



Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria Informatica

A.A. 2020/2021

Sessione di Laurea di Luglio 2021

Realtà Virtuale per l'addestramento alla gestione del rischio idrogeologico

Simulazione fisica degli strumenti e interazione multiutente

Relatore:

Fabrizio Lamberti

Candidato:

Luca Cavallini

Ringrazio la mia famiglia, per essermi stata vicina nei momenti più felici e quelli più tristi, e che mi ha supportato incondizionatamente.

Ringrazio Melissa Fanelli, che mi sopporta da ormai cinque anni oltre ogni ragionevolezza e una persona cui non potrei fare a meno, che si è occupata dell'impaginazione di questa tesi.

Ringrazio il mio correlatore, Federico De Lorenzis, che ha saputo coadiuvare il suo ruolo istituzionale con parole di conforto e impegno concreto affinché questo progetto potesse essere portato a termine.

Ringrazio tutti i miei amici, una lista troppo lunga per essere citata per esteso, ma con i quali ho condiviso risate e preoccupazioni.

Ringrazio tutti gli altri studenti e ricercatori presso il laboratorio LAB2 del Dipartimento di Automatica e Informatica, che si sono sempre dimostrati gentili e disponibili nei miei confronti.

Per ultimo, ma non meno importante, ringrazio il mio relatore, il prof. Fabrizio Lamberti, che ha creduto in me e mi ha permesso di portare a compimento questo progetto.

Sommario

Il lavoro di tesi è stato sviluppato per fornire un ulteriore strumento per la gestione delle emergenze nelle zone di frontiera, situazioni che il progetto PITEM RISK affronta e analizza all'interno dell'ambito del programma transfrontaliero INTERREG Italia-Francia ALCOTRA. I sotto-progetti PITEM RISK FOR e PITEM RISK ACT poi mettono in collaborazione la Regione Piemonte, la Protezione Civile e il Politecnico di Torino per sviluppare la parte formativa e di addestramento del piano, indirizzando diverse aree di criticità e di rischio come gli incendi boschivi e le emergenze idrogeologiche. Per quest'ultima area il Dipartimento di Automatica e Informatica del Politecnico di Torino e il Coordinamento Regionale della Protezione Civile hanno quindi discusso, ideato e realizzato lo strumento didattico che si andrà a delineare in queste pagine: una simulazione verosimile in Realtà Virtuale (o Virtual Reality, VR), basata sulle procedure e protocolli adottati nel primo intervento a seguito di emergenze idrogeologiche secondo il modulo europeo High-Capacity Pumping. Questa simulazione si compone di diversi fattori e quelli realizzati nell'ambito di questo lavoro di tesi sono la simulazione fisica e le interazioni con alcuni degli strumenti utilizzati dagli operatori (in particolare manichette e tubi di varia natura) e la sincronizzazione dell'esperienza di VR tra due utenti per permettere l'addestramento da remoto

Indice

Sommario	V
Elenco delle figure	XI
1.Introduzione	1
1.1 Contesto	1
1.1.1 PITEM RISK	1
1.1.2 Protezione Civile	2
1.1.3 Modulo HCP	2
1.1.4 Rischio idrogeologico	3
1.1.5 Realtà Virtuale	4
1.2 Obiettivi della tesi	4
1.2.1 Organizzazione delle attività	5
1.3 Struttura del documento	6
2.Stato dell'arte	9
2.1 Realtà Virtuale	9
2.1.1 Concetti base	9
2.1.2 Storia	11
2.1.3 Casi d'uso	16
2.2 VR nell'addestramento	19
2.2.1 Vantaggi e motivazioni dell'approccio	20
2.2.2 Svantaggi	22
2.3 Simulazioni nell'ambito di PRODIGE	22
2.3.1 Rischio esondazione a Cuneo	23
2.3.2 Incidente stradale a Saint Paul-sur-Ubaye	24
2.3.3 Dispersione sostanze chimiche sul colle Maddalena	24
2.3.4 Nube tossica radioattiva in avvicinamento a Torino	25

3.Progettazione	29
3.1 Acquisizione del materiale	29
3.1.1 Progetto PITEM RISK e Protezione Civile	29
3.1.2 Esercitazioni e sopralluoghi	30
3.2 Documento di design	31
3.2.1 Decostruzione dell'esperienza	31
3.2.2 Pianificazione delle risorse necessarie	33
3.3 Plugin di supporto	34
3.4 Obi	35
3.4.1 ObiSolver	35
3.4.2 ObiConstraints	37
3.4.3 ObiRope	39
3.5 Mirror	41
3.5.1 Network Manager	42
3.5.2 Network Identity	42
3.5.3 Authority	43
3.5.4 Network Behaviour	43
3.5.5 Comunicazioni	43
3.5.6 Network Transform e Network Transform Child	44
3.5.7 Protocolli di trasporto	44
3.6 Altri strumenti	45
3.6.1 OpenVR	45
3.6.2 Final IK	45
4.Realizzazione	47
4.1 Simulazione fisica degli strumenti	47
4.1.1 Modellazione delle manichette e logica base in Unity	47
4.1.2 Integrazione con Obi Rope	49
4.1.3 Meccanismo di aggancio	51

4.1.4 Gestione manichette	53
4.1.5 Corda del filtro	53
4.2 Sincronizzazione tra utenti	55
4.2.1 NetworkManager_Custom	55
4.2.2 Menu Online	56
4.2.3 Modifiche per la simulazione online	59
4.2.4 Script per la comunicazione client-server	61
4.2.5 Elementi di valutazione specifici	63
4.3 Altri sistemi	63
4.3.1 Sistema di interazione	63
4.3.2 Logica di automazione degli NPC	63
4.3.3 Ambiente di gioco	64
4.3.4 Voice over e valutazione	64
 5. Conclusione e sviluppi futuri	 67
5.1 Conclusioni	67
5.2 Sviluppi futuri	68
 Bibliografia	 71

Elenco delle figure

2.1 Reality-Virtuality Continuum con esempi, Paul Milgram, 1994	10
2.2 Sensorama, promo e brevetto, Morton Heilig, 1961	11
2.4 Spada di Damocle, Ivan Sutherland, 1968	12
2.3 Brevetto Telesphere Mask, 1963	12
2.5 EyePhone, Jaron Lanier, 1989	13
2.6 SX-Sega VR, Sega, 1993 DX-Virtual Boy, Nintendo, 1995	14
2.7 Palmer Luckey con Oculus Development Kit, 2011	15
2.8 Concetto dello screen-door effect	16
2.9 Esempio di simulazione VR per vigilli di fuoco	18
2.10 Sistema prototipo di RV multimodale simulante la cabina di un AOP	20
2.11 Prova dello scenario del Parco Gesso e Stura con pedana Virtualizer	23
2.12 Dettaglio soccorsi scenario Paul-sur-Ubaye	24
2.13 Scenario dell'incidente sul Colle della Maddalena	25
2.14 Dettaglio dell'area di lavoro nello scenario della nube tossica	25
3.1 Esercitazione HCP Nodo idraulico d'Ivrea	31
3.2 Motopompa Godwin in funzione	33
3.3 Scenario all'interno di Unity con tutti i modelli presenti	34
3.4 Parametri dell'ObiSolver	36

3.5 Editing delle particelle di ObiRope durante la simulazione	39
3.6 Piramide dei componenti in Mirror	41
3.7 Schema delle comunicazioni tra client e server	44
4.1 Esempio di modello scenico manichette appiattibili e modello tubo curvo rigido	48
4.2 Gerarchia apertura in Blender e Unity	48
4.3 Impostazione dell'ObiSolver in game	50
4.4 Esempio di aggancio della manichetta alla pompa con supporto da parte del client	51
4.5 Boccia_var M1 (sx) e Apertura_var M5(dx) sono agganciate tra loro, visibile dal campo...	52
4.6 Manichette agganciate in-game e Parent Constraint Boccia_var M1	52
4.7 List Tubes con la catena di aspirazione formata dalle due manichette M1 e M5	53
4.8 Tiraggio della corda del filtro	54
4.9 Script della corda del filtro con i parametri di valutazione	54
4.10 Visione da edito di Network Manager Custom	56
4.11 Processo di connessione dell' host al menu online	57
4.12 Processo di connessione del client al menu online	58
4.13 Esempio della simulazione esclusiva	59
4.14 Esempio in editor e codice del passaggio di fase sincronizzato	60
4.15 Elenco degli script Network Behaviour sul player prefab	62

1.Introduzione

1.1 Contesto

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), ampliata con interazione multiutente.

La collaborazione del Coordinamento Regionale della Protezione Civile per il Piemonte e il Politecnico di Torino ha dato il via allo sviluppo di un progetto didattico innovativo basato sull'utilizzo della tecnologia VR, scelta con lo scopo di ricreare 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 Protezione Civile nel primo intervento in caso di rischio idrogeologico e la loro implementazione in un contesto multiutente.

1.1.1 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 mira a migliorare la risposta alle emergenze, consentendo una maggior comprensione dei rischi e dei giusti comportamenti da adottare in caso di crisi.

All'interno del PITEM RISK si sviluppano due sotto-progetti specifici nell'ambito della prevenzione e addestramento denominati PITEM RISK FOR [1] e PITEM RISK ACT [3]: questi prevedono una collaborazione tra la Regione Piemonte, la Protezione Civile 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" 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.

1.1.2 Protezione Civile

La Protezione Civile (PC) è un ente pubblico che svolge attività comprendenti la tutela delle 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 Regionale [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.

1.1.3 Modulo HCP

Il modulo High-Capacity Pumping (HCP) [6], 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 PC 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) [7] 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 [8], 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 PITTEM RISK.

1.1.4 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); anche se l'uso corretto del termine si riferirebbe soltanto allo studio delle 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 PC 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 [9];
- rischio idraulico, nel caso di superamento dei livelli idrometrici dei corsi d'acqua maggiori e dei relativi effetti sul territorio [9]

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. 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.

1.1.5 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 considerati come 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.

Con "Realtà Virtuale" si definisce un'interazione uomo-macchina tramite simulazione in tempo reale rappresentante un ambiente realistico fruibile attraverso particolari interfacce che riproducono molteplici sensi.

Lo scopo è quindi di "ingannare" l'utente fino al punto di fargli credere di trovarsi in un ambiente verosimile che segue le stesse leggi proprie del mondo reale in cui è possibile il movimento e l'interazione. 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.

Sono molti e vari i campi di utilizzo della VR: addestramento, educazione, medico, militare, industriale e psicologico. 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.

1.2 Obiettivi della tesi

Il lavoro di tesi illustrato in questo documento si inserisce e finalizza un'attività più ampia che aveva lo scopo di sviluppare un'applicazione VR per supportare l'addestramento di operatori 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, accompagnato solamente da Personaggi non Giocanti o Non-Player Character (NPC) o anche con l'aiuto un altro utente e opererà nell'utilizzo di motopompe autoadescanti nelle fasi di posizionamento, messa in sicurezza, montaggio manichette, accensione, controllo e spegnimento.

In particolare, il presente documento andrà a presentare le fasi di progettazione e realizzazione del sistema di interazione per le manichette e la loro fisica e la sincronizzazione via rete remota delle azioni degli utenti, che integrano quanto realizzato e descritto in due precedenti lavori di tesi.

Il sistema di gestione delle manichette ha richiesto l'utilizzo di un plugin per simulare la fisica di corpi tubulari con vario grado di flessibilità e il meccanismo di interblocco tra due capi del tubo, oltre che la simulazione dei problemi di trasporto e disposizione nell'ambiente.

L'aggiunta di una modalità multiutente rappresenta invece un'estensione rispetto all'applicazione originale, che permette a due utenti di interagire in contemporanea nella stessa area di lavoro in ruoli diversamente coperti da NPC.

1.2.1 Organizzazione delle attività

Per prima cosa 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 prof. Fabrizio Lamberti (responsabile del progetto per il Dipartimento di Automatica e Informatica del Politecnico di Torino) e dei 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.

A seguire dopo una prima versione dell'applicazione in cui era disponibile solo la modalità utente singolo per l'apprendimento delle procedure e per una valutazione in solitaria si è sviluppata, attraverso il presente lavoro, una modalità di valutazione multiutente in cui la squadra verrà valutata sia per comportamenti individuali sia per come i suoi componenti avranno interagito insieme sul campo.

1.3 Struttura del documento

Il documento di tesi è suddiviso nei seguenti capitoli.

- **Capitolo 1: Introduzione**

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

- **Capitolo 2: Stato dell'arte**

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

- **Capitolo 3: Progettazione**

Resoconto delle fasi di progettazione del lavoro di tesi, comprendenti l'analisi del materiale fornito e la stesura del documento di design, seguiti dalle strategie seguite al momento dell'ideazione dello strumento delle manichette e integrazione nella simulazione di queste, oltre che la teoria riguardante le architetture client-server.

- **Capitolo 4: Realizzazione**

Analisi delle fasi di sviluppo della simulazione approfondendo l'implementazione passo-passo della fisica delle manichette e delle comunicazioni a distanza.

- **Capitolo 5: Conclusioni e sviluppi futuri**

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

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 o device con abbastanza potenza computazionale (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 Concetti base

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.

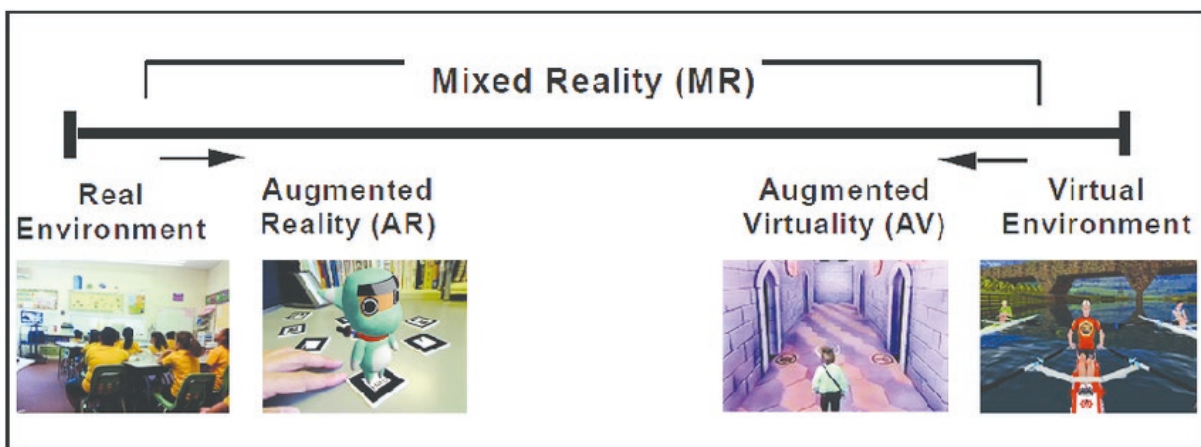
Per questo motivo nel campo della VR, sono stati definiti due concetti in grado di valutare fattori quali gli stimoli, le interazioni e l'ambiente dal punto di vista del realismo e della naturalezza delle sensazioni suscitate nell'utente: immersione e presenza.

