

# POLITECNICO DI TORINO

Laurea Magistrale  
in Ingegneria Informatica

Tesi di Laurea Magistrale

**Realtà Virtuale per l'addestramento alla  
gestione del rischio idrogeologico:**

*Sistema di interazione, logica per il controllo degli  
agenti e strategia di valutazione*



**Relatore**

Prof. Fabrizio Lamberti

**Candidato**

Gisella Vignola

Sessione di Luglio 2020



# Sommario

Il progetto *PITEM RISK* nasce all'interno del programma *INTERREG Italia-Francia ALCOTRA*, con il fine di sviluppare strumenti adatti alla gestione delle emergenze nelle zone di frontiera, consentendo una maggior comprensione dei rischi e dei giusti comportamenti da adottare in caso di crisi.

All'interno del *PITEM RISK* si sviluppa un sotto-progetto specifico nell'ambito della prevenzione e addestramento denominato *PITEM RISK FOR*: questo prevede una collaborazione tra la Regione Piemonte, la Protezione Civile, ed il Politecnico di Torino con lo scopo di sviluppare la parte formativa del piano in una rete condivisa.

L'intero progetto andrà quindi a fornire diversi metodi di apprendimento indirizzati ad utenti differenti in grado di formare questi ultimi sulle procedure adatte a più tipologie di rischio, tra cui emergenze di tipo idrogeologico e connesse agli incendi boschivi.

La necessità di creare un sistema di addestramento efficiente e immediato per la gestione in particolare del rischio idrogeologico determina l'obiettivo di questa tesi che si realizza attraverso una simulazione didattica in Realtà Virtuale o *Virtual Reality*.

La collaborazione con il Coordinamento Regionale della Protezione Civile ha portato così allo sviluppo di uno strumento didattico innovativo basato sull'utilizzo delle suddette tecnologie, scelte con lo scopo di realizzare al meglio un'esperienza verosimile per gli operatori in addestramento. Il progetto si traduce in una simulazione incentrata sulle operazioni e sui protocolli attuati dalla Protezione Civile nel primo intervento, in caso di rischio idrogeologico, tramite l'utilizzo del modulo europeo denominato *High Capacity Pumping*.

In particolare il presente lavoro di tesi si focalizza su una parte del summenzionato strumento, analizzando nello specifico la progettazione e la realizzazione del sistema di interazione, la logica d'automazione dei Personaggi Non-Giocanti (*Non-Playing Character*) e sul sistema di valutazione delle azioni dell'utente.



# Ringraziamenti

Ringrazio tutte le persone che mi sono state vicino, supportata e sopportata durante questi anni.

In particolare la mia famiglia, che non mi ha mai fatto mancare nulla, e il mio cane Artù, che mi ha tenuto calda d'inverno.



# Indice

<b>1</b>	<b>Introduzione</b>	<b>15</b>
1.1	Il Progetto PITEM RISK . . . . .	15
1.2	Protezione Civile . . . . .	16
1.2.1	Rischio idrogeologico . . . . .	17
1.2.2	Il Modulo Europeo HCP . . . . .	18
1.3	La Realtà Virtuale . . . . .	19
1.3.1	Definizione di Realtà Virtuale . . . . .	19
1.3.2	Immersione e presenza . . . . .	20
1.4	Obiettivi del progetto e della tesi . . . . .	21
1.5	Organizzazione delle attività . . . . .	21
1.6	Struttura del documento . . . . .	22
<b>2</b>	<b>Stato dell'arte</b>	<b>24</b>
2.1	Realtà Virtuale . . . . .	24
2.1.1	Storia . . . . .	24
2.1.2	Campi d'utilizzo . . . . .	28
2.1.3	Serious Game . . . . .	30
2.1.4	Realtà Virtuale multi-utente . . . . .	31
2.2	La Realtà Virtuale per l'addestramento . . . . .	31
2.2.1	Vantaggi e motivazioni dell'approccio . . . . .	32
2.2.2	Svantaggi dell'approccio . . . . .	34
2.3	Realtà Virtuale e gestione delle emergenze . . . . .	35

2.3.1	Rischio esondazione nel parco fluviale Gesso e Stura a Cuneo . . .	36
2.3.2	Incidente stradale presso St.Paul-sur-Ubaye . . . . .	37
2.3.3	Dispersione di sostanze chimiche sul Colle della Maddalena . . .	37
2.3.4	Nube tossica da sostanza radioattiva in avvicinamento a Torino .	38

### **3 Tecnologie** 40

3.1	Hardware . . . . .	40
3.1.1	HTC Vive . . . . .	41
3.1.2	Oculus Rift . . . . .	42
3.1.3	Altre tecnologie sul mercato . . . . .	44
3.2	Software . . . . .	44
3.2.1	Game Engine . . . . .	44
3.2.2	Unity . . . . .	45
3.2.3	Microsoft Visual Studio . . . . .	47
3.2.4	Blender . . . . .	47
3.2.5	MakeHuman e Mixamo . . . . .	48

### **4 Progettazione** 49

4.1	Acquisizione del materiale . . . . .	49
4.1.1	Progetto PITEM, Protezione Civile e Regione Piemonte . . . . .	49
4.1.2	Esercitazioni e sopralluoghi . . . . .	50
4.2	Documento di design . . . . .	51
4.2.1	Il soggetto . . . . .	51
4.2.2	Modalità supportate . . . . .	53
4.2.3	Il design dell'ambiente . . . . .	53

4.2.4	Il Modulo HCP . . . . .	55
4.2.5	Personaggi non giocatori . . . . .	58
4.2.6	Interfaccia di gioco . . . . .	59
4.2.7	Voice-over, indizi ed errori . . . . .	61
4.2.8	Valutazione . . . . .	61
4.2.9	Logica di simulazione . . . . .	63

## 5 Realizzazione 66

5.1	Sistema d'interazione . . . . .	66
5.1.1	Steam VR . . . . .	66
5.1.2	Open VR . . . . .	67
5.1.3	Raycasting . . . . .	69
5.1.4	Gestione delle collisioni in Unity . . . . .	70
5.1.5	Locomozione . . . . .	73
5.1.6	Line Renderer . . . . .	75
5.1.7	Implementazione . . . . .	76
5.2	Logica d'automazione degli NPC . . . . .	79
5.2.1	Macchine a Stati Finiti . . . . .	80
5.2.2	Animazione . . . . .	82
5.2.3	Cinematica inversa . . . . .	83
5.2.4	Final IK . . . . .	85
5.2.5	NavMesh . . . . .	87
5.2.6	Implementazione . . . . .	88
5.3	Strategia di Valutazione . . . . .	96
5.4	Altri sistemi . . . . .	98
5.4.1	Ambiente . . . . .	98
5.4.2	Gestione delle manichette . . . . .	99

5.4.3 Voice-over . . . . . 101

**6 Conclusioni e sviluppi futuri** 103

6.1 Conclusioni . . . . . 103

6.2 Sviluppi futuri . . . . . 104

# Elenco delle figure

2.1	Sensorama, Morton Heilig, 1961 . . . . .	25
2.2	La spada di Damocle, Ivan Sutherland, 1968 . . . . .	25
2.3	PowerGlove, Nintendo, 1989 . . . . .	26
2.4	SEGA VR, SEGA, 1993 . . . . .	27
2.5	Virtual Boy, Nintendo, 1995 . . . . .	28
2.6	Dettaglio di una simulazione VR in ambito medico-chirurgico . . . . .	32
2.7	Dettaglio di una simulazione VR in ambito aeronautico . . . . .	34
2.8	Scenario Parco Gesso e Stura di PRODIGE con pedana Virtualizer . . . . .	36
2.9	Dettaglio dello scenario St. Paul-sur-Ubaye di PRODIGE . . . . .	37
2.10	Scenario dell'incidente sul Colle della Maddalena di PRODIGE . . . . .	38
2.11	Dettaglio dello scenario relativo alla propagazione della nube tossica di PRODIGE . . . . .	39
3.1	HMD del sistema VR HTC Vive . . . . .	41
3.2	Controller del HTC Vive . . . . .	42
3.3	HMD e controller dell'Oculus Rift . . . . .	43
3.4	Interfaccia di Unity 2019.3.2 . . . . .	46
3.5	Interfaccia Blender 2.82 . . . . .	48
4.1	Fiume e pozza modellati nello scenario di simulazione . . . . .	54
4.2	Modulo HCP posizionato nell'ambiente di simulazione . . . . .	55
4.3	Motopompa Godwin (fotografia e ricostruzione 3D) . . . . .	56

4.4	Equipaggiamento di un Modulo HCP . . . . .	57
4.5	Equipaggiamento di un operatore della PC per rischio idrogeologico . . . . .	58
4.6	Sala operativa PC . . . . .	59
4.7	Flow chart del menù iniziale . . . . .	60
5.1	Input Binding UI . . . . .	68
5.2	Esempio d'utilizzo del raycasting e parametri modificabili . . . . .	69
5.3	Esempi di Collider . . . . .	70
5.4	Componente Rigidbody . . . . .	71
5.5	Tabella degli eventi generabili: OnCollision (X) OnTrigger (Y) . . . . .	72
5.6	Esempio di Teleporting . . . . .	73
5.7	Esempio di Arm Swinging . . . . .	74
5.8	Esempio di Line Renderer . . . . .	76
5.9	Outline implementato per l'interazione con gli oggetti . . . . .	77
5.10	Sagoma della pompa durante il suo posizionamento tramite Raycasting . . . . .	78
5.11	Line Renderer utilizzato per la distensione del nastro . . . . .	79
5.12	Esempio di FSM . . . . .	81
5.13	Interfaccia Animator . . . . .	83
5.14	Esempio di catena cinematica . . . . .	84
5.15	Componente Full Body Biped IK . . . . .	86
5.16	Target per Right End Effector . . . . .	86
5.17	Esempio di NavMesh . . . . .	87
5.18	Componente NavMesh Agent . . . . .	88
5.19	Stato di Grab . . . . .	89
5.20	Stato di Carry . . . . .	90

5.21	FSM implementata . . . . .	90
5.22	Aspirazione Master . . . . .	92
5.23	Aspirazione Slave . . . . .	93
5.24	Mandata Slave . . . . .	94
5.25	Controllo perdite . . . . .	95
5.26	Gestione del quadro di comando . . . . .	96
5.27	Componente Terrain . . . . .	99
5.28	Esempi di utilizzo di Obi Rope . . . . .	100
5.29	Esempi di icone e highliner implementati . . . . .	102



# Capitolo 1

## Introduzione

La necessità di creare un sistema di addestramento efficiente e immediato per la gestione del rischio idrogeologico, così diffuso del nostro paese, determina l'obiettivo di questa tesi che si realizza attraverso una simulazione didattica in *Realtà Virtuale* o *Virtual Reality* (VR).

La collaborazione del Coordinamento Regionale della Protezione Civile (PC) per il Piemonte ha portato così allo sviluppo di un progetto didattico innovativo basato sull'utilizzo dell'emergente tecnologia VR, scelta con lo scopo di realizzare al meglio un'esperienza verosimile per gli operatori in addestramento. Il progetto si traduce in una simulazione incentrata sulle operazioni e sui protocolli attuati dall'organizzazione della PC nel primo intervento in caso di rischio idrogeologico.

### 1.1 Il Progetto PITEM RISK

Il progetto *PITEM RISK* [1] nasce all'interno del programma *INTERREG Italia-Francia ALCOTRA* [2], con il fine di sviluppare strumenti adatti alla gestione delle emergenze nelle zone di frontiera, tra le regioni di Liguria, Piemonte, Valle d'Aosta, Auvergne Rhône-Alpes e Provenza-Alpi-Costa Azzurra.

Il progetto tenta di migliorare la risposta alle emergenze, consentendo una maggior comprensione dei rischi e dei giusti comportamenti da adottare in caso di crisi. Inoltre si rivolge non solo agli operatori della PC ma anche ai volontari del corpo *Antincendi Boschivi*, civili e altri tecnici impiegati nella sicurezza del territorio.

All'interno del *PITEM RISK* si sviluppa un sotto-progetto specifico nell'ambito della prevenzione e addestramento denominato *PITEM RISK FOR* [3]: questo prevede una collaborazione tra la Regione Piemonte, la PC regionale ed il Politecnico di Torino con lo scopo di sviluppare la parte formativa del piano in una rete condivisa.

La rete sarà poi resa disponibile all'intero territorio *ALCOTRA* attraverso "centri complementari e interconnessi VR, che lavoreranno su una base software condivisa" [3] e verrà utilizzata nei campi della:

- formazione degli interessati (compresi i civili) nelle procedure confacenti ai ruoli che gli competono;
- cooperazione tra i diversi servizi (emergenza, amministratori civili, ecc.) per il miglioramento della collaborazione.

Il progetto andrà quindi a fornire un diverso metodo di apprendimento indirizzato ad utenti diversi e in grado di supportare la formazione sulle procedure per più tipologie di rischio, tra cui emergenze di tipo idrogeologico e incendio boschivo.

## 1.2 Protezione Civile

La PC è un ente pubblico che svolge attività comprendenti l'insieme delle procedure messe in atto con lo scopo di tutelare le persone, gli insediamenti, i beni, l'ambiente e gli animali dal pericolo o dai danni causate da calamità naturali sia a livello nazionale che locale.

Il *Servizio Nazionale della Protezione Civile* (istituito dalla legge 24 febbraio 1992, n. 225) [4] è una struttura del governo della Repubblica Italiana, facente capo alla Presidenza del Consiglio dei Ministri, con la responsabilità nella gestione della prevenzione, previsione, gestione e superamento delle calamità e nell'assistenza e soccorso delle popolazioni in pericolo operando a livello Nazionale in collaborazione con le Regioni e Province autonome.

La legge n. 225/92 individua le attività e gli obiettivi della PC concentrandole nella tutela e nella salvaguardia di persone e ambiente e vengono suddivise in:

- previsione, valutazione degli scenari di rischio, monitoraggio degli eventi attesi;
- allertamento, pianificazione, diffusione delle conoscenze attraverso formazioni ed esercitazioni;
- soccorso e prima assistenza alle popolazioni colpite da calamità;
- superamento dell'emergenza fino alla ripresa delle normali condizioni di vita.

Il progetto *Colonna Mobile Nazionale* [5], con D.G.R. del 12.01.2008 n. 12-8046, rappresenta il quadro di riferimento per tutte le strutture del Sistema Regionale di PC con particolare accento sulle procedure operative standard; tale colonna è strutturata tramite una serie di moduli funzionali standard (stabiliti in base alla tipologia di emergenza in atto), materiali e

squadre operative in dotazione alla Regione stessa o alle organizzazioni di volontariato associate.

### 1.2.1 Rischio idrogeologico

Con dissesto idrogeologico si intendono tutti quei fenomeni e i danni causati dall'acqua in ogni sua forma (superficiali, sotterranee, in forma liquida o solida); il termine viene usato in modo improprio in quanto l'idrogeologia è quella parte delle scienze geologiche che studia solamente le acque sotterranee.

Il dissesto idrogeologico in Italia è un fenomeno esteso alla quasi totalità della nazione e tanto frequente da rappresentare un problema notevole nella determinazione di alluvioni, frane, valanghe, erosione degli argini e delle coste.

Il *Servizio Nazionale della Protezione Civile* definisce il rischio in questione differenziandolo in:

- *rischio idrogeologico*, nel caso in cui si presenti il superamento dei livelli pluviometrici critici lungo i versanti, dei livelli di smaltimento delle acque piovane, dei livelli idrometrici dei corsi d'acqua minori e dei conseguenti effetti sul territorio [6];
- *rischio idraulico*, nel caso di superamento dei livelli idrometrici dei corsi d'acqua maggiori e dei relativi effetti sul territorio [6].

Il *Modulo rischio idraulico* costituisce il modulo funzionale adibito al rischio idrogeologico inserito nella colonna mobile della PC precedentemente menzionata. Il modulo comprende strumenti come motopompe, autopompe, tubi, manichette di diversa portata, diametro e lunghezza e squadre di operatori addestrati all'utilizzo del modulo stesso.

A livello nazionale si è andato successivamente a definire un *Modulo specialistico rischio idraulico standard*, atto a fronteggiare le emergenze nazionali. Il modulo ha il principale compito di aspirare e scaricare grandi quantità d'acqua, intervenendo al mantenimento delle condizioni stabili alla vita sul territorio.

L'asset è idoneo ad operare su eventi di rilievo nazionale (tipo C) e ad eventi su scala provinciale o regionale (tipo B).

La Regione Piemonte, presentando un territorio con conformazione fisica e livelli di piovosità idonei al rischio idrogeologico, si è trovata in molte occasioni ad affrontare eventi drammatici quali le alluvioni del Tanaro nel 1994 e 2000 o del Villar Pellice nel 2008 e 2016. Per tal motivo, la Regione Piemonte ha molto investito nella struttura della PC regionale, con il principale scopo di rafforzare le risorse della colonna mobile per il soccorso e il contenimento dei danni dovuti ad eventi di tipo idrogeologico.

### 1.2.2 Il Modulo Europeo HCP

Il modulo *High-Capacity Pumping* (HCP) [7], a seguito del trattato di Lisbona 20/12/2007 – 2008/73/ CE Euratom, definisce lo standard per il Modulo di rischio idrogeologico per Colonna Mobile atto a lavorare in concomitanza al meccanismo di PC dell'Unione Europea. Il modulo consiste in una squadra di operatori specializzati e pronti ad intervenire celermente, con il compito di supportare le autorità e organizzazioni locali nella gestione di calamità di livello europeo e, in genere, diretto dal Centro di Coordinamento della Protezione Civile Europea.

Il modulo segue le direttive internazionali stabilite dalle Nazioni Unite svolgendo le principali attività di drenaggio, monitoraggio delle acque e approvvigionamento idrico per le operazioni antincendio. Il modulo, a livello regionale, deve essere:

- pronto alla partenza entro 12 ore dall'attivazione (6 ore per l'attrezzatura fondamentale e 12 per le rimanenti componenti di supporto);
- attrezzato con motopompe autoadescanti con portata minima di 1200 m/ora e 20 m di dislivello;
- presentare differenza di altezza massima 40 m;
- capace di pompare acqua fangosa al 5% di elementi solidi con dimensioni minime di 40 mm;
- pompare l'acqua fino a un minimo 2500 m utilizzando manichette flessibili con giunti *Storz* (tipo di raccordo largamente impiegato sulle manichette antincendio);
- possibilità di spostamento entro un raggio minimo di 3000 km da Torino (Italia) via terra e via mare;
- autosufficiente per un tempo minimo di 96 ore.

Ogni anno vengono organizzate da *EUROMODEX* (organizzazione non governativa che si occupa della prevenzione dei rischi, naturali e non, a livello europeo) [8] diverse esercitazioni internazionali a cui anche la PC della Regione Piemonte ha partecipato; sono un esempio le esercitazioni del 2017 in Repubblica Ceca e del 2019 in Montenegro [9], in cui è stato possibile svolgere simulazioni di rischio di tipo idrogeologico utilizzando il Modulo HCP in collaborazione con team provenienti da tutta Europa.

La criticità del rischio idrogeologico nella Regione Piemonte e quindi l'importanza del modulo HCP per la colonna mobile regionale, ha indirizzato alla scelta di questo modulo come riferimento per lo sviluppo della simulazione VR nel progetto *PITEM RISK FOR*. La simulazione si pone infatti l'obiettivo della formazione nell'utilizzo della pompa appartenente a tale modulo: trasporto dell'attrezzatura, valutazione, posizionamento della stessa nell'area operativa e messa in opera.

## 1.3 La Realtà Virtuale

Ponendosi l'obiettivo di realizzare una simulazione di addestramento il più efficace possibile nell'insegnamento ai volontari delle corrette procedure all'utilizzo del Modulo HCP, si è reso necessario porre una particolare attenzione sul realismo dello scenario di rischio, il complesso schema delle procedure da rispettare, l'utilizzo di un gran numero di strumenti e l'interazione con altri operatori appartenenti alla squadra.

Per poter implementare al meglio delle possibilità tecnologiche questi punti chiave, dispositivi come monitor, tastiere e mouse sono stati trovati limitanti nel realismo soprattutto per quanto riguarda l'interazione con l'ambiente e la strumentazione.

La simulazione esige quindi una tecnologia in grado di permettere il movimento e l'esplorazione dell'ambiente, un'interazione il più possibile naturale con altri membri del team e in generale un grado di immedesimazione e libertà tale da portare l'utente a credere di trovarsi in una reale situazione di rischio e a comprendere le complicate operazioni proprie di un modulo come l'HCP.

### 1.3.1 Definizione di Realtà Virtuale

Con il termine "*Realtà Virtuale*" Gregor Burdea definisce un'interazione uomo-macchina tramite simulazione rappresentante un ambiente realistico fruibile attraverso particolari interfacce. La simulazione deve essere in tempo reale, permettere interazioni con l'ambiente e multisensoriale [10].

Lo scopo è quello di "ingannare" l'utente fino al punto di fargli credere di trovarsi in un ambiente realistico che segue le leggi della fisica, logiche e sociali proprie del mondo reale in cui è possibile il movimento e l'interazione.

Esiste infatti una grande contrapposizione nei termini "reale" e "virtuale": con il primo si intende tutto ciò che esiste ed è vero, mentre il termine virtuale descrive ambienti immaginari creati con la tecnologia [11].

Gli stimoli sensoriali sono simulati artificialmente in modo che il cervello umano non percepisca la differenza da quelli naturali; in questo modo stimoli come quelli visivi o uditivi vanno a concorrere al mantenimento della finzione e del grado d'immersione dell'utente [12].

Inoltre è anche molto importante considerare la sfera emotiva dell'utente: infatti gli stimoli e le interazioni devono essere in grado di suscitare reazioni e sentimenti naturali, sempre con lo scopo di mantenere un elevato livello di coinvolgimento.

Sono molti e vari i campi di utilizzo della VR: addestramento, educazione, medico, militare, industriale e psicologico (*Virtual Reality Exposure Therapy* o VRET). Nel prossimo futuro, dato l'attuale interesse tecnologico e dei nuovi campi d'applicazione, ci si aspetta che la tecnologia VR diventi sempre più immersiva fino a coinvolgere tutti i sensi umani, seppur attualmente si tratti solo di un obiettivo teorico.

### 1.3.2 Immersione e presenza

Per poter mantenere livelli di immedesimazione e realismo accettabili, così da impedire la *dissociazione* dell'utente dall'esperienza virtuale, è necessario definire fattori in grado di valutare questi parametri.

Per questo motivo nel campo della VR, sono stati definiti due concetti in grado di valutare fattori quali gli stimoli, le interazioni e l'ambiente dal punto di vista del realismo e della naturalezza delle sensazioni suscitate nell'utente: *immersione* e *presenza*.

- Con *immersione* si va ad indicare la percezione fisica, attraverso gli stimoli sensoriali, di trovarsi effettivamente all'interno di un mondo reale. Il livello di immersione dipende quindi dagli stimoli visivi, uditivi, aptici ed inerziali. Inoltre aumenta con l'aumentare della qualità della simulazione (risoluzione, grafica, tempo di latenza, realistica delle interazioni). L'immersione è quindi una misura oggettiva in grado di indicare quanto gli stimoli percepiti durante una simulazione VR siano realistici per i vari sensi [13].
- Con *presenza* si intende, invece, la percezione mentale di trovarsi in un'esperienza realistica; si parla quindi di quanto l'esperienza sia capace di "ingannare la mente" dell'utente. Mentre l'immersione è una grandezza oggettiva legata alla simulazione, la presenza è per lo più una misura soggettiva, legata più all'utente che alla simulazione in cui è immerso. Il termine si lega inoltre al concetto di "*sospensione dell'incredulità*", indicando un utente che decide di sospendere il proprio giudizio critico in modo da non essere condizionato dalle differenze tra simulazione e realtà [14].

Data la natura ancora relativamente giovane della tecnologia VR, i due concetti vengono spesso reinterpretati o ridefiniti; Mel Slater recentemente fornito una definizione alternativa per i due concetti, utilizzando *lace illusion* al posto di immersione (ridefinendo così la percezione fisica) e *plausibility illusion* invece di presenza (ad indicare la percezione mentale) [15].

## 1.4 Obiettivi del progetto e della tesi

Come detto, il progetto di tesi ha lo scopo di sviluppare un'applicazione VR con il fine di supportare l'addestramento di volontari della PC nell'utilizzo del modulo europeo HCP e delle relative procedure standard, in uno scenario di rischio idrogeologico.

L'utente opererà in uno scenario fittizio, in solitaria o aiutato solamente da *Personaggi non Giocanti* o *Non-Player Character* (NPC) e opererà nell'utilizzo di motopompe autoadescanti nelle fasi di posizionamento, messa in sicurezza, montaggio manichette e accensione.

In particolare, il presente lavoro di tesi andrà a presentare le fasi di progettazione e realizzazione del sistema di interazione (strumenti, utente, NPC e reciproca), della logica d'automazione degli NPC e della strategia di valutazione.

Lo sviluppo del sistema di interazione si è svolto a partire dalla modellazione degli oggetti interagibili (ad esclusione delle manichette) fino all'integrazione del sistema VR.

Per quanto riguarda l'automazione degli NPC, si è andati a lavorare con la *cinematica inversa* o *Inverse Kinematics* (IK) e l'animazione sugli stessi, per integrarli in una *Macchina a Stati Finiti* o *Finite State Machine* (FSM) con la quale viene gestita la logica del controllo delle automazioni.

Infine la valutazione, elemento essenziale per una corretta verifica dell'operato dell'utente, si basa sui criteri raccolti durante gli incontri con la PC che riportano le criticità e gli errori più comuni commessi durante l'utilizzo del Modulo HCP.

## 1.5 Organizzazione delle attività

In primo luogo si è andati a definire lo scopo e i contenuti del progetto attraverso raccolta di materiale fornito dalla PC e dalle riunioni svoltesi in presenza del professor Fabrizio Lamberti (responsabile del progetto per il Dipartimento di Automatica e Informatica del Politecnico di Torino) e dai rappresentanti della PC Regione Piemonte, presso il Dipartimento Energia (DENERG) del Politecnico di Torino e la sede della PC di Torino.

Le riunioni sono state utili a definire lo scopo finale dell'applicazione, l'ambito d'applicazione (rischio idrogeologico) e alla collezione di materiale testuale e audiovisivo dagli archivi della PC. Inoltre, in un raduno di volontari organizzato durante la settimana nazionale della PC (13-19 ottobre 2019) presso San Mauro Torinese (TO), è stata possibile la raccolta di altro materiale sul campo.

Definito l'argomento didattico della simulazione (Modulo HCP), è stato possibile effettuare sopralluoghi presso le sedi della PC a Druento e Vercelli, organizzati con lo scopo di visualizzare l'attrezzatura equipaggiata sul modulo e una prima visione del funzionamento e delle fasi di messa in opera e utilizzo.

Contemporaneamente alla raccolta e analisi dei dati, si è proceduto ad una prima stesura di un documento di design, alla creazione di uno storyboard, alla modellazione 3D, allo studio della struttura logica della simulazione ed alla scelta degli asset più adeguati.

Successivamente è stato possibile organizzare un'esercitazione incentrata sul Modulo HCP presso il Nodo Idraulico di Ivrea. L'esperienza aveva lo scopo di visualizzare le operazioni di messa in sicurezza, utilizzo e smontaggio del modulo stesso in modo da raccogliere esperienza di prima mano da cui trarre informazioni.

A questo punto è stato possibile iniziare la realizzazione dello scenario prototipale con le interazioni fondamentali (manichette, NPC, motopompa), mantenendo sempre una costante comunicazione con i responsabili della PC per feedback e aggiornamenti.

## 1.6 Struttura del documento

Il documento di tesi è suddiviso nei seguenti capitoli.

- Capitolo 1: Introduzione

Include i casi studio su cui si basa il progetto di tesi (il progetto PITEM RISK, l'organizzazione della PC, il rischio idrogeologico e il modulo europeo HCP), breve excursus sulla VR e i suoi punti chiave, le attività svolte e gli obiettivi prefissati per il lavoro di tesi.

- Capitolo 2: Stato dell'arte

Approfondimento della VR concentrato su storia, campi di applicazione e addestramento. Include inoltre un'analisi delle applicazioni già sviluppate in ambito di gestione delle emergenze.

- Capitolo 3: Tecnologie

Descrizione tecnica delle tecnologie, hardware e software, utilizzate nello sviluppo del progetto di tesi.

- Capitolo 4: Progettazione

Resoconto delle fasi di progettazione del lavoro di tesi comprendente l'analisi del materiale fornito e la stesura del documento di design, del testo per il voice-over, della struttura e dei parametri di valutazione.

- Capitolo 5: Realizzazione

Analisi delle fasi di sviluppo della simulazione approfondendo la scelta degli elementi utilizzati e la logica della simulazione.

- Capitolo 6: Conclusioni e sviluppi futuri

Si riportano le conclusioni al lavoro svolto, gli obiettivi raggiunti, i limiti riscontrati ed i miglioramenti apportabili in futuro.

# Capitolo 2

## Stato dell'arte

### 2.1 Realtà Virtuale

Il concetto di VR comprende tutte le metodologie con cui è possibile simulare una situazione realistica tramite l'interazione uomo-macchina, con il fine ultimo di portare l'utente in una realtà alternativa ingannando il più possibile tutti i suoi sensi.

Teoricamente l'immersione potrebbe essere completa ma nella pratica, con le tecnologie fino ad ora implementate, vengono considerate VR solo applicazioni fruibili attraverso computer (come i videogiochi) o interfacce apposite (come visori, tute o guanti).

Nella VR non è importante che la simulazione sia fedele alla realtà, ma piuttosto che essa possa essere abbastanza realistica e interattiva da permettere all'utente di immergersi e dimenticare di non trovarsi nel mondo reale.

#### 2.1.1 Storia

La VR si è sviluppata molto soprattutto nell'ultimo decennio, ma non è una vera novità nel campo delle tecnologie.

