Systematic Review. PLoS ONE, 7(12), p.e48469.

- [24] Garcia-Palacios, A., Hoffman, H., Carlin, A., Furness, T. and Botella, C., 2002. Virtual reality in the treatment of spider phobia: a controlled study. *Behaviour Research and Therapy*, 40(9), pp.983-993.
- [25] Slater, M., Pertaub, D., Barker, C. and Clark, D., 2006. An Experimental Study on Fear of Public Speaking Using a Virtual Environment. *CyberPsychology & Behavior*, 9(5), pp.627-633.
- [26] Brian Chau, D., 2020. Immersive Virtual Reality Therapy With Myoelectric Control For Treatment-Resistant Phantom Limb Pain: Case Report. [online] PubMed Central (PMC). Available at:
<http://www.ncbi.nlm.nih.gov/pmc/articles/pmc5880370/> [Accessed at June 2020].
- [27] Wouters, P., van Nimwegen, C., van Oostendorp, H. and van der Spek, E., 2013. A meta-analysis of the cognitive and motivational effects of serious games. *Journal of Educational Psychology*, 105(2), pp.249-265.
- [28] Tanaka, E., Paludo, J., Bacchetti, R., Gadbem, E., Domingues, L., Cordeiro, C., Giraldi, O., Alcarde Gallo, G., Mendes da Silva, A. and Cascone, M., 2017. Immersive virtual training for substation electricians. 2017 IEEE Virtual Reality (VR),.
- [29] Clifford, R., Jung, S., Hoermann, S., Billinghamurst, M. and Lindeman, R., 2019. Creating a Stressful Decision Making Environment for Aerial Firefighter Training in Virtual Reality. 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR),.
- [30] Jeon, S., Han, J., Jo, Y. and Han, K., 2019. Being More Focused and Engaged in Firefighting Training: Applying User-Centered Design to VR System Development. 25th ACM Symposium on Virtual Reality Software and Technology,.
- [31] Alfalah, S., Falah, J., Alfalah, T., Elfalah, M., Muhaidat, N. and Falah, O., 2018. A comparative study between a virtual reality heart anatomy system and traditional medical teaching modalities. *Virtual Reality*, 23(3), pp.229-234.

- [32] Saunders, J., Davey, S., Bayerl, P. and Lohrmann, P., 2019. Validating Virtual Reality as an Effective Training Medium in the Security Domain. 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR),.
- [33] Lang, B., 2020. Boeing Is Using Varjo Headsets To Train Astronauts For An ISS Docking Mission In 2021. Road to VR. Pubblicato a giugno 2020. Consultato a giugno 2020: <https://www.roadtovr.com/boeing-varjo-virtual-reality-astronaut-training-starliner/>
- [34] Hafsia, M., Monacelli, E. and Martin, H., 2018. Virtual Reality Simulator for Construction workers. Proceedings of the Virtual Reality International Conference - Laval Virtual on - VRIC '18,.
- [35] Progetto - Prodige. Consultato a Novembre 2019: <http://www.pro-prodige.eu/>
- [36] European Environment Agency. Living In A Changing Climate. Pubblicato nel giugno 2015. Consultato nel giugno 2020: <https://www.eea.europa.eu/signals/signals-2015/articles/living-in-a-changing-climate>
- [37] Anthes, C., Garcia-Hernandez, R., Wiedemann, M. and Kranzlmuller, D., 2016. *State of the art of virtual reality technology*. 2016 IEEE Aerospace Conference.
- [38] HTC Vive: <https://www.vive.com/eu/>
- [39] Vive Specifications: <https://developer.vive.com/resources/knowledgebase/vive-specs/>
- [40] Oculus Rift: https://www.oculus.com/rift/?locale=it_IT#oui-csl-rift-games=mages-tale
- [41] Oculus Device Specifications: <https://developer.oculus.com/design/oculus-device-specs/>
- [42] Unity: <https://unity.com/>
- [43] Unity Manual: <https://docs.unity3d.com/Manual/UnityManual.html>

- [44] Unity Asset Store: <https://assetstore.unity.com/>
- [45] Visual Studio: <https://visualstudio.microsoft.com/it/vs/>
- [46] Blender: <https://www.blender.org/>
- [47] MakeHuman Community: <http://www.makehumancommunity.org/>
- [48] Mixamo: <https://www.mixamo.com/>
- [49] Obi Rope: <https://assetstore.unity.com/packages/tools/physics/obi-rope-55579>
- [50] Unity Terrain Editor: <https://learn.unity.com/tutorial/working-with-the-terrain-editor#>
- [51] Cascade - River, Lake, Waterfall and more:
<https://assetstore.unity.com/packages/tools/terrain/cascade-river-lake-waterfall-and-more-106072>
- [52] SteamVR plugin:
<https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647>
- [53] SteamVR 2.0 e Action Sets: <https://connect.unity.com/p/steamvr-2-0-starter-migration-action-sets>
- [54] Final IK sull'Asset Store di Unity:
<https://assetstore.unity.com/packages/tools/animation/final-ik-14290>
- [55] RootMotion: <http://root-motion.com/>
- [56] Navigation System in Unity: <https://docs.unity3d.com/Manual/nav-NavigationSystem.html>