- 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. 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.
- 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.

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

Un altro concetto che può essere utile a illustrare il rapporto della VR con la realtà è il reality-virtuality continuum[10] (Figura 2.1), introdotto da Paul Milgram.



2.1 Reality-Virtuality Continuum con esempi, Paul Milgram, 1994

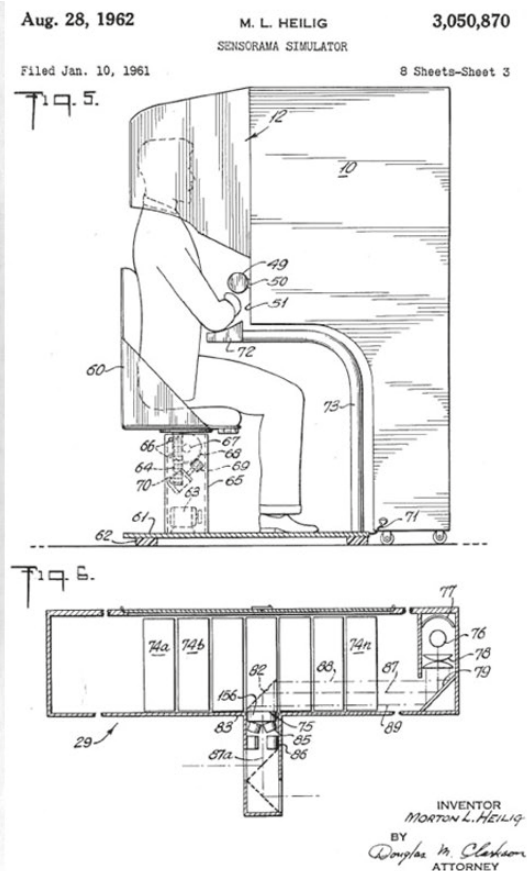
Praticamente si tratta di un asse sui cui estremi sono collocati da un lato la realtà e dall'altro lato la completa virtualità (ovvero un ambiente totalmente artificiale fruibile attraverso l'esecuzione su computer e mediante interfacce meccaniche come mouse e tastiera). Tutte le sfumature di questo spettro in mezzo tra questi due estremi costituiscono l'insieme della Mixed Reality (MR), ovvero l'interazione tra oggetti del mondo fisico con quelli del mondo puramente virtuale che porta a una commistione dei due. Più vicino all'estremo della realtà vi è la Augmented Reality (AR), in cui si introducono elementi virtuali in sovrapposizione ad uno spazio di gioco reale, in generale posizionati mediante punti di riferimento, mentre dall'altro lato vi è l'Augmented Virtuality (AV) in cui in un ambiente virtuale possono essere introdotti nuovi oggetti in real time partendo da elementi reali (come oggetti o persone) catturati mediante scansioni 3D o altri dispositivi di cattura immagine. La VR si colloca più nell'ambito della virtualità in quanto l'utente interagisce in un ambiente virtuale, sebbene utilizzi controller e interfacce che permettano di utilizzare il pieno spettro dei movimenti umani all'interno di un'area di spazio fisico delimitata e che riproducano il sistema di vista umano. I sistemi VR moderni sono in grado di restituire i 6 DoF (Degrees of Freedom o Gradi di Libertà), 3 per la posizione e 3 per l'orientamento, necessari a collocare precisamente il cono di visuale all'interno del mondo virtuale.

2.1.2 Storia

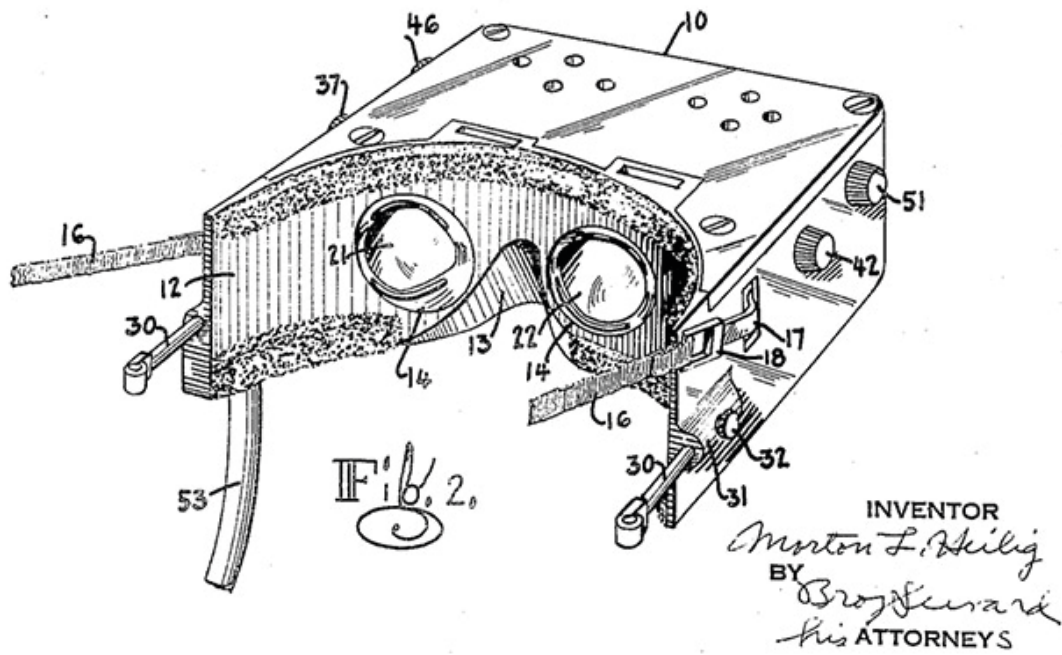
La VR si è sviluppata molto soprattutto nell'ultimo decennio, ma non è una vera novità nel campo delle tecnologie [11]. Si possono sostanzialmente identificare cinque periodi di progressione della VR:

1. 1950-1978: le radici e primi esperimenti;
2. 1979-1989: ulteriore studio nei laboratori;
3. 1982-1997: commercializzazione e iniziale interesse per la tecnologia;
4. 1997-2010: realizzazione dei limiti e abbandono parziale da parte del pubblico;
5. 2010-corrente: risorgimento e diffusione capillare.

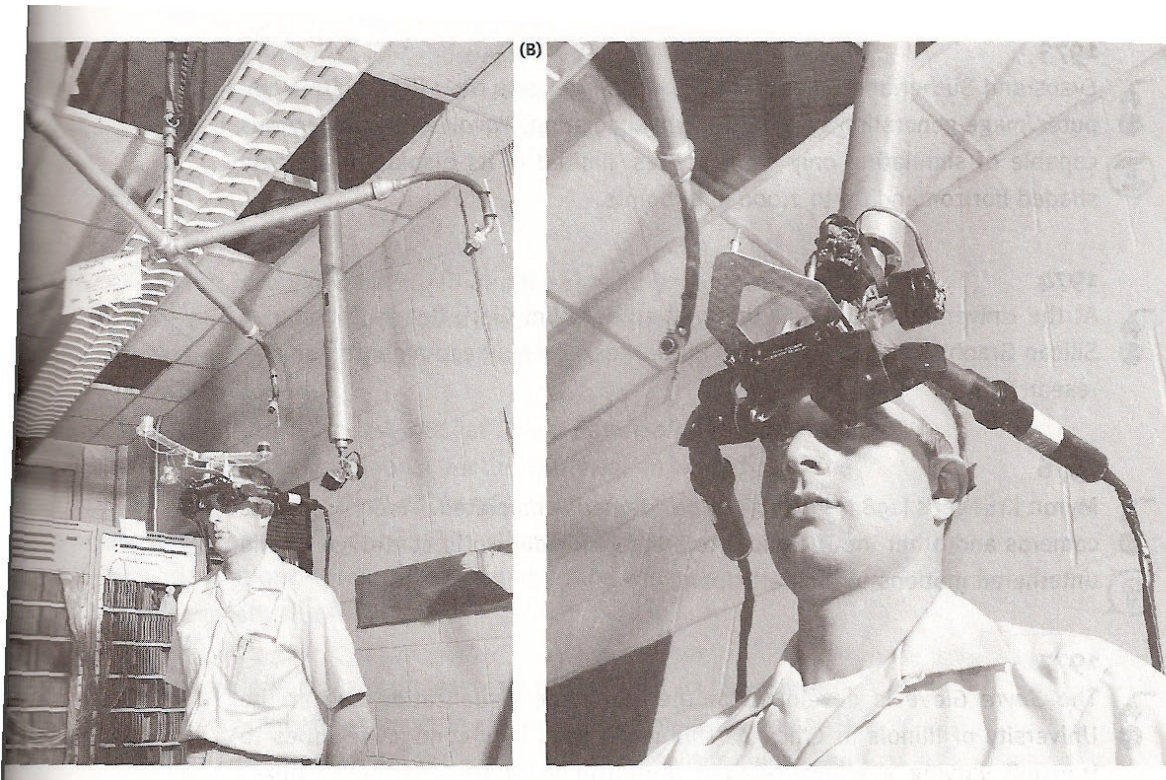
Già nel 1961 Morton Heilig, videomaker statunitense, inventò un dispositivo chiamato Sensorama [12] (Figura 2.2). Questo permetteva di riprodurre un film che non coinvolgeva solo vista e udito ma anche tatto e olfatto, in un'esperienza che chiamò Experience Theater. Il dispositivo comprendeva un monitor 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. Sempre da parte del videomaker fu anche il deposito del brevetto della prima idea di Head Mounted Display (HDM)[13] (Figura 2.3)



2.2 Sensorama, promo e brevetto, Morton Heilig, 1961



2.3 Brevetto Telesphere Mask, 1963



2.4 Spada di Damocle, Ivan Sutherland, 1968

È solo nel 1968, però, che venne effettivamente realizzato quello che ora viene considerato il primo visore di tipo Head-Mounted Display, progettato al MIT da Ivan Sutherland e dal suo studente Bob Sproul, così pesante che doveva essere appeso al soffitto. Questa caratteristica ispirò il suo nome, La Spada di Damocle[14][Figura 2.4] 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). 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.



2.5 EyePhone, Jaron Lanier, 1989

Ma è solo nel 1987 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 (Figura 2.5) e DataSuit.

Il DataGlove venne venduto nel 1989 alla Nintendo diventando PowerGlove [18], il primo di-

spositivo 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. 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[19] (Figura 2.6) 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. La Nintendo nel 1995 tentò di rifarsi del flop dei DataGlove sviluppando il Virtual Boy [20] (Figura 2.6). 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. Uscirono sul mercato anche visori pensati per l'utilizzo con sistema operativo Microsoft Windows come Forte VFX1, Philips Scuba VR e VFX3D.



2.6 SX-Sega VR, Sega, 1993 DX-Virtual Boy, Nintendo, 1995

Dopo questi primi decenni di grandi aspettative per la nuova, promettente tecnologia, gli anni 2000 furono caratterizzati da un brusco calo dell'interesse a causa delle difficoltà ritenute insormontabili con le tecnologie dell'epoca, ovvero la scarsa risoluzione, basso contrasto, bassa profondità di campo, alto input lag e gli esosi requisiti computazionali, che causavano problemi di nausea oltre che economici. La VR sembrava destinata a essere relegata a una mera idea attuabile solo in laboratorio o nella finzione del cinema di fantascienza, e non un prodotto effettivamente commerciabile.

Dal 2010 la prospettiva sarebbe cambiata. [15] Palmer Luckey, un giovane ingegnere della University of South California, nel tempo libero aveva sviluppato diversi prototipi per un nuovo headset

VR che però sarebbe stato ridotto come bacino d'utenza, ipotizzando di vendere appena 100 copie come kit fai-da-te per appassionati tramite la piattaforma di crowdfunding Kickstarter, con una campagna pianificata per il 2012. Fu l'intervento di John Carmack, sviluppatore di id Software e di titoli iconici come Castle Wolfenstein, Doom e Quake, interessato al progetto del ragazzo, Oculus Rift, che lo mise in contatto con Gabe Newell e con il team di Valve per portare a realtà il progetto su scala ben più ampia[16] (Figura 2.7).



2.7 Palmer Luckey con Oculus Development Kit, 2011

Dal 2012, col successo della campagna Kicstarter di Oculus, l'interesse per la VR commerciale crebbe nuovamente. Si aggiunsero altri competitor alla scena, come Google con il suo CardBoard VR, Sony con il PlayStation VR e Valve stessa con l'HTC Vive, per citarne alcuni. Dal 2018, con il rilascio di Oculus Go e Oculus Quest, si dà il via ai cosiddetti standalone VR headset, ovvero visori che non necessitano di una connessione via cavo a un PC e sensori esterni per delimitare lo spazio di attività, ma possono essere appunto usati senza altri dispositivi, ospitando una cpu integrata e camere sul visore stesso per ottenere informazioni sull'ambiente circostante (anche se rimane la possibilità di collegarli a una postazione di lavoro per permettere prestazioni migliori). Invece la VR da dispositivo mobile (ovvero via smartphone) è andata via via scemando, non avendo attecchito per via degli esosi requisiti computazionali.