Già nel 1961 Morton Heilig, videomaker statunitense, inventò un dispositivo chiamato *Sensorama* (Figura 2.1) [16]. Questo permetteva di riprodurre un film che non coinvolgeva solo vista e udito ma anche tatto e olfatto, in un'esperienza che chiamò *Experience Theater*.

Il dispositivo comprendeva un monitor a colori stereoscopico, un emettitore di odori, una ventola per la simulazione del vento, un sistema audio stereo e un seggiolino mobile.

Nella prima simulazione testata da Heilig, l'utente aveva la possibilità di sperimentare un viaggio in moto per le strade di Brooklyn.

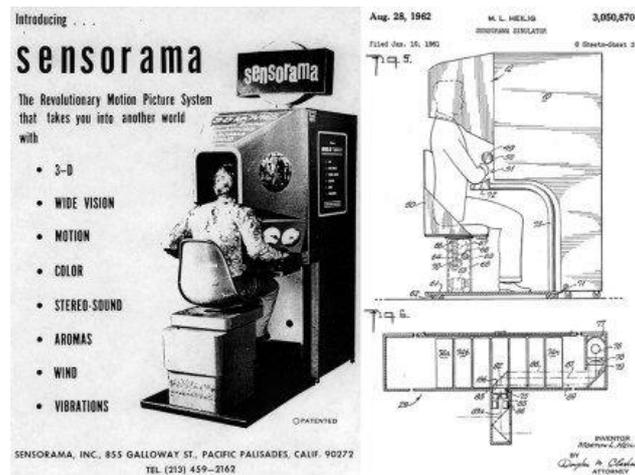


Figura 2.1: Sensorama, Morton Heilig, 1961

È solo nel 1968, però, che venne inventato quello che ora viene considerato il primo visore di tipo *Head-Mounted Display* (HMD). Il dispositivo, progettato al MIT da Ivan Sutherland e dal suo studente Bob Sproul, era così pesante che doveva essere appeso al soffitto. Questa caratteristica ispirò il suo nome, *La Spada di Damocle* (Figura 2.2) [17].

Due tubi catodici, uno per occhio, permettevano di proiettare immagini 3D sovrapponendole alla visione reale, rendendolo quindi per lo più un visore di *Realtà Aumentata*, o *Augmented Reality* (AR).



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

Negli anni '70 si fecero dei piccoli passi avanti con l'invenzione dei primi mondi virtuali navigabili e dei primi HMD di tipo *LEEP (Large Expanse, Extra Perspective)* che permettevano di coprire maggiormente il campo visivo.

Ma è solo nel 1982 che il termine venne coniato da Jaron Lanier, uno dei suoi più famosi pionieri. Lo scienziato 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 dava lavoro ad altri pionieri della VR come Thomas Zimmerman, inventore dei *DataGlove* che, utilizzando sensori a fibre ottiche, riuscivano a rilevare ben 256 posizioni delle dita.

A seguito della chiusura del laboratorio, a causa della crisi di mercato dei videogiochi, Lanier e i suoi colleghi fondarono il *VPL Research*, produttore di *DataGlove*, *EyePhone* e *DataSuit*.

Il *DataGlove* venne venduto nel 1989 alla Nintendo diventando *PowerGlove* (Figura 2.3) [18], il primo dispositivo per la VR disponibile al pubblico.

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



Figura 2.3: PowerGlove, Nintendo, 1989

Negli anni '90, a seguito dei precedenti importanti traguardi, ci si concentrò sul rendere disponibili al consumatore (quindi a prezzi abbordabili) le tecnologie fino ad allora inventate, soprattutto per quanto riguarda il campo dell'intrattenimento.

La SEGA nel 1993 annunciò il suo rivoluzionario HMD, chiamato *SEGA VR* (Figura 2.4) [19] con quattro giochi abbinati. Il visore presentava schermi LCD, cuffie stereo ed era capace di tracciare i movimenti della testa dell'utente. Il visore non vide mai la luce a causa dei forti mal di testa ed altri malesseri che procurava.



Figura 2.4: SEGA VR, SEGA, 1993

La Nintendo nel 1995 tentò di rifarsi del flop dei *DataGlove* sviluppando il *Virtual Boy* (Figura 2.5) [20]. La console si presentava come una sorta di elmetto da appoggiare su un tavolo collegato ad un controller. Il dispositivo presentava grosse limitazioni: un effetto 3D non particolarmente impressionante, un display monocromatico, l'assenza di portabilità, un prezzo elevato e la comparsa di senso di nausea e mal di testa. Divenne il secondo flop nel campo della VR dell'azienda.



Figura 2.5: Virtual Boy, Nintendo, 1995

Uscirono sul mercato anche visori pensati per l'utilizzo con sistema operativo Microsoft Windows come *Forte VFX1*, *Philips Scuba VR* e *VFX3D*.

Dopo questi primi decenni di grandi aspettative per la nuova, promettente tecnologia, gli anni 2000 furono caratterizzati da un parziale calo dell'interesse a causa delle difficoltà ritenute insormontabili con le tecnologie dell'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. Si parlerà maggiormente delle attuali tecnologie nel Capitolo 3.

L'obiettivo ultimo della VR, di immersione e presenza completa, non è però ancora stato del tutto raggiunto. Si tratta quindi di una tecnologia che presenta ancora molti margini di sviluppo per gli anni a venire.

### 2.1.2 Campi d'utilizzo

La VR è comunemente applicata al campo dell'intrattenimento, in quanto si tratta di uno dei pochi settori nei quali è possibile renderla disponibile alle masse.

I video 3D vengono comunemente utilizzati nei 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 invece la vendita di piattaforme VR per uso personale.

Nel campo delle belle arti la VR presenta molti campi d'utilizzo. È possibile ricreare la visita ad un museo in modo da poterne usufruire senza muoversi da casa, per esempio per le persone inabili al movimento. Si può utilizzare la VR anche per ricreare esperienze impossibili nel mondo reale, 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 punto di vista irraggiungibile (ad esempio la visita in volo della cupola della *Cappella della Sacra Sindone*, Torino [22]) o per ricreare i siti ormai perduti (ad esempio *Ullastret* città iberica di 2200 anni fa [23]).

Nella psicologia, la VR viene comunemente utilizzato per ricreare situazioni particolari in un ambiente controllato. 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. Per esempio nella cura delle fobie è possibile ricreare scene in cui il paziente deve affrontare le proprie paure (terapia dell'esposizione) [24].

La VR risulta molto utile anche nell'aiutare certi gruppi sociali, come gli anziani o i soggetti diversamente abili, rendendo possibile la simulazione di esperienze per loro impossibili da vivere nel mondo reale. Gli studi dimostrano infatti che è molto efficace nel combattere la depressione e la solitudine portate da certe inabilità.

Con risultati ancora scarsi si è anche tentato di applicare la VR alla riabilitazione fisica, risultando meno efficace dei metodi tradizionali ma presentando margini di miglioramento. Si continua invece ad utilizzarla 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 [25].

Nell'ingegneria e nell'architettura, la VR si è rivelata utile nelle fasi di progettazione. È possibile portare progetti di edifici ed altre strutture dalla carta alla VR, in modo da potervi interagire prima della loro realizzazione, oppure ricreare eventi particolari, come ad esempio terremoti, per osservarne il comportamento.

Ultimamente, si sta tentando di integrare sistemi VR per il commercio elettronico, ma è un campo che richiederà ancora molti anni prima di risultare efficace dovendo anche combattere la reticenza degli utenti finali.

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 sintetico, 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, ecc. Si parlerà più nel dettaglio di questi ambiti nella Sezione 2.2.

Nonostante i vari campi d'utilizzo la VR è ancora relativamente impiegata, anche se si sta riscontrando un aumento significativo dell'interesse che promette una costante crescita nei prossimi anni.

### 2.1.3 Serious Game

Il termine nasce nel 1970 riferito a qualsiasi tipologia di giochi che utilizzi il divertimento, non per intrattenere, ma per educare, con applicazioni nel campo della salute, della comunicazione strategica, dell'istruzione, ecc.

Non esiste una singola definizione di *Serious Game* (SG) e, benché in genere ormai si colleghi il termine ai giochi digitali, li si può ricollegare a giochi di qualsiasi genere, utilizzando qualsiasi tecnologia e per qualsiasi piattaforma.

Il primo SG sviluppato è considerato *Battlezone* [26] del 1980 e sviluppato da *Atari*. Il gioco era pensato per l'addestramento militare, ma fu un fallimento.

È solo negli anni '90 che gli studiosi iniziarono a pensare seriamente ai diversi utilizzi dei videogiochi oltre l'intrattenimento.

Oggi 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.

I vantaggi dei SG sono molteplici.

In questi giochi è possibile ricreare qualsiasi elemento del mondo reale: strumentazione, veicoli, macchinari, edifici tutto ad una spesa irrisoria in confronto agli stessi oggetti reali. In questo modo è possibile renderli accessibili all'utente senza che questi debba realmente mettere le mani su quegli stessi costosissimi strumenti. Inoltre è anche possibile ricreare qualsiasi scenario reale o immaginario, come battaglie o terremoti, senza essere realmente presenti in un evento di questo genere.

La costruzione e l'uso di simulatori tradizionali ha infatti un costo molto alto in confronto allo sviluppo di un videogioco.

Numerosi studi scientifici hanno inoltre dimostrato l'utilità dei SG nell'apprendimento, osservando come gli studenti che utilizzano SG memorizzano i concetti più velocemente e facilmente di quelli che studiano con metodo tradizionale [27]. Questo è dovuto per lo più al fatto che i SG sono in grado di promuovere l'interesse, l'impegno e il pensiero di ordine superiore degli studenti, in aggiunta all'assenza della "paura del fallimento" e delle conseguenze che ne derivano.

#### 2.1.4 Realtà Virtuale multi-utente

Più l'esperienza VR risulta coinvolgente e immersiva, più aumenta il senso di isolamento dalla realtà e la solitudine, in quanto si condivide la simulazione solo con l'intelligenza artificiale.

Le applicazioni multi-utente possono essere utili non solo per combattere questo aspetto psicologico, condividendo le esperienze con amici, familiari e colleghi, ma anche in quanto interessanti esperimenti sociali.

Molti studi sono stati effettuati per questo ambito (con lo sviluppo di applicazioni come *VRChat* [28]) con risultati che evidenziano le potenzialità sociali della tecnologia [29], tanto da far interessare Facebook a soluzioni di *social VR*.

Per quanto riguarda le applicazioni VR per l'addestramento e l'educazione, il multi-utente può portare un grande vantaggio, permettendo di condividere l'esperienza educativa con altre persone. È possibile perciò andare a creare esperienze impossibili da svolgere in solitaria e promuovere il lavoro di gruppo.

Un'applicazione multiutente di successo deve permettere agli utenti di ricreare interazioni sociali reali; gli utenti devono poter mantenere il proprio spazio personale, interagire l'uno con l'altro e osservarsi fisicamente.

Nonostante ciò, la maggior parte delle esperienze di VR ora in commercio sono in genere per l'utente singolo.

## 2.2 La Realtà Virtuale per l'addestramento

Come già descritto nella Sezione 2.1.3, i SG stanno sempre di più sviluppandosi come nuovi metodi d'apprendimento, sfruttando la capacità immersiva della tecnologia VR per comunicare in modo più diretto ed efficace le informazioni. Sono di particolare interesse per enti governativi, istituzioni accademiche e privati legati al settore tecnico che cercano metodi alternativi alla formazione dei propri tecnici specializzati.

Un addestramento si può definire tanto più efficace quanto maggiore è la qualità dell'insegnamento, la capacità di verifica delle competenze acquisite e la copertura di situazioni particolari ed imprevisti che possono insorgere durante l'esecuzione dei compiti assegnati. Un sistema educativo di qualità ha perciò bisogno di grandi investimenti di denaro utilizzati per sviluppare metodi personalizzati, come lezioni frontali od online, esercitazioni dal vivo o strumenti ad hoc. Gli svantaggi sono però numerosi: le lezioni dal vivo od online non favoriscono l'approccio psicofisico necessario ad un tipo di addestramento pratico e le

esercitazioni, organizzate per ovviare al primo svantaggio, possono rivelarsi molto costose, sia in termini di tempo che di denaro, per risultare quindi insufficienti. Infatti, sebbene forniscano esempi validi, in alcuni casi possono non essere abbastanza d’impatto per trasmettere la “lezione” in modo efficace; è questo il caso di situazioni ad alto rischio come ad esempio operazioni chirurgiche (in cui un errore può portare alla morte del paziente) (Figura 2.6) come pure la simulazione di eventi naturali (pericolosi per gli operatori o impossibili da simulare).



*Figura 2.6: Dettaglio di una simulazione VR in ambito medico-chirurgico*

Tra i due casi si può così andare ad inserire la VR che, sebbene ancora molto sotto-impiegata tranne che per rare eccezioni (come nel campo medico o aerospaziale), è stata capace di rivelare la propria utilità in molteplici settori sia pubblici che privati. Di seguito si andranno ad approfondire i relativi punti di forza e criticità.

### 2.2.1 Vantaggi e motivazioni dell’approccio

Come già delineato, il principale motivo per sviluppare esercitazioni in VR è proprio quella di evitare le esercitazioni dal vivo. Oltre alla quantità di soldi e tempo investiti per l’organizzazione di questa tipologia di eventi, in un’esperienza reale si ha il problema della potenziale pericolosità dell’interazione con le attrezzature reali e dall’impatto delle variabili

ambientali. Inoltre, un errore da parte del personale in addestramento ancora inesperto, potrebbe causare guasti anche di rilevante importanza su macchinari costosi o disservizi su infrastrutture pubbliche e private, causando quindi ulteriori costi.

Per questo motivo in genere si tende a fornire preparazione, con conseguente verifica delle abilità, su un sottoinsieme ristretto delle operazioni e dei macchinari che, rimanendo pur sempre una formazione standard, potrebbe causare momenti d'indecisione in situazioni reali con relative conseguenze [30].

Tramite la VR è possibile invece offrire formazioni complete e sicure, costituite da simulazioni comprendenti le diverse casistiche e l'utilizzo completo delle apparecchiature in un'ambiente realistico.

Per quanto riguarda l'addestramento degli operatori adibiti alla gestione di eventi catastrofici, tutto ciò è doppiamente importante, ed a quanto detto si aggiunge inoltre il fattore dello stress psicologico che caratterizza una situazione d'emergenza. Quindi, è molto importante effettuare addestramenti il più possibile simili ad una reale situazione di rischio, poiché la familiarità con uno scenario può fare la differenza nell'operato di ogni addetto. Spesso, però, non è possibile replicare fedelmente un evento, non solo per la componente ambientale, ma anche per la sicurezza degli operatori che, essendo ancora inesperti, non possono neanche assistere ad un reale evento naturale senza il pericolo d'intralciare le squadre operative [31].

Anche in questo campo torna utile l'applicazione della VR nell'addestramento, poiché capace non solo di replicare fedelmente il reale evento catastrofico ma anche di introdurre elementi che possono rendere più realistica la simulazione, come impedimenti visivi, uditivi, folle di civili, ecc., il tutto senza costi eccessivi, garantendo la sicurezza del personale ed il miglioramento della risposta in situazioni stressanti.

Grazie alla VR è inoltre possibile la misurazione dello stato emotivo e dello stress percepito dall'operatore nelle diverse fasi della simulazione, la registrazione dei risultati e delle prestazioni, così da poter valutare l'operato in un secondo momento (oltre che analizzare i punti di forza e debolezze della simulazione, in modo da poter apportare eventuali miglioramenti) [32].

Tramite numerosi esperimenti, in cui si comparano le esperienze educative tradizionali con quelle VR, si è riscontrata una forte somiglianza sia per quanto riguarda l'apprendimento di nozioni [33] sia per quanto riguarda le esperienze sul campo [34], nelle quali è persino possibile fornire un feedback diretto e immediato alle azioni degli utenti.

Un'altra delle opportunità offerta da una simulazione in VR è la possibilità di rendere l'esperienza scalabile, cioè il poter variare le condizioni dell'addestramento, come ad esempio la difficoltà, in base all'esperienza del personale in addestramento e dalle esigenze di chi sta richiedendo la formazione.

Al costo di un singolo sistema VR è quindi possibile effettuare infinite esercitazioni senza dover affrontare spese e la difficoltà dell'organizzare esercitazioni su media e larga scala; inoltre la facilità d'utilizzo e i rapidi miglioramenti della tecnologia permettono di adottarla sempre più facilmente e raggiungere livelli di qualità tali da poterla persino utilizzare per le esercitazioni degli astronauti, che necessitano di alti livelli di dettaglio e fedeltà (Figura 2.7) [35].

Un'altra possibilità è quella di organizzare esercitazioni anche a distanza, rompendo così le barriere geografiche.



Figura 2.7: Dettaglio di una simulazione VR in ambito aeronautico

Quelli illustrati sono solo alcuni dei vantaggi della VR applicata all'addestramento, e come affiancamento alle lezioni frontali ed alle esercitazioni pratiche svolte con modalità tradizionali.

## 2.2.2 Svantaggi dell'approccio

La tecnologia VR, tuttavia, non ha solo caratteristiche positive ma, come ogni cosa, svantaggi intrinseci.

Per quanto lo sviluppo e l'applicazione della tecnologia sembri ultimamente molto diffusa, in certi settori è ancora molto sotto utilizzata. Inoltre, le persone inesperte che si avvicinano alla tecnologia per la prima volta, hanno spesso bisogno di un allenamento preventivo

nell'utilizzo efficace della VR. In assenza di questo, l'addestramento potrebbe risultare inefficace.

Un altro elemento svantaggioso è la mancanza di un sistema di copertura completa dei sensi; con un'attrezzatura classica composta da un visore e due controller, sono trasmettibili solamente feedback aptici (tramite vibrazione dei controller), visivi e sonori. Mancano soprattutto i feedback naturali derivanti dall'interazione con gli oggetti e l'ambiente, a meno che non si stiano utilizzando delle tute aptiche in grado di stimolare ogni parte del corpo [36].

Uno dei maggiori svantaggi della VR (che è anche tra i più comuni sperimentabili), è il malessere fisico provocato dal suo uso prolungato: nausea da cinetosi, affaticamento visivo, mal di testa, vomito, pallore, sudorazione, stanchezza, vertigini, sonnolenza, disorientamento e apatia.

Infine, a seconda della tipologia di simulazione, dell'attrezzatura utilizzata e della capacità di calcolo dell'hardware utilizzato, le caratteristiche della simulazione potrebbero risultare limitate nell'accuratezza, dando così origine ad un'applicazione efficiente ma non abbastanza efficace.

## 2.3 Realtà Virtuale e gestione delle emergenze

Nell'ambito del programma *INTERREG ALCOTRA* tra Italia e Francia, nasce il progetto *PRODIGE* (PROteggere i cittadini, Difendere le Infrastrutture, Gestire i Grandi Eventi), elaborato allo scopo di migliorare la collaborazione tra le autorità addette alla gestione delle emergenze italiane e francesi nei protocolli congiunti messi in opera in caso di rischi naturali e artificiali tramite metodi di addestramento alternativi basati sulla tecnologia VR [37].

Ispirato dai risultati del progetto *PRODIGE* nasce il progetto *PITEM RISK*, sviluppato nel contesto della prevenzione dei disastri ambientali (come frane, esondazioni, ecc.) sempre più frequenti e intensi a causa dei cambiamenti ambientali (come previsto dalla *Agenzia Europea dell'Ambiente*) [38]. Per migliorare la risposta ai rischi nelle zone transfrontaliere, spesso le più vulnerabili a causa della cattiva cooperazione tra le autorità delle diverse nazioni, sono stati sviluppati strumenti educativi innovativi basati su tecnologia VR nell'ambito dell'integrazione di nuove ed innovative tecnologie al sistema della PC.

A questo proposito, il progetto si propone di migliorare il sistema educativo e d'addestramento degli operatori e introdurre nuove strategie di comunicazione in tempo reale con la popolazione, non solo per comunicare informazioni ma anche per ricevere dati utili. Queste strategie sono mirate sia agli "addetti ai lavori" sia ai civili, così da aumentarne la consapevolezza.

Essendo quindi i due progetti strettamente collegati, ci si è molto basato sulle precedenti simulazioni sviluppate all'interno di PRODIGE per lo sviluppo del progetto di tesi, essendo queste un buon esempio di simulazioni in VR finalizzate alla gestione delle emergenze.

Il documento di presentazione del programma *PRODIGE* è stato utilizzato per trarre i seguenti estratti.

### 2.3.1 Rischio esondazione nel parco fluviale Gesso e Stura a Cuneo

Lo scenario simula una possibile inondazione del fiume Gesso su un'area del Parco. Nella simulazione viene descritto un costante peggioramento della situazione attraverso l'utilizzo di livelli d'allerta che, inizialmente a livello 2 (preallarme e criticità moderata), passa a livello 3 (allarme e criticità elevata) e all'allertamento della PC. L'operatore arriva sul posto e procede alla chiusura degli accessi e all'evacuazione delle persone dall'aerea. In seguito, sempre mantenendosi in contatto con il Centro Operativo Comunale, si posiziona presso gli ingressi in modo da controllare gli accessi.

La particolarità della simulazione consiste nel sistema di navigazione implementato attraverso una pedana omnidirezionale con corsa sul posto (Figura 2.8) e nel sistema automatico di valutazione degli errori.



*Figura 2.8: Scenario Parco Gesso e Stura di PRODIGE con pedana Virtualizer*

### 2.3.2 Incidente stradale presso St.Paul-sur-Ubaye

La simulazione descrive un incidente stradale causato dall'improvvisa esondazione del fiume Riou Sec, passante al di sotto di un ponte, presso St Paul-sur-Ubaye (Figura 2.9). Il mezzo, intrappolato a metà del ponte a causa dell'acqua che lo spinge verso la balaustra, viene inondato dall'acqua con ancora il conducente a bordo ed impossibilitato ad aprire le portiere a causa della forte pressione. Allertati i soccorsi, il veicolo viene prontamente trainato via dal ponte da parte di operatori dotati di imbracatura e corda di sicurezza e il conducente portato via dall'elisoccorso della PC.



*Figura 2.9: Dettaglio dello scenario St. Paul-sur-Ubaye di PRODIGE*

### 2.3.3 Dispersione di sostanze chimiche sul Colle della Maddalena

Lo scenario descrive un possibile incidente con dispersione di sostanze chimiche trasportate da un veicolo pesante sul territorio francese vicino al colle della Maddalena (Figura 2.10). Il camion, finendo fuori strada in prossimità di un ponte nel paese di Larche, finisce nel fiume rovesciandosi; l'incidente causa una perdita di sostanze chimiche inquinanti dalla cisterna che finiscono per riversarsi nel fiume. Il conducente rimane intrappolato e privo di sensi all'interno della cabina finché non viene salvato dal corpo dei Vigili del Fuoco. Per limitare l'area inquinata, ormai allargatasi dalla continua fuoriuscita delle sostanze chimiche,

vengono attivate le procedure per la delimitazione dell'area e del contenimento della perdita.

Sia quest'ultima simulazione che quella precedente, presentano uno scenario georeferenziato, una modalità multiutente, la guida di veicoli (sia terra che aria), pacchetti di oggetti 3D legati all'equipaggiamento della PC ed un centro di comando fittizio utilizzato per la modifica dello scenario durante la simulazione.



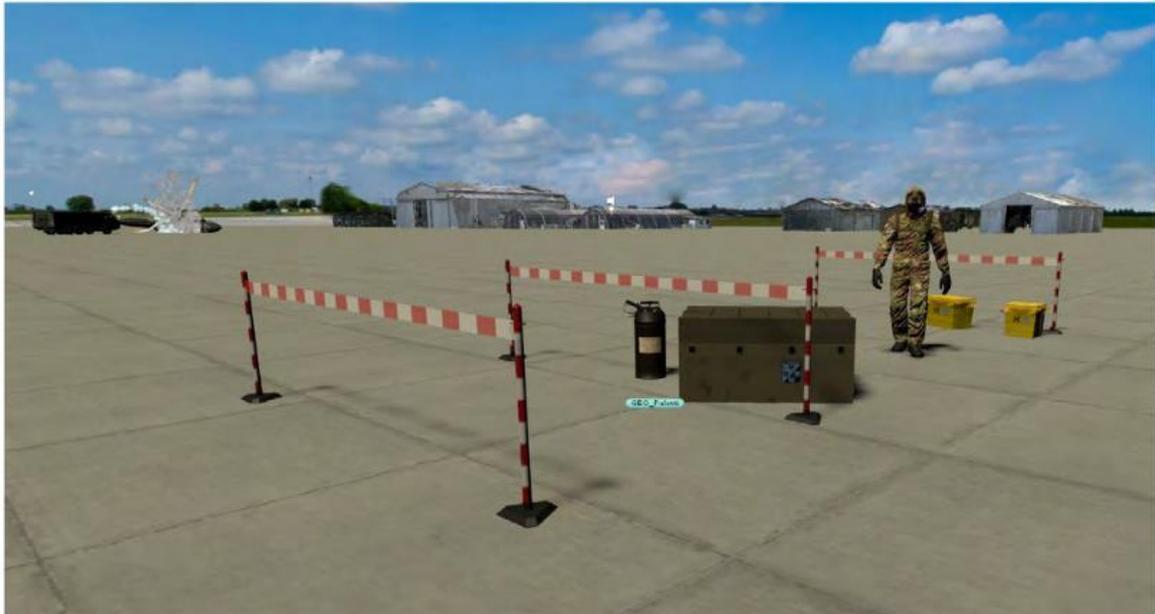
*Figura 2.10: Scenario dell'incidente sul Colle della Maddalena di PRODIGE*

#### 2.3.4 Nube tossica da sostanza radioattiva in avvicinamento a Torino

L'ultimo scenario prevede due scene sequenziali. Nella prima un elicottero che sta trasportando un carico di Torio 232 cade durante il decollo su una cisterna di carburante causando un incendio. Gli operatori *CBRN* (*Chemical Biological Radiological Nuclear*) devono procedere al contenimento dell'incendio ed alla delimitazione dell'area dell'incidente dove ora si sprigiona una nube radioattiva (Figura 2.11). Nella seconda scena la nube raggiunge una zona del parco Dora, nella città di Torino, nel quale si sta svolgendo una manifestazione musicale che ha raccolto una numerosa folla di persone. Gli agenti della Polizia Municipale della Città di Torino già sul posto, vengono allertati dalla Centrale Operativa dell'imminente minaccia.

Grazie alla collaborazione con L'Aeronautica Militare per la progettazione del primo scenario, è stato possibile studiare le operazioni attuate dall'esercito in caso di crisi e creare una simulazione nella quale PC, Pronto Soccorso e anti-incendio sono a capo alla stessa entità, il

comandante dell'aeroporto, operando così in un contesto simile a quello francese e favorendo così la creazione di sinergie tra i due paesi.



*Figura 2.11: Dettaglio dello scenario relativo alla propagazione della nube tossica di PRODIGE*

Le caratteristiche principali che rendono la simulazione efficace sono la grafica immersiva, la possibilità di scegliere il ruolo da interpretare, la simulazione verosimile dell'incendio e l'integrazione di procedure di comunicazione con i civili.

# Capitolo3

## Tecnologie

### 3.1 Hardware

Con l'aumentare negli ultimi anni dell'interesse, da parte delle aziende, nel settore della VR, si è riscontrato un aumento nel numero di tecnologie offerte da questo mercato [30]; questo ha conseguentemente portato ad una categorizzazione dei dispositivi: tipologia di visore e sistema di tracking, piattaforme di destinazione, complessità del sistema implementato e strumenti di input sensoriale.

Per usufruire di un'esperienza VR immersiva, è strettamente necessario avere a disposizione un visore indossabile sulla testa (HMD), dispositivi di input per l'interazione con lo scenario (possono essere indossabili come guanti *VR Gloves*, controller trattenuti nelle mani oppure sensori applicati a parti del corpo o oggetti) e sensori in grado di tracciare i movimenti dell'utente in tempo reale (*motion tracking*) denominati *base station*. Solo con il tracciamento in tempo reale, l'utente può sentirsi immerso nell'ambiente, usufruendo di una visione a 360° dello scenario e percependo i feedback prodotti dalla sua interazione con esso.

In quanto il senso più sviluppato nell'essere umano è la vista e quindi quello più difficile da ingannare, gli strumenti più importanti in una tecnologia VR risultano essere i visori che possono essere classificati in base alla presenza o meno di capacità di calcolo autonoma. Gli HMD più semplici, sia dal punto di vista hardware che di potenza di calcolo, sono definiti *mobile HMD* che si appoggiano alle capacità dei dispositivi mobili a cui sono collegati (come gli smartphone) mentre quelli più complessi, i *wired HMD*, presentano display ad alta risoluzione, sensori e richiedono un processore di calcolo esterno (come un computer *VR-ready*).

Le interfacce più diffuse e utilizzate sono in generale provviste di *hand controller*. Questi sono in grado di permettere le interazioni con il mondo virtuale e tramettere i feedback aptici conseguenti ad essi. Sebbene siano presenti sul mercato anche altri dispositivi più elaborati (come i *VR Gloves*) che permettono di aumentare il realismo e la portata dei feedback, risultano eccessivamente costosi e poco accessibili.

In particolare, i dispositivi che sono stati maggiormente utilizzati nella realizzazione del progetto di tesi in questione, sono l'*HTC Vive* e l'*Oculus Rift*.

### 3.1.1 HTC Vive

L'*HTC Vive* [40], realizzato da *Valve* in collaborazione con *HTC*, è un dispositivo VR commercializzato nel 2016. La tecnologia è costituita da un visore HMD, due controller e due sensori *base station* (*Lighthouses*) utilizzati per tracciare i movimenti e la posizione dell'utente, per riprodurli nella simulazione virtuale, creando un reticolo ad infrarossi nel mondo reale chiamato *Room-Scale*. Le *base station* comunicano con i 70 sensori presenti sull'HMD e sui controller, permettendo il movimento in un'area 5 m x 5 m. I sensori, di tipo ottico, possono subire interferenze a causa di dispositivi elettronici o dalla presenza di una terza *base station* [41].