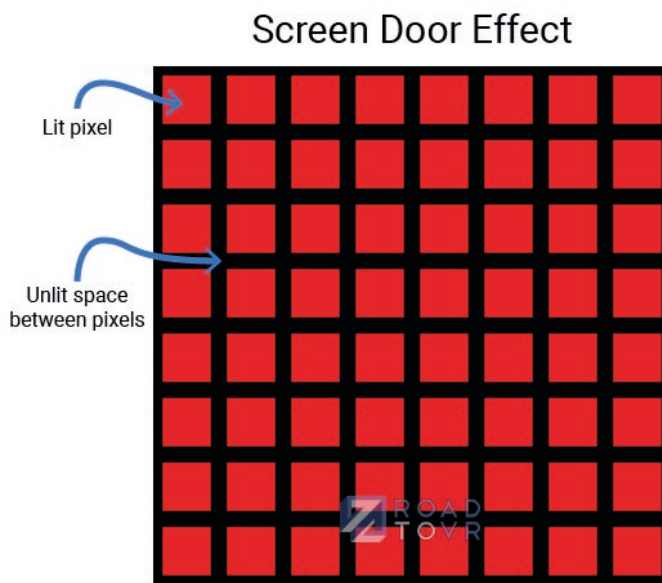
Il futuro dei visori VR inevitabilmente porterà a potenziare ulteriormente le caratteristiche principali per una esperienza più confortevole per l'utente finale, ovvero:

- La risoluzione degli schermi per ogni occhio, che evita effetti di pixellation e screen-door effect, che fanno apparire l'immagine meno nitida e suddivisa da una griglia in quanto os-

servata così da vicino. L'accentuazione di questo effetto dipende anche se le lenti del visore sono composte da LCD RGB, che restituiscono un'immagine più nitida, o da OLED, che hanno una resa del colore e dei neri più vivida a minor costo ma con meno risoluzione [17] (Figura 2.8).

- Il refresh rate, che aiuta a simulare un'esperienza più fluida ed evitare effetti di nausea. Sotto i 60 Hz questi effetti sono molto marcati, mentre con visori che raggiungono i 90+ Hz di refresh rate permettono di simulare molto meglio il rateo di acquisizione dell'occhio umano.

Il field of view orizzontale, che devono coprire almeno il cono di circa 110 gradi in cui avviene la visione stereoscopica per l'occhio umano.



2.8 Concetto dello screen-door effect

2.1.3 Casi d'uso

La VR è oggi ampiamente diffusa nell'industria dell'intrattenimento, in quanto si tratta di uno dei pochi settori nei quali è possibile renderla disponibile alle masse e mostrare immediatamente le sue potenzialità.

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. Ulteriormente, con le restrizioni dovute dall'epidemia corrente,

le potenzialità per integrare vecchie modalità di consumo con questa nuova tecnologia sono sempre maggiori [18].

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. Nella cura delle fobie è possibile ricreare scene in cui il paziente deve affrontare le proprie paure (terapia dell'esposizione).

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.

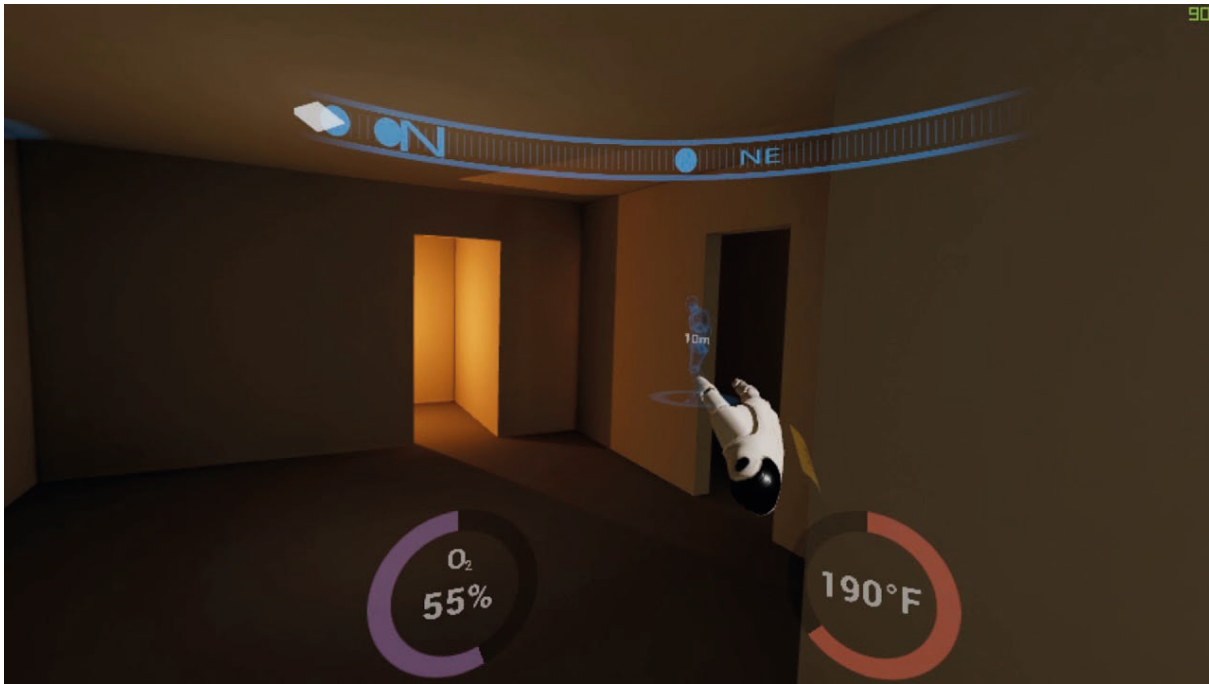
Con risultati ancora scarsi si è anche tentato di applicare la VR alla riabilitazione fisica, risultando meno efficace dei metodi tradizionali ma presentando margini di miglioramento. Si continua invece ad utilizzarla per il trattamento del dolore agli arti fantasma, con la creazione di stanze 3D in 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 successiva. Queste applicazioni vengono anche chiamate Serious Game (SG).

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 SG e, benché in genere ormai si colleghi il termine ai giochi digitali, lì si può ricollegare a giochi di qualsiasi genere, utilizzando qualsiasi tecnologia e per qualsiasi piattaforma.



2.9 Esempio di simulazione VR per vigilli di fuoco

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, in quanto presentano diversi vantaggi. 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, così come simulare situazioni di pericolo o difficilmente replicabili (Figura 2.9). La costruzione e l'uso di simulatori tradizionali o la ricreazione di eventi simulati ha infatti un costo molto alto in confronto allo sviluppo di un SG.

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. 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.

L'ultimo ambito di applicazione in cui la VR presenta sfoci interessanti sono quelle dell'ambito sociale. Si possono quindi creare interazioni multiutente per condividere le esperienze con amici, familiari e colleghi e renderli partecipi dell'immersività del mezzo.

Molti studi sono stati effettuati per questo ambito (con lo sviluppo di applicazioni come VRChat [19] con risultati che evidenziano le potenzialità sociali della tecnologia, tanto da far interessare Facebook a soluzioni di social VR.

Per quanto riguarda le applicazioni VR per l'addestramento e l'educazione, il multiutente può portare un grande vantaggio, permettendo di condividere l'esperienza educativa con altre persone. È perciò possibile 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 VR nell'addestramento

Come precedentemente accennato, sempre maggiore attenzione viene posta nell'impiego della VR per scopi non solo ludici, e in particolare caso nella sua interazione con l'universo dei SG. Mescolare infatti l'immersione della simulazione con l'aspetto educativo e informativo di questi ultimi può aiutare ad apprendere più efficacemente le informazioni e procedure che si vogliono comunicare agli operatori e inserirle in un contesto verosimile alla mansione svolta. Queste possibilità sono d'interesse sia per enti governativi, istituzioni accademiche e privati legati al settore tecnico che cercano metodi alternativi e/o complementari a quelli tradizionali, e la VR sembra poter colmare alcune di queste lacune.

Alcuni degli elementi che contraddistinguono un addestramento efficace sono la qualità dell'insegnamento, la frequenza nel verificare le competenze impartite e la copertura di molte situazioni ed imprevisti che possono sorgere nell'esecuzione della mansione preposta. Tuttavia, i principali investimenti per ottenere queste caratteristiche sono rivolti verso metodi di training mediante lezioni frontali od online e con esercitazioni dal vivo che simulano i casi affrontati durante l'insegnamento. Questi sono sicuramente protocolli consolidati, ma hanno la loro serie di svantaggi: l'apprendimento in aula o via web non aiuta ad approcciarsi con lo sforzo psicofisico effettivo necessario a portare a termine l'intervento nei migliori dei modi possibili; di contro le esercitazioni dal vivo richiedono ingenti spese di tempo e risorse per la loro organizzazione ed esecuzione, e sebbene forniscano un esempio della situazione in cui il personale si troverà ad operare, nei casi di simulazione di emergenze naturali o altri eventi ad alto impatto questo esempio può essere insufficiente, perché si deve garantire la sicurezza degli studenti in training e riprodurre eventi simili minerebbe la loro incolumità, oltre che essere impossibile.

La VR può così inserirsi come agente intermedio tra queste due modalità di training e, sebbene l'adozione sia ancora agli albori, con rare eccezioni (per esempio nell'ambito medico e aerospaziale), già dallo scorso decennio la potenziale utilità di questi sistemi viene indagata in maniera sempre più estesa da molteplici settori nel campo pubblico e privato. È quindi opportuno illustrare quali possono essere i motivi per implementare una soluzione del genere, e i relativi punti di forza e criticità.

2.2.1 Vantaggi e motivazioni dell'approccio

Come illustrato, una delle possibili motivazioni per cui si è portati a implementare un sistema di VR per l'addestramento è l'investimento richiesto per le esercitazioni dal vivo. Le lezioni frontali non possono preparare all'impatto con la strumentazione reale e l'approccio sul campo, ma questi possono presentare gravi rischi per la salute degli operatori se affrontati con leggerezza. Inoltre, se il training prevede l'interazione con una infrastruttura pubblica o macchinario privato un errore da parte del personale in addestramento, ancora inesperto e non totalmente saldo sulle proprie capacità, potrebbe causare un disservizio o rovinare l'apparecchiatura su cui si stanno verificando le capacità apprese durante l'insegnamento in aula, causando quindi ulteriori costi oltre quelli già stanziati.

Per evitare questa situazione, si tende a testare le abilità dell'individuo su un sottoinsieme ristretto di operazioni e macchinari, fornendo quindi una preparazione parziale che potrebbe portare a momenti di indecisione sul campo[20]. La VR è quindi uno strumento utile per poter fornire l'intero intervallo di casistiche e la piena funzionalità delle apparecchiature in una simulazione, fornendo un training completo e al tempo stesso evitando di danneggiare le strutture reali (Figura 2.10).



2.10 Sistema prototipo di RV multimodale simulante la cabina di un AOP (Air Observation Post)

Lo stesso discorso può essere fatto per il personale adibito alla risposta ad eventi catastrofici e naturali, a cui bisogna aggiungere un altro elemento particolarmente impattante sulle prestazioni sul campo: lo stress psicologico e fisico a cui si viene sottoposti. La familiarità che l'operatore ha con

lo scenario può fare la differenza sulla qualità dell'intervento, ma molte situazioni di pericolo non possono essere facilmente replicate nell'addestramento anche per incolumità stessa degli operatori, e la loro occorrenza naturale può essere estremamente infrequente da non permettere un'esplorazione accurata dell'effetto sulle abilità di decisione e l'abilità di comunicazione e coordinamento tra i membri della squadra di primo intervento.[21] Anche qui la simulazione in VR aiuta l'addestramento potendo non solo simulare questi edge-case, ma introducendo elementi che possono rendere l'esperienza più verosimile, come impedimenti visivi, uditivi, folla di cittadini da evacuare, ecc. senza incorrere nei costi e nella pericolosità di provare a riprodurli in una simulazione dal vero, garantendo la salute del personale in training e migliorando la loro risposta in condizioni stressanti.

Collegato a questo argomento può essere la misurazione dello stato emotivo e psicologico degli operatori durante la simulazione, verificando che lo stress percepito e la risposta ad esso siano comparabili con la situazione reale e le esercitazioni dal vivo.[22] I risultati e le prestazioni ottenute durante il training in VR sono registrabili per apportare anche modifiche correttive al sistema di addestramento, virtuale e non. Un altro aspetto positivo che si è riscontrato in diversi esperimenti di training in VR è la somiglianza tra l'efficacia dell'insegnamento fornito dai metodi tradizionali con quelli in VR, sia per aspetti nozionistici [23] che di procedure da applicare sul campo[24]. È possibile anche fornire feedback diretto e immediato alle azioni degli utenti, permettendo loro di assimilare più facilmente le procedure corrette e correggere quelle errate.

Un sistema in VR ha anche il vantaggio di poter scalare il proprio grado di difficoltà in base alle competenze del personale in addestramento, e dimensionato in base alle esigenze dell'organizzazione che richiede il sistema di training in VR. Inoltre, questo può essere utilizzato per standardizzare un protocollo di risposta a situazioni di pericolo, o valutare eventuali lacune e difetti in quelli già impiegati per comprendere come migliorarli e renderli più estensivi.

Il costo relativamente basso di un visore per la VR e la facilità d'uso che migliorano col progredire della tecnologia permettono di adottare questa soluzione sempre più facilmente rispetto alle costose esercitazioni su media e larga scala, rendendo più comoda e ripetibile la pianificazione di simulazioni nel sistema di VR, che raggiungono livelli di qualità da mostrare anche i minimi dettagli della strumentazione che in casi estremi come le cabine di pilotaggio degli astronauti, in cui è richiesto un alto livello di dettaglio e fedeltà.[25] Una strumentazione ubiqua a costo basso permette inoltre di superare le limitazioni e barriere geografiche, permettendo al personale di addestrarsi anche remotamente e in caso di piattaforme di training consistenti e sincronizzate è possibile sviluppare anche la capacità di comunicazione anche con gli altri membri della squadra operativa, specialmente per quegli eventi che chiedono una risposta geograficamente distribuita.

Questi sono alcuni dei vantaggi che implementare una soluzione in VR può fornire nell'affiancare le normali esercitazioni dal vivo e le lezioni frontali.

2.2.2 Svantaggi

Sebbene le premesse sembrino estremamente positive, è opportuno presentare anche gli svantaggi insiti nella tecnologia allo stato attuale.

La VR sta diventando sempre più diffusa ma la sua adozione per molti settori è ancora allo stadio embrionale, pertanto la capacità degli operatori di adoperare i controller e i sistemi in VR richiedono molto spesso un allenamento preventivo per risultare competenti ed efficaci nel loro utilizzo, e ciò potrebbe precluderne l'implementazione, oltre al fattore del costo iniziale per la modellazione degli scenari e la programmazione delle logiche di simulazione.

Un altro punto a sfavore è la mancanza di feedback fisico, a meno di un utilizzo di dispositivi all'avanguardia come tute a corpo intero; in generale con head mounted display e controller classici la simulazione può trarre vantaggio solo da stimoli sonori e visivi, con limitati input aptici. Ciò potrebbe portare a sottostimare alcuni sforzi richiesti invece nell'evento sul campo. [26]

In base alla specificità dell'applicazione, la dimensione dell'insieme degli operatori specializzati che possono fornire consigli sulla correttezza delle procedure implementate e sull'accuratezza della simulazione può essere limitata.

2.3 Simulazioni nell'ambito di PRODIGE

Nell'ambito del programma INTERREG di cooperazione transfrontaliera ALCOTRA tra Francia e Italia sorge il progetto PRODIGE (PROteggere i cittadini, Difendere le Infrastrutture, Gestire i grandi Eventi), volto a “rafforzare la collaborazione tra le Autorità italiane e francesi preposte alla gestione delle emergenze, incrementando la capacità di eseguire efficaci operazioni di risposta congiunta ai rischi naturali e antropici attraverso la definizione di protocolli innovativi di training degli operatori basati sulla realtà virtuale”. [27]

Così questo progetto funge da precursore al progetto PITEM RISK, nato in risposta ai forti cambiamenti climatici e i relativi disastri ambientali che possono portare (frane, smottamenti, dissesti idrogeologici, ecc.), con l'aumento della loro frequenza e intensità previsto dalla European Environment Agency [28]. In particolare le zone transfrontaliere sono le più vulnerabili a fenomeni di questo genere e per garantire la miglior cooperazione possibile per una risposta operativa pronta e l'assistenza ai cittadini, a fronte di differenze in ambito gestionale, linguistico e organizzativo, si sono sviluppati strumenti innovativi basati sulla VR che sono ora a disposizione per agevolare la transizione verso un nuovo approccio alla PC basato sulla combinazione di tecnologie, organizzazione e formazione.

Il progetto si propone inoltre di migliorare i percorsi di formazione ed addestramento degli operatori di PC e sviluppare nuove strategie di informazione e comunicazione in tempo reale con la popolazione trasformandoli in sensori locali e sorgenti di dati. Questa iniziativa mira a raggiungere sia gli “addetti ai lavori”, sia i cittadini, con l'obiettivo di informare, ma anche di creare consapevo-

lezza circa le potenzialità della VR per un mondo più sicuro.

I quattro scenari qui presentati sono stati in qualche modo d'ispirazione per l'applicazione che si è andata a sviluppare all'interno del progetto PTTEM RISK per il rischio idrogeologico e che rappresenta il soggetto di questa tesi. Le loro descrizioni sono tratte dal documento di presentazione di PRODIGE.

2.3.1 Rischio esondazione a Cuneo

Questo scenario simula un rischio idrogeologico per una possibile esondazione nell'area del Parco. In una situazione di allerta 2 (ovverosia di preallarme e criticità moderata) la pista ciclabile che costeggia il torrente Gesso rimane aperta al pubblico. A seguito di un improvviso peggioramento della situazione meteo la soglia di allerta viene innalzata a 3 (cioè di allarme e criticità elevata): vengono quindi attivate le procedure di PC per la chiusura al pubblico e l'evacuazione dell'area. L'operatore arriva sul posto e procede all'evacuazione delle persone presenti e alla chiusura degli accessi. Dopo aver perlustrato l'area, presidia gli ingressi mantenendosi in contatto con il Centro Operativo Comunale.

Le tecniche di rilievo introdotte da questo scenario sono la navigazione mediante pedana treadmill e corsa sul posto e il sistema di valutazione automatica degli errori (Figura 2.11).



2.11 Prova dello scenario del Parco Gesso e Stura con pedana Virtualizer

2.3.2 Incidente stradale a Saint Paul-sur-Ubaye

Lo scenario prevede il recupero del conducente di un veicolo da diporto, trascinato da un'esondazione improvvisa verso un ponte a metà del fiume Riou Sec, nei pressi di St. Paul-sur-Ubaye. Le pareti del ponte impediscono al mezzo di cadere nel fiume e l'autista è bloccato all'interno, poiché la forza dell'acqua impedisce l'apertura della porta. Vengono allertati i soccorsi, un equipaggio di servizio dotato di imbracatura e corda si occupa di ormeggiare il veicolo, mentre il conducente viene prelevato da un elicottero della PC (Figura 2.12).



2.12 Dettaglio soccorsi scenario Paul-sur-Ubaye

2.3.3 Dispersione sostanze chimiche sul colle Maddalena

Lo scenario simula l'incidente di un veicolo pesante che trasporta sostanze chimiche altamente inquinanti su territorio francese in prossimità del Colle della Maddalena. Il veicolo esce dalla carreggiata e cade su un fianco nel fiume all'ingresso del paese di Larche, riversando lentamente nell'acqua il prodotto chimico trasportato. Il conducente, privo di sensi, rimane all'interno della cabina finché non viene estratto e messo in salvo dai Vigili del Fuoco. La sostanza chimica, nel frattempo, si diffonde attraverso il fiume, causando inquinamento: vengono quindi attivate le procedure per delimitare l'area e contenere la perdita (Figura 2.13).

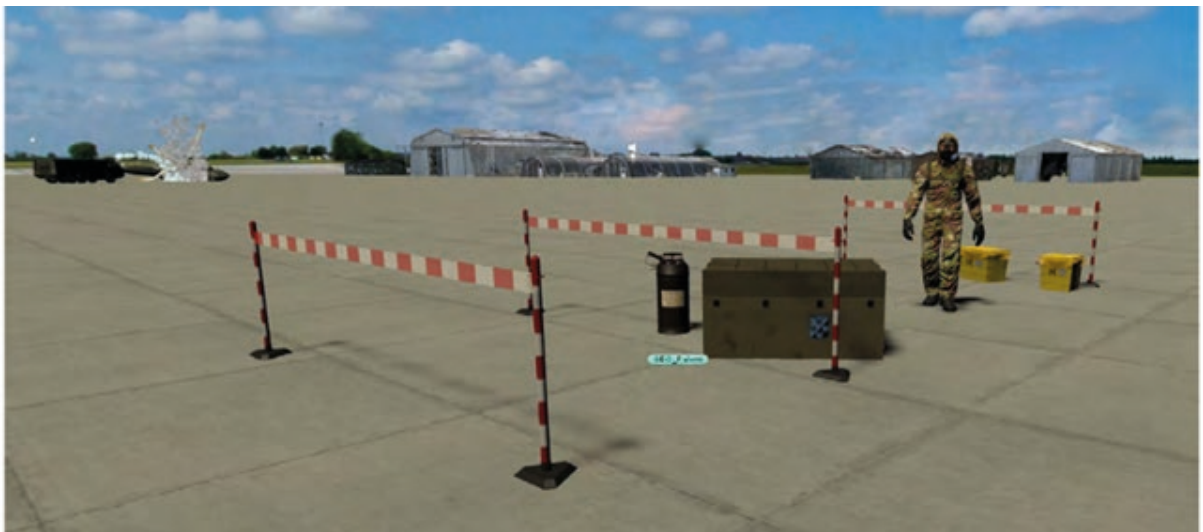
Sia questo che lo scenario precedente presentano un modello georeferenziato dello scenario, interazione simultanea multiplayer, guida di veicoli (sia terra che aria), librerie di oggetti 3D legati all'equipaggiamento della PC e una "cabina di regia" dove poter modificare lo scenario durante il suo svolgimento.



2.13 Scenario dell'incidente sul Colle della Maddalena

2.3.4 Nube tossica radioattiva in avvicinamento a Torino

Il quarto scenario è l'unico costituito da due momenti consequenziali. Inizialmente un elicottero carico di torio 232 sbaglia la manovra di decollo e finisce sopra una cisterna di rifornimento, causando un incendio. Gli operatori CBRN per contenere l'incendio e delimitare l'area radioattiva, dalla quale nel frattempo si sprigiona una nube tossica. Il secondo momento vede la nube tossica che investe il Parco Dora, nella città di Torino. Gli operatori di Polizia Municipale della Città di Torino, già presenti sul posto per un evento musicale che raduna centinaia di persone, vengono allertati dalla Centrale Operativa del possibile arrivo della nube (Figura 2.14).



2.14 Dettaglio dell'area di lavoro nello scenario della nube tossica

Per la realizzazione della prima parte dello scenario è stata attivata una cooperazione con l'Aeronautica Militare, che ha inoltre consentito ai partner PRODIGE di operare in un contesto in cui le tre responsabilità di PC, Pronto Soccorso ed Anti-incendio sono in capo alla stessa entità, ossia il comandante dell'aeroporto: ciò ha permesso di confrontarsi con un modello concettuale più vicino al sistema francese, in cui l'autorità dello SDIS incanala in sé esattamente queste tre funzioni operative.

Questo scenario si contraddistingue per la grafica immersiva, possibilità di scelta di un ruolo, l'integrazione delle procedure di comunicazione coi cittadini e di software specializzati per la simulazione degli incendi.

3. 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 multiutente e la simulazione fisica degli strumenti.

3.1 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 RISK, la modalità di lavoro della PC, l'area di applicazione finale della simulazione e le procedure attuate in caso di rischio idrogeologico.

3.1.1 Progetto PITEM RISK e Protezione Civile

Come già descritto nel Capitolo 1, la simulazione nasce nel contesto del progetto PITEM RISK 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 trasporta-

bile, 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.

3.1.2 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 Albatross e dell'attrezzatura ad essa collegata (rimorchio, manichette, altri attrezzi). Dal momento che la pompa Albatross 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 (Figura 3.1).

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.



3.1 Esercitazione HCP Nodo idraulico d'Ivrea

3.2 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. La descrizione della logica, il design dell'ambiente, l'interfaccia utente e delle modalità di gioco, sono state descritte ampiamente in altre tesi. In questo documento invece si preferirà delineare le modalità e sistemi aggiunti a quelli già esistenti, oltre delineare il funzionamento di quelli modificati per la modalità multiutente.

3.2.1 Decostruzione dell'esperienza

Sulla base del materiale raccolto ed in accordo con la PC si sono delineate quattro 'fasi' in cui dividere l'esperienza, sia per le simulazioni guidate che per quella valutata utente singolo e multiutente.

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 auto-adescente 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 0: Vestizione dei DPI e sopralluogo

La prima parte prevede che utenti indossino i rispettivi Dispositivi di Protezione Individuale (DPI), comprendenti casco e salvagente dotato di moschettone; la squadra poi deve svolgere un sopralluogo della zona per decidere le zone d'aspirazione, mandata e posizionamento della pompa più adeguate.

Fase 1: Allestimento campo operativo

Si prosegue con il posizionamento della pompa e delle manichette in un'area ottimale, utilizzando un puntatore. Successivamente si procede con la delimitazione dell'area operativa con paletti in legno e nastro di segnalazione, seguito dal posizionamento del telo di plastica protettivo degli argini, fissandolo con i picchetti al terreno.

Fase 2: Stesura manichette

Ora gli operatori della squadra, a coppie, montano le manichette di aspirazione e di scarico (mandata); la catena di aspirazione è composta in genere da tubi semirigidi partendo dalla pompa fino alla zona dove si è verificato il versamento. Gli ultimi elementi ad essere incorporati sono il filtro e la corda per il suo recupero, poi si procede all'immersione del capo nella pozza (con l'operatore agganciato a 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 dei picchetti, per prevenirne movimenti erratici al fluire dell'acqua.

Fase 3: Avviamento, monitoraggio e smontaggio

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 (Figura 3.2). Nel caso di perdita la pompa deve essere spenta il più velocemente possibile. 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).