Figura 3.1: HMD del sistema VR HTC Vive

Il visore HMD (Figura 3.1) è cablato e si collega ad un computer tramite presa USB 3.0 e HDMI e alla corrente elettrica tramite presa. Inoltre anche le base station sono provviste di prese di corrente e della possibilità di collegarsi tra loro per migliorarne la sincronizzazione. I controller funzionano tramite semplici batterie interne.

Il visore è costituito da due schermi 1080 x 1200 AMOLED (*Active Matrix Organic Emitting Diode*), con lenti *Fresnel* (lenti di grandi dimensioni con piccola lunghezza focale in grado di regolare la distanza interpupillare) quasi circolari, frequenza di aggiornamento di 90 Hz e un campo visivo (*field of view*) di 110°. L'HMD è anche dotato di un accelerometro, un giroscopio, sensori a 360° e una telecamera in grado di calcolare l'area sicura (poi visualizzata nella simulazione tramite una griglia) in cui l'utente può muoversi nella realtà in un sistema di sicurezza chiamato *Chaperone*.

I controller (Figura 3.2) sono dotati di 24 sensori ciascuno e permettono l'interazione con l'ambiente virtuale. Sono inoltre dotati di 3 tasti e un touchpad cliccabile, 6 gradi di libertà (*Degree of Freedom* o DoF) e un sistema aptico chiamato *Force Feedback*. Nel mondo virtuale della simulazione, i controller possono essere visualizzati tramite modelli di mani virtuali o oggetti oppure con modelli che li riproducono fedelmente.



Figura 3.2: Controller del HTC Vive

Oltre ai sensori posti su visore e controller, la tecnologia *Vive* mette a disposizione i *Vive Tracker*: dispositivi di tracciamento per oggetti fisici o parti del corpo.

Le *base station*, fornite di emettitori laser con frequenza di 120 Hz e led ad infrarossi a 60 Hz, permettendo un tracking a 360°. È possibile così rilevare la posizione di ogni dispositivo *Vive* presente nell'area coperta dalle *base station*, basandosi sul numero di flash che colpiscono un sensore e su quando questi sono stati emessi.

*Valve* offre un SDK (*Software Development Kit*, pacchetto di strumenti e informazioni utili alla programmazione di software) a supporto di tutti i propri dispositivi chiamato *OpenVR* e utilizzato anche come API (*Application Programming Interface*) di *SteamVR* (software di supporto alla VR per la piattaforma *Steam*) per gestire la connessione dei dispositivi VR collegati. *Unity* dà la possibilità di includere *OpenVR* come SDK predefinito per lo sviluppo di applicazioni VR, fornendo asset (*SteamVR asset*) per facilitare il lavoro dello sviluppatore.

### 3.1.2 Oculus Rift

L'Oculus Rift (Figura 3.3) [42], distribuito da *Oculus VR* nel 2016 e acquistato da Facebook Inc. nel 2014, è un dispositivo VR tra i più diffusi sul mercato. La tecnologia consiste di un visore cablato, da collegare ad un PC tramite presa USB e HDMI, da due controller per l'interazione con l'ambiente e da due sensori per il *motion tracking* pensati per essere posizionati su una superficie (in genere la scrivania ai lati del computer) e collegati al PC tramite cavi USB [43].

Il dispositivo per quanto comodo da indossare e facile nell'utilizzo, comporta limitazioni dal punto di vista del tracciamento: i sensori, posizionati allo stesso livello, parallelamente e occlusi dal piano d'appoggio, spesso causano perdite di segnale a livello del terreno, in posizioni troppo alte o nel caso in cui i sensori indossati dall'utente vengono coperti (per esempio dando la schiena i sensori).



*Figura 3.3: HMD e controller dell'Oculus Rift*

L'HMD è composto da due schermi *OLED* (*Organic Light Emitting Diode*) da 1080 x 1200 pixel, frequenza di 90 Hz, accelerometro, giroscopio (similarmente all'*HTC Vive*) e un campo visivo di 110° verticale e orizzontale.

A differenza dell'*HTC Vive* il visore *Oculus* risulta più comodo da indossare, soprattutto per utilizzo prolungato, pesando solo 470 g contro i 555 g del primo.

I controller *Oculus* risultano molto simili a quelli *HTC Vive*, presentando tuttavia una forma più ergonomica chiamata *Hafli Moon* in quanto somigliante appunto ad una mezza luna. Per quanto riguarda il numero e tipologia di input, non si distacca molto da quelli tipici trovabili su altri controller VR (trigger, grip, levetta analogica corrispondete al touchpad del *Vive*) utilizzati comunemente per l'interazione con menù, navigazione e interazioni con oggetti della scena.

L'*Oculus*, infine, è caratterizzato da un sistema audio integrato per suono direzionale 3D; il visore è provvisto di cuffie smontabili e sostituibili (diversamente dalle auricolari disponibili con dispositivi *HTC Vive*).

### 3.1.3 Altre tecnologie sul mercato

Esclusi i dispositivi precedentemente descritti e i loro più recenti modelli, il mercato VR presenta altri dispositivi che, tuttavia, presentano limitazioni che li rendono poco fruibili alla realizzazione della simulazione.

Possiamo citare, ad esempio, *PlayStation VR*, prodotto per essere integrato con l'omonima console di giochi e i principali mobile HMD, che sfruttano i software smartphone (tra cui *Samsung GearVR* e *Google Cardboard*).

Inoltre, negli ultimi anni, sono stati distribuiti nuovi modelli dei dispositivi *HTC* e *Oculus*: *HTC Vive Pro*, *Oculus Rift* e *Oculus Rift S*. I tre nuovi dispositivi tuttavia presentano documentazioni limitate rispetto alle loro versioni precedenti che, essendo anche disponibili nel laboratorio VR@Polito del DAUIN del Politecnico di Torino, si è deciso di utilizzare per il progetto di tesi in esame.

## 3.2 Software

### 3.2.1 Game Engine

Ogni fase dello sviluppo di un'applicazione VR è caratterizzata dalla gestione, praticamente simultanea e in tempo reale, di un gran numero di elementi. Per questo motivo esistono librerie software con lo scopo di supportare lo sviluppatore nella gestione di questi elementi; dagli aspetti grafici (*Direct X* e *Open GL*), gestione dei contenuti audio (*DirectSound*), gestione della fisica (*Open Dynamics Engine*), ecc.

L'integrazione tra queste librerie, per andare poi a sviluppare un'unica applicazione, risulta però oneroso sia in termini di tempo che di denaro, in quanto spesso queste risultano anche molto diverse tra loro. Gli sviluppatori quindi tendono ad utilizzare software, detti Motore Grafico o *Game Engine*, che andranno a costituire il nucleo dell'applicazione e che, fornendo tutte le tecnologie di base, semplificano lo sviluppo rendendole compatibili anche a piattaforme diverse come pc, mobile, console, dispositivi VR e di AR.

In alternativa lo sviluppatore avrebbe dovuto integrare le librerie a mano in un'unica applicazione (di solito in C++) con conseguente diminuzione dell'ottimizzazione e l'aumento del rischio di errori.

Inoltre un Game Engine permette di ridurre i costi, i tempi, gli errori e mette a disposizione dello sviluppatore un'interfaccia su cui gestire al meglio i vari componenti dell'applicazione (quali contenuti visivi, audio, sistemi particellari, effetti visivi, avatar virtuali) limitando così le componenti di coding.

### 3.2.2 Unity

*Unity* [44] è un Game Engine sviluppato da *Unity Technologies* nel 2005 e il software scelto per lo sviluppo del progetto di tesi in oggetto. L'applicazione è multiplatforma (anche se inizialmente pensata solo per macOS) ed in grado di realizzare applicazioni in 2D, 2.5D, 3D, VR e AR. L'interfaccia, anche personalizzabile, facilita la gestione delle risorse con specifici pannelli di controllo e permette una facile integrazione di contenuti audio, video, modelli 3D e script C#.

Purché sviluppato per il campo videoludico, Unity è spesso anche utilizzato nell'ambito del cinema, dell'animazione, dell'architettura e del design di interni e industriale. Il software inoltre supporta numerose piattaforme come *Android*, *Windows*, *macOS*, *iOS*, *Xbox One*, *PlayStation 4*, *HTCVive*, *GearVR*, *Oculus Rift*, *Microsoft HoloLens*, *Apple TV* e *Samsung Smart TV* [45].

Unity presenta un'interfaccia (Figura 3.4) suddivisa in sezioni:

1. il pannello *Project*, che permette di gestire i file all'interno della cartella di progetto;
2. la *Hierarchy*, è l'elenco degli asset utilizzati nella scena;
3. l'*Inspector*, con il quale è possibile modificare gli attributi di ogni oggetto presente nella scena. Questi possono riguardare le componenti fisiche (posizione, rotazione e scala), le collisioni, le texture, i materiali, e l'illuminazione. Gli attributi possono anche essere custom, cioè creati dallo sviluppatore tramite script C#;
4. la *Scene*, permette la visualizzazione in 3D o 2D degli oggetti attivi inseriti nella Hierarchy, la navigazione dello spazio 3D (similmente ai comuni programmi di computer grafica come Maya, Blender, Houdini, Nuke), la modifica delle caratteristiche fisiche degli oggetti. Quando una scena viene caricata, agli oggetti precedentemente presenti vengono sostituiti quelli della Hierarchy corrispondente alla scena attiva;
5. Il pannello *Game*, garantisce una prima rappresentazione del prodotto finale, così che lo sviluppatore possa testare, in modo veloce, l'applicazione senza doverla esportare.

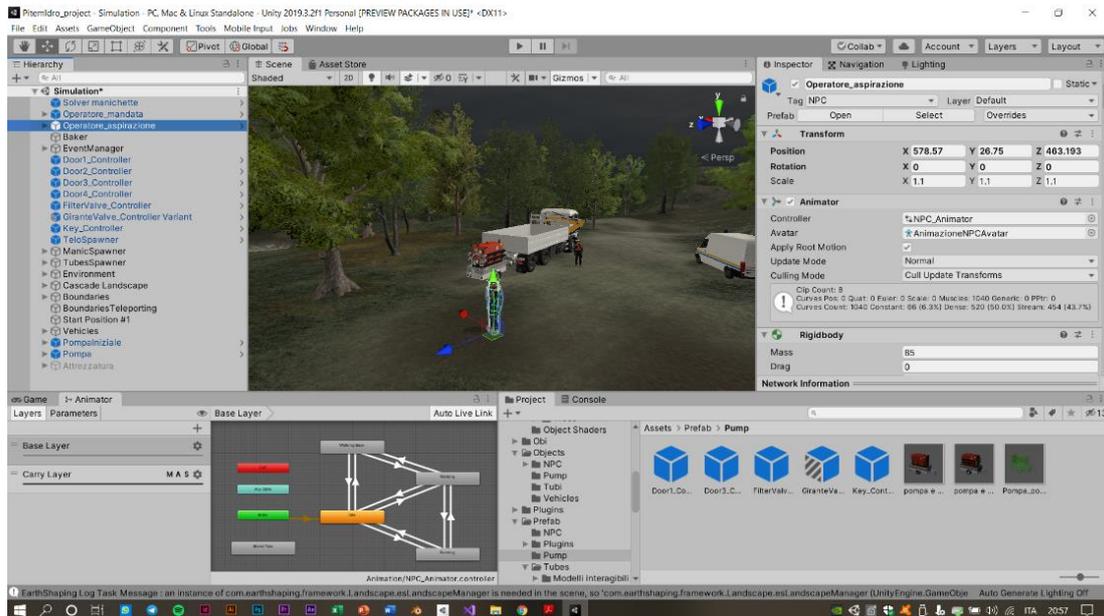


Figura 3.4: Interfaccia di Unity 2019.3.2

Unity è caratterizzato da quattro paradigmi: *GameObject*, *Component*, *Prefab* e *Script*. Su di questi il software basa tutta la sua logica di sviluppo.

Con *GameObject* si intende ogni elemento presente in una scena (avatar, elementi di background, effetti particellari, elementi di illuminazione, camere).

Ogni *GameObject* è a sua volta caratterizzato da *Component*, cioè attributi, che possono essere build-in di Unity o creati dall'utente tramite script e aggiunti e modificati tramite Inspector.

Un *GameObject* può poi essere salvato in un file di progetto (chiamato *Prefab*) e reso disponibile all'utente tramite le cartelle nel pannello Project in modo che possa essere importato in qualsiasi scena.

I component creati ad-hoc dallo sviluppatore vengono implementati tramite *Script* in C#. Insieme al C++, il C#, rappresenta infatti lo standard di programmazione dell'industria videoludica, in quanto risultano molto versatili e presentano una documentazione accurata.

Unity presenta inoltre un proprio store (*Asset Store* [46]) sotto forma di un ampio database di script, modelli e prefab gratuiti o a pagamento resi disponibili ai programmatori per facilitare lo sviluppo di progetti complessi. Nello sviluppo dello stesso progetto di tesi in questione, sono stati utilizzati asset reperiti dall'Asset Store di Unity e utilizzati nella creazione dello scenario, nella gestione della IK, nella fisica dei tubi e nello sviluppo della logica multiutente.

### 3.2.3 Microsoft Visual Studio

Il software Unity si appoggia su altre applicazioni per la scrittura di codice, il debugging e la creazione di contenuti grafici.

L'aspetto relativo al coding, nel progetto oggetto di questa tesi, è stato gestito tramite il software *Visual Studio* (versione 2019) [47]. Si tratta di un *Ambiente di Sviluppo Integrato* (IDE) appartenente a Microsoft, in grado di riconoscere svariati linguaggi di programmazione come Visual Basic, C#, C++, XML, JavaScript, HTML5 e CSS.

Il software permette la scrittura di codice, il testing, il debugging e la produzione di contenuti per qualsiasi piattaforma quali PC, mobile e console; le innumerevoli estensioni permettono, inoltre, di utilizzarlo anche per la scrittura con linguaggi meno comuni tra cui Python e Node.js e gli add-on ne permettono l'utilizzo anche tramite programmi esterni quali, appunto, lo stesso Unity.

Unity permette infatti l'impostazione di un IDE predefinito e di importare librerie specifiche per facilitarne il collegamento e il debugging. È possibile inoltre collegare l'IDE a *Git* (software di controllo versione rilasciato da *Linus Torvalds*) facilitando la condivisione del progetto durante il suo sviluppo.

### 3.2.4 Blender

Nel progetto oggetto di questa tesi, si è deciso di utilizzare il software open source *Blender* (versione 2.82) [48] come programma di modellazione, animazione, rigging, rendering e texturing.

Similarmente ad ogni software di grafica 3D, l'interfaccia di Blender (Figura 3.5) è suddivisa in pannelli personalizzabili; tra i più importanti troviamo la *Viewport* (per la navigazione della scena), lo *Shading* (tramite cui è possibile gestire i nodi dei materiali o shader), l'*UV mapping* (per gestire la disposizione della texture sulla superficie dell'oggetto), la *Timeline* (per animare tramite keyframe) e il pannello di *Rendering*.

Tramite Blender, inoltre, è possibile gestire l'intera pipeline di realizzazione di contenuti CGI, compreso il compositing (la simulazione di VFX e sistemi particellari), la Motion Tracking, il montaggio video e l'illustrazione e animazione 2D/3D.

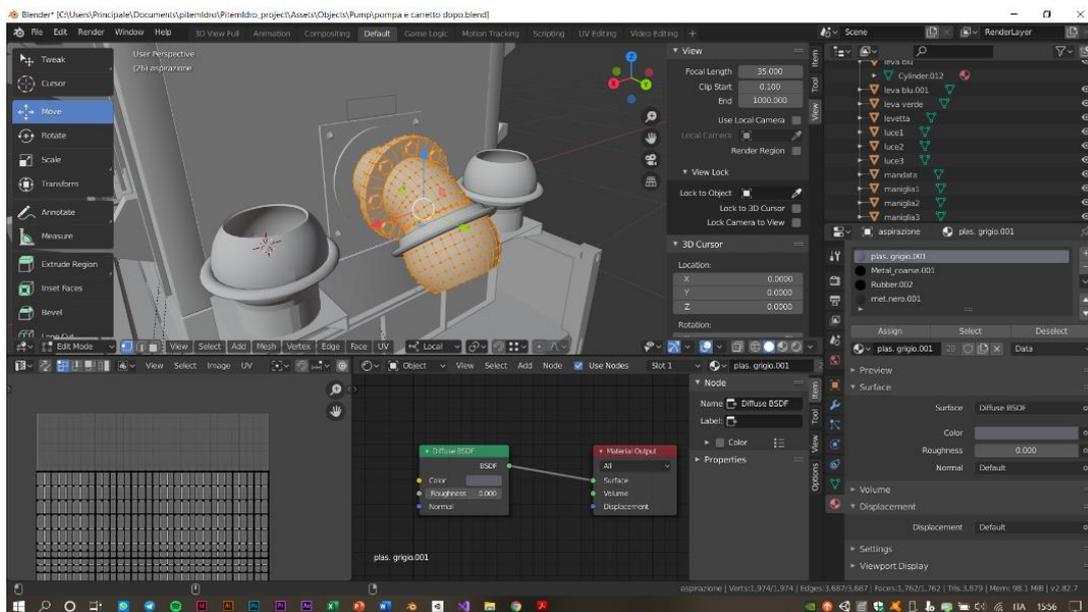


Figura 3.5: Interfaccia Blender 2.82

Per quanto riguarda il progetto di tesi in questione, si è utilizzato Blender nella modellazione dell'attrezzatura in dotazione al Modulo HCP, tra cui la motopompa Godwin, il modello dei personaggi non giocabili, le animazioni principali, l'UV mapping e la retopology dei modelli.

### 3.2.5 MakeHuman e Mixamo

*MakeHuman* [49] è un software gratuito per la realizzazione di modelli 3D umanoidi. Il programma permette di utilizzare il *morphing*, trasformando un modello umanoide standard in una grande varietà di personaggi tramite l'interpolazione lineare. L'applicazione è stata utilizzata nel progetto di tesi in questione per la creazione dei modelli degli NPC.

Con lo stesso scopo si è utilizzato il servizio online di animazione 3D *Mixamo* [50]. La piattaforma web fornisce un database gratuito di animazioni 3D esportabili. È possibile poi integrarli al modello 3D tramite programmi esterni come *Blender*, *Autodesk Maya*, *3ds Max* e altri. In alternativa è possibile anche importare il modello 3D da animare direttamente sulla piattaforma che ne creerà un *Rig* (scheletro del modello) a cui associare l'animazione.

# Capitolo 4

## Progettazione

La progettazione è una fase fondamentale per lo sviluppo di ogni progetto e consiste in tutte quelle attività che andranno a definire il risultato finale prima della vera e propria realizzazione. Nel progetto di tesi in oggetto la progettazione è stata utilizzata per definire, a partire dal materiale fornito ed estrapolato dagli incontri con la PC, un documento di design in cui vengono descritti tutti gli aspetti della simulazione.

In questo capitolo si andrà quindi a definire questa fase del lavoro di tesi, approfondendo i motivi legati alle scelte attuate nella simulazione andando ad esaminare il materiale audio/video raccolto, la struttura applicativa, lo scenario, il voice-over e i parametri di valutazione.

### 4.1 Acquisizione del materiale

Il materiale raccolto nelle fasi iniziali di progettazione consiste in materiale scritto, audio e video raccolto durante riunioni ed esercitazioni. Il materiale è stato utilizzato per andare a definire gli obiettivi del progetto *PITEM FOR*, la modalità di lavoro della PC, l'area di applicazione finale della simulazione e le procedure attuate in caso di rischio idrogeologico.

#### 4.1.1 Progetto PITEM, Protezione Civile e Regione Piemonte

Come già descritto nel Capitolo 1, la simulazione nasce nel contesto del progetto *PITEM FOR* per la gestione del rischio idrogeologico, definendo come obiettivo l'addestramento efficace ed intuitivo alle procedure messe in atto dalle organizzazioni di primo soccorso identificate, nella Regione Piemonte, dal Dipartimento della PC Regionale.

I primi incontri con la PC si sono quindi concentrati sul funzionamento del Dipartimento in caso di emergenza, delle procedure standard attuate in caso di rischio idrogeologico e la designazione della sotto-procedura su cui concentrarci nello sviluppo della simulazione, il tutto coadiuvato dal materiale reperito negli archivi della PC o raccolto su internet dal sito della Regione Piemonte. Il materiale è stato utile a comprendere i reali bisogni della PC, il concetto di “colonna mobile” (descritto nel Capitolo 1), il funzionamento dei singoli moduli e quali fossero i disastri legati al rischio idrogeologico più comuni nella Regione.

È emerso come la PC non disponga ancora di un programma di addestramento formalizzato per il modulo (HCP) e ciò, aggiunto al fatto che l’attrezzatura è molto costosa e non facilmente trasportabile, ha fatto ricadere la scelta dell’argomento della simulazione sul modulo stesso.

Successivamente, è stato possibile concentrare le ricerche sul modulo in questione, raccogliendo ulteriore materiale specifico direttamente dagli archivi della PC, ricavando video di esercitazioni svolte nel passato e documentazione sugli standard europei per il modulo.

#### 4.1.2 Esercitazioni e sopralluoghi

Contemporaneamente al processo di progettazione è stato possibile partecipare ad esercitazioni e sopralluoghi alle sedi della PC per esaminare la strumentazione utilizzata e visionare il suo utilizzo sul campo.

- Settimana nazionale della PC, San Mauro Torinese (13-19 ottobre 2019). L’esercitazione, incentrata sul rischio idrogeologico, è stata utile per avere una prima impressione sulle procedure della PC e sugli standard messi in atto in questa tipologia di emergenza.
- Sede della PC Druento. Presso la sede è stato possibile ispezionare il Modulo HCP equipaggiato con motopompa autoadescante *Godwin*, l’attrezzatura con cui è equipaggiato e l’utilizzo della gru. Oltre a materiale audio e video sugli strumenti (per la modellazione) e sulle procedure di montaggio (per stilare una lista delle fasi della simulazione), è stato possibile raccogliere informazioni direttamente dagli operatori che utilizzano il modulo, così da avere un parere diretto sulle difficoltà e sugli errori comuni commessi durante l’operatività. Si è scelto di utilizzare questa tipologia di modulo (in termini di pompa ed attrezzatura correlate) per la simulazione.
- Sede della PC Vercelli. Durante il sopralluogo è stato possibile raccogliere ulteriore materiale audio e video riguardante una diversa tipologia di Modulo HCP attrezzato con motopompa autoadescante di tipologia *Albatros* e dell’attrezzatura ad essa

collegata (rimorchio, manichette, altri attrezzi). Dal momento che la pompa *Albatros* ha un peso maggiore della *Godwin*, il suo trasporto avviene tramite camion con cassone scarrabile (non considerato nella simulazione per quanto detto sopra).

- Esercitazione HCP, Nodo Idraulico d'Ivrea. L'esercitazione, organizzata appositamente, si è focalizzata sull'utilizzo sul campo della pompa di tipologia *Godwin*, per delineare meglio le fasi di messa in sicurezza, montaggio, utilizzo e smontaggio del modulo. Inoltre, è stato possibile anche visualizzare l'area del Nodo Idraulico che, essendo una zona spesso soggetta ad allagamenti, è risultato un ottimo esempio su cui basare il design dell'ambiente da inserire nella simulazione.

Il materiale audio/video raccolto è stato integrato alle operazioni descritte sui manuali e documenti raccolti precedentemente sulle procedure standard e sul Modulo HCP, per andare a definire una lista delle fasi operative e, conseguentemente, lo storyboard utilizzato la realizzazione della simulazione.

## 4.2 Documento di design

Il risultato finale della fase di progettazione è un documento contenente tutte le informazioni necessarie alla realizzazione della simulazione e alla visualizzazione del prodotto finale. Contiene, ad esempio, la descrizione della logica, del design dell'ambiente, della interfaccia utente e delle modalità di gioco, in modo da fornire una rappresentazione il più dettagliata possibile di tutti i componenti della simulazione.

### 4.2.1 Il soggetto

La simulazione si svolge in una zona boschiva lungo l'argine di un fiume in piena che esondando ha causato la formazione di un versamento che dovrà essere prosciugato. I volontari della PC hanno il compito di dragare la pozza tramite l'utilizzo del Modulo HCP attrezzato con motopompa autoadescante e scaricare l'acqua in un canale parallelo. Gli operatori andranno così a svolgere tutte le procedure necessarie per la messa in sicurezza, il montaggio e l'utilizzo della pompa.

Le operazioni si basano sulle reali procedure utilizzate dalla PC che vengono suddivise nelle seguenti fasi.

### Fase 1: Allestimento campo operativo

La squadra deve svolgere un sopralluogo della zona per decidere le zone d'aspirazione, mandata e posizionamento della pompa più adeguate. Una volta deciso e verificato lo stato dei Dispositivi di Protezione Individuale (DPI) di ciascun membro della squadra, si prosegue con il posizionamento della pompa e delle manichette in un'area ottimale. Per ultimo, si delimita la zona operativa con pompa e manichette e si monta il telone su cui avverrà lo scarico dell'acqua.

### Fase 2: Stesura manichette

Ora i membri della squadra, due per volta, montano le manichette di aspirazione e di scarico (mandata); la mandata è composta in genere da tubi semirigidi partendo dalla pompa fino alla zona di aspirazione nella quale, dopo l'aggiunta del filtro e della corda per il suo recupero, si procede all'immersione del capo nella pozza (l'operatore deve essere assicurato ad una corda di sicurezza). Il montaggio della mandata, che si svolge in egual modo ma in senso opposto, deve essere composta inizialmente da manichette semirigide o tubi rigidi per lo scarico della pressione e, in seguito, da manichette appiattibili o tubi rigidi in casi particolari (curve, ostacoli), fino a poggiarne l'estremità sul telone blu. L'ultimo capo della catena deve poi essere assicurato con una corda ad alcuni picchetti.

### Fase 3: Avviamento e monitoraggio

La squadra si prepara all'avviamento della pompa. Un operatore si posiziona a lato del pannello di controllo (aprendo le ante dell'armatura insonorizzante), controlla che le valvole della girante e del filtro siano chiuse, quindi gira la chiave per avviare il motore (si suppone che la pompa sia automatica, pertanto non ci sono operazioni da svolgere per portarla in pressione). L'operatore controlla la catena di aspirazione e di mandata in cerca di pozze d'acqua o zampilli che provengono dal tubo e che possono essere state causate da una manichetta sganciata o da un guasto. Nel caso di perdita la pompa deve essere spenta il più velocemente possibile.

### Fase 4: Smontaggio

Una volta che la pompa è spenta, l'addetto vicino al pannello di controllo apre le valvole del filtro e della girante, facendo entrare aria. Una volta che l'acqua è defluita, è possibile tirare a riva il filtro mediante la corda. Per le manichette appiattibili (mandata) si aprono le semisfere nei punti di caduta (ad esempio, in fondo a una discesa).

## 4.2.2 Modalità supportate

Il singolo utente impersona un membro della PC, interagendo da solo con l'ambiente o con operatori controllati dal calcolatore (NPC).

La simulazione supporta due modalità, descritte di seguito.

- Simulazione didattica guidata con voice-over: l'utente svolge una simulazione guidata in cui compie tutte le operazioni che costituiscono l'esercitazione, con il supporto di un operatore (NPC) che lo aiuta nel trasporto dell'attrezzatura e nel montaggio delle manichette e dei tubi. L'utente segue le istruzioni dategli passo-passo da un voice-over in concomitanza all'evidenziazione degli oggetti. L'utente può decidere quali fasi e rispettivi ruoli svolgere, ma la sua libertà d'azione è limitata all'istruzione da eseguire in quel momento senza quindi la possibilità di commettere errori.
- Simulazione con valutazione finale: la simulazione viene svolta in autonomia, con la sola supervisione del caposquadra (come nel caso precedente), con la differenza che non vi saranno segnalazioni di errori, e l'intero operato dell'utente verrà valutato al termine della simulazione in base a specifiche norme e regole decise in precedenza. La modalità ha lo scopo di aumentare il realismo, in modo che l'utente possa sentirsi libero di agire e possa commettere eventualmente degli errori come in una reale esperienza sul campo.

Le due modalità differiscono quindi sulla sola presenza o meno della segnalazione di errore durante l'operato dell'utente e dalla conseguente valutazione finale.

Ogni modalità è inoltre suddivisa in fasi selezionabili tramite menù, in modo che l'utente possa scegliere quali fasi svolgere e quali lasciare.

## 4.2.3 Il design dell'ambiente

Inizialmente, il progetto prevedeva che la simulazione rappresentasse un'ipotetica esondazione della diga del Moncenisio. Il rischio in questione è stato però ritenuto troppo singolare e particolare per essere abbastanza realistico da utilizzare in una simulazione didattica; il rischio infatti doveva essere abbastanza comune da poter essere ritenuto un buon esempio su cui basare il design dell'ambiente.