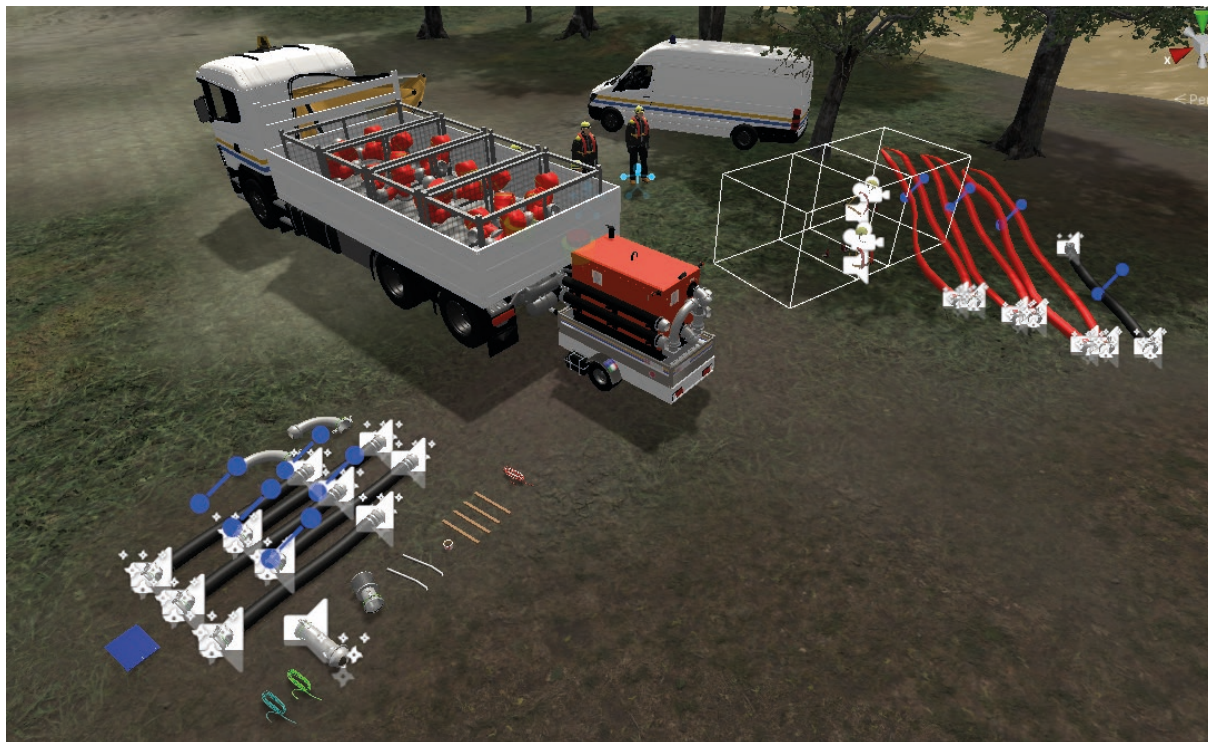


3.2 Motopompa Godwin in funzione

3.2.2 Pianificazione delle risorse necessarie

In seguito, ci si è interrogati su quali modelli e sistemi si sarebbero dovuti creare. Per quanto riguarda i modelli 3D, sarebbero serviti (Figura 3.3):

- motopompa Godwin CD 150m, con carro di rimorchio per il trasporto e con dispositivi di stabilizzazione incorporati;
- camion con gru (per il trasporto delle manichette appiattibili e dei tubi lunghi), modello già fornito dalla PC;
- officina mobile (per il trasporto dell'attrezzatura minore), modello già fornito dalla PC;
- manichette appiattibili e semirigide (per le operazioni di trasporto e scarico dell'acqua);
- tubi rigidi curvi e dritti (per il pompaggio dell'acqua in presenza di ostacoli, dossi e curve strette);
- strumento di supporto alla rotazione della leva delle aperture, come un tubo ricurvo o una chiave inglese (per aiutare gli operatori nella fase di aggancio delle manichette);
- filtro pescante (per l'aspirazione dell'acqua nella zona allagata);
- telo blu e picchetti (per la protezione della zona di mandata dal getto d'acqua erogato dalla motopompa);
- paletti e nastro (per la delimitazione dell'area operativa);
- casco e salvagente con moschettone indossabili;
- corda di sicurezza, corda di recupero del filtro e corda di fissaggio per l'ultimo tubo della mandata;
- operatori umani (NPC);
- ambiente in cui si sarebbe svolta la simulazione.



3.3 Scenario all'interno di Unity con tutti i modelli presenti

Invece per quanto riguarda i sistemi si sarebbero dovuti implementare:

1. locomozione dell'avatar dell'utente in VR;
2. presa e trasporto di oggetti;
3. navigazione e logica degli operatori NPC;
4. simulazione fisica delle manichette semirigide e appiattibili;
5. meccanismo di chiusura e aggancio tra manichette;
6. gestore del flusso di gioco e delle attività;
7. gestore delle connessioni e script di comunicazione tra client e server;
8. sistema di valutazione;
9. interfaccia utente e menu

In questo documento vedremo la realizzazione dettagliata per i punti 4, 5, 7 e 9. Il resto verrà illustrato brevemente, in quanto discusso nelle tesi precedenti (per quanto riguarda il punto 7 si è deciso di realizzare una simulazione online con solo due utenti).

3.3 Plugin di supporto

Mentre si stavano concependo i vari strumenti, si era consci che implementare una soluzione da zero avrebbe richiesto molto tempo; pertanto, per alcuni aspetti ci si è affidati a funzionalità già esistenti, aggiunte mediante plugin (alcuni anche a pagamento, per i quali erano disponibili opportune licenze) ed integrate nel flusso di lavoro.

3.4 Obi

Obi è un plugin che racchiude un motore di simulazione fisica di particelle CPU-based in tempo reale per Unity, che presenta un framework unificato per tessuti interattivi, fluidi, corde e softbody, disponibili in quattro asset chiamati rispettivamente Obi Cloth, Obi Fluid, Obi Rope e Obi Softbody. Utilizza gli ultimi ritrovamenti in ambito di ricerca e tecnologia all'avanguardia, strutturato con un'architettura estensibile e modulare che lo rendono estremamente performante, anche grazie a un solver multithread.[29] è stato realizzato dallo studio spagnolo Virtual Method[30]. Nello specifico, si è adoperata la versione 4.3 del plugin.

La particolarità di Obi è che tutte le simulazioni e gli oggetti creati sono ottenute mediante interazioni tra particelle, invece che interagire con mesh e deformazioni. Le particelle si influenzano a vicenda e possono essere influenzate dagli altri oggetti attraverso l'uso di vincoli. Assieme a Obi, che gestisce l'allocazione delle particelle e il setup dei vincoli, si affianca Oni, una libreria indipendente dall'engine scritta in C++ che calcola la simulazione fisica. In questa struttura, Obi agisce quindi da interfaccia tra Unity e Oni. Obi poi è formato da tre componenti principali:

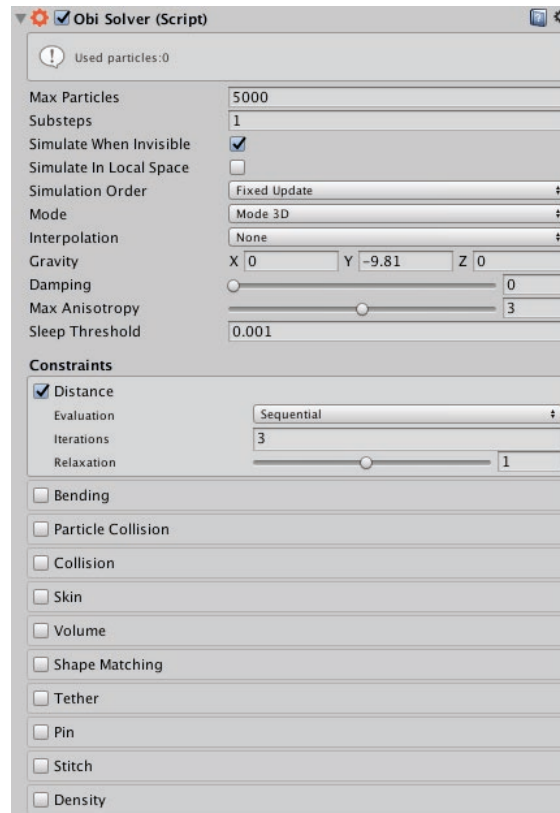
- **ObiSolver:** Interfaccia principale che espone i parametri della simulazione fisica all'utente e che permette di attivare/disattivare il calcolo di determinati vincoli e con quanti step di risoluzione. Va aggiunto alla scena.
- **ObiActor:** Classe base per ogni entità che desidera usare la fisica delle particelle, e che prende dei parametri di input e deve essere inizializzato manualmente, generando così le sue particelle e resettando i vincoli.
- **ObiConstraints:** Classe base che gestisce i vincoli per ciascun gruppo di particelle, e che è poi implementata in diverse tipologie che vanno aggiunte all'ObiSolver per essere calcolate.

Una volta aggiunti questi elementi alla scena e impostati i parametri, la pipeline di simulazione prevede che l'Obi Actor prelevi un certo numero di particelle dall'Obi Solver e imposti le loro proprietà iniziali. Poi chiede ai propri componenti ObiConstraint di generare i vincoli di diverso tipo tra loro, e a loro volta ogni singolo componente ObiConstraint invia la richiesta all'ObiSolver di mettere in atto i rispettivi vincoli. Poi, ObiSolver inoltra tutte le computazioni a Oni, che simula tutte le particelle attive e vincoli, senza sapere a chi appartengano, permettendo quindi di essere slegato dalle definizioni di Obi [31].

Verranno ora illustrati nel dettaglio questi tre componenti; per quanto riguarda l'ObiActor, si farà riferimento unicamente a quello usato, ObiRope.

3.4.1 ObiSolver

L'ObiSolver è il componente principale per la simulazione, che deve conoscere tutti gli ObiActor su cui dovrà agire, per permettere loro di gestire i loro vincoli e interagire tra di loro, anche tra ObiActor di natura diversa. Può essere aggiunto a qualsiasi GameObject nella scena, e più ObiSolver possono coesistere nella stessa scena, ma ObiActor assegnati a diversi ObiSolver non possono interagire tra loro.



3.4 Parametri dell'ObiSolver

I parametri presenti sono quelli riportati di seguito (Figura 3.4).

1. **Max Particles:** Indica il numero di particelle che il solver può gestire contemporaneamente e di conseguenza la memoria allocata, considerando tutti gli ObiActor che si sottoscrivono prenderanno in prestito un certo numero di queste particelle quando necessario.
2. **Substep:** Il numero di volte che ObiRope viene aggiornato con ciascuno step fisico. Il valore di default è 1, con 2,3 e 4 lasciato per simulazioni più accurate, causando però un maggior calcolo computazionale. È equivalente a ridurre il fixed timestep del Time Manager di Unity.
3. **Simulate when invisible:** Indica se l'ObiSolver deve continuare le simulazioni dei vari ObiActor anche quando non sono inclusi nel frustum di vista della camera di Unity.
4. **Simulate in local space:** Decide se simulare le posizioni e velocità delle particelle avendo come riferimento l'origine del mondo (più semplice, ma può dare problemi se gli attori sono molto distanti da essa e non può controllare le forze) oppure la posizione del solver stesso (più complesso e richiede di imparentare gli attori al solver, ma permette un controllo più preciso).
5. **Simulation order:** indica se la simulazione venga calcolata durante Fixed Update di Unity, dopo o durante Late Update.

6. Mode: 3D o 2D (solo sul piano XY).
7. Interpolation: modalità per simulare le particelle, ovvero nessuna interpolazione (più veloce) o interpolate (migliore resa).
8. Gravity: direzione e velocità della forza di gravità applicata alle particelle di questo solver.
9. Damping: effetto di smorzamento sulle velocità delle particelle.
10. Max anisotropy: il massimo rapporto tra i raggi dell'ellissoide con cui le singole particelle sono costruite. Il valore di default, 1, indicherà particelle sferiche, mentre uno maggiore permette alle particelle di essere schiacciate e adattarsi meglio alle collisioni, ottimo per le simulazioni di fluidi.
11. Sleep threshold: indica una soglia di energia cinetica sotto la quale la particella sarà considerata ferma e non reagirà a variazioni più piccole di questa.
12. Constraint: ObiSolver presenta una lista di vincoli che possono essere attivati o meno. Nel paragrafo successivo saranno descritte le varie tipologie, ma qui si indicano brevemente i loro parametri comuni.
 - a. Enabled: se i vincoli di questa tipologia vanno inclusi nella simulazione o meno, se spenti o se nessun attore li utilizza impatteranno meno a livello di prestazioni
 - b. Iterations: quante volte per substep il vincolo deve essere valutato, più alto sarà più realistico sarà il risultato, più basso più le prestazioni saranno migliori, con valore di default 3.
 - c. Evaluation order: il modo in cui i vincoli sono valutati. Se impostato come Sequential mode, i vincoli vengono risolti nell'ordine in cui sono stati creati e partono dalle modifiche realizzate da quelli prima di loro. Converge più rapidamente a una soluzione corretta con un minor numero di interazioni, ma è meno stabile. Invece, se si è scelto Parallel mode, tutti i vincoli sono valutati assieme ma non applicati immediatamente alle particelle, e invece sono mediati con tutti i risultati delle successive iterazioni. Garantisce una simulazione più stabile, ma sono necessari più iterazioni, e di conseguenza un calo di prestazioni.
 - d. SOR factor: per migliorare la velocità di convergenza, si può permettere un fattore di rilassamento fino al doppio del valore di default (1, ovvero nessun rilassamento), ma può causare problemi di stabilità.

3.4.2 ObiConstraints

Ogni ObiActor che vuole simulare un particolare comportamento deve aggiungere il relativo componente al proprio GameObject. In generale ogni template di ObiActor (Rope, Cloth, Softbody

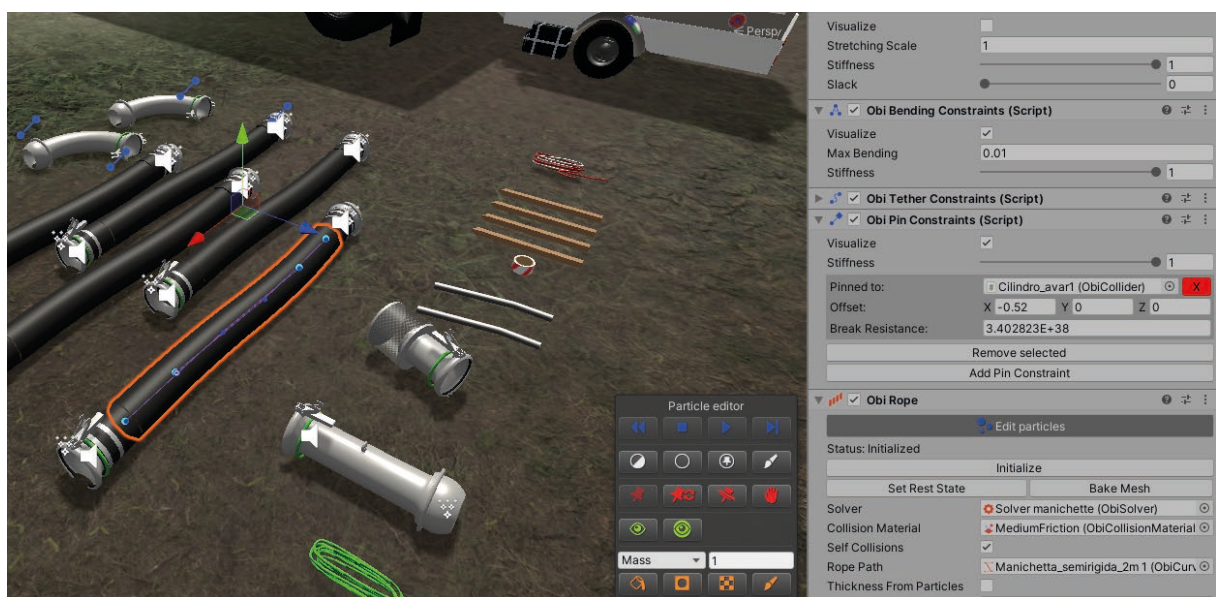
e Fluid) ha già aggiunto i vincoli che servono per simulare il relativo oggetto. A ogni frame, Obi sposta le particelle secondo le proprie velocità e poi modifica le loro posizioni per soddisfare tutti i vincoli contemporaneamente, con diverse iterazioni che si avvicinano al risultato ottimo. L'elenco dei constraint, in ordine di risoluzione, sono:

- **Distance Constraint:** l'obiettivo di questo vincolo è di tenere due particelle a una distanza fissa l'una dall'altra. Con un numero di interazioni basse, si restituirà un comportamento elastico, mentre con un numero alto sarà più rigido. Includono poi altri parametri come la scala di allungamento, la rigidità e la resistenza alla compressione.
- **Bending Constraint:** questo vincolo lavora su trii di particelle, cercando di allineare quella centrale alle altre due lungo una linea retta, ed è spesso usato in tessuti e corde per resistere ai piegamenti. Un numero alto di iterazioni permette di raggiungere una rigidità maggiore. Il parametro di max bending indica quanta piega è permessa prima che il vincolo sia attivato.
- **Collision Constraint:** si occupano di calcolare le collisioni tra le particelle e i collider degli oggetti nella scena. Per essere rilevato, all'oggetto basta aggiungere un component ObiCollider (che copia le informazioni dei normali collider di Unity) e impostare il parametro Phase a un numero diverso dello stesso parametro delle particelle dell'ObiActor con cui vogliamo interagire. Le collisioni sono "speculative" e calcolate prima che l'attuale collisione avvenga, tuttavia, risolvere i punti di contatto in maniera sequenziale può causare alcuni punti di ritornare a essere collisioni e dare un aspetto 'tremolante' al risultato visivo.
- **Particle Collision Constraint:** come per i vincoli di collisione, questi calcolano le collisioni ma tra particelle di ObiActor diversi ma sempre gestiti dallo stesso solver
- **Skin Constraint:** questa tipologia di vincoli vengono usate per tenere in forma gli abiti e ogni altro tessuto animato mediante rig. Possono essere usati solo in congiunzione con Skinned-MeshRenderer
- **Volume Constraint:** ciascun vincolo di questo tipo prende un gruppo di particelle posizionate ai vertici di una mesh e cerca di mantenere il volume della mesh. Usato per simulare corpi morbidi e palloncini, laddove un numero maggiore di interazioni permette una pressione interna maggiore e mantenere meglio il volume. Possiede inoltre un parametro pressure, che indica quale percentuale dello spazio originale deve occupare.
- **Shape Matching Constraint:** questo vincolo collega un gruppo di particelle assieme definito "cluster" e cerca di mantenerne la stessa posizione e orientamento l'una relativa all'altra, mentre permette al gruppo nel suo insieme di traslare e ruotare. Può essere impostato per simulare una deformazione elastica o più rigida, e altri parametri che invece stabiliscono quando da deformazione elastica si passi a una deformazione plastica e quanto tempo serve per tornare allo stato originario, se possibile.
- **Tether Constraint:** Simili ai vincoli di distanza, ma usati per evitare tensioni eccessive in lunghe catene di Distance constraint, che possono accumulare errori se non sottoposti a un alto numero di iterazioni che diventano particolarmente evidenti alla fine della catena. Devono essere costruiti tra particelle "fisse" e quelle libere di muoversi.

- **Pin Constraint:** applica forze a una coppia particella-rigidbody (con ObiCollider) così che mantengano la loro relativa posizione e si influenzino a vicenda. A differenza di tutti gli altri vincoli, questi non sono creati automaticamente dall'ObiActor ma devono essere impostati manualmente dal programmatore, specificando il loro offset dall'origine del rigidbody calcolato in coordinate locali. La particella e il rigidbody cercano poi di spostarsi per fare in modo che la particella selezionata si sposti nel punto indicato. Se per caso questo fosse all'interno del volume del rigidbody, bisognerà impostare entrambi sulla stessa fase in modo da ignorarne le collisioni e non causare jittering.
- **Density Constraint:** usati nei fluidi, servono a mantenere la massa in un determinato intorno di particelle simile a una certa costante, evitando incrementi o diminuzioni di questo valore di massa, con numeri alti di iterazioni rendendo i fluidi meno compressibili e quindi meno simili alla gelatina.

3.4.3 ObiRope

ObiRope è un ObiActor generato come una linea di particelle collegate tra loro che le forzano a stare una certa distanza l'una dall'altra (Figura 3.5). Le particelle all'interno della corda mantengono informazioni solo sulla posizione, non sulla rotazione reciproca (in quel caso si avrebbe un ObiRod). Per generare una corda, basta aggiungere il component ObiRope a un oggetto, e assieme a esso verranno aggiunti questi altri componenti: ObiCurve, ObiRope Renderer, e Distance, Bending, Tether e Pin Constraint.



3.5 Editing delle particelle di ObiRope durante la simulazione

ObiCurve è la curva di riferimento che la corda utilizza nel proprio parametro RopePath per generare il proprio percorso iniziale. In generale ogni corda deve avere un component ObiCurve separato, altrimenti verrebbero generate l'una sopra l'altra. Per manipolare la corda si aggiungono/rimuovono/spostano i punti di controllo come si fa in una normale curva di Bézier.

Il Renderer è il component che si occupa di disegnare la corda nell'ambiente, e in base al tipo può generarla come:

- estrusione continua di una sezione fornita dal programmatore;
- un lineRenderer, simile a un nastro sempre rivolto verso la camera (molto leggero dal punto di vista prestazionale);
- una catena, composta da prefab negli spazi tra particelle che simulano gli anelli di una catena e di cui si può indicare anche una torsione di ogni anello;
- una lista di mesh personalizzate disposte lungo la lunghezza della catena, il metodo più flessibile ma anche più esoso dal punto di vista computazionale.

Il component ObiRope ha inoltre diversi parametri che ne gestiscono il comportamento e l'aspetto:

1. Pooled particles: ammontare di particelle extra che la corda può usare in caso venga tagliata o allungata.
2. Rope Path: Spline usata per creare la forma iniziale della corda.
3. Thickness: raggio delle particelle usate per generare la corda.
4. Resolution: Specifica quante particelle verranno usate per generare la corda. Un valore di 1 indica lasciare uno spazio pari al valore thickness, mentre 0.5 il doppio di quel valore, lasciando che le particelle si tocchino. Valori minori lasciano gap più ampi tra l'una e l'altra.
5. Smoothing: indica se vanno aggiunti edge loop extra tra particelle per simulare una corda più liscia
6. Tearable: indica se si vuole rendere la corda tagliabile.
7. Tear resistance multiplier: quanta tensione deve essere applicata alla corda prima che si rompa.
8. Start ed End prefab: istanzia un prefab ai capi della corda, che però sono solo rappresentazioni visuali e non interagiscono con la corda – altrimenti vanno usati i pin constraint.
9. Section twist: torsione in gradi applicata a ciascuna sezione della corda.

Questo conclude la spiegazione delle funzionalità di Obi per quanto concerne le funzionalità all'interno della simulazione. Nel capitolo successivo si vedranno poi come sono stati impostati questi parametri.

3.5 Mirror

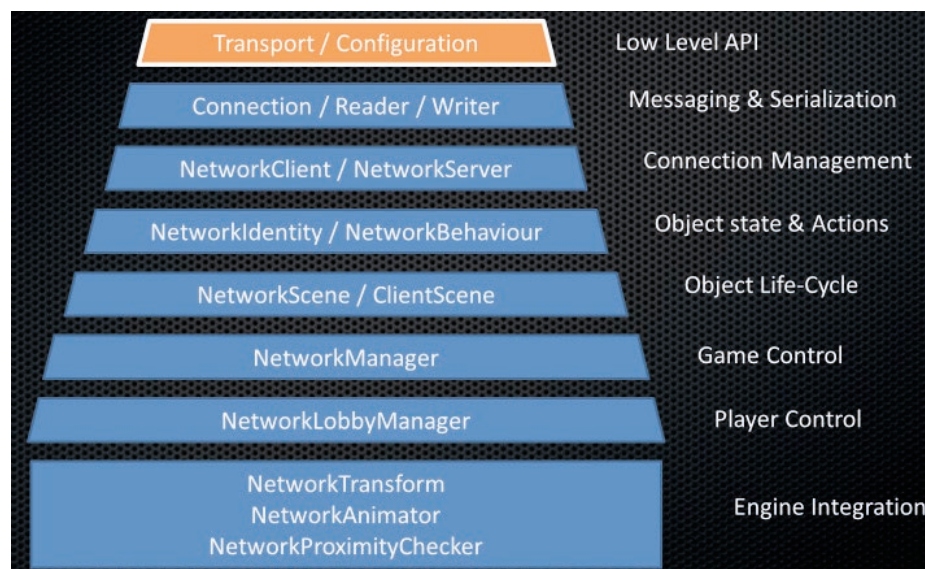
Mirror è un'API Networking di alto livello di Unity totalmente gratuita e open source, che supporta diversi protocolli di trasporto di basso livello e che è ottimizzata per facilità di utilizzo ed esperienze multigiocatore ad alto numero di partecipanti.[32], [33].

A seguito della discontinuità di UNET, la libreria built-in di Unity per gestire le connessioni, la squadra dietro Mirror ha realizzato una libreria che ne calcasse il funzionamento e che fosse di maggior semplice utilizzo e stabile. È costruito sullo strato di comunicazione e trasporto in tempo reale di basso livello e gestisce molte delle interazioni comuni necessarie per giochi multigiocatore, prevedendo però un'architettura server-authoritative (in cui però un client può agire anche da server allo stesso tempo).

Fornisce funzionalità quali:

- handler dei messaggi;
- serializzazione general-purpose ad alte prestazioni;
- gestione di oggetti distribuita;
- sincronizzazione dello stato;
- classi network.

Mirror include il codice per gestire server e client sulla stessa macchina, senza dover creare codice separato per i due. Il server è un'istanza di gioco a cui tutti gli altri giocatori si connettono e che tiene traccia di diversi stati e trasmette le variazioni di dati ai client connessi. Può essere un server dedicato oppure un server host, in cui un client crea un'istanza dell'applicazione che funge sia da server e da client. Invece le istanze client normalmente si connettono da macchine diverse da quella dove corre il server e questa può avvenire su rete locale o da remoto (Figura 3.6).



3.6 Piramide dei componenti in Mirror

3.5.1 Network Manager

Il Network Manager è il nucleo del gioco multiplayer ed è realizzato interamente con le funzioni dell'API, quindi offre diverse funzionalità in un unico component, che può essere eventualmente espanso per permettere una flessibilità maggiore.

Permette di specificare il tipo di protocollo di trasporto come component separato dal Network Manager in base alle esigenze dell'applicazione e del programmatore, ad esempio TCP, UDP, WebGL e Steam. Può essere lanciato in tre modalità – come client, server dedicato o host – e si connette all'indirizzo e porta indicati dai campi Network Address e Transport Port se client, o ascoltare sulla porta indicata se host/server.

Gestisce lo spawn dei player prefab, ovvero di quegli oggetti associati al giocatore che utilizza per interagire con l'ambiente e che vengono istanziati ogniqualvolta un nuovo client si connette alla scena, previa registrazione del prefab nella lista di possibile prefab spawnabili. Lo spawn utilizzerà le coordinate di posizione e rotazione salvate nella transform del prefab, ma se si specifica un metodo di spawn (Random o Round Robin) assieme ai componenti Network Start Position nella scena.

La gestione delle scene di base prevede la definizione di una scena offline e una scena online. Quando il server o l'host viene avviato, la scena online viene caricata e ogni client che si connette ha l'obbligo di caricare quella stessa scena. Quando invece si interrompe la connessione, sia che sia il server a essere fermato o per una disconnessione del client, si carica invece la scena offline. Eventualmente è anche possibile spostarsi tra diverse scene online mediante il metodo `ServerChangeScene`, che sposta tutti i client attualmente connessi alla nuova scena e distrugge tutti gli oggetti della scena precedente. È importante che l'oggetto con sopra il Network Manager abbia la proprietà `Don't Destroy on Load` così che possa sopravvivere al passaggio di scene.

È possibile personalizzare ulteriormente la funzionalità del Network Manager implementando i metodi virtuali in una classe derivata e avendo premura di espletare le funzionalità fornite dall'implementazione base.

3.5.2 Network Identity

Il component Network Identity è il cuore dell'API di alto livello. Assegna a ogni oggetto sulla rete un identificativo univoco e usa quell'identificativo per essere riconosciuto dal sistema di rete. Ha un unico parametro, `Server Only`, che se selezionato impone che quell'oggetto sia istanziato solo sul server e non sui client.

Gli oggetti con una Network Identity possono essere spawnati mediante istanziazione, e in quel caso devo essere registrati nel Network Manager come prefab, oppure siano caricati assieme alla scena. In quest'ultimo caso, Unity disabilita tutti gli oggetti con questo componente, e non appena un client si connette al server, il server risponde al client quali oggetti della scena abilitare e quali sono le loro informazioni di sistema aggiornate. Questo per assicurare la coerenza tra la scena tra

client e server.

Inoltre, con ogni Network Identity vengono generati tre identificativi: l'Asset Id, che utilizza il codice GUID di Unity per conoscere il percorso su disco dell'oggetto, lo Scene Id, un uint che viene assegnato durante la `OnPostProcessScene` e che permette di differenziare gli oggetti di una scena tra loro (cosa che Unity non ha), la Network Instance Id, che viene usata come passaggio di parametro nelle chiamate remote tra client e server, e per ultima la Player Controller ID che identifica a quale player è assegnato questo oggetto.