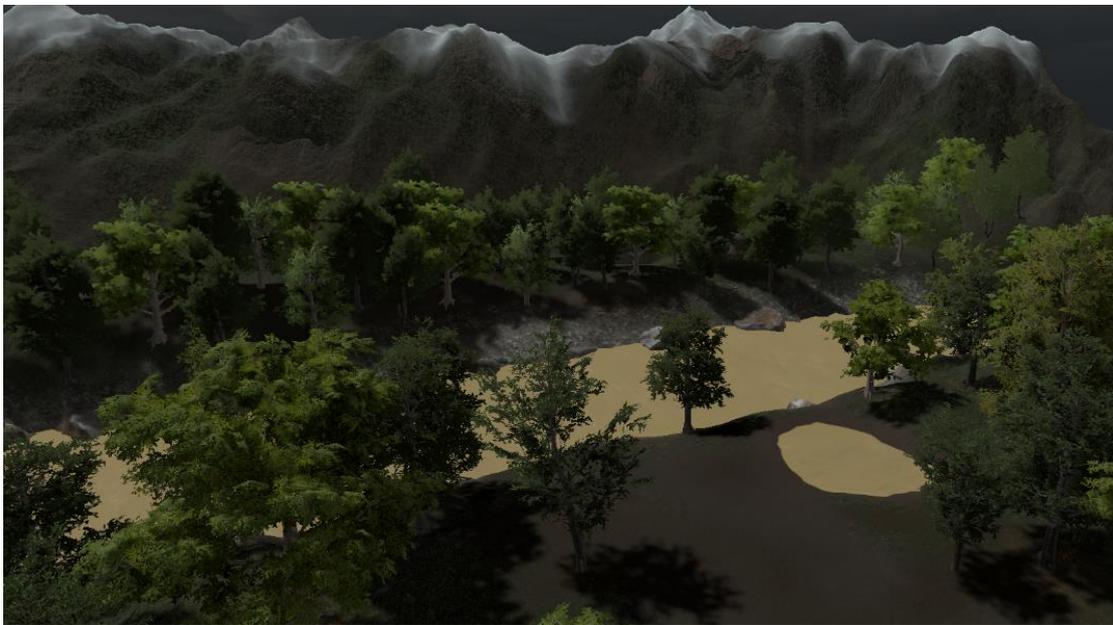
Si è deciso così di andare a studiare quali fossero i reali rischi idrogeologici nei quali la PC della Regione Piemonte si ritrova ad agire ogni anno.

In seguito alle ricerche svolte ed alle riunioni con la PC si è deciso di individuare il Nodo Idraulico d'Ivrea come base per il design dell'ambientazione.

L'esercitazione HCP al Nodo sopra menzionata è stata, quindi, anche molto utile ai fini della raccolta di materiale fotografico e video dell'area e delle sue caratteristiche che la rendono così vulnerabile al rischio idrogeologico. Il Nodo è infatti una zona della città d'Ivrea lungo la Dora Baltea; l'argine in questo punto è talmente basso da causare frequenti esondazioni del fiume e costringere la PC a numerosi interventi come quelli del 1994 e del 2000 [51].

Si è deciso di non andare a ricreare fedelmente il Nodo, ma di utilizzarlo come esempio per il design. Infatti, essendo la simulazione destinata non solo alla Regione Piemonte ma, potenzialmente, anche ad altri territori, si è preferito creare un ambiente generico e scalabile che potesse quindi risultare familiare a qualsiasi fruitore.

Partendo da queste premesse si è proceduto a definire il design dell'ambiente; la simulazione si svolge in una zona boschiva in una valle percorsa da un fiume. La squadra agisce in una radura su uno dei margini del fiume che, esondando, ha creato una pozza d'acqua (Figura 4.1).



*Figura 4.1: Fiume e pozza modellati nello scenario di simulazione*

Parallelamente al fiume si trova un canale secondario, rappresentante il punto di mandata, in cui si dovrà poi scaricare l'acqua aspirata dalla pompa. Nell'area operativa saranno già disposti i mezzi e l'attrezzatura del Modulo HCP (Figura 4.2).

L'area sarà inoltre priva di ostacoli come alberi, buche o zone in pendenza, poiché lo scopo della simulazione è quello di creare una situazione standard così che l'insegnamento possa avvenire nel modo più semplice e privo di imprevisti possibile. Si previene così la possibile

confusione dovuta alle numerose configurazioni alternative dovute alle caratteristiche del territorio ed è possibile concentrare l'attenzione sullo svolgimento corretto ed in sicurezza delle procedure.



*Figura 4.2: Modulo HCP posizionato nell'ambiente di simulazione*

#### 4.2.4 Il Modulo HCP

Il modulo europeo HCP (Figura 4.3) è costituito a partire dalle regole stabilite dal cosiddetto Meccanismo Europeo di PC con lo scopo di uniformare i moduli preesistenti e stabilire standard comuni a tutti i dipartimenti PC nazionali così che, in caso di crisi internazionale, tutti i moduli utilizzati presentino le stesse caratteristiche ed attrezzature.

Come detto, tra le motopompe prese in considerazione per la simulazione si è scelto di utilizzare il modello *Godwin* (portata 150m) poiché più spesso adoperata per costo e comodità d'utilizzo, al posto ad esempio della più potente ma ingombrante *Albatros*.



Figura 4.3: Motopompa Godwin (fotografia e ricostruzione 3D)

Il Modulo HCP comprende l'attrezzatura riportata di seguito (Figura 4.4).

- Camion e furgone officina: utilizzati per il trasporto delle manichette, del rimorchio con la pompa, della squadra operativa e dell'attrezzatura. I mezzi sono di colore bianco e presentano il logo e le strisce gialla e blu tipiche della PC. I modelli 3D sono stati direttamente forniti dalla PC.
- Pompa: la pompa scelta per la simulazione è del tipo autoadescante modello *Godwin*. La pompa si trova all'interno di una cassa isolante di colore rosso ed è trasportata tramite un rimorchio. Tra gli strumenti con cui la pompa è attrezzata ci sono il quadro di comando e le due valvole del filtro e della girante.

- Tubi rigidi grigi: il modulo ne comprende uno lungo 1m e due curvi a 90°. Vengono in genere utilizzati per il superamento degli ostacoli o per compiere delle curve strette nella catena di mandata.
- Manichette semirigide nere: utilizzati per lo più nella catena d'aspirazione o all'inizio della catena di mandata in modo da diminuire la pressione dell'acqua. Il modulo ha cinque tubi semirigidi da 4m e uno da 6m.
- Manichette appiattibili rosse: questa tipologia di manichetta viene solamente utilizzata nella catena di mandata, è lunga 17m e nel modulo ne sono presenti due.
- Attrezzo: il modulo comprende due tubi curvi con la punta rossa utilizzati per effettuare la chiusura delle manichette.
- Filtro: il modulo ne comprende uno da collegare alla fine della catena d'aspirazione.
- Paletti e nastro: quattro paletti ed un nastro sono utilizzati per effettuare la delimitazione dell'area operativa.
- Corde: le corde servono per legare le manichette della catena d'aspirazione e il filtro, per facilitarne l'estrazione dall'acqua, assicurare l'ultima manichetta della catena di mandata al telo blu e assicurare a terra l'operatore durante l'operazione di posizionamento del filtro in acqua.
- Telo blu: il telo deve essere posizionato sotto l'ultima manichetta della catena di mandata così da smorzare la forza dell'acqua uscente. Il telo è blu, in modo da distinguerlo dal telo giallo utilizzato per altri scopi.

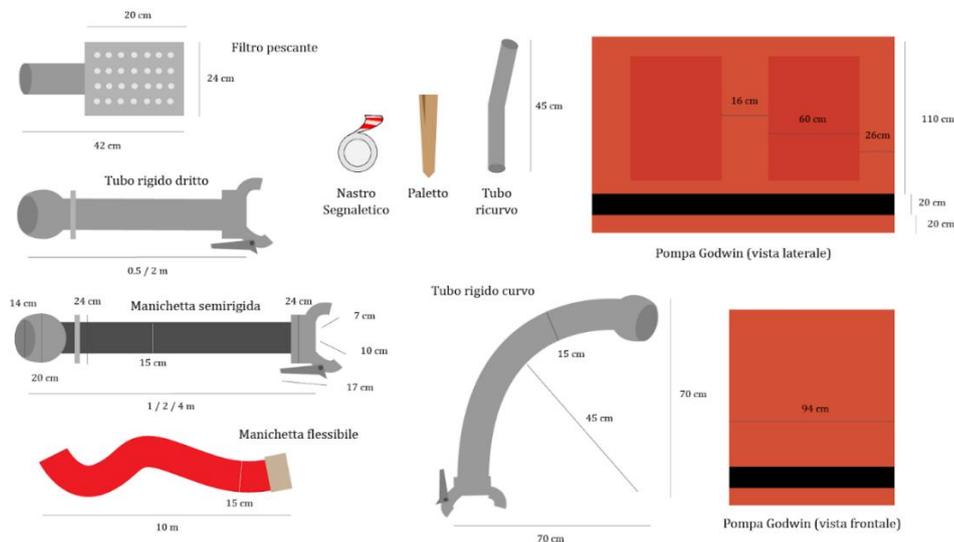


Figura 4.4: Equipaggiamento di un Modulo HCP

#### 4.2.5 Personaggi non giocanti

I NPC hanno un ruolo fondamentale nella simulazione. Infatti, la squadra HCP è in genere composta da cinque o sei operatori, le operazioni sono parallelizzabili (alcune persino impossibili da svolgere singolarmente), e l'attrezzatura è troppo pesante ed ingombrante per essere trasportata da soli. È stato così necessario sviluppare un sistema in grado di gestire operatori fittizi controllati dal sistema così che durante la simulazione l'utente sia sempre accompagnato da un NPC impersonante un compagno di squadra, implementato per andare in soccorso all'utente nelle operazioni impossibili da completare in solitaria.

In particolare l'aiuto di un NPC è fondamentale nelle fasi di montaggio delle manichette, in quanto è impossibile sollevarle e trasportarle da soli ed è necessario che, durante l'aggancio di due manichette tra loro, ci sia qualcuno a mantenere ferme le due aperture.

Inoltre, si è andati ad utilizzare gli NPC anche per svolgere in autonomia, con una modalità automatizzata, le operazioni non svolte dall'utente (cioè quelle fasi non selezionate sul menù prima dell'inizio della simulazione) come ad esempio le fasi di montaggio delle manichette di aspirazione e di mandata.

Infine, si è deciso di basare il design degli NPC sugli stessi reali operatori del Modulo HCP (Figura 4.6) per i quali è stato possibile raccogliere diverso materiale fotografico durante i sopralluoghi alle sedi della PC piemontese e durante le esercitazioni. Ogni operatore deve indossare una certa tipologia di attrezzatura a seconda del rischio sul qual sta operando poiché sono destinati alla salvaguardia e sicurezza dell'operatore stesso; questo tipo di attrezzatura è definita DPI.



*Figura 4.5: Equipaggiamento di un operatore della PC per rischio idrogeologico*

Il DPI di un operatore HCP comprende:

- tuta standard con strisce catarifrangenti;
- scarponi;
- giubbotto di salvataggio;
- caschetto con il logo della PC e torcia;
- imbragatura con gancio per moschettone;
- radio.

#### 4.2.6 Interfaccia di gioco

L' interfaccia di gioco è particolarmente importante in qualsiasi applicazione essendo in effetti il primo elemento visualizzato dall'utente. Per questo motivo deve risultare il più allettante e funzionale possibile.

Il menù iniziale è costituito da una scena ispirata ad una reale sala di comando (Figura 4.6) e, tramite un pannello contenente pulsanti selezionabili tramite raicasting, permette all'utente di personalizzare la propria esperienza (nel modo illustrato dalla Figura 4.6). La prima opzione offerta è quella della modalità di gioco: didattica guidata o con valutazione finale.

Entrambe le alternative porteranno alla conseguente scelta delle fasi e ruoli che si vogliono interpretare durante la simulazione. L'utente può decidere di saltare particolari operazioni o addirittura intere fasi che saranno poi effettuare automaticamente dall'applicazione o dal sistema d'automazione degli NPC.



*Figura 4.6: Sala operativa PC*

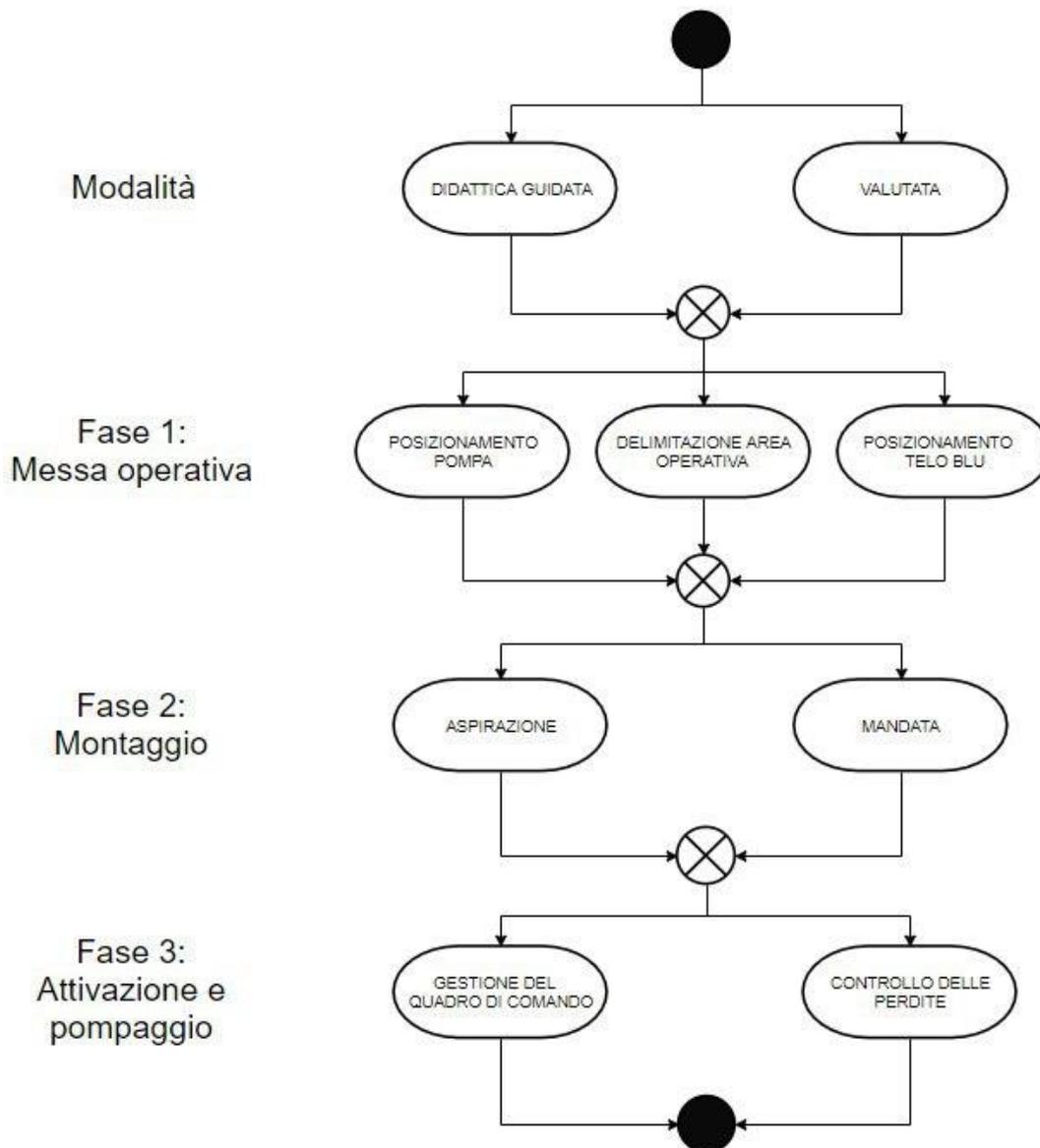


Figura 4.7: Flow chart del menù iniziale

Durante la simulazione, l'interfaccia d'insegnamento si basa per la maggior parte su feedback di tipo uditivo (voice-over) trasmessi attraverso una radio fittizia basandosi sul reale sistema di comunicazione adottato dalla PC tra gli operatori stessi. Si parlerà più nel dettaglio di questa interfaccia nella sezione 4.2.7.

Infine, nella modalità valutata, al completamento della simulazione verrà visualizzata una schermata di resoconto contenente tutti gli errori commessi durante le operazioni, le dimenticanze e il comportamento generale adottato dall'utente andando così a valutarne l'operato secondo i parametri concordati con la PC.

#### 4.2.7 Voice-over, indizi ed errori

Il voice-over è uno dei punti cardine della simulazione, essendo proprio questo a guidare l'utente in tutte le fasi delle procedure da attuare diventando quindi l'insegnante virtuale del volontario.

È stata posta particolare attenzione al testo del voice-over in quanto, essendo la simulazione atta all'insegnamento, è di particolare importanza la sua correttezza. Oltre ad un'iniziale parte di ricerca, il testo è stato anche più volte sottoposto a fasi di validazione e correzione da parte degli operatori della PC con il fine di sviluppare una versione la più corretta, diretta ed esplicativa possibile, evitando istruzioni errate e ambigue.

Il voice-over è fondamentale quindi nella modalità didattica guidata, nella quale, incarnando il ruolo del caposquadra, va a spiegare ogni passo delle procedure da seguire nelle diverse fasi, l'utilizzo corretto di ogni strumento in dotazione al modulo e gli errori comuni da evitare.

Per facilitare l'apprendimento, nella scena saranno presenti indizi, o segnalatori 3D che andranno ad evidenziare le zone adatte al posizionamento dell'attrezzatura, gli oggetti di riferimento e le zone in cui spostarsi.

Il sistema è inoltre capace di rilevare i possibili errori commessi dall'utente, in modo da poterli sottolineare a quest'ultimo, sempre tramite clip audio, così da correggere il suo comportamento.

Come detto, si è scelto di sfruttare la presenza del caposquadra come "voce" del voice-over, veicolandola attraverso una fittizia radio in dotazione all'operatore, piuttosto che altre metodologie di comunicazione (ad esempio di pannelli scritti) in quanto risulta maggiormente realistico e aumenta il concetto di immersione per l'utente.

Nella modalità di valutata, il voice-over viene utilizzato solamente per fornire informazioni generiche all'utente, in quanto si suppone che quest'ultimo sappia già effettuare le procedure imparate nella modalità guidata, lasciandogli così completa libertà d'azione.

#### 4.2.8 Valutazione

Alla fine della modalità valutata, verrà visualizzato un pannello di valutazione compilato in base a criteri definiti dalla fase e dal ruolo interpretati durante la simulazione. I criteri di valutazione sono stati concordati con l'aiuto degli operatori della PC e comprendono punti critici delle procedure ed errori comuni compiuti dai volontari addetti al Modulo HCP.

I criteri generali oppure specifici per le diverse fasi, sono riportate di seguito.

## Generali

- Ogni operatore può effettuare solo le operazioni assegnategli all'inizio della simulazione per permettere una corretta collaborazione di tutti i membri.
- Ogni operatore deve attendere che il compagno abbia completato il suo compito prima di passare al successivo.
- Le operazioni devono essere svolte nell'ordine spiegato nella simulazione guidata (posizionamento pompa, delimitazione area, montaggio aspirazione/montaggio mandata, controllo valvole, accensione pompa, monitoraggio, spegnimento, apertura valvole).
- Le operazioni devono essere svolte completamente.
- Azioni estranee all'operazione in atto non sono permesse.

## Sopralluogo e DPI

- Ogni operatore deve effettuare un sopralluogo adeguato avvicinandosi alle zone di mandata e aspirazione.
- Ogni operatore deve indossare casco e giubbotto di salvataggio.

## Fase 1: Messa operativa

- Posizionamento della pompa:
  - distanza della pompa dalla pozza < 8m;
  - la zona deve essere libera da ostacoli;
  - aperture della pompa ruotate verso la pozza per evitare la distorsione delle manichette appiattibili e semirigide.
- Delimitazione dell'area operativa:
  - utilizzo di tutti e quattro i paletti e nastro;
  - delimitazione dell'intera zona comprendente pompa e attrezzatura.
- Posizionamento del telo blu:
  - utilizzo di tutti i picchetti nei punti giusti;
  - posizionamento in una zona libera da ostacoli.

## Fase 2: Montaggio

- Aspirazione:
  - utilizzo di sole manichette semirigide;
  - montaggio del filtro come ultimo componente della catena;
  - inizio del montaggio a partire dalla pompa;
  - posizionamento del filtro non troppo lontano dalla riva ma abbastanza distante per essere completamente coperto dall'acqua;

- tutti i tubi chiusi prima dell'accensione;
- filtro e catena assicurati alla corda per l'estrazione;
- assicurato alla corda di sicurezza prima di entrare in acqua.
- Mandata:
  - utilizzo di manichette semirigide o rigide come primi pezzi della catena;
  - utilizzo di sole manichette appiattibili in seguito;
  - utilizzo di una manichetta semirigida come ultimo componente della catena;
  - fissaggio dell'ultimo tubo ai picchetti del telo;
  - posizionamento dell'estremità finale della catena sul telo blu;
  - tutti i tubi chiusi prima dell'accensione;
  - assicurazione dell'ultimo capo della mandata con corda e picchetti

### Fase 3: Attivazione e pompaggio

- Gestione del quadro di comando:
  - valvole e porte chiuse;
  - catene di aspirazione e mandata montate correttamente;
  - pompa spenta celermente in caso di perdite;
  - valvole e porte non aperte durante il funzionamento;
  - valvole aperte prima dello smontaggio;
- Controllo delle perdite:
  - segnalazione celere di eventuali perdite;
  - manichette non spostate durante il funzionamento;
  - manichette non aperte durante il funzionamento;
  - controllo della catena di mandata e pozza.

#### 4.2.9 Logica di simulazione

La simulazione è stata divisa in fasi seguendo la reale suddivisione delle procedure analizzate nella Sezione 4.2.1, condensando le fasi 3 e 4 in una sola non potendo andare simulare un lungo utilizzo prolungato della pompa.

A loro volta, le fasi sono state suddivise in "ruoli", cioè le singole azioni da compiere per completare la procedura in esame nella fase.

Come detto, fasi e azioni possono essere scelte tramite il menù iniziale, dando così la possibilità all'utente di eseguire solamente specifiche procedure; fasi e ruoli non selezionati saranno svolti automaticamente dal sistema o degli NPC. Qualsiasi simulazione sarà comunque preceduta da una fase di sopralluogo e controllo dei DPI.

Nel seguito, vengono approfondite le diverse fasi.

## Sopralluogo e DPI

- Gli operatori compiono un sopralluogo dell'area operativa, muovendosi nello scenario e raggiungendo i punti di aspirazione e di scarico (segnalati da icone). Devono inoltre indossare i DPI retti da un NPC vicino alla pompa.

### Fase 1: Messa operativa

- Posizionamento della pompa: La pompa deve essere posizionata in una zona piana priva di ostacoli a meno di 8 m di distanza dalla zona d'aspirazione. Mediante il controller nella mano destra, si genera una proiezione tridimensionale della pompa direttamente nell'ambiente, così da deciderne orientamento e posizione nello spazio. Tutte le operazioni di posizionamento fisico della pompa e dell'attrezzatura non sono svolte manualmente dalla squadra ma sono gestite automaticamente dalla simulazione, facendo direttamente comparire la pompa, i tubi e le manichette nel punto del terreno scelto dal caposquadra.
- Delimitazione dell'area operativa: Prima di iniziare le operazioni di montaggio, è necessario delimitare la zona comprendente pompa e tubi tramite i paletti (inserendoli direttamente nel terreno) e il nastro (da collegare ai paletti) in dotazione.
- Posizionamento del telo blu: l'estremità della catena di mandata deve essere posta sopra ad un telo cerato di colore blu, utilizzato per smorzare la forza del flusso d'acqua in uscita. Il telo deve quindi essere posizionato nella zona di mandata (il telo si posizionerà automaticamente quando posato) e fissato tramite picchetti (da inserire nei fori appositi segnalati).

### Fase 2: Montaggio

- Aspirazione: Due operatori (l'utente ed un NPC) si occupano di trasportare i tubi e le manichette e di agganciarle in modo da formare una catena che parte dalla pompa e che raggiunge il punto di aspirazione. Le manichette sono in genere del tipo semirigido nero e vanno agganciate collegando un'estremità maschio a quella femmina dell'altro e agganciando i due capi con l'aiuto di un tubo ricurvo da inserire nella leva della femmina. La catena deve essere terminata con il filtro da inserire all'interno della pozza da aspirare. Prima di entrare nell'acqua l'operatore deve assicurarsi ad una corda di sicurezza trattenuta da un NPC. Il filtro e i tubi devono essere collegati tramite una corda (da collegare al filtro e ad ogni apertura successiva) che servirà a facilitare l'estrazione della catena dall'acqua.
- Mandata: Si procede come già descritto in precedenza per la catena di aspirazione andando ad utilizzare tubi semirigidi neri e rigidi grigi all'inizio della catena fino ad arrivare a terra (così da scaricare la pressione dell'acqua) e procedendo poi con manichette appiattibili rosse. Si conclude infine la catena con un tubo semirigido da assicurare tramite una corda ai picchetti del telo.

### Fase 3: *Attivazione e pompaggio*

- **Gestione del quadro di comando:** Prima dell'accensione bisogna controllare che le valvole di aspirazione e mandata e che le porte, ad esclusione di quella del quadro di comando, siano chiuse. A questo punto è possibile accendere la pompa girando la chiave nel pannello. Lo spegnimento può essere richiesto a causa di una perdita o allo svuotamento della pozza entrambi segnalati da un NPC con un'alzata di braccio. Dopo lo spegnimento è necessario andare ad aprire le valvole del filtro e della girante.
- **Controllo delle perdite:** Durante il funzionamento un operatore deve controllare che non vi siano perdite, dovute allo sgancio di tubi o ad una rottura, percorrendo le catene di manichette. In caso di perdita o di svuotamento della pozza, l'operatore dovrà avvertire l'NPC addetto alla gestione del quadro di comando alzando il braccio e premendo il trigger. In seguito allo spegnimento si può procedere all'apertura delle valvole del filtro e della girante.

# Capitolo 5

## Realizzazione

In questo capitolo si andranno a valutare le scelte implementative attuate nella realizzazione del progetto di tesi, con particolare riguardo al sistema d'interazione, alla logica d'automazione degli agenti ed il sistema di valutazione.

### 5.1 Sistema d'interazione

Dato il livello di dettaglio richiesto dall'intera simulazione, l'interazione con ogni strumento deve essere il più possibile realistico e simile alle reali procedure attuate dagli operatori della PC. Per questo motivo, come già detto in precedenza, si è deciso di sviluppare il lavoro basandosi su tecnologia VR con lo scopo appunto di aumentare il livello di realismo e di immersione. Tutto ciò ha portato alla realizzazione di un intero sistema d'interazione ad hoc che cerca di essere più dettagliato possibile e basato su tecnologia VR.

Nel seguito verrà illustrato come è stato realizzato nel dettaglio il sistema d'interazione player-ambiente, player-oggetti, NPC-oggetti, oggetti-oggetti.

Nonostante l'impiego del plugin *Steam VR* (descritto qua di seguito), il sistema d'interazione è quasi completamente stato implementato da zero, utilizzando le funzionalità del plugin solamente per la mappatura degli elementi tracciati (visore e controller), dei pulsanti sui controller e per alcune funzionalità specifiche.

#### 5.1.1 Steam VR

Steam è una piattaforma digitale sviluppata da *Valve Corporation* (2003) [52] che si occupa della distribuzione on-line di videogiochi e dei relativi supporti, e si tratta inoltre del programma più utilizzato del suo genere.

La piattaforma mette a disposizione numerosi tool di supporto scaricabili attraverso l'apposito store.

Uno di questi tool, *Steam VR* [53], è stato appositamente sviluppato in modo da permettere ai giocatori di utilizzare strumenti VR direttamente sulla piattaforma, permettendo così al software di commercializzare anche videogiochi della suddetta tipologia.

Il tool è in grado di supportare, esclusivamente su un sistema *VR Ready* (termine che indica dispositivi aventi i requisiti necessari all'utilizzo della VR), innumerevoli dispositivi VR come *HTC Vive*, *Oculus Rift* e *Windows Mixed Reality*.

### 5.1.2 Open VR

Valve ha reso inoltre disponibile un plugin chiamato *Open VR* [54] e scaricabile dallo store di Unity, utile ad interfacciare il programma direttamente con Steam VR. Il plugin rende quindi possibile la creazione efficiente di applicazioni in VR e consiste in una cartella dentro alla quale è possibile trovare molti elementi immediatamente utilizzabili come modelli 3D dei controller, player già impostati, script per il movimento e l'interazione ed una vasta e approfondita documentazione.

Di particolare utilizzo sono le scene già allestite che danno la possibilità di provare le varie opzioni messe a disposizione dal tool.

Steam VR mette inoltre a disposizione un'API (Application Programming Interface) tramite la quale è possibile andare a gestire i propri dispositivi hardware VR collegandoli automaticamente al sistema, senza dover fare affidamento su SDK (Software Development Kit) dello specifico produttore. L'API rende possibile un facile ed intuitivo controllo di aspetti come i modelli 3D dei controller, la gestione degli input e la mappatura degli elementi tracciati.

Unity si appoggia ad Open VR per ottenere, tramite triangolazione dei sensori, la posizione e l'orientamento del visore per poi applicarli alla camera principale della scena attiva. Successivamente, Unity invia l'immagine "ripresa" dalla telecamera al visore. In pratica il plugin funziona come un ponte che effettua la traduzione tra gli output VR e Unity inviandoli poi alla controparte.

Dalla versione 2.0 il programmatore, attraverso un'interfaccia online apposita chiamata *Input Binding UI* (Figura 5.1), è libero di personalizzare le impostazioni per le interazioni con i controller utilizzando i set predefiniti o creandone di propri e andando poi a salvarne la configurazione in quello che viene definito un *Action Set*.



Figura 5.1: Input Binding UI

L'API è suddivisa in sette interfacce principali, descritte di seguito.