3.5.3 Authority

Il concetto di Authority è un metodo per decidere chi possiede un oggetto e chi ne ha il controllo. Di default il server ha autorità su ogni oggetto che non sia quello usato dal giocatore, in quel caso si parla di Client Authority. Se un client ha autorità su un oggetto (e di default ha autorità sull'oggetto con cui viene generato, il cui termine tecnico è “spawn”) può effettuare chiamate remote al server da quell'oggetto e se si disconnette l'oggetto verrà distrutto assieme al player. L'autorità su un oggetto può essere concessa in diversi modi e rimossa, ma non si può rimuovere l'autorità che il client ha sul suo player Object, se non rimpiazzandolo.

3.5.4 Network Behaviour

Sostituisce il `MonoBehaviour` di Unity su oggetti che hanno una Network Identity e permette loro di effettuare chiamate remote di alto livello. Può avere accesso a variabili sincronizzate, callback di rete, funzioni di server e client, inviare comandi e ricevere chiamate RPC. Come proprietà invece ha informazioni se l'oggetto è avviato sul server, se è sul client, se rappresenta l'oggetto creato per il giocatore, se il client ha autorità su quell'oggetto, la `netId` e la Network Identity dell'oggetto, la connessione al server (se su client) e la connessione al client (validi su oggetti del giocatore sul server).

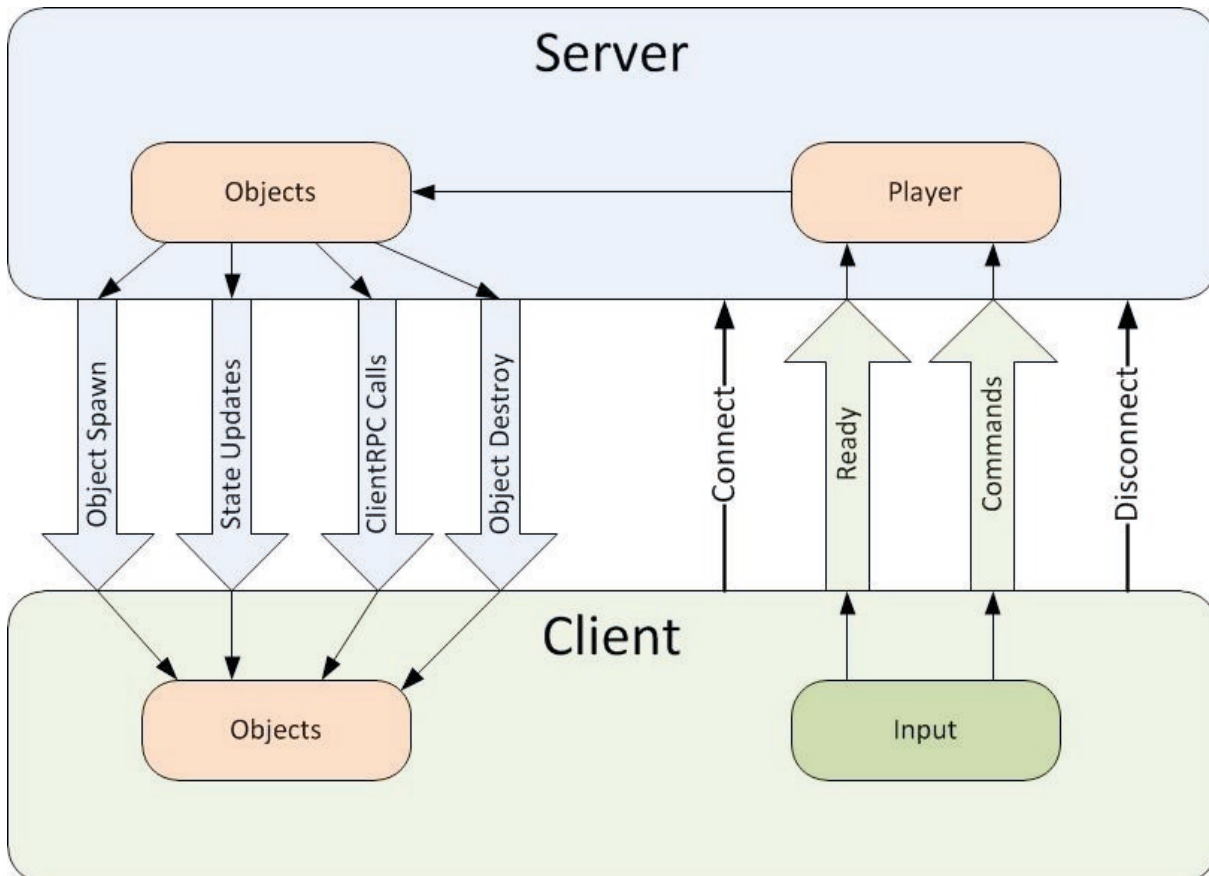
3.5.5 Comunicazioni

Ci sono diversi metodi con cui il server e client possono comunicare tra loro (Figura 3.7).

- Azioni remote: Chiamate anche Remote Procedure Calls (RPC), ce ne sono di due tipologie, le Command, che sono chiamate dal client ed eseguite sul server, e le Client RPC che invece sono chiamate dal server ed eseguite sul client. Le Command normalmente possono essere solo inviate dal proprio oggetto giocatore, ma si può ignorare con un opportuno attributo o se si assegna l'autorità del client a quell'oggetto. Le RPC verso i client possono sempre essere inviate in quanto il server ha sempre autorità su ogni oggetto; ce ne sono di due tipi, le Client RPC, che vengono inviate a ogni client, se invece vogliono essere inviati a una connessione specifica si utilizzerà una Target RPC.
- Callback del Network Manager: queste funzioni vengono chiamate quando si verificano determinati eventi durante un gioco multiutente, come l'avvio dell'host, l'unione o l'abbandono di un giocatore.
- Callback del Network Behaviour: simile a quelle precedenti, ma si può specificare il compor-

tamento per singolo oggetto derivato da NetworkBehaviour in risposta a questi eventi.

- Messaggi Network: una forma di comunicazione più a basso livello, che permette di inviare dati direttamente mediante scripting tra client e server. Si possono inviare tipi di dati base di sistema e di Unity.



3.7 Schema delle comunicazioni tra client e server

3.5.6 Network Transform e Network Transform Child

Questi component sincronizzano le informazioni di posizione, rotazione e scala degli oggetti con Network Identity attraverso la rete. Di default è server authoritative, ma se si spunta il parametro Client Authority allora i cambi di posizione sono inviate dal client al server. Si possono impostare poi i parametri di sensibilità per indicare la soglia minima per generare un messaggio, la modalità di sincronizzazione (ovvero se le modifiche devono essere propagate agli altri) e l'intervallo di sincronizzazione. Quelle con child servono per la posizione di eventuali figli con posizioni variabili rispetto al padre.

3.5.7 Protocolli di trasporto

Mirror ha il supporto di tre protocolli standard di trasporto: TCP, realizzata mediante la libreria Telepathy, UDP, mediante la Low Level API di Unet, e Websocket mediante Ninja.Websockets.

3.6 Altri strumenti

In questa sezione sono stati raccolti altri strumenti che sono stati già descritti nelle altre tesi che hanno preceduto il presente lavoro.

3.6.1 OpenVR

Valve ha reso inoltre disponibile un plugin chiamato Open VR [34] e scaricabile dallo store di Unity, utile ad interfacciare il programma direttamente con Steam VR, lo strumento per interfacciare i visori VR con la nota piattaforma. 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.

3.6.2 Final IK

Per gestire la cinematica inversa degli NPC si è utilizzato Final IK, una libreria di script prodotta dalla RootMotion, molto diffusa tra gli sviluppatori che utilizzano questo sistema grafico. Final IK è stato utilizzato per il controllo delle mani del modello, in modo da rendere più realistico il posizionamento delle dita sull'oggetto preso e di semplificare il lavoro, limitando la creazione delle animazioni ai movimenti di corsa, camminata e idle (stato di riposo dell'NPC).[35]

4. Realizzazione

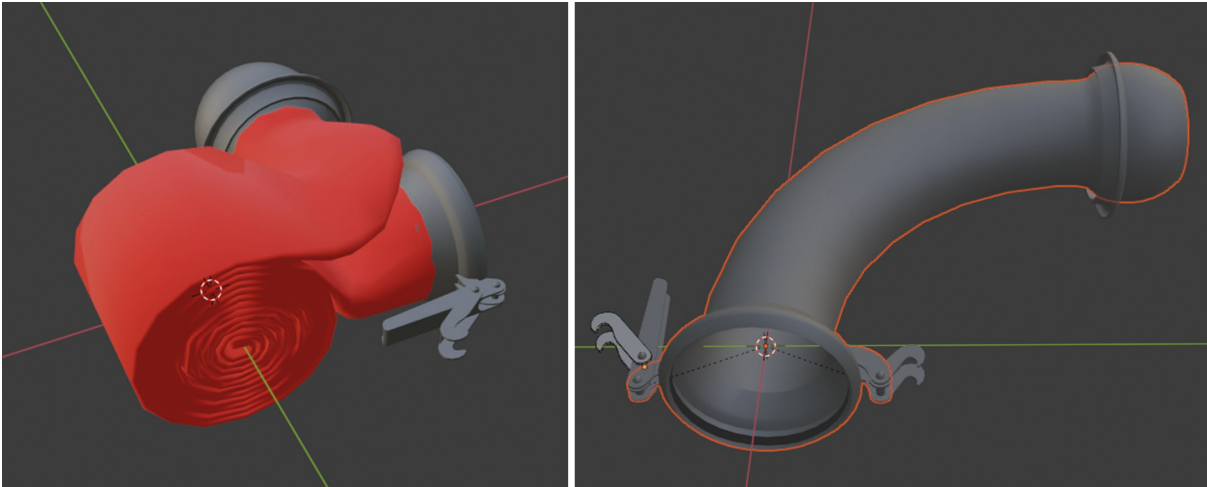
Questo capitolo è dedicato a illustrare il processo di traduzione delle idee raccolte durante la progettazione in oggetti di gioco concreti utilizzabili dagli utenti. In particolare, ci si soffermerà sulla fisica e il comportamento logico degli strumenti centrali a questa simulazione (le manichette e i tubi) che sono stati adoperati anche nella versione utente singolo dell'applicazione, e poi la conversione dell'intera esperienza in un progetto network-compliant. Gli altri elementi dell'applicazione (il sistema di interazione VR, la FSM di automazione degli NPC, la modellazione dell'ambiente di gioco e la logica di voice over e valutazione) sono stati analizzati nel dettaglio nelle tesi precedenti e ci si limiterà quindi a una breve descrizione delle caratteristiche principali.

4.1 Simulazione fisica degli strumenti

Come precedentemente descritto, la PC impiega tre tipologie di tubature durante le esercitazioni col modulo HCP. Il sistema atto a far interagire questi strumenti tra loro ha richiesto diverse sperimentazioni e iterazioni nella loro realizzazione, fino a raggiungere quella che è l'implementazione attuale.

4.1.1 Modellazione delle manichette e logica base in Unity

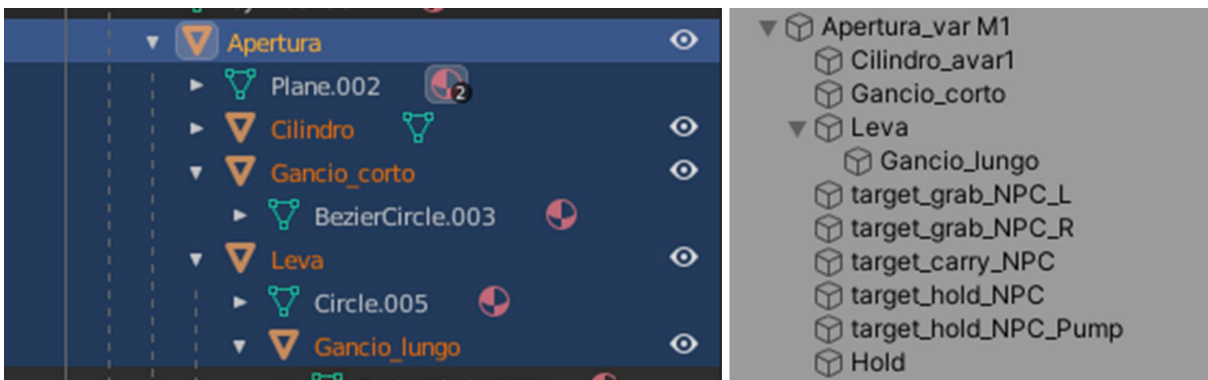
Tramite il software di Blender sono state realizzate le manichette, prendendo a modello quelle della PC. In particolare, si è cercato di rispettare le proporzioni reali per quanto riguardava lunghezza, diametro e forma. Per le manichette semirigide e quelle appiattibili si sono costruiti due modelli: uno “scenico”, utilizzato nelle situazioni iniziali soltanto come elemento di scena e senza alcuna logica di gioco applicata, mentre il modello di gioco è in realtà diviso in due: uno per l'estremità contenente la femmina (la forma a cono con il meccanismo di chiusura di ganci e leva) e uno per il maschio (la forma a sfera con l'anello di aggancio). Si è avuta l'accortezza di avere il pivot dell'oggetto al centro del diametro di ambo le estremità. A questa si aggiunge una breve sezione della manichetta, sia essa rossa o nera, che serve come punto di appoggio per ObiRope. Per quanto riguarda i tubi rigidi, si è preferito invece realizzarli con un'unica mesh che connettesse le due estremità, invece che tenere le due separate. Il filtro invece ha una sola estremità, una femmina, siccome sarà l'ultimo elemento della catena di aspirazione (Figura 4.1).



4.1 Esempio di modello scenico manichette appiattibili e modello tubo curvo rigido

Una volta importati in Unity, è stata rispettata la stessa gerarchia che si era impostata nel file Blender per quanto riguardava la suddivisione del modello (Figura 4.2). Sul livello più alto della gerarchia sono stati piazzati tutti gli script di interazione, mentre su quelli inferiori si sono andati ad aggiungere funzionalità minori (spesso integrative per quelle superiori). Tra questi ci sono i collider per l'interazione e rigidbody (laddove il collider per le collisioni col terreno si trova sul cilindro), gli script per poter essere afferrato dal giocatore e dagli NPC, oltre a Network Identity, Network Transform e Network Transform Child e Animator per elementi quali i ganci e leva.