- *Sistema IVRS*: interfaccia principale per la gestione degli elementi.
- *IVRChaperone*: fornisce l'accesso al sistema *Chaperone* (sistema di sicurezza dell'HMD che visualizza l'area sicura nel quale muoversi liberamente nel mondo reale).
- *Compositore IVR*: consente l'esecuzione del rendering di contenuti 3D tramite il compositore VR.
- *IVROverlay*: consente l'esecuzione del rendering di contenuti 2D tramite il compositore VR.
- *IVRRenderModels*: consente l'accesso ai modelli di rendering.
- *IVRScreenshots*: consente ad un'applicazione di accedere ai modelli di rendering.
- *IVRInput*: consente la personalizzazione, la creazione, la modifica e la condivisione di associazioni azioni/dispositivo.

Il plugin è risultato quindi molto utile allo sviluppo della simulazione, in quanto ha consentito di evitare di sviluppare da zero un sistema di tracking e di gestione dei sensori su controller e visore, cosa che sarebbe risultata molto dispendiosa sia in termini di tempo che d'impegno. Inoltre, la vasta documentazione e il frequente utilizzo da parte di sviluppatori in grado di fornire supporto tecnico ha fatto ricadere la scelta su questo tool al posto di altri come *Oculus VR* (API esclusiva per *Oculus*).

Prendendo come riferimento la tecnologia *HTC Vive*, si descrivono i comandi possibili e in particolare come questi siano stati assegnati nel sistema d'interazione del progetto di tesi.

- **Locomozione**: pulsante Grip con movimento delle braccia.

- Sollevamento, trasporto, rotazione e rilascio degli oggetti: pulsante Trigger.
- Apertura porte, valvole e chiave: pulsante Trigger.
- Interazione con il menù: *raycasting* (tecnica nella quale si va ad intersecare un raggio ad una superficie, descritta nel seguito) e pulsante Trigger.
- Posizionamento e rotazione pompa: raycasting e Trackpad destro.
- Segnalazione spegnimento: pulsante Trigger con sollevamento del braccio al di sopra della testa.
- Ripetizione ultima istruzione: trackpad sinistro.
- Passaggio alla fase successiva (solo nella modalità valutata): trackpad destro e sinistro.

### 5.1.3 Raycasting

Con raycasting si intende una tecnica utilizzata all'interno di ambienti 3D per la creazione di raggi che partono da un punto specificato e continuano all'infinito in una direzione specifica.

I raggi possono essere visibili o invisibili e vengono utilizzati per controllare le collisioni con gli oggetti che lo intersecano poiché l'idea è proprio quella di tracciare il raggio che parte da un oggetto, un pixel alla volta, fino a rilevare l'oggetto intersecato più vicino.

In poche parole è come se fosse l'oggetto stesso avesse la capacità di vedere.

La tecnica viene utilizzata con variati scopi: creare linee di tiro, controllare la visuale di un NPC, valutare la distanza tra due oggetti, ecc.

Unity utilizza la classe *Ray*, a cui si assegna l'origine, l'orientamento e la lunghezza (tutti gli oggetti posizionati oltre a questa distanza saranno "invisibili"), per la generazione del raggio, e *RaycastHit* per ottenere gli oggetti intersecati dallo stesso. Inoltre, è anche possibile inserire degli elementi dell'applicazione in una lista apposita (*LayerMask*) in modo che la collisione del raggio con gli stessi venga ignorata (Figura 5.2).

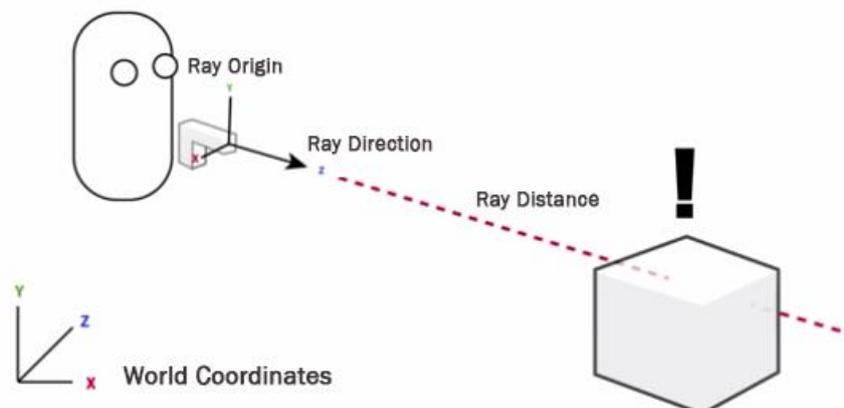


Figura 5.2: Esempio d'utilizzo del raycasting e parametri modificabili

Per la gestione del raycasting nel progetto di tesi, si è deciso di impiegare un sistema già implementato e reso disponibile dal plugin Open VR chiamato *Laser Pointer*. La scelta è stata guidata dall'efficienza del sistema e dalla complessità di un sistema basato sulla VR.

Il metodo è stato utilizzato nella scelta della zona per il posizionamento della pompa e per le interazioni con i pulsanti del menù iniziale.

#### 5.1.4 Gestione delle collisioni in Unity

Una parte importante della verosimiglianza di una simulazione 3D è il sistema di gestione delle collisioni. Infatti, senza di questo, il player e gli oggetti nell'ambiente restano intangibili, potendo persino passare l'uno attraverso l'altro.

Unity mette perciò a disposizione una serie di componenti a tale scopo.

Il componente *Collider* crea un volume tangibile intorno all'oggetto a cui viene applicato, e rileva le collisioni con altri collider inviando le informazioni agli eventi associati ed invocabili tramite script (*OnCollisionEnter*, *OnCollisionExit*, *OnCollisionStay*).

I Collider esistono in diverse forme (Figura 5.3) a seconda del volume che si vuole creare, ed ognuno di essi è modificabile nel centro di generazione e nei propri specifici parametri:

- *BoxCollider*: a forma di rettangolo, con base e altezza;
- *SphereCollider*: a forma di sfera, con raggio;
- *CapsuleCollider*: a forma di capsula, con raggio e lunghezza;
- *MeshCollider*: questo tipo di Collider prende la forma della mesh a cui viene applicato come una sorta di copertura aderente; attivando il parametro *convex*, invece di un Collider identico alla mesh, se ne ottiene uno semplificato e più semplice quindi da calcolare.
- *Altri*: esistono anche Collider per oggetti 2D ed altri specifici per alcune tipologie di oggetto.

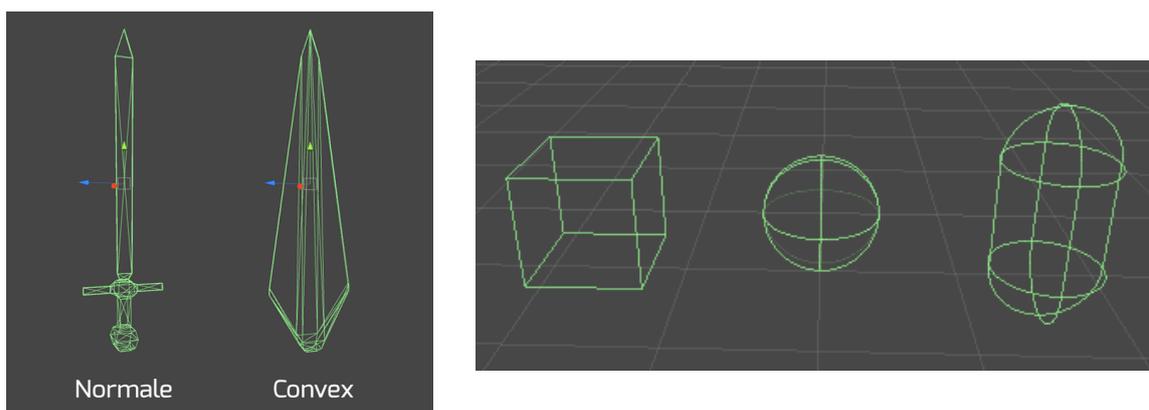


Figura 5.3: Esempi di Collider

I *Trigger* sono una versione alternativa di *Collider* e vengono utilizzati quando si vogliono rilevare le collisioni ma mantenere l'oggetto intangibile. Per attivarli basta spuntare la casella apposita nella componente *Collider* assegnata all'oggetto in questione nell'Inspector di Unity. Come i *Collider* classici, emettono eventi specifici alle collisioni (*OnTriggerEnter*, *OnTriggerExit*, *OnTriggerStay*). Un *Collider* non *Trigger* viene definito *Static*.

Essendo "invisibili", i *Collider Trigger* in genere vengono utilizzati per controllare eventi nei quali si vuole rilevare l'entrata o l'uscita di oggetti da una zona come l'ingresso del player in una stanza.

Bisogna notare che se un oggetto collide con un altro durante un movimento (e quindi durante la modifica del componente *Transform*), la collisione non verrà rilevata proprio come se non avesse incontrato nessun oggetto sulla sua traiettoria.

In questi casi bisogna aggiungere all'oggetto il componente chiamato *Rigidbody*, che indica che l'oggetto in questione potrà essere sottoposto alle forze fisiche implementate.

Senza il componente *Rigidbody* l'oggetto non sarebbe soggetto a forze come la gravità, l'attrito o la resistenza al movimento, e non reagirebbe alla collisione dopo un urto, dando così come risultato un movimento irrealistico.

I parametri impostabili su un *Rigidbody* (Figura 5.4) sono:

- *Mass*: massa dell'oggetto;
- *Drag*: resistenza opposta dall'oggetto durante uno spostamento;
- *Angular Drag*: resistenza opposta alla rotazione;
- *Use Gravity*: indica se l'oggetto è sottoposto o no alla gravità;
- *Is Kinematic*: rende statico il componente facendo così in modo che l'oggetto possa mantenere il *Rigidbody* senza però essere soggetto alle forze esterne.

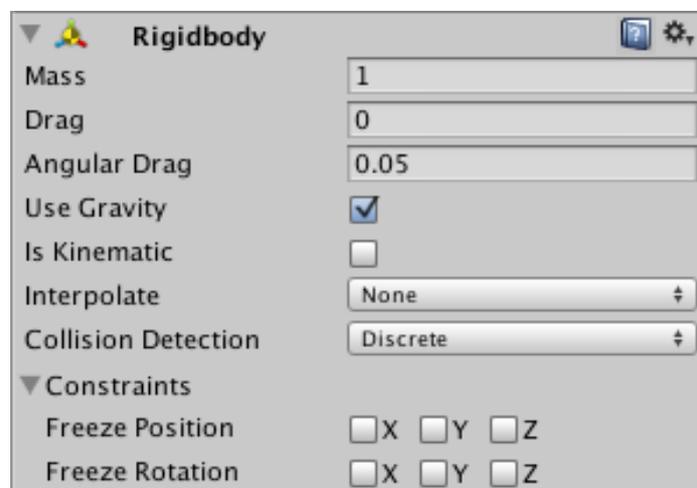


Figura 5.4: Componente *Rigidbody*

Quando due oggetti collidono possono verificarsi eventi diversi a seconda delle configurazioni dei Rigidbody e dei Collider su di essi, o addirittura nessun tipo di evento (Figura 5.5). Perciò, durante la realizzazione di un progetto Unity, bisogna anche tener conto di questo fatto e progettare la configurazione dei componenti in base alle interazioni che si vogliono implementare prima di iniziarne la realizzazione.

	Static Collider	Rigidbody Collider	Kinematic Rigidbody Collider	Static Trigger Collider	Rigidbody Trigger Collider	Kinematic Rigidbody Trigger Collider
Static Collider		X			Y	Y
Rigidbody Collider	X	X	X	Y	Y	Y
Kinematic Rigidbody Collider		X		Y	Y	Y
Static Trigger Collider		Y	Y		Y	Y
Rigidbody Trigger Collider	Y	Y	Y	Y	Y	Y
Kinematic Rigidbody Trigger Collider	Y	Y	Y	Y	Y	Y

*Figura 5.5: Tabella degli eventi generabili: OnCollision (X) OnTrigger (Y)*

Nella simulazione, il sistema è stato utilizzato in tutte le tipologie d'interazione.

- Oggetti interagibili: Static Collider e Rigidbody non Kinematic, dato che le mani del player hanno assegnato un Trigger Collider, questa assegnazione permette il rilevamento delle collisioni tra mano e oggetto. Ad ogni collisione con un oggetto interagibile si va a formare una sorta di contorno di colore giallo per sottolineare il fatto che l'oggetto è utilizzabile.
- Oggetti statici: Static Collider, dal momento che oggetti che devono rimanere fermi non hanno infatti bisogno di Rigidbody perché non soggetti a forze esterne.
- Player: Static Collider e Rigidbody non Kinematic con mani Trigger Collider e Rigidbody Kinematic, in quanto il player deve potersi muovere sottostando alle leggi della fisica, mentre le mani devono essere resistenti ad esse, in modo da poter interagire con gli oggetti senza, per esempio, essere costrette a terra dalla gravità.
- NPC: Static Collider e Rigidbody Kinematic.
- Terreno: Terrain Collider, un Collider specifico.

### 5.1.5 Locomozione

Nonostante l'alto grado di coinvolgimento sperimentabile tramite la tecnologia VR, questa mantiene sempre alcune limitazioni comuni ai videogiochi che, accostate alla realistica di altri elementi VR, risultano maggiormente alienanti del normale; una di queste è il sistema di locomozione.

Se non si dispongono di strumenti avanzati come ad esempio le pedane omni-direzionali, i metodi di movimento sono limitati a poche soluzioni che implicano comunque il rimanere, nella realtà, nella stessa area di pochi metri quadrati.

L'opzione più banale è il movimento tramite l'uso del joystick e dei pulsanti sui controller che, anche se accettabile nelle prime esperienze con questa tecnologia, oggi non fa altro che limitare l'esperienza VR.

Un altro metodo che implica l'uso dei controller è il *Teleporting* (Figura 5.7). Questo in genere utilizza il raycast per indicare il punto in cui si vuole andare per poi essere "teletrasportati" automaticamente sul luogo. È possibile così per lo sviluppatore creare facilmente delle zone permesse e non, o punti specifici in cui potersi spostare; per quanto possa essere un metodo veloce ed intuitivo, risulta molto estraniante per un'esperienza che dovrebbe essere quanto più immersiva possibile, e risultando perciò una soluzione accettabile solamente per la navigazione in ambienti limitati come ad esempio una stanza. script di Teleporting sono presenti nell'SDK messo a disposizione da Valve (Open VR).

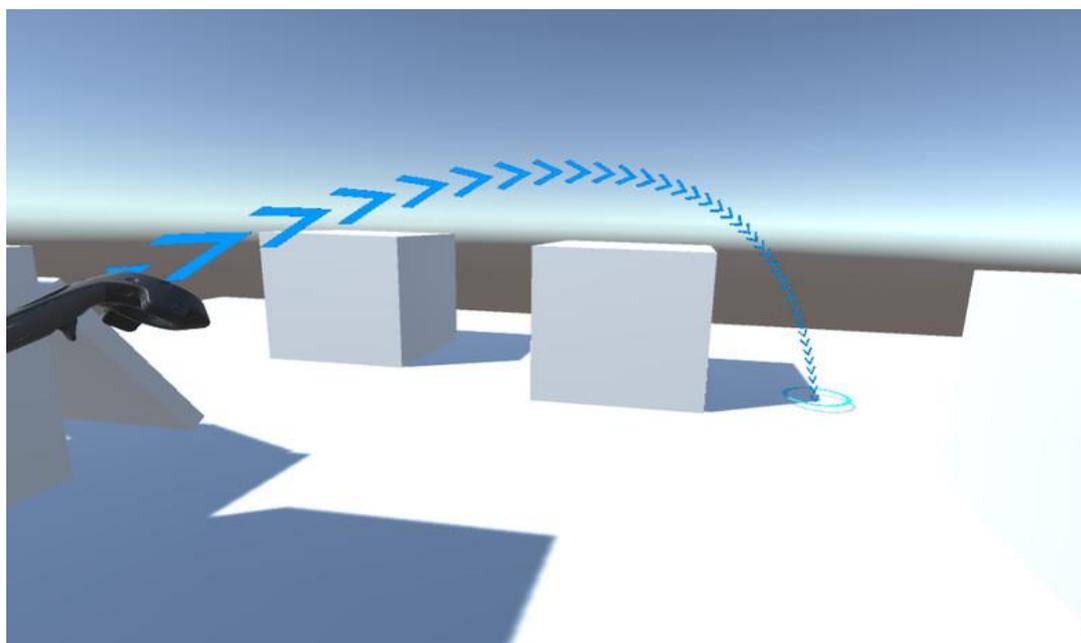


Figura 5.6: Esempio di Teleporting

Il *Room Scaling* è un paradigma VR che permette all'utente il movimento libero nella simulazione tramite la mappatura del movimento reale nell'ambiente virtuale. I sensori monitorano costantemente la posizione dell'utente nello spazio reale tridimensionale tramite triangolazioni, ed il sistema la traduce automaticamente nelle coordinate del mondo virtuale. Ciò consente all'utente di navigare il mondo virtuale esattamente come farebbe in quello reale.

Le esperienze di questo tipo richiedono però una grande quantità di spazio libero nel quale l'utente possa muoversi liberamente senza la paura di scontrarsi con ostacoli. Non è possibile quindi utilizzare questa tipologia di movimento in una casa o ufficio di dimensioni standard, risultando perciò poco pratica. Dal momento che gli sviluppatori non possono stimare lo spazio a disposizione dell'utente, non è neanche possibile scalare l'ambiente virtuale in funzione di quello reale.

È possibile anche far indossare sistemi di tracciamento ai polpacci dell'utente così che questi, simulando una camminata su posto a varie velocità, possa spostarsi nel mondo virtuale con intensità simili (*Foot Swinging*). Per quanto la soluzione sia realistica, il peso dell'attrezzatura oltre allo sforzo fisico e mentale la rendono molto faticosa da utilizzare per l'utente medio.

Per ultimo l'*Arm Swinging* (Figura 5.8), sviluppato da *Electric Night Owl* [55], risulta essere una soluzione con il giusto mix tra naturalezza e funzionalità. L'utente deve semplicemente oscillare le braccia, ruotandole come in una reale camminata, per navigare nella direzione verso cui le mani sono orientate, il tutto senza dover muovere gambe o testa. Il movimento risulta quindi sufficientemente naturale e comodo senza il fastidio di dover utilizzare ulteriori sensori.

Il tool è disponibile gratuitamente sullo store di Unity con una vasta e dettagliata documentazione.

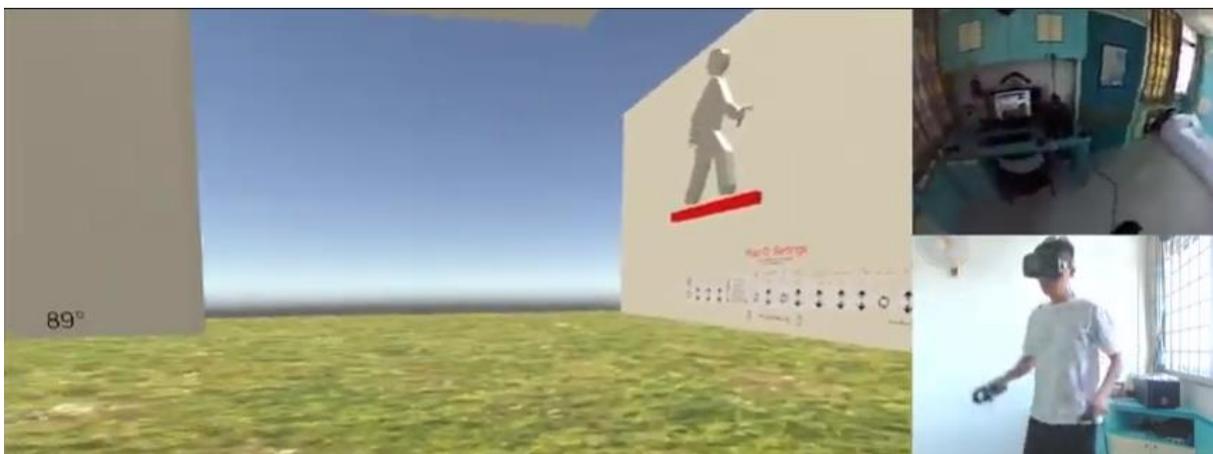


Figura 5.7: Esempio di Arm Swinging

Per quanto riguarda il progetto di tesi analizzato in questo elaborato, esso presenta un'ambientazione, per quanto limitata, molto estesa, richiedendo un metodo di locomozione efficiente e comodo da utilizzare per lungo tempo. Inoltre, la tipologia di simulazione creata richiede anche un metodo di locomozione che sia abbastanza realistico da non ridurre il senso d'immersione.

Tra i metodi descritti solamente l'ultimo, l'Arm Swinging, è risultato un compromesso accettabile tra realismo e comodità per la navigazione su un territorio esteso come quello implementato nella simulazione, escludendo perciò il Teleporting e il Room Scaling. Inoltre, a causa della durata della simulazione, il metodo deve risultare anche comodo da utilizzare per lungo tempo, in modo da evitare la stanchezza causata da metodi come il Foot Swinging.

Il sistema Arm Swinging utilizzato è stato implementato a partire dal sistema già messo a disposizione da Steam VR e modificato opportunamente per il caso in esame. L'Arm Swinging viene attivato tramite il movimento delle braccia e tenendo premuti i pulsanti Grip dei controller; il player, a questo punto si dirigerà verso la direzione indicata dall'orientamento delle mani mantenendo una costante altezza dal terreno e iniziando un'animazione che simula i movimenti della testa durante una camminata reale in prima persona sul player. L'intensità del movimento delle braccia viene tradotta nella velocità di camminata.

### 5.1.6 Line Renderer

Questo componente standard di Unity permette di disegnare linee 2D (in uno o più segmenti) nella scena tramite un array di punti rappresentanti i vertici per i quali deve passare la linea.

Il componente è assegnabile tramite *Inspector*, attraverso il quale è anche possibile gestirne i parametri (Figura 5.9); tra i più importanti si trovano:

- *Positions*: tabella nella quale si vanno ad inserire la posizione dei vertici della linea da disegnare;
- *Loop*: indica se la linea deve essere chiusa o meno;
- *Width*: tabella sulla quale è possibile gestire lo spessore della linea in ogni suo segmento;
- *Materials*: è possibile scegliere i materiali da assegnare alla linea;
- *Alignment*: tramite questo menù si può decidere se la linea deve essere statica e parallela all'asse delle Z oppure seguire l'orientamento della camera in modo che sia sempre rivolta alla vista dell'utente e dando così l'impressione di essere un oggetto 3D;
- *Use World Space*: indica se bisogna gestire il Line Renderer nel sistema di riferimento globale o locale;
- *Lighting*: per la gestione delle ombre.

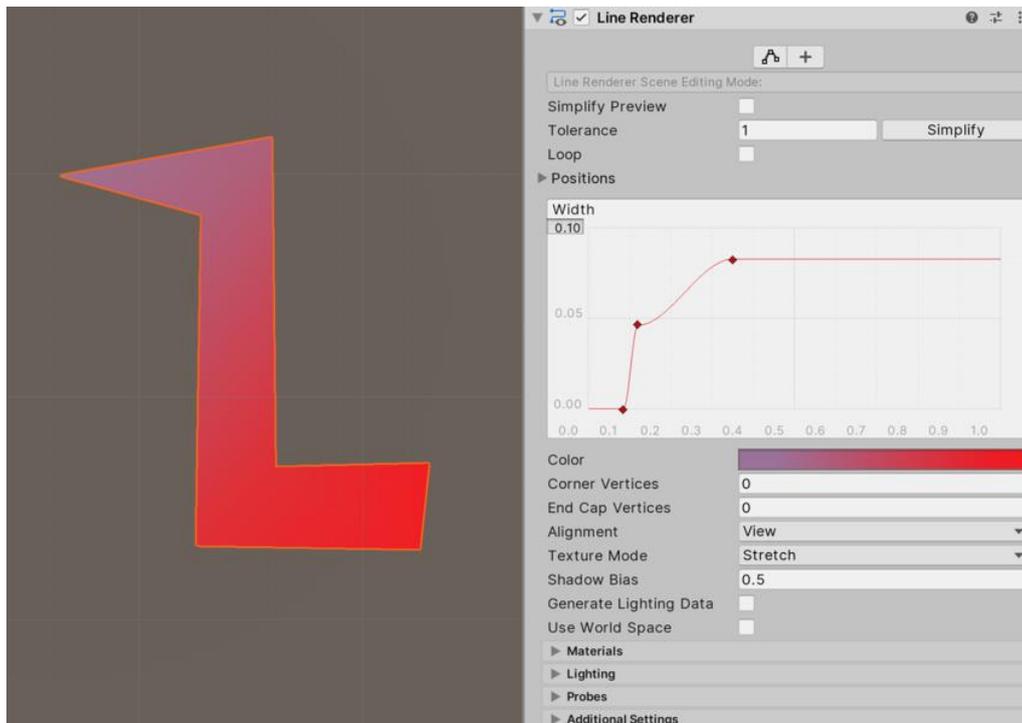


Figura 5.8: Esempio di Line Renderer

Il componente è stato utilizzato nella simulazione per la creazione delle corde e dei nastri, come ad esempio quello con cui occorre interagire per la recinzione della zona d'aspirazione o il fissaggio dell'ultima manichetta di mandata.

### 5.1.7 Implementazione

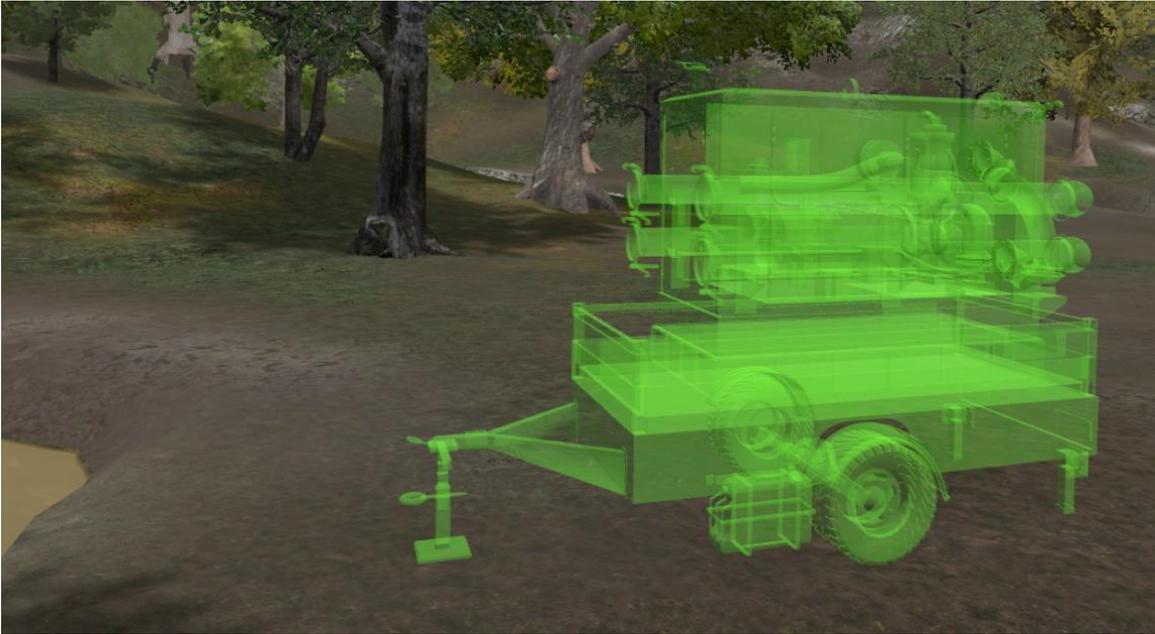
Verrà ora analizzato nel dettaglio come è stato gestito ogni elemento del sistema d'interazione tramite i tool e gli asset descritti in precedenza.

- *Hand*: alla base del sistema d'interazione vi è uno script denominato *Hand* assegnato alle mani del player. Lo script rileva la collisione con ogni oggetto interagibile, fa comparire un contorno giallo intorno allo stesso, detto *outline* (Figura 5.10), che va a sottolineare il suo stato e, identificando la tipologia di oggetto, permette di interagire con lo stesso in modo adeguato alla situazione e al contesto.



Figura 5.9: Outline implementato per l'interazione con gli oggetti

- *Grabbing/Dropping*: la raccolta degli oggetti è stata implementata tramite uno script chiamato appunto *Grabbing* e al click del pulsante Trigger. Lo script è stato assegnato a tutti gli oggetti raccogliibili e fa in modo che lo stesso diventi figlio della mano che sta cercando di raccogliere l'oggetto in modo che, durante il trasporto, questo possa seguire il suo proprietario. Il Grabbing finisce al rilascio del tasto Trigger che de-parenta l'oggetto lasciandolo cadere a terra (*Dropping*).
- *Porte*: interagendo con una porta, tenendo premuto il tasto Trigger e ruotando il braccio come se si stesse aprendo una porta reale attiva lo script Door che, assegnato alla stessa, instaura una rotazione sull'asse y tramite un'animazione.
- *Valvole e chiave*: l'apertura delle valvole e la rotazione della chiave d'accensione della pompa è semplicemente gestita tramite animazioni azionate dal click sul tasto Trigger;
- *Posizionamento pompa*: il posizionamento della pompa in una zona adeguata al suo stazionamento è stata implementata grazie alla combinazione di raycasting e rilevamento di collisioni. Puntando il controller destro verso terra viene attivato il sistema di raycasting, che genera nella zona collisa una sorta di sagoma semi-trasparente della pompa (Figura 5.11) che è possibile anche ruotare tramite il trackpad del controller. Non tutte le zone sono permesse: infatti sono stati posizionati dei Trigger Collider all'interno dell'ambiente che fungono da limiti. Normalmente la pompa "fantasma" è di colore verde ma, alla collisione con uno dei Collider limite, diventa di colore rosso a segnalare l'impossibilità di posizionamento. Infine, con il click sul pulsante Trigger la pompa viene posizionata correttamente nella zona scelta insieme all'attrezzatura (nella zona libera più vicina).



*Figura 5.10: Sagoma della pompa durante il suo posizionamento tramite Raycasting*

- *Allestimento zona di mandata*: la zona di mandata consta di tre elementi: il telo blu, i picchetti e la corda di fissaggio del tubo finale della catena. Il telo blu è disponibile al Grabbing nella zona vicina alla pompa e, una volta rilasciato nella zona desiderata (che possiede un Trigger Collider), si andrà a dispiegare automaticamente insieme ai picchetti. Questi ultimi, con il medesimo metodo, sono posizionabili negli anelli segnalati per il fissaggio sicuro del telo al terreno. Infine, dopo aver costruito la catena di mandata, bisogna assicurare l'ultima manichetta ai picchetti tramite una corda che, trattenuta con il Grabbing e passata nei picchetti ed ai lati del tubo, genererà una corda implementata tramite Line Renderer che andrà automaticamente a fissarsi agli oggetti indicati.
- *Delimitazione area operativa*: l'area deve essere delimitata tramite quattro picchetti e del nastro segnaletico teso tra gli stessi (Figura 5.12). I picchetti possono essere raccolti tramite Grabbing e posizionati nel terreno infilandoli direttamente nello stesso come se fossero paletti reali. Infatti questi, sentendo la collisione con il terreno e dopo una certa soglia di distanza tra il terreno e la punta, diverranno Kinematic e si forzerà il Dropping in posizione. Il nastro, in modo analogo alla corda, genera un Line Renderer ogniqualvolta venga rilevata la collisione tra lo stesso e uno dei paletti.



*Figura 5.11: Line Renderer utilizzato per la distensione del nastro*

Esistono inoltre altri script implementati per la gestione delle interazioni e dei suoi limiti come ad esempio uno script che limita la caduta degli oggetti attraverso il terreno (facendoli ricomparire sullo stesso).

## 5.2 Logica d'automazione degli NPC

Come descritto nel Capitolo 4, gli NPC sono una parte fondamentale della simulazione in quanto vanno a sostituire i membri della squadra che dovrebbero nella realtà collaborare allo svolgimento delle operazioni.

Infatti, operazioni come il trasporto o l'aggancio dei tubi, sono impossibili da portare a compimento senza l'aiuto di un secondo operatore in quanto l'attrezzatura del Modulo HCP risulta spesso troppo pesante, ingombrante e dal difficile utilizzo per una sola persona.

Per rendere la simulazione il più realistica possibile, gli NPC devono muoversi ed agire come farebbero reali operatori della PC andando quindi ad agire nella scena e ad interagire con la strumentazione in modo, per quanto più possibile, naturale.

Per questo motivo, animazioni 3D preimpostate per la gestione della cinematica delle mani, braccia e gambe sono risultate inefficienti per una simulazione nella quale un modello può prendere infinite potenziali posizioni, in quanto richiederebbero un lungo lavoro di disposizione degli arti specializzato per l'interazione con ogni elemento.

Il sistema inoltre non consta solamente di NPC predisposti all'aiutare l'utente, ma anche di un sistema di automazione nel quale gli agenti vanno ad eseguire quelle operazioni che in genere sono fatte da altri membri della squadra parallelamente al lavoro del singolo operatore.

Di seguito si andranno a descrivere i tool e gli asset utilizzati per la gestione di questa funzionalità, e come questi siano stati impiegati nella pratica.

### 5.2.1 Macchine a Stati Finiti

Come accennato in precedenza, gli NPC devono agire nel modo più verosimile possibile, imitando nelle azioni i reali operatori della PC. Perché ciò avvenga, non solo si richiede un'animazione realistica che vada ad imitare i movimenti reali, ma anche un'esecuzione autonoma.

Per questo motivo, si sono utilizzati i modelli 3D degli operatori per costruire un sistema di Intelligenza Artificiale o *Artificial Intelligence* (AI) in grado di controllare le operazioni di agenti automatizzati in grado di modificare il proprio comportamento in base alle azioni dell'utente o ai cambiamenti ambientali.

Al fine della realizzazione di un sistema di AI, sono state impiegate Macchine a Stati Finiti o *Finite State Machine* (FSM) dedicate al controllo dei flussi d'esecuzione, nei quali ogni nodo definisce lo stato temporaneo in cui si trova l'NPC nello svolgimento delle operazioni.

In una FSM, solamente uno stato per volta può essere attivo. Ciò vuol dire che un'azione completa si può comporre anche da diversi stati tra i quali il flusso transiterà costantemente a seconda dello stimolo o dal *trigger* ricevuto. I trigger possono essere di vario genere, come ad esempio una precisa azione dell'utente, un cambiamento ambientale o un evento temporale.

Le FSM sono comunemente utilizzate nell'informatica, nelle scienze e nell'ingegneria per la rappresentazione di flussi d'esecuzione di sistemi fisici o reali e il loro scopo è quindi quello di scomporre il comportamento complesso di un oggetto in una serie di stadi più semplici.

Una FSM è rappresentabile attraverso grafici costituita da nodi (cerchi) collegati da transizioni (archi) che rappresentano rispettivamente lo stato in cui si trova l'NPC e il trigger che causa il cambiamento di stato.

Nella Figura 5.13 è possibile osservare una semplice FSM che controlla il comportamento di una formica. Lo stato iniziale è *Find Leaf* (trova la foglia) che rimarrà attivo finché la foglia non verrà trovata passando a questo punto allo stato *Go Home* (vai a casa). Una volta arrivata a destinazione e quindi completata l'azione caratterizzante lo stato, la FSM tornerà su *Find Leaf* in uno stato ciclico infinito. Solo se ci si trova nello stato *Find Leaf* e si avvicina il puntatore del mouse alla formica si innesca un trigger che fa transitare la FSM allo stato *Run Away* (scappa) fino a quando questa non si è allontanata facendo così tornare la FSM nel suo stato normale. Poiché non esistono transizioni tra *Run Away* e *Go Home*, solo quando la formica si trova in quest'ultimo stato la si può catturare.

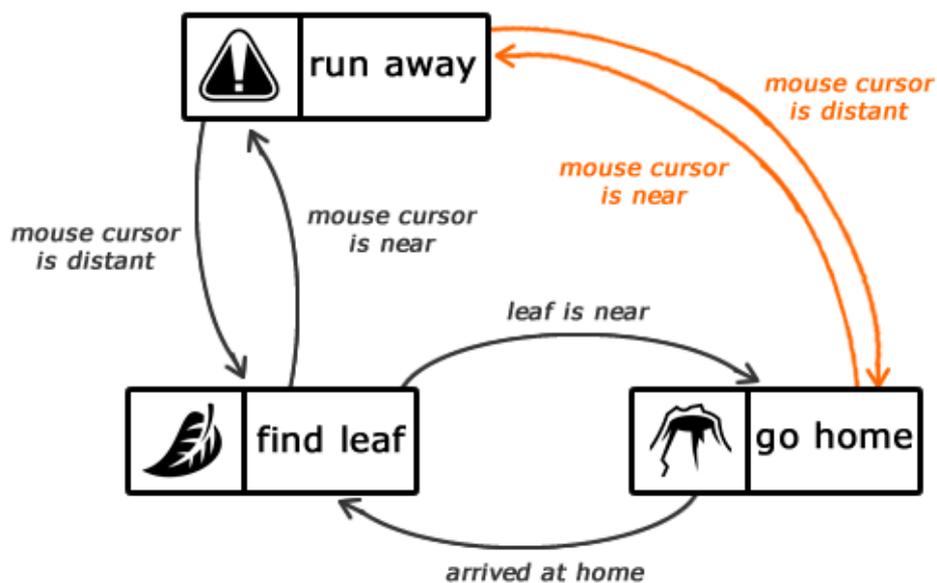


Figura 5.12: Esempio di FSM

Esistono due tipologie di FSM:

- macchine di Moore, in cui l'output dipende solo dallo stato corrente, e le azioni vengono eseguite quando ci si trova in uno stato;
- macchine di Mealey, in cui, diversamente dal caso precedente, l'output dipende sia dallo stato corrente che dall'input, e le azioni vengono eseguite quando avviene la transizione (ci saranno quindi un minor numero di stati).

Unity mette a disposizione più metodi per implementare FSM: tool appositi, animator e script.

I tool sono scaricabili sotto forma di asset attraverso lo store di Unity ma spesso si trattano di soluzioni quasi eccessivamente complicate.

*Animator* è una FSM messa a disposizione da *Mecanim*, il sistema d'animazione di Unity. In genere questa è utilizzata per la gestione delle animazioni assegnate ai nodi, però è possibile assegnare anche altre tipologie di elementi utilizzandola quindi come una vera e propria FSM AI che offre persino una comoda interfaccia rappresentante il grafico corrispondente alla FSM programmata.

In questo modo è possibile associare ai nodi animazioni o script che rappresentano l'azione da compiere nello stato in questione, ed alle transizioni la tipologia di trigger (bool, int, ecc.) associata al cambiamento di stato.

Ultima possibilità è quella di scrivere una FSM tramite codice su script sfruttando i costrutti di C# come Enum, Switch-Case e Coroutine. Per quanto questa opzione possa risultare scomoda da mantenere, complicata e lunga da implementare, è l'alternativa che offre più libertà.

Tutte le FSM implementate devono comunque essere controllate da uno script AI superiore, programmato appunto per gestire i trigger da inviare alle FSM.

Nel progetto di tesi si è utilizzato sia una FSM Animator, per la gestione delle animazioni di movimento (corsa, camminata), sia una implementata tramite script, nella quale le variazioni di stato dipendono dal ruolo assegnatogli durante la simulazione, sia che si tratti dell'aiutante del player sia che debba svolgere in autonomia le fasi che l'utente non ha eseguito. Tutte le FSM sono state programmate con lo scopo di progettare una struttura condivisa da tutti gli NPC presenti nella scena.

### 5.2.2 Animazione

Nella gestione degli agenti si è deciso di sviluppare due tipologie di animazioni: tradizionali e real-time.

Con tradizionale si intendono tutte quelle animazioni pianificate, implementate e salvate in modo che possano essere attivate in qualsiasi momento tramite un trigger, mentre con real-time le animazioni che devono essere implementate sul momento poiché le loro caratteristiche dipendono dal contesto.

Per quanto riguarda la prima tipologia di animazioni si è utilizzato il programma *Mixamo* (Capitolo 3) tramite il quale è stato possibile creare un *Rig* (scheletro artificiale) sul modello importato e applicarvi una serie di animazioni predefinite. Il tutto è stato poi importato su *Blender* per effettuare modifiche al Rig e alle animazioni che, essendo appunto predefinite, non erano state realizzate per lo specifico modello.

Il progetto *.blend*, importato su Unity, è stato quindi utilizzato per assegnare ogni animazione ad uno stato sul *Mecanim Animation System* nel quale è possibile costruire FSM con nodi (animazioni) e archi (trigger). Il sistema così implementato è visualizzabile e gestibile dal pannello *Animator* apposito.

Nella simulazione, questo approccio è stato utilizzato per gestire le animazioni di movimento come *Idle* (animazione utilizzata quando l’NPC è fermo), *Camminata* e *Corsa* (Figura 5.14).

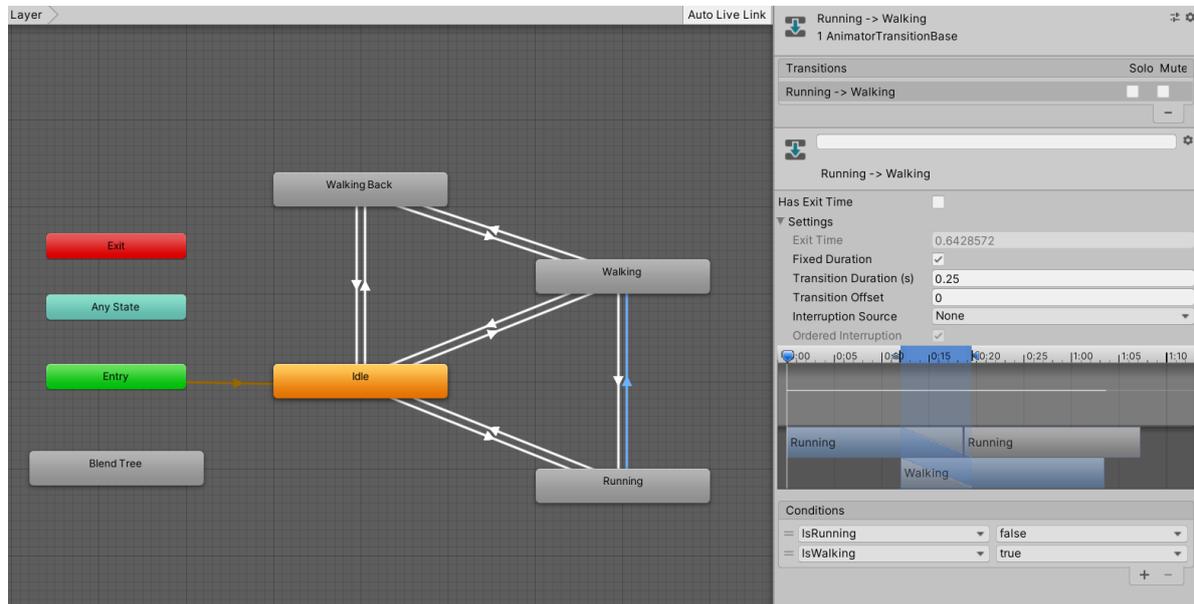


Figura 5.13: Interfaccia Animator

### 5.2.3 Cinematica inversa

Nell’implementazione delle animazioni real-time è stata utilizzata la cinematica inversa o *Inverse Kinematics* (IK).

Con l’IK è possibile manipolare una catena cinematica (nel caso specifico, il Rig del modello) fino al raggiungimento di una specifica posa solamente tramite il riposizionamento delle estremità finali.

Infatti, il modello 3D dell’NPC presenta un Rig composto da segmenti rigidi (detti ossa) collegati tramite giunti che vanno a mappare un realistico scheletro umano e che va a costituire la cosiddetta catena cinematica. Ogni osso può essere animato variando i cosiddetti angoli articolari (angolo formato da due ossa in corrispondenza di un giunto), nei limiti del movimento realistico, fino a raggiungere la posa desiderata, con una tecnica di cinematica diretta, o *Forward Kinematics* (FK). Il problema è che non esistendo una soluzione analitica al problema, risulta particolarmente difficile trovare la giusta combinazione di angoli

per eseguire l'animazione desiderata, in quanto, ad esempio per il semplice spostamento di una mano, bisogna andare a calcolarsi non solo l'angolo del polso ma anche quello del gomito e della spalla se si vuole che il braccio segua la mano.

Tramite l'IK è invece possibile andare a definire l'animazione di un osso solamente tramite l'impostazione della posizione iniziale e finale dello stesso poiché sarà infatti il sistema a calcolare gli angoli articolari che serviranno al raggiungimento della nuova posizione. In questo modo è semplicemente possibile disporre la figura nello spazio e spostare le sue parti nelle posizioni iniziali e finali desiderate per ottenere un'animazione completa.

In altre parole, con la FK l'animatore (o il programmatore) deve dire al sistema come effettuare il movimento per ogni componente che compone il modello, mentre con la IK deve semplicemente dire dove si vuole arrivare.

Nell'esempio riportato Figura 5.15, si può osservare come l'utilizzo dell'IK è stato utilizzato nel controllo di un braccio composto da 6 giunti e 5 segmenti. Il sistema, nel raggiungere il target con il giunto finale (*End-Effector*) della catena, effettua calcoli per determinare l'angolo di ogni giunto ed utilizza soluzioni numeriche per la posizione dei punti intermedi. È quindi possibile ottenere più di una configurazione per lo stesso *End-Effector*.

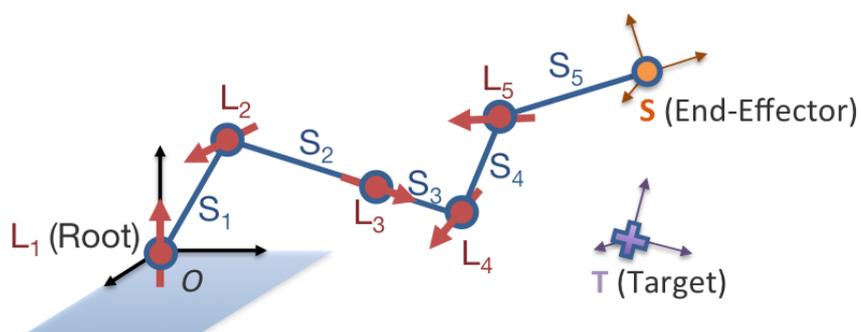


Figura 5.14: Esempio di catena cinematica

Anche la IK ha però i suoi difetti; infatti, visto che è il sistema a calcolare in automatico gli angoli articolari per il raggiungimento della posizione finale, è possibile ottenere animazioni poco realistiche causate dal movimento inverosimile dei giunti. Non è quindi possibile controllare ogni istante dell'animazione.

È possibile in questi casi procedere utilizzando metodi procedurali tenendo conto che gli angoli più lontani dall'End-Effector saranno quelli più influenzati mentre quelli più vicini possono essere utilizzati per regolazioni più fini. È utile inoltre definire un piano sul quale far giacere la spalla e il gomito del braccio.

Per quanto riguarda gli NPC utilizzati nella simulazione si è proceduto importandone il modello 3D su Unity e generando un Rig a partire dalla geometria. A questo punto il piano sopra descritto viene calcolato automaticamente da Unity tramite appositi tool.

Animazioni di tipo IK sono state realizzate tramite l'asset *Final IK*.

## 5.2.4 Final IK

Si tratta di un asset molto utilizzato dagli sviluppatori e disponibile sullo store di Unity [56]. Il tool consta di una libreria di script utilizzata nella gestione della IK e prodotta dalla *RootMotion*.

All'interno della cartella è possibile trovare script per il controllo delle catene cinematiche, modelli preimpostati e scene d'esempio in cui è possibile osservare tutte le soluzioni offerte da Final IK.

L'asset offre:

- *Full Body Biped IK*: solutore, o *solver*, leggero e potente in grado di modificare le animazioni procedurali in tempo reale e introdurre numerosi effetti d'animazione;
- *VR IK*: solver IK dedicato alla VR in grado di supportare fino a 6 punti di tracciamento (testa, mani, piedi, corpo);
- *Grounder*: sistema automatico di posizionamento verticale del personaggio e correzione dell'allineamento;
- *Sistema d'interazione*: progettato per creare in tempo reale interazioni procedurali;
- *Aim IK*: strumento finalizzato alla creazione di sistemi di mira per i personaggi;
- *Limiti di rotazione*: utilizzato per creare complesse strutture di vincolo per il *Rig*;
- *Algoritmi*: Final IK utilizza una vasta gamma di solver impieganti algoritmi diversi.

Nonostante la disponibilità di molti asset sviluppati con il medesimo scopo, si è scelto d'impiegare Final IK poiché molto utilizzato dagli sviluppatori e poiché presenta inoltre una vasta e particolareggiata documentazione.

Final IK è stato utilizzato nella gestione delle animazioni IK real-time, in modo da rendere più realistica l'interazione con gli oggetti e potersi concentrare sulla creazione di animazioni tradizionali solamente per il movimento (corsa e camminata).

Importato il modello dell'NPC su Unity, si è utilizzato Full body Biped IK per generare un'armatura estremamente flessibile e destinata esclusivamente alla realizzazione delle animazioni real time tramite script. Il componente offre inoltre un'interfaccia (Figura 5.16) per la gestione di ogni *effector* (giunzione) che possono essere di tre tipi: End-Effector (mani e piedi), Mid-body Effector (spalle e cosce) e Multi-effector (corpo). Tramite l'interfaccia è possibile anche assegnare target agli effector e il peso che ognuno di essi deve avere durante il compiersi del movimento.

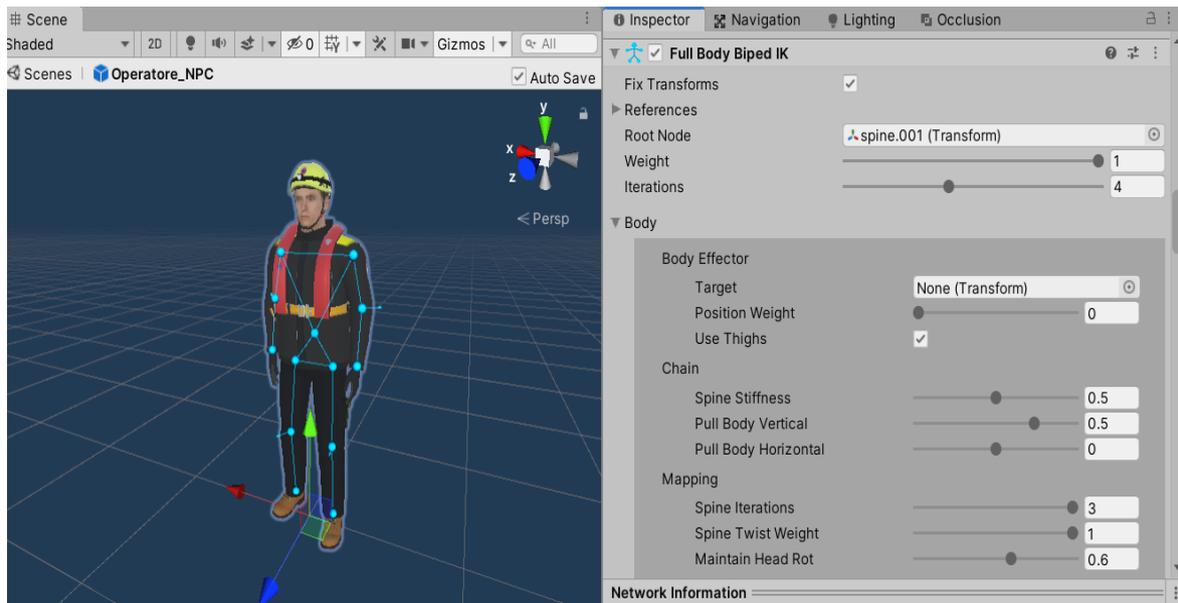


Figura 5.15: Componente Full Body Biped IK

Questo sistema è stato utilizzato in particolare con le mani dell’NPC alle quali si è assegnato un target riprodotto il loro esatto posizionamento durante il Grabbing. A questo scopo si è duplicata l’armatura posta sulle mani, posizionata correttamente su ogni oggetto interagibile (con l’aggiunta del componente *Interaction Object*), aggiunto il componente *Interaction Target* (che indica appunto il suo stato di target) e assegnata tramite script al Left Hand Effector o Right Hand Effector al momento opportuno (Figura 5.17).

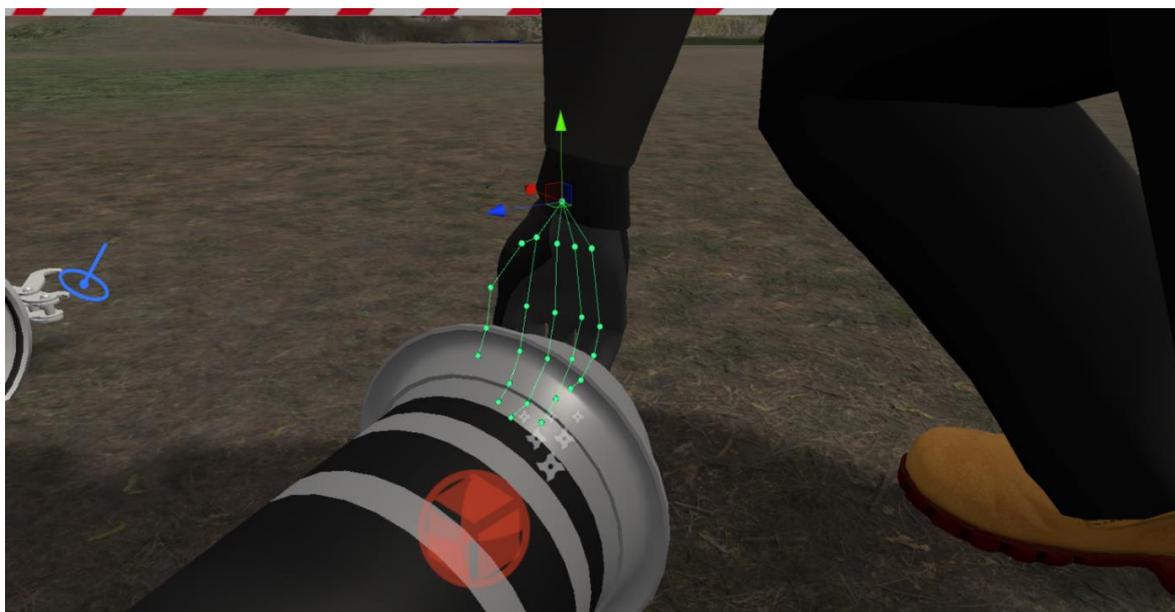


Figura 5.16: Target per Right End Effector

## 5.2.5 NavMesh

Unity mette a disposizione una funzione di navigazione o *Navigation* [57] integrata, la quale consente di creare personaggi in grado di spostarsi nella scena in modo intelligente.

Il sistema è costituito da:

- *NavMesh (Navigational Mesh)*: struttura dati generata automaticamente a partire dalla geometria del terreno che descrive le superfici percorribili dai giocatori, cioè quelle prive di ostacoli come pietre, alberi, depressioni (Figura 5.18); viene quindi utilizzata per calcolare il percorso più breve per raggiungere un punto sul terreno;
- *NavMesh Agent*: componente da assegnare ai personaggi che andranno poi a muoversi sul terreno in grado di renderli oggetti intelligenti capaci di calcolare i percorsi di movimento, eludere gli ostacoli ed evitare di scontrarsi gli uni con gli altri;
- *Off-Mesh Link*: componente in grado di creare percorsi comprendenti zone non coperte dalla Navmesh, ad esempio per saltare un fossato o una recinzione;
- *NavMesh Obstacle*: componente da assegnare agli oggetti in movimento nella simulazione (e quindi non calcolati dalla NavMesh) che gli agenti dovrebbero evitare.

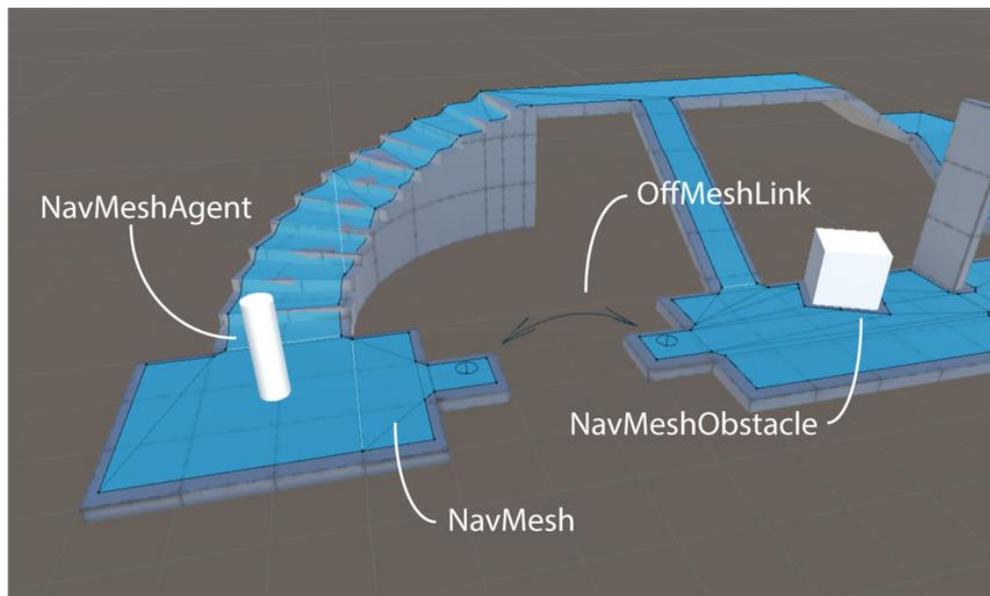


Figura 5.17: Esempio di NavMesh

Il sistema di navigazione è stato quindi utilizzato nella gestione del movimento degli NPC, in modo che questi potessero muoversi autonomamente nello scenario modellato rilevando i tragitti percorribili privi di ostacoli (fiumi, montagne, rocce, alberi).

Dapprima si sono impostate le caratteristiche dell’NPC quali altezza, raggio e parametri di salto e successivamente quelli della NavMesh, come ad esempio la risoluzione e l’accuratezza. A questo punto è stato possibile procedere con il *Baking* (generazione) della

NavMesh e ad aggiungere il componente NavMesh Agent al modello (Figura 5.19). Inoltre a tutti gli oggetti dinamici della scena (pompa, altri NPC, ecc.), non potendo essere inseriti nella mappa, è stato assegnato il componente NavMesh Obstacle.

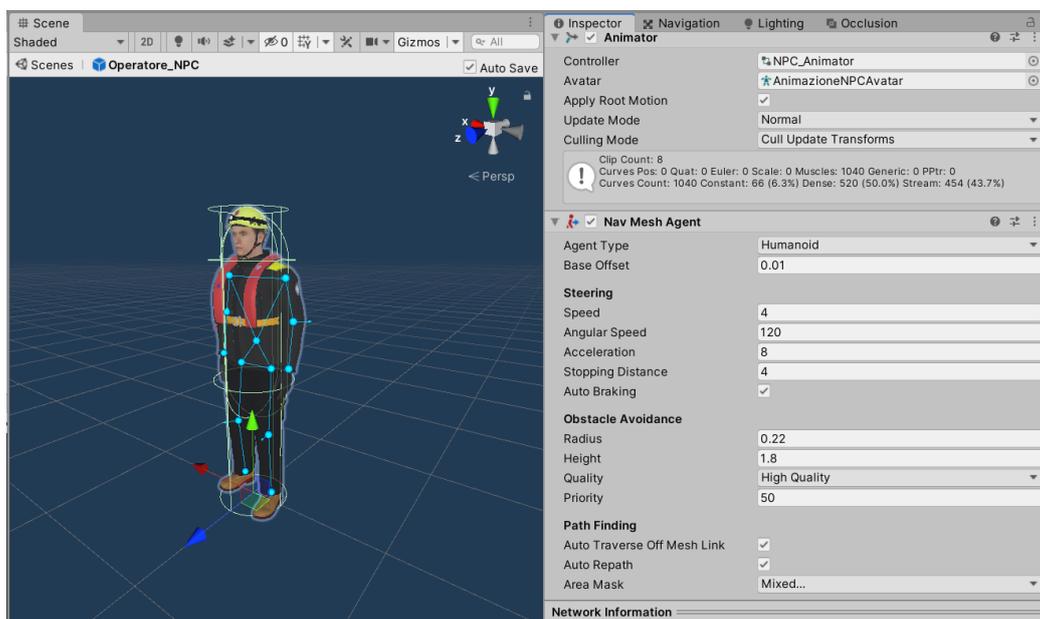


Figura 5.18: Componente NavMesh Agent

## 5.2.6 Implementazione

Verrà ora illustrato nel dettaglio come ogni tool è stato gestito e come è stato impostato ogni aspetto del sistema.