Gli script però fondamentali per tutte le funzionalità sono essenzialmente due: lo script Apertura e lo script Boccia, che sono collocati rispettivamente sulle estremità femmina e maschio di ogni manichetta, tubo e filtro. Essi sanno quale altro appartiene alla stessa unità “logica” (ovvero lo stesso tubo) e quando ne toccano uno del tipo opposto di un altro tubo. Si vedrà meglio il loro funzionamento nel paragrafo 4.1.3.



4.2 Gerarchia apertura in Blender e Unity

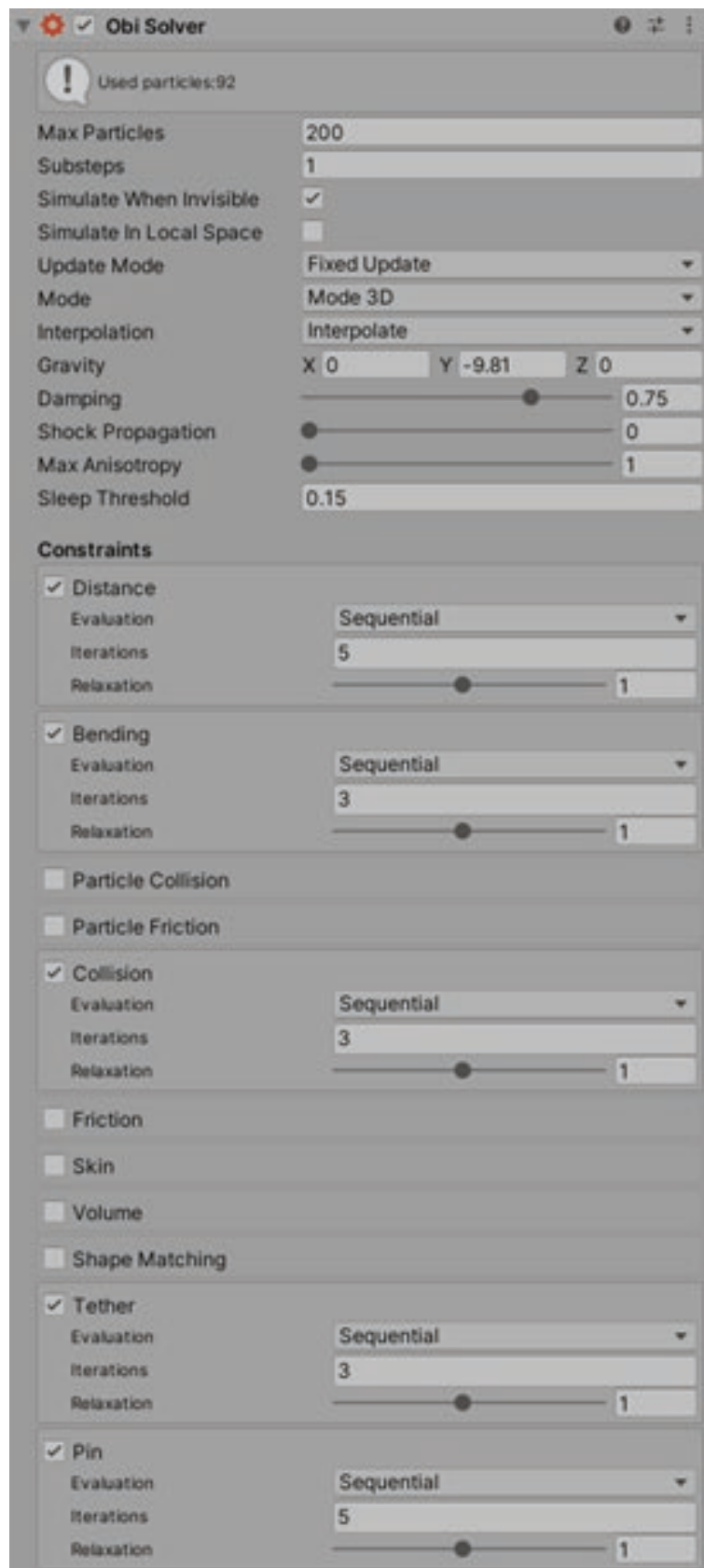
4.1.2 Integrazione con Obi Rope

Sebbene come unità logica una coppia di estremità femmina-maschio siano legate, bisognava renderle collegate anche da un punto di vista fisico, laddove non lo fossero già come nel caso dei tubi rigidi e del filtro pescante. Si sono quindi aggiunte le manichette per connettere i due estremi, realizzante mediante il plugin di ObiRope. Per primo si sono realizzati i prototipi di una manichetta semirigida da 2m, poi quella da 4m e poi quelle da 20 m rossa appiattibile. Per poterle collegare si è aggiunto un Obi RigidBody sulla cima della gerarchia delle estremità e un Obi Collider sul cilindro sottostante, impostato alla stessa fase delle particelle, ovvero 1. In base alla lunghezza della manichetta e al numero di particelle da cui è composta, si sono impostati gli offset dei Pin Constraint da questo ObiCollider, siccome essendo elementi separati questa impostazione non poteva essere salvata come prefab. Ogni altro oggetto contro cui le manichette possono collidere, come il terreno, la pompa, ecc. è stato dotato di ObiCollider con fase 0.

All'inizio si è utilizzata una Rope Section a 24 punti di controllo, messa a forma di C chiusa, in modo tale che simulasse sia il rivestimento esterno che interno quando osservato lungo il foro delle estremità, ma poi si è optato per una più semplice sezione a 8 punti in modo che risultasse di minor impatto dal punto di vista computazionale. Per le manichette rosse si è usata una sezione più schiacciata a rappresentare la sezione quando vuota.

Dato che le manichette semirigide hanno un basso grado di flessibilità, per simularle si è abbassato molto il parametro Resolution di ObiRope, così che il vincolo di Bending possa gestire molte meno particelle e simulare al meglio questa scarsa flessibilità. In aggiunta si sono impostati i Tether Constraint per aiutare l'intera struttura a essere coesa e non estensibile. Invece per le manichette rosse, siccome possono essere piegate e avvolte su loro stesse, non si sono abilitati i vincoli di Bending e Tether.

L'impostazione del solver ha previsto un utilizzo di 200 particelle (numero più che sufficiente per le esigenze dell'applicazione), la necessità di continuare a simulare la fisica delle manichette anche quando non inquadrare, un alto grado di damping e sleep threshold per simulare il peso e l'inerzia di queste manichette. A livello di vincoli si sono tenuti attivi solo quelli necessari a ObiRope per funzionare, ovvero i Distance, Bending, Collision, Tether e Pin Constraint (Figura 4.3).



4.3 Impostazione dell'ObiSolver in game

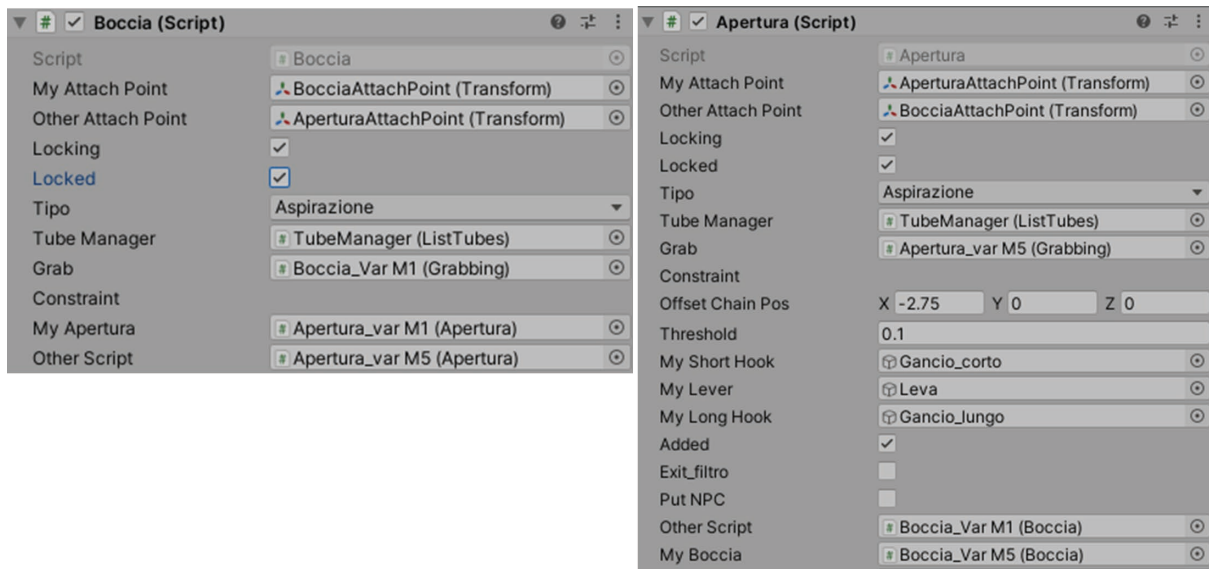
4.1.3 Meccanismo di aggancio

Come accennato precedentemente, la meccanica di aggancio inizia quando un'estremità avente lo script Apertura viene avvicinato a un'estremità con lo script Boccia (per comodità ci si riferirà a questi GameObject direttamente come apertura e boccia d'ora in poi) e sorretto con entrambe le mani di un utente, a simulare le procedure di sicurezza che la PC deve adoperare quando si appresta a collegare due manichette tra loro. Se si verificano queste condizioni, l'apertura e la boccia entrano in fase di locking, ovvero sono pronte a essere agganciate. Si attivano gli Animator dei ganci e della leva, portando i ganci vicino all'anello della boccia e la leva in alto. Sia l'apertura che la boccia cambiano lo stato del proprio Rigidbody in kinematic, in modo che non si spostino. A questo punto, possono verificarsi due casi: se l'utente toglie le mani o tira indietro l'apertura dalla boccia, le due estremità resettano il loro stato e tornano allo stato di scollegate. Il secondo caso è che un utente operatore proceda con l'operazione di aggancio e prenda lo strumento adibito alla chiusura e lo infili sulla leva alzata (Figura 4.4). In quel caso lo strumento, tramite il proprio script Allineamento, attiva il proprio component Parent Constraint e simula di essere imparentato alla leva. A questo punto, però, per essere girato dipende dalla rotazione attorno al pivot della leva, e tramite lo script calcola la differenza tra la sua posizione attuale e quella della mano di chi lo sta adoperando. La differenza in gradi è inviata all'apertura che ruota la leva dei gradi necessari. Se raggiunge l'estremità alta, lo strumento si scollega e le estremità rimangono nella precaria situazione di locking.



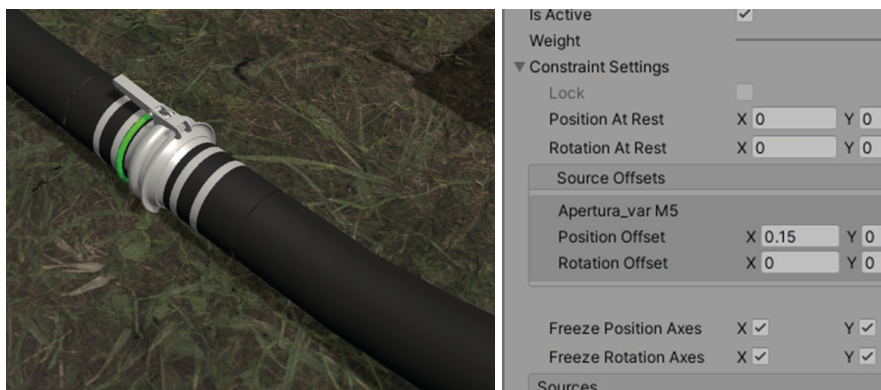
4.4 Esempio di aggancio della manichetta alla pompa con supporto da parte del client

Se invece la leva viene abbassata fino a raggiungere una certa soglia, lo strumento si scollega ma ora l'apertura e la boccia passano alla fase successiva, lo stato di locked. In questa fase si considerano agganciate e non possono essere staccate (Figura 4.5). Questo “aggancio” è reso con un Parent Constraint sulla boccia (ma in alcuni casi anche sull'apertura, in caso delle bocce che partono dalla pompa o dei tubi rigidi) che segue la Transform dell'altra estremità sia per posizione e rotazione (Figura 4.6).



4.5 Boccia_var M1 (sx) e Apertura_var M5(dx) sono agganciate tra loro, visibile dal campo Other-script

Dopo è possibile spostare lentamente la coppia di estremità agganciate in quanto il rigidbody dell'apertura torna non kinematic, questo per permettere operazioni di correzione della catena di tubi che si viene a creare. Eventualmente è poi possibile scollegare queste due estremità procedendo in senso inverso: reinserendo lo strumento nella leva abbassata e poi alzandola indica alle due estremità di tornare alla fase di locking, e da qui possono essere separate normalmente.

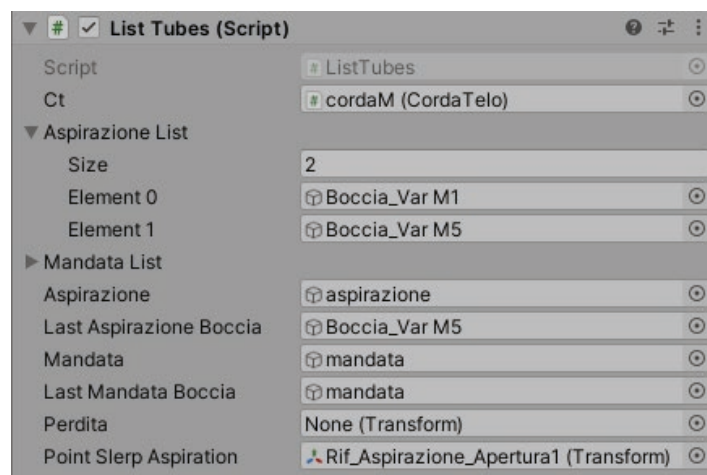


4.6 Manichette agganciate in-game e Parent Constraint Boccia_var M1

4.1.4 Gestione manichette

Dato che si sarebbero dovuto gestire due catene separate di manichette (una per l'aspirazione, una per la mandata) sarebbe servito un sistema di controllo che indicasse se gli utenti si fossero dimenticati di chiudere alcune estremità tra loro o avessero sbagliato la tipologia di manichetta