L'automazione è costituita dalla presenza di due agenti che operano in modo coordinato; il primo, chiamato *Master*, è il "capo" e colui che compie quelle operazioni che possono essere eseguite in solitaria (come ad esempio l'accensione della pompa), il secondo, denominato *Slave*, è quello che segue le direttive del Master aiutandolo nelle operazioni che non può compiere da solo come il sollevamento e il trasporto dei tubi.

Ogni operatore automatizzato è controllato da una serie di script appositi che vanno a gestire ogni componente assegnato. Lo script principale contiene una FSM che definisce ogni stato in cui l'NPC può trovarsi durante la simulazione.

Gli stati sono quelli illustrati di seguito.

- *Idle*: stato nel quale l'NPC rimane fermo ed animato solamente da una routine apposita che fa muovere la testa e il corpo in modo casuale per evitare un'immobilità innaturale.
- *Grab*: stato attivato quando l'NPC deve sollevare una manichetta per andare a costruire la catena di tubi di mandata o d'aspirazione. Attivando il Grab, l'NPC corre

(azionando quindi l'animazione di corsa) verso il tubo da raccogliere, si china in modo che la mano possa raggiungere il target posto sul capo (tramite Final IK) e risollevarsi. Sia il Master (per la femmina) che lo Slave (per il maschio) attendono che la controparte sia sul posto prima di procedere (Figura 5.20).



*Figura 5.19: Stato di Grab*

- *Drop*: In questa fase, attivata a seguito di un Grab, l'NPC si reca camminando (animazione di camminata) verso il punto nel quale il tubo deve essere agganciato, il Master posa il tubo attaccandolo al maschio precedente e attende che lo Slave, che a sua volta stava attendendo che il Master finisse il Drop, posi la femmina a terra prima di lasciar andare il capo del tutto.
- *Carry*: stato del solo Slave che viene attivato durante il trasporto del tubo. Lo stato fa in modo che lo Slave segua il Master tenendo sempre una specifica distanza e rotazione massima (Figura 5.21).
- *Follow*: stato implementato solamente per l'NPC Slave. L'agente segue il Master quando la distanza tra di loro supera una certa soglia.



Figura 5.20: Stato di Carry

- *Go-To-Point*: passando all’NPC la posizione di un punto dello scenario, l’NPC si affretta a raggiungerlo per poi tornare in stato Idle.
- *Pump*: stato dell’NPC che si occupa della gestione dell’apertura delle porte, dell’accensione e dello spegnimento della pompa (gestito tramite Final IK). Si tratta di uno stato assegnabile unicamente al Master.
- *Puddle-Inspection*: stato del Master nel quale l’NPC si reca presso la pozza e, nel caso sia vuota, alza il braccio (Final IK) per segnalarlo all’operatore addetto alla pompa che procederà con il suo spegnimento.

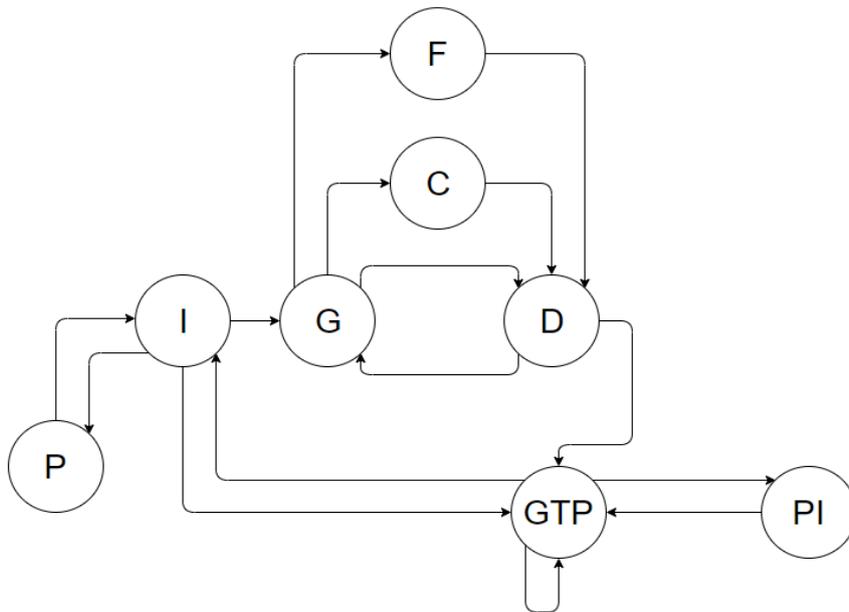


Figura 5.21: FSM implementata

Per gestire il componente Final IK è stato assegnato all’NPC uno script apposito che, attraverso una serie di Switch-Case, permette di variare i parametri del componente a seconda del gesto eseguito tramite l’IK. Ad esempio, il movimento che l’NPC compie durante il Grab di un oggetto, è diverso da quello usato per sollevare la mano, e i parametri aiutano a rendere il movimento più naturale (come la rotazione della spalla o il piegamento delle ginocchia).

Ogni NPC ha poi uno script denominato *MovementController* che gestisce i movimenti compiuti dall’NPC: velocità di spostamento, velocità angolare, tipologia d’animazione da attivare, ecc. Per questo motivo sono stati implementati diversi metodi, uno per ogni tipologia di movimento, nei quali si sono impostati i valori che ogni parametro deve avere per quello specifico caso. Inoltre, prima di iniziare qualsiasi movimento, lo script va a calcolare il percorso da utilizzare per raggiungere il punto indicato tramite la Navmesh del terreno.

A ciascun capo di ogni manichetta è stato inoltre assegnato uno script chiamato *RelativePosition* che va a gestire tutti i parametri che devono essere variati durante l’azione di Grab e di Drop. Lo script si trova unicamente sulle manichette poiché sono gli oggetti più difficili da gestire ed è maggiormente importante che la loro coordinazione sia precisa in quanto elemento fondamentale alla simulazione.

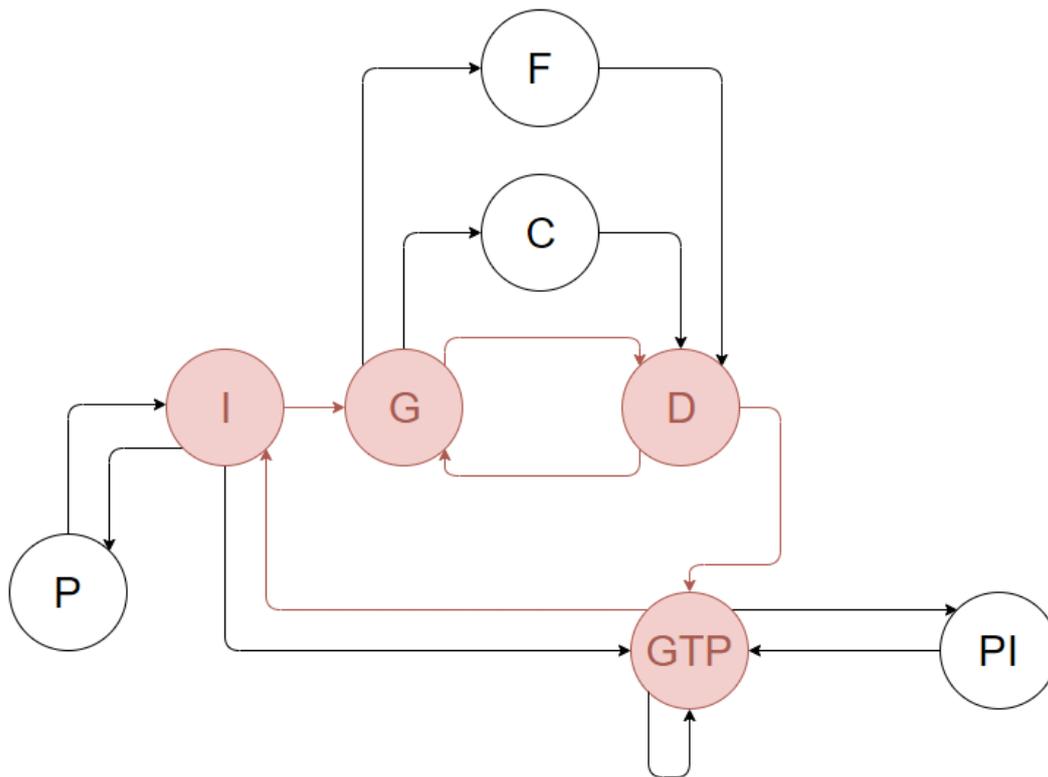
Tutti questi script e in particolare la FSM sono poi gestiti da una struttura esterna chiamata *Automation Controller*, che coordina i trigger che vanno poi a provocare le transizioni di stato sulla FSM stessa.

Ogni metodo dell’Automation Controller va a gestire una particolare routine, tutte combacianti con le fasi e le operazioni che il sistema d’automazione degli agenti deve gestire.

Nel seguito viene descritta nel dettaglio ogni fase automatica.

- Aspirazione:

quando viene attivata l’aspirazione, il Master dallo stato Idle passa al Grab. L’NPC corre alla prima manichetta assegnatagli, aspetta che anche lo Slave raggiunga il target e raccoglie il capo transitando allo stato Drop. In questo stato, il Master raggiunge il target in cui il tubo deve essere depositato, aspetta che anche lo Slave abbia concluso il movimento, posa il capo e, dopo che anche lo Slave ha completato il Drop, torna nello stato Grab. Il processo continua così in modo ciclico, fino a quando la catena è conclusa con il Grab e il Drop del filtro. A questo punto si attiva lo stato Go-to-Point, per far tornare l’NPC vicino alla pompa e tornare in stato Idle.



*Figura 5.22: Aspirazione Master*

Lo Slave segue lo stesso tipo di processo con l'eccezione che, durante il trasporto delle manichette, l'agente passa nello stato Carry che garantisce che l'NPC segua il Master mantenendo sempre una certa distanza e rotazione. Solo quando il Master ha concluso la posa del suo capo, lo Slave passa in stato Drop posando il maschio. Finito transita nuovamente su Grab. Giunti al Grab del filtro da parte del Master, lo Slave, non essendo d'aiuto, attiva il Go-To-Point e successivamente l'Idle.

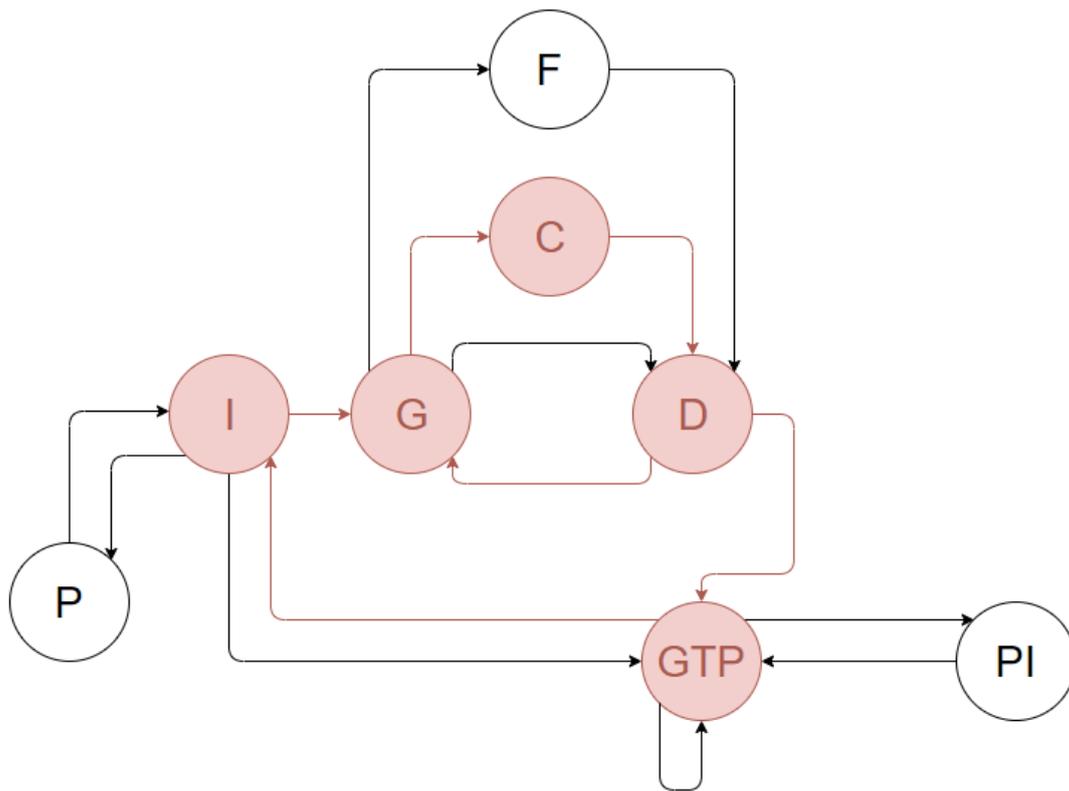


Figura 5.23: Aspirazione Slave

- Mandata:

anche per la costruzione della mandata il Master segue lo stesso tipo di processo. Lo Slave si differenzia nella parte di trasporto del tubo transitando in questo caso su Follow. Infatti, non dovendo più mantenere rigidamente la stessa distanza e soprattutto la stessa rotazione, grazie alla maggiore lunghezza della manichetta, non è necessario utilizzare uno stato così limitante come è il Carry. Anche il Drop è diverso dal caso precedente, in quanto non basta solo posare il capo a terra ma anche distendere la manichetta verso il punto di mandata.

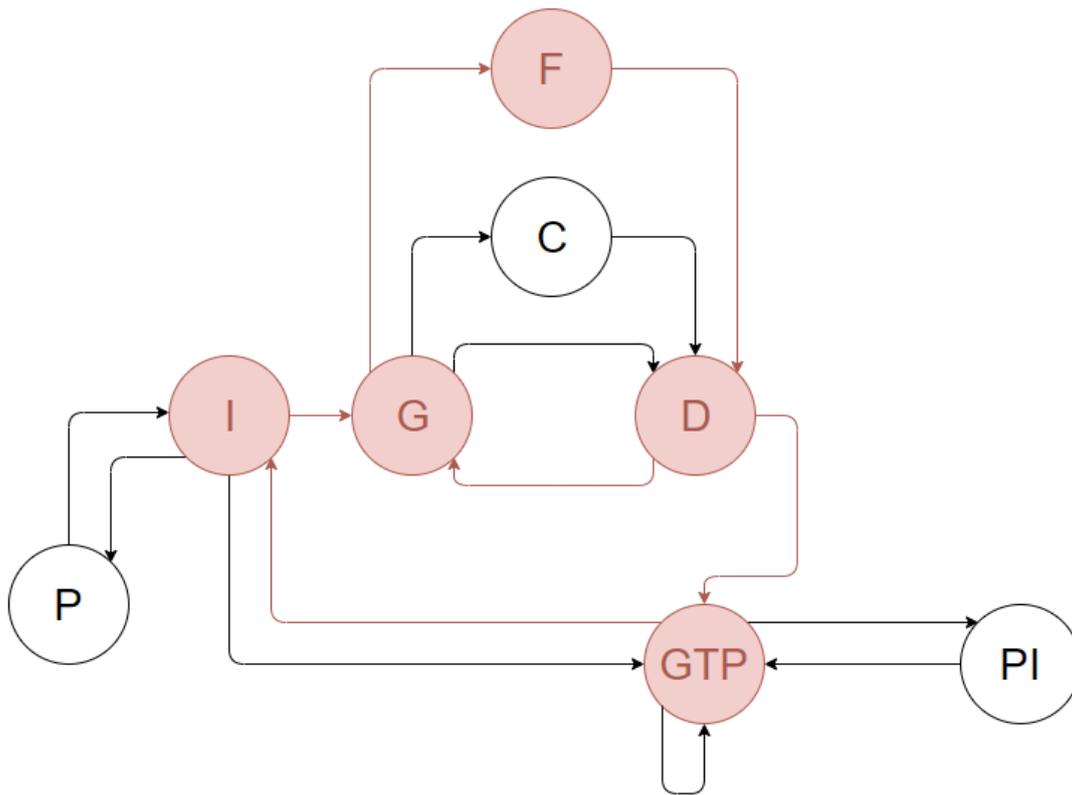


Figura 5.24: Mandata Slave

- Controllo perdite:

in questa fase l’NPC deve correre lungo la catena di mandata per controllare che non vi siano perdite. A questo scopo, si attiva lo stato Go-To-Point assegnandogli come target la prima manichetta della catena e, una volta raggiunto il punto, si riattiva lo stesso stato passandogli la seconda manichetta e così via. Una volta che la pozza è vuota, si passa allo stato Puddle-Inspection che induce l’NPC ad avvicinarsi alla pompa e ad alzare il braccio per segnalare la possibilità di spegnimento. Spenta la pompa, si transita su Go-To-Point, tramite il quale l’agente avvicina ai mezzi, per passare poi in Idle.

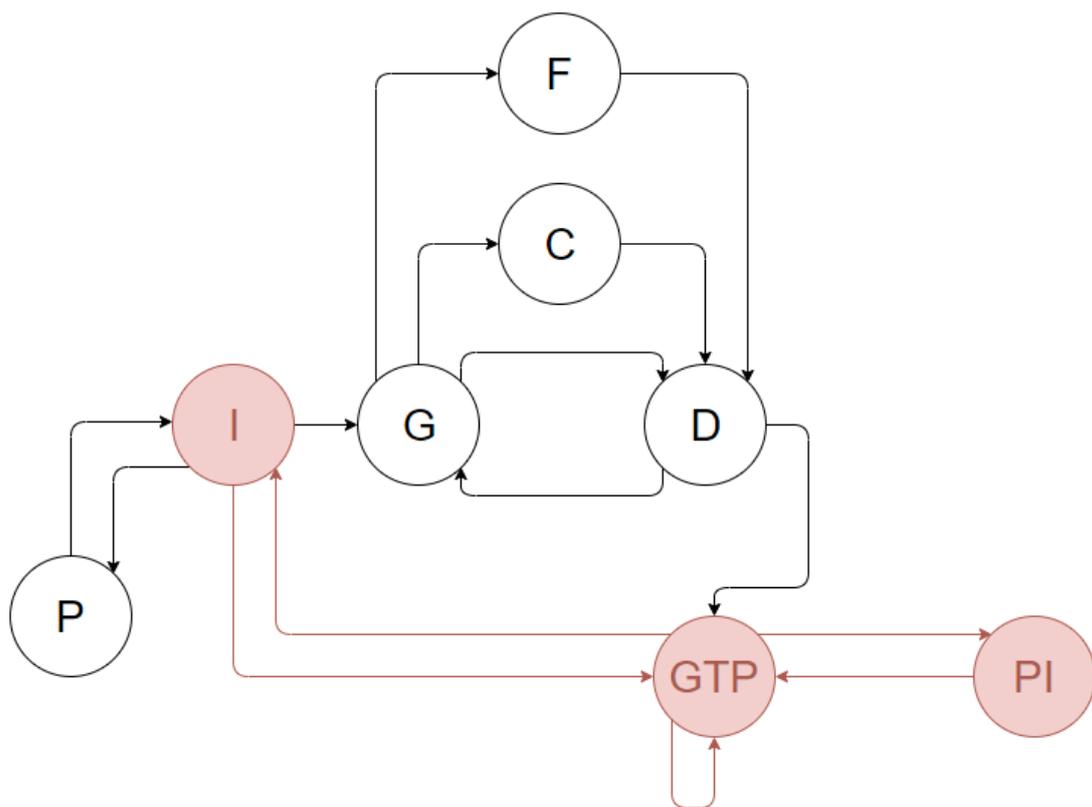


Figura 5.25: Controllo perdite

- Gestione del quadro di comando:

passando dallo stato Idle a Pump, l’NPC si reca vicino alla pompa, apre la porta del pannello e gira la chiave d’accensione. A questo punto transita allo stato Idle restando vicino al pannello ad attendere il segnale di spegnimento. Quando il player dà il segnale (alzando il braccio e premendo il trigger), si passa di nuovo allo stato Pump e l’NPC spegne la pompa transitando poi ad Idle.

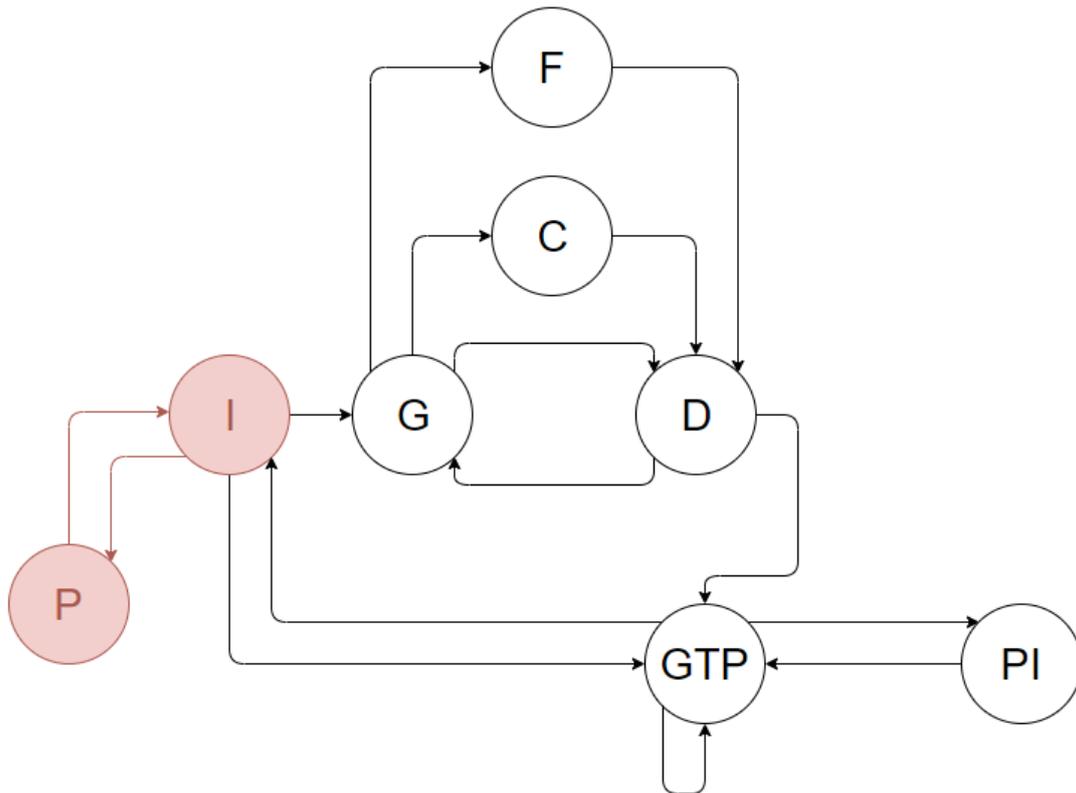


Figura 5.26: Gestione del quadro di comando

Occorre infine sottolineare che quanto sopra descritto è stato implementato tenendo conto dell'ambiente 3D (nel quale si sono posti dei limiti di movimento e di cui si è calcolata la NavMesh) e del sistema VR (per il quale sono stati creati script appositi per la gestione delle interferenze del giocatore).

### 5.3 Strategia di valutazione

Per quanto riguarda la modalità valutata, l'utente ha completa libertà di movimento nel cercare di riprodurre le corrette operazioni insegnategli durante la modalità didattica poiché, alla fine della sessione, il suo operato sarà valutato. Per questo motivo, è stato implementato un sistema in grado di verificare che le azioni dell'utente non violino i parametri di valutazione concordati con la PC.

Si descrivono ora nel dettaglio le soluzioni impiegate per la valutazione:

- Sopralluogo:

L'utente deve avvicinarsi alle zone d'aspirazione e mandata (si calcola la distanza tra il player e i target) ed indossare i DPI (casco e giubbotto di salvataggio).

- Posizionamento pompa:

La pompa deve essere posta ad una distanza massima di 8m dalla pozza (si calcola la distanza tra le due) e avere le bocchette ruotate verso la pozza (la pompa deve avere una rotazione all'interno di un intervallo stabilito).

- Delimitazione area operativa:

L'area in cui è stata posizionata la pompa e l'attrezzatura deve essere delimitata correttamente (utilizzo di tutti i paletti e del nastro), in modo che questa non sia né troppo grande né troppo piccola (si calcola la distanza tra gli estremi della zona e i paletti, la distanza reciproca dei paletti e che i paletti non si trovino all'interno della zona) e che il nastro sia collegato in modo adeguato (si devono collegare tutti i paletti, il nastro non deve incrociarsi, il lato aperto deve essere quello rivolto verso la pozza).

- Posizionamento telo blu:

Il telo deve essere posizionato in una zona libera da ostacoli (tre opzioni) e fissato con tutti e quattro i picchetti.

- Aspirazione:

La catena d'aspirazione deve essere costituita esclusivamente da manichette semirigide e terminata con il filtro da posizionare in acqua (sono state create delle liste che tengono conto di ogni manichetta utilizzata per comporre la catena). Il posizionamento del filtro può avvenire solamente in seguito all'assicurazione con il cavo apposito (booleani).

- Mandata:

La catena di mandata deve essere costituita inizialmente da manichette semirigide o tubi rigidi, proseguire con manichette appiattibili, e terminare con un'ulteriore manichetta semirigida da posizionare sul telo blu (anche in questo caso esistono liste che tengono conto delle manichette della catena). Il maschio dell'ultima manichetta deve essere assicurato ai picchetti nel modo corretto (la corda non deve incrociarsi e deve essere collegata ai picchetti in modo corretto).

- Controllo manichette:

Durante l'accensione occorre controllare che lungo la catena di mandata non vi siano perdite e monitorare la pozza fino al suo completo drenaggio (calcolo delle distanze). In questo caso si deve procedere a segnalarlo all'operatore addetto al pannello, e solo dopo lo spegnimento, procedere all'apertura delle valvole (booleani). Durante il funzionamento nessuna manichetta può essere scollegata (liste).

- Accensione pompa:

Prima dell'accensione tutte le manichette, le valvole e le porte devono essere chiuse (gestito tramite liste e booleani). Una volta avviata la pompa, questa non può essere spenta prima di aver ricevuto il segnale. Una volta spenta le valvole devono essere aperte.

I risultati vengono registrati e salvati sul *GameManager*, oggetto istanziato all'apertura della simulazione e distrutto solamente alla sua chiusura. L'oggetto tiene traccia della configurazione di simulazione scelta dall'utente, è responsabile del passaggio da una scena all'altra e memorizza tutte le informazioni generate durante la simulazione, tra le quali, appunto, i risultati della valutazione.

Al termine della simulazione l'applicazione torna al menù iniziale, sul quale viene visualizzato un pannello contenente tutti gli errori commessi.

## 5.4 Altri sistemi

Vengono ora descritti più in generale altri elementi degni di nota implementati per la simulazione.

### 5.4.1 Ambiente

Come detto in precedenza, si è deciso di ambientare la simulazione in una radura situata in una valle racchiusa da montagne, vicino alla quale un fiume in piena ha formato una pozza d'acqua. Per fare in modo che il territorio risultasse soddisfacentemente realistico, si sono utilizzati una serie di tool e asset, riportati di seguito.

- **Terrain:** tool incorporato a Unity [58] che permette di creare e poi modificare uno scenario naturale. Il tool crea un ambiente tridimensionale sul quale il modellatore può andare a compiere un'operazione di *sculpting* utilizzando il brush, messo a disposizione dal componente, per modellarlo a proprio piacimento creando rilievi e depressioni. Il tool permette inoltre di variare la forma, la dimensione, l'intensità, la rotazione e la spaziatura del brush così da poter creare la forma desiderata (Figura 5.28). Tramite il brush è anche possibile disporre una o più texture sovrapposte sul terreno.
- **Cascade:** l'asset [59] è stato utilizzato per la creazione del fiume, del canale e della pozza tramite una struttura a nodi orientabili; i nodi fungono da puntatori che permettono di modificare le proprietà fisiche del piano d'acqua. *Cascade* permette anche di modificare le proprietà dello shader applicato per simulare l'acqua come ad esempio la rifrazione, la luminosità, la frequenza e l'intensità delle onde.
- **Particellari:** Unity mette a disposizione un componente per la gestione dei sistemi particellari. Tramite Inspector è possibile variare una serie di parametri tra i quali numero di particelle, velocità di movimento, frequenza e velocità di generazione.

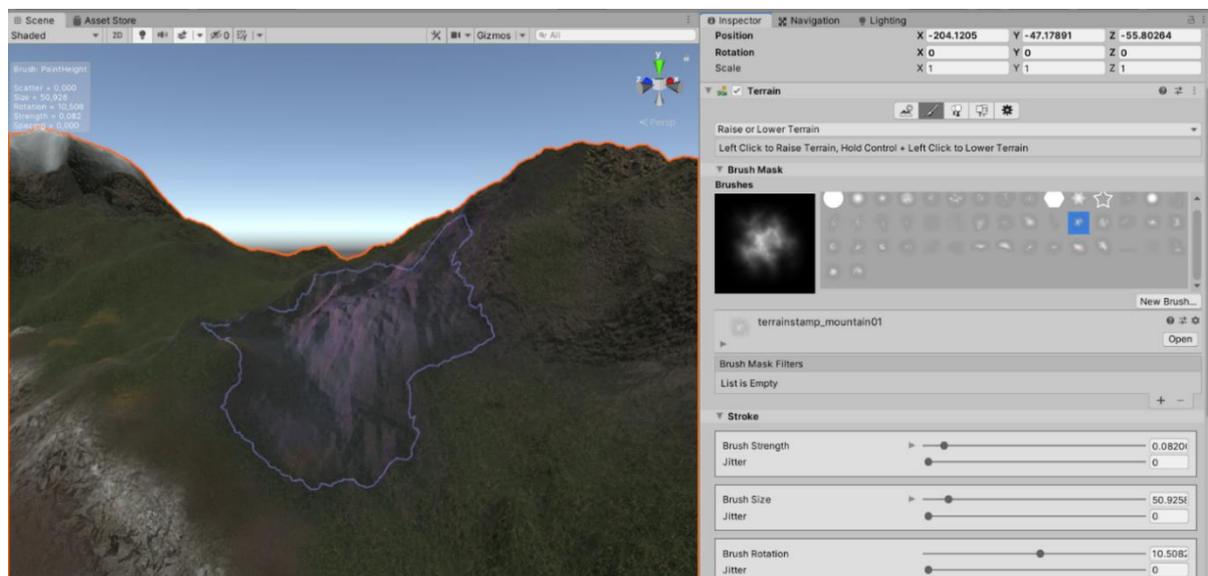


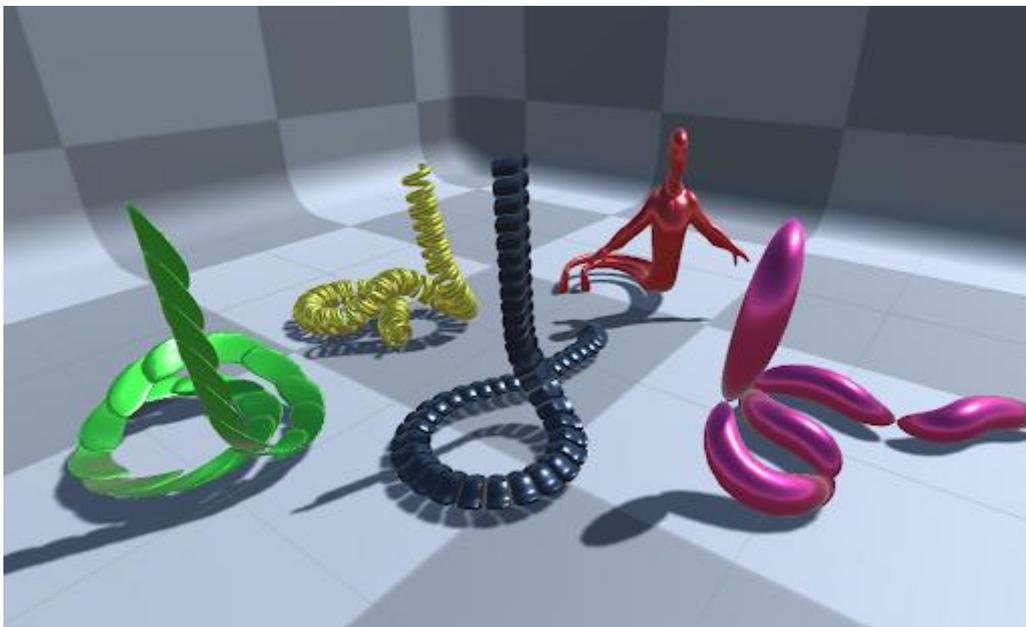
Figura 5.27: Componente Terrain

## 5.4.2 Gestione delle manichette

*Obi* [60] è una famiglia di plugin fisici basati su particelle per Unity che fornisce un motore di simulazione fisica in tempo reale CPU-based per realizzare tessuti, corde, catene, fluidi e *SoftBody*. La famiglia si scompone poi in singoli asset (*Obi Cloth*, *Obi Rope*, *Obi Fluid* e *Obi Softbody*) che sfruttano lo stesso solver fisico di base e ognuno di questi oggetti è realizzato mediante particelle che si relazionano tra di loro mediante vincoli. L'*Obi Solver* (che si occupa

di far convergere le equazioni fisiche che influenzano le particelle) è *multi-threaded* e *SIMD accelerated*, e cerca di essere il più esaustivo possibile sulle sue funzionalità senza nascondere all'utente i parametri che governano il suo comportamento. I singoli oggetti gestiti dal solver vengono invece chiamati *Obi Actor* e partendo da un *blueprint* che contiene informazioni di base (particelle e vincoli) le istanzia poi nell'ambiente in modo che il solver possa simularne il comportamento mediante riferimento.

All'interno del progetto di tesi si è impiegato l'asset *Obi Rope* (Figura 5.29), in particolare la versione 4.3, normalmente utilizzato per simulare corde e cavi, in questo caso invece adoperato per rendere corretto il comportamento fisico delle manichette necessarie a costruire le catene di aspirazione e mandata. Per implementarne la funzionalità innanzitutto si è portato nell'ambiente il solver, che presenta vari parametri, tra cui il massimo numero di particelle simulate, il tempo di aggiornamento delle posizioni delle particelle, la gravità e la soglia di energia cinetica sotto la quale ignorare la modifica allo stato degli attori. Sempre nel solver vi è anche una lista di quali "regole" far valere tra le particelle, come la distanza, l'angolo, la frizione, la densità (nel caso dei fluidi) e i vincoli di fissaggio.



*Figura 5.28: Esempi di utilizzo di Obi Rope*

Come Obi Actor si è invece utilizzato Obi Rope, che costruisce una catena di particelle secondo una curva spline chiamata Obi Curve, che ha punti di controllo che possono essere manipolati tramite editor. Una volta fornita la curva che la manichetta dovrà seguire a riposo all'Actor, si decide la sezione della corda, ovvero la forma che verrà estrusa lungo questa curva e che costituisce il profilo della manichetta, e si provvede a fornire un riferimento al solver. Una volta fatto ciò, si istanzia l'Actor che quindi crea le particelle e le dispone nell'ambiente secondo il grado di risoluzione voluto (più la risoluzione è alta, più particelle

vengono impiegate e più accurata è la simulazione). Il singolo Actor è stato poi modificato ulteriormente nel suo comportamento mediante la configurazione di parametri delle singole particelle come raggio e massa, oppure di vincoli di distanza, di vincoli di piegamento e di fissaggio. Per questi ultimi si sono realizzati due modelli 3D delle estremità del tubo e per renderli interagibili con Obi Rope sono stati aggiunti componenti come Obi Collider e Obi RigidBody, che permettono al solver di tenere conto anche di quegli oggetti all'interno della simulazione.

La logica principale dell'interazione tra le manichette si è sviluppata poi mediante script sulle diverse estremità, ovvero l'apertura femmina e maschio, dove la femmina presenta un meccanismo di chiusura per vincolare le due parti assieme durante il funzionamento della pompa. Anche questa procedura di bloccaggio è stata quindi realizzata da zero, e va ripetuta per ogni nuova manichetta aggiunta ad una delle catene. Le catene vengono poi salvate mediante un *Manager* che controlla il tipo di manichette impiegate, se sono state correttamente bloccate, a quale catena appartengono e in che ordine si susseguono. Sono poi stati creati altri modelli 3D per simulare oggetti come tubi e filtri per l'aspirazione, che usano gli stessi script di interazione poiché le loro estremità sono le stesse delle manichette ma senza la necessità di essere vincolati a istanze di Obi Rope per simularne il comportamento fisico.

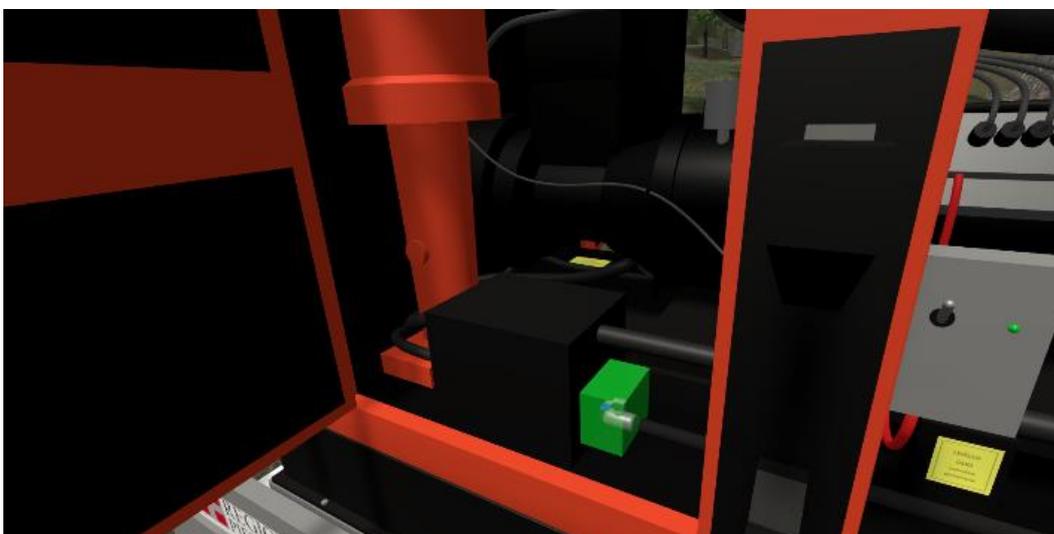
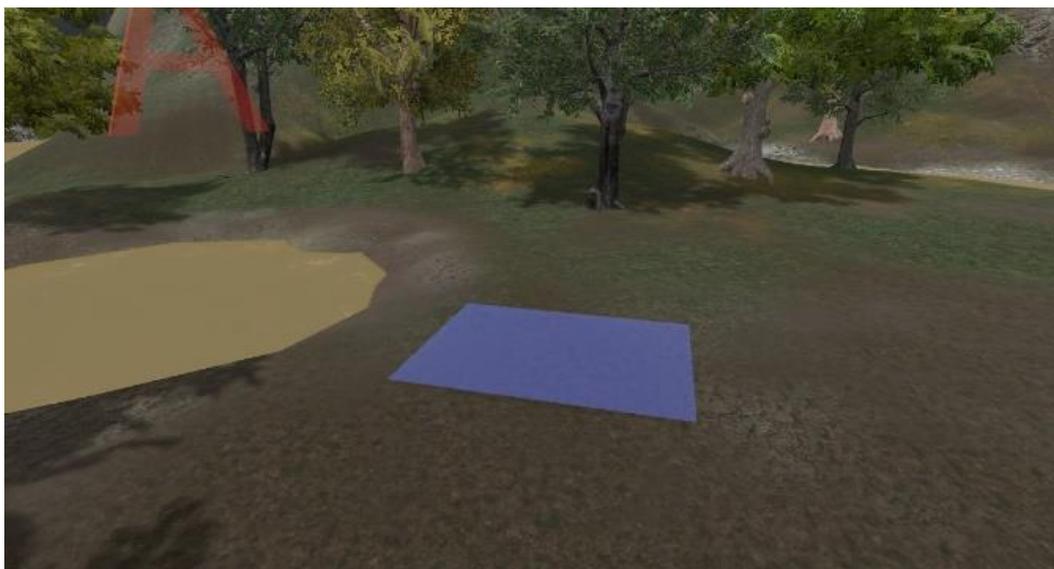
### 5.4.3 Voice-over

Come detto in precedenza, è stato necessario sviluppare un sistema di voice-over per comunicare all'utente le procedure e le istruzioni necessarie al completamento del compito.

In effetti, durante la simulazione l'utente è sottoposto a più stimoli sensoriali:

- *icone e highliner* colorati permettono di indicare ed evidenziare gli oggetti nella scena (Figura 5.30);
- *suoni binaurali* utilizzati per simulare i rumori naturali e generati dall'attrezzatura, utilizzati per aumentare l'immersione nella simulazione;
- *clip audio* riferiscono le istruzioni per l'esecuzione delle operazioni all'utente, simulando la voce di un fittizio caposquadra trasmessa attraverso una radio in dotazione.

Il voice-over è stato gestito tramite *AudioSource* (sorgente sonora che va a riprodurre le clip) e *AudioListener* (in grado di gestire l'audio 3D).



*Figura 5.29: Esempi di icone e highliner implementati*

# Capitolo 6

## Conclusioni e sviluppi futuri

Nel presente capitolo verranno analizzati i punti di forza, le debolezze e i possibili miglioramenti futuri attuabili nel lavoro svolto fino a questo punto e descritti in questo lavoro di tesi.

### 6.1 Conclusioni

Il lavoro di tesi ha dato vita ad una simulazione in VR finalizzata all'addestramento sul Modulo HCP per i volontari della PC della Regione Piemonte. La simulazione è stata progettata per essere utilizzabile sia da operatori già addestrati che da inesperti, in quanto essa andrà a costituire un tool innovativo a supporto del programma educativo e di addestramento della PC per il rischio idrogeologico.

Purtroppo non è stato possibile condurre una fase di sperimentazione e d'analisi dei dati (a causa della pandemia da *Covid-19*). Per questo motivo, non si sono raccolti dati sull'efficacia educativa e sulle debolezze rilevate da persone esterne alla realizzazione della stessa.

Nonostante ciò, il lavoro realizzato è stato valutato dalla PC stessa ed i membri del team di lavoro hanno potuto osservare una serie di problemi e limitazioni che, allo stato attuale, non è stato possibile superare o comunque migliorare.

- Ottimizzazione: si sono riscontrati problemi legati alla frequenza di riproduzione dei fotogrammi (*Frames Per Second*, o FPS). Per essere utilizzabile, una simulazione VR deve mantenere un valore FPS al di sopra di una certa soglia per evitare che la stessa funzioni a scatti e che possa causare disturbi fisici all'utente (ad esempio, *motion sickness*).

La simulazione in questione contiene molti elementi pesanti dal punto di vista computazionale, come i sistemi particellari, la gestione della simulazione dell'acqua, il

sistema di gestione delle manichette e molti altri elementi che contribuiscono a rallentare il sistema.

Si è quindi proceduto ad attuare una corposa attività di ottimizzazione, che ha compreso fasi di “pulizia” degli script (eliminando quelle funzioni che appesantiscono la simulazione) e di *retopology* degli elementi e del terreno (metodo utilizzato per ricreare una superficie 3D esistente ottimizzando la sua geometria). Nonostante ciò il problema permane ancora in parte.

- Gestione delle manichette: nonostante i significativi passi avanti fatti nel sistema di gestione delle manichette, non è stato possibile ricreare alla perfezione il loro utilizzo, a causa delle numerose limitazioni poste dai tool utilizzati per la simulazione delle corde (ad esempio l'impossibilità di far compiere alle manichette degli spostamenti troppo bruschi). Infatti, *Obi Rope* è un asset progettato per la gestione di piccoli oggetti di contorno mentre la simulazione oggetto della tesi, incentrata per lo più sulla gestione delle stesse manichette, utilizza l'asset in un modo molto più complesso e lontano dall'idea per la quale il tool è stato progettato. Per aggirare queste limitazioni sono state applicate soluzioni forse non sempre realistiche ma sicuramente più efficienti.
- Durata e complessità: l'intera simulazione, dopo la prima reale implementazione di una struttura logica e conseguente sperimentazione in condizioni di laboratorio, è risultata troppo lunga per essere svolta in un'unica sessione. Ciò ha portato alla decisione di suddividerla in fasi e sotto-fasi, in modo che l'utente potesse decidere quali svolgere e quali lasciare al sistema. Ogni fase non eseguita dall'operatore deve comunque essere completata per permettere all'utente di proseguire con i suoi compiti. Per questo motivo, è stato creato un intero sistema d'automazione volto unicamente ad eseguire in autonomia ogni fase, attraverso l'uso di espedienti (come le dissolvenze al nero utilizzate per coprire la comparsa o scomparsa di oggetti) o da NPC (svolgono le operazioni in tempo reale come le farebbe l'utente). Tutto lo sviluppo della logica descritta è stata molto complicata e ha impiegato una gran parte del tempo a disposizione.

## 6.2 Sviluppi futuri

Sono numerosi i miglioramenti e le possibili aggiunte che si potrebbero apportare alla simulazione attuale.

- **Modalità multi-utente:** La simulazione è stata inizialmente progettata per funzionare anche in una modalità multi-utente, che avrebbe dovuto simulare la cooperazione tra diversi operatori facenti parte della stessa squadra HCP. Infatti, in genere le operazioni non vengono compiute in solitaria ma tramite l'aiuto dei colleghi facenti parte di una squadra (tanto che si ha il sistema di NPC nella simulazione).  
Per questo motivo, tutta la realizzazione della struttura è stata eseguita tenendo in mente i requisiti necessari ad una gestione multi-utente, risultando così già pronta per il suo futuro inserimento.
- **Utilizzo della gru:** Per effettuare lo scarramento della pompa e delle manichette, per il caso di pompa Godwin, in genere si procede utilizzando la gru con la quale il camion è attrezzato. Nella simulazione si è deciso di omettere questa procedura in quanto non utile allo scopo della stessa ma in futuro potrebbe risultare utile inserire anche questa procedura, in modo da addestrare gli operatori all'utilizzo dell'intero modulo.

I miglioramenti sopra menzionati sono quelli più immediati e già discussi con la PC e, quindi, anche immediatamente realizzabili.

Altri a più lungo termine potrebbero essere quelli elencati in seguito.

- **Nuovi scenari:** la simulazione è stata progettata per essere eseguita in uno scenario extraurbano, ma sarebbe utile poter inserire altri scenari, come ad esempio l'allagamento di un sottopassaggio o, più in generale, di una zona urbana.
- **Nuovi casi d'uso:** le procedure descritte sono quelle standard all'utilizzo di una motopompa autoadescante; sarebbe interessante inserire nello scenario casi particolari, come la presenza di ostacoli o l'insorgere di un guasto, in modo da insegnare agli operatori come procedere anche in questi casi e coprire così tutti i possibili problemi che potrebbero insorgere.
- **Nuove procedure:** per questa simulazione si è deciso di seguire le procedure del Modulo HCP per l'utilizzo di una motopompa autoadescante Godwin ma il modulo può essere attrezzato anche con altre tipologie di pompa (ad esempio la pompa Albatros o ad immersione) o dover procedere utilizzando ulteriori procedure determinate dalla situazione (ad esempio la costruzione di argini di fortuna con pile di sacchi o barriere apposite, l'utilizzo di vasche, l'evacuazione di civili, ecc.). Sarebbe pertanto utile dare la possibilità di simulare, all'interno dello strumento realizzato, anche altre procedure.



# Bibliografia

[1] *PITEM RISK*

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

[2] *ALCOTRA*

URL: <http://www.interreg-alcotra.eu/it/decouvrir-alcotra/les-projets-finances/risk-resilienza-informazione-sensibilizzazione-e>

[3] *PITEM RISK FOR*

URL: <https://www.regione.piemonte.it/web/temi/fondi-progetti-europei/programmi-progetti-europei/cooperazione-territoriale-europea-piemonte/pitem-risk-progetto-risk-for>,  
[http://www.regione.piemonte.it/governo/bollettino/abbonati/2019/43/attach/dda1810003358\\_1040.pdf](http://www.regione.piemonte.it/governo/bollettino/abbonati/2019/43/attach/dda1810003358_1040.pdf)

[4] *Protezione Civile*

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

[5] *Colonna Mobile Regionale*

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

[6] *Rischio idrogeologico*

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

[7] *Modulo HCP*

URL: [http://protezionecivile-coordinamentovc.it/e107\\_plugins/content/content.php?content.28](http://protezionecivile-coordinamentovc.it/e107_plugins/content/content.php?content.28),  
<https://www.laprotezionecivile.com/quaderni/quaderno8.pdf>

[8] *EUROMODEX*

URL: <http://www.apell-euromodex.eu/>

[9] *Esercitazione Montenegro 2019*

URL: [http://protezionecivile-coordinamentovc.it/e107\\_plugins/content/content.php?content.96](http://protezionecivile-coordinamentovc.it/e107_plugins/content/content.php?content.96)

[10] Grigore C. Burdea, Philippe Coiffet. *“Virtual Reality Technology”*. Wiley, 2003.

[11] William Sherman, Alan Craig. *“Understanding Virtual Reality: Interface, Application and Design”*. Morgan Kaufmann, 2002.

[12] Steven M. LaValle. *“Virtual reality”*. Cambridge University Press, 2015.

[13] Daniel Mestre, Jean-Louis Vercher. *“Immersion and presence”*. Computer Science, 2011.

[14] ISPR. International Society for Presence Research. *“The Concept of Presence: Explication Statement”*. 2000.

[15] Mel Slater. *“Immersion and the illusion of presence in virtual reality. British Journal of Psychology”*. 2018, 109(3).

[16] *Sensorama*.

URL: <https://en.wikipedia.org/wiki/Sensorama>

[17] *La Spada di Damocle*.

URL: [https://en.qwe.wiki/wiki/The\\_Sword\\_of\\_Damocles\\_\(virtual\\_reality\)](https://en.qwe.wiki/wiki/The_Sword_of_Damocles_(virtual_reality))

[18] *PowerGlove*

URL: [https://en.wikipedia.org/wiki/Power\\_Glove](https://en.wikipedia.org/wiki/Power_Glove)

[19] *SEGA VR*

URL: [https://en.wikipedia.org/wiki/Sega\\_VR](https://en.wikipedia.org/wiki/Sega_VR)

[20] *Virtual Boy*

URL: [https://it.wikipedia.org/wiki/Virtual\\_Boy](https://it.wikipedia.org/wiki/Virtual_Boy)

[21] Brendan Cassidy, Gavin Sim, David Wayne Robinson, Devlin Gandy. *“A Virtual Reality for Analyzing Remote Archeological Sites”*. 2019.

URL: [https://www.researchgate.net/publication/332979647\\_A\\_Virtual\\_Reality\\_Platform\\_for\\_Analyzing\\_Remote\\_Archaeological\\_Sites](https://www.researchgate.net/publication/332979647_A_Virtual_Reality_Platform_for_Analyzing_Remote_Archaeological_Sites)

[22] *Cappella della Sacra Sindone*.

URL: [https://it.wikipedia.org/wiki/Cappella\\_della\\_Sacra\\_Sindone](https://it.wikipedia.org/wiki/Cappella_della_Sacra_Sindone)

[23] Albert Sierra, Gabriel de Prado, Isis Ruiz Soler, Ferran Codina. *“Virtual Reality And Archaeological Reconstruction: Be There, Back Then”*. 2017.

URL: <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/>

[24] Raquel Goncalves, Ana Lucia Pedrozo, Evandro Silvia Freire Coutinho, Ivan Figueira, Paila Ventura. *“Efficacia della terapia dell'esposizione alla realtà virtuale nel trattamento della PTSD: una revisione sistematica”*. 2012.

URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3531396/>

[25] Brian Chau, Ivan Phelan, Philip Ta, Sarah Humbert, Justin Hata, Duc Tran. *“Immersive Virtual Reality Therapy with Myoelectric Control for Treatment-resistant Phantom Limb Pain: Case Report”*. 2017.

URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5880370/>

[26] *Battlezone*.

URL: <https://it.wikipedia.org/wiki/Battlezone>

[27] Pieter Wouters, Christof van Nimwegen, Herre van Oostendorp, and Erik D. van der Spek. *“A Meta-Analysis of the Cognitive and Motivational Effects of Serious Games”*. 2013.

URL: [https://web.archive.org/web/20170919031754/http://ocw.metu.edu.tr/pluginfile.php/11786/mod\\_resource/content/1/A%20MetaAnalysis%20of%20the%20Cognitive%20and%20Motivational%20Effects%20of%20Serious%20Games.pdf](https://web.archive.org/web/20170919031754/http://ocw.metu.edu.tr/pluginfile.php/11786/mod_resource/content/1/A%20MetaAnalysis%20of%20the%20Cognitive%20and%20Motivational%20Effects%20of%20Serious%20Games.pdf)

[28] *VRChat*.

URL: <https://en.wikipedia.org/wiki/VRChat>

[29] Jesper Vang Christensen, Mads Mathiesen, Joakim Have Poulsen, Ea Ehrnberg Ustrup, Martin Kraus. *“Player Experience in a VR and Non-VR Multiplayer Game”*. 2018.

URL: [https://www.researchgate.net/publication/329231623\\_Player\\_Experience\\_in\\_a\\_VR\\_and\\_Non-VR\\_Multiplayer\\_Game](https://www.researchgate.net/publication/329231623_Player_Experience_in_a_VR_and_Non-VR_Multiplayer_Game)

[30] E. H. Tanaka et al. *“Immersive virtual training for substation electricians”*, 2017, IEEE Virtual Reality (VR), Los Angeles, CA, pagg. 451–452, doi: 10.1109/VR.2017.7892374.

[31] R. M. S. Clifford, S. Jung, S. Hoermann, M. Billinghamurst, e R. W. Lindeman. *“Creating a Stressful Decision Making Environment for Aerial Firefighter Training in Virtual Reality”*, 2019, IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, pagg. 181–189, doi: 10.1109/VR.2019.8797889.

[32] S.-G. Jeon, J. Han, Y. Jo, e K. Han. *“Being More Focused and Engaged in Firefighting Training: Applying User-Centered Design to VR System Development”*, in 25th ACM

Symposium on Virtual Reality Software and Technology, Parramatta NSW Australia, nov. 2019, pagg. 1–11, doi: 10.1145/3359996.3364268.

[33] S. F. M. Alfalah, J. F. M. Falah, T. Alfalah, M. Elfalah, N. Muhaidat, e O. Falah. “A comparative study between a virtual reality heart anatomy system and traditional medical teaching modalities”, *Virtual Real.*, vol. 23, n. 3, pagg. 229–234, set. 2019, doi: 10.1007/s10055-018-0359-y.

[34] J. Saunders, S. Davey, P. S. Bayerl, e P. Lohrmann. “Validating Virtual Reality as an Effective Training Medium in the Security Domain”, 2019, IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, pagg. 1908–1911, doi: 10.1109/VR.2019.8798371.

[35] B. Lang. “Boeing is Using Varjo Headsets to Train Astronauts for an ISS Docking Mission in 2021”, *Road to VR*, giu. 11, 2020.

URL: <https://www.roadtovr.com/boeing-varjo-virtual-reality-astronaut-training-starliner/>.

[36] M. Hafsia, E. Monacelli, e H. Martin. “Virtual Reality Simulator for Construction workers”, in *Proceedings of the Virtual Reality International Conference - Laval Virtual on - VRIC '18*, Laval, France, 2018, pagg. 1–7, doi: 10.1145/3234253.3234298.

[37] *Progetto - Prodige*

URL: <http://www.pro-prodige.eu/>.

[38] “Living in a changing climate — European Environment Agency”.

URL: <https://www.eea.europa.eu/signals/signals-2015/articles/living-in-a-changing-climate>.

[39] Christoph Anthes, Rubén Jesùs Garcia-Hernàndez, Markus Wiedemann, Dieter Kranzlmuller. “State of the art of virtual reality technology”. 2016.

URL: <https://ieeexplore.ieee.org/document/7500674>

[40] *Vive*

URL: <https://www.vive.com/eu/>

[41] *Documentazione Vive*

URL: <https://developer.vive.com/resources/knowledgebase/vive-specs/>

[42] *Oculus Rift*

URL: [https://www.oculus.com/rift/?locale=it\\_IT#oui-csl-rift-games=mages-tale](https://www.oculus.com/rift/?locale=it_IT#oui-csl-rift-games=mages-tale)

[43] *Documentazione Oculus Rift*

URL: <https://developer.oculus.com/design/oculus-device-specs/>

[44] *Unity*

URL: <https://unity.com/>

[45] *Documentazione Unity*

URL: <https://docs.unity3d.com/Manual/UnityManual.html>

[46] *Asset Store di Unity*

URL: <https://assetstore.unity.com/>

[47] *Visual Studio*

URL: <https://visualstudio.microsoft.com/it/vs/>

[48] *Blender*

URL: <https://www.blender.org/>

[49] *MakeHuman*

URL: <http://www.makehumancommunity.org/>

[50] *Mixamo*

URL: <https://www.mixamo.com/>

[51] *Nodo Idraulico d'Ivrea*

URL: [http://www.adbpo.it/PAI/1%20-%20Relazione%20generale/1.3%20-%20All.1-%20Analisi%20dei%20principali%20punti%20critici/Volume1/16\\_SCHEDADB02.pdf](http://www.adbpo.it/PAI/1%20-%20Relazione%20generale/1.3%20-%20All.1-%20Analisi%20dei%20principali%20punti%20critici/Volume1/16_SCHEDADB02.pdf)

[52] *Steam*

URL: <https://store.steampowered.com/?l=italian>

[53] *Steam VR*

URL: : <https://store.steampowered.com/steamvr?l=italian>

[54] *Open VR*

URL: <https://docs.unity3d.com/Manual/XRPluginArchitecture.html>

[55] *Arm Swinger*

URL: <https://github.com/ElectricNightOwl/ArmSwinger>

[56] *Final IK*

URL: <https://assetstore.unity.com/packages/tools/animation/final-ik-14290>

[57] *Navigation System*

URL: <https://docs.unity3d.com/Manual/nav-NavigationSystem.html>

[58] *Terrain Editor*

URL: <https://learn.unity.com/tutorial/working-with-the-terrain-editor#>

[59] *Cascade*

URL: <https://assetstore.unity.com/packages/tools/terrain/cascade-river-lake-waterfall-and-more-106072>

[60] *Obi*

URL: <http://obi.virtualmethodstudio.com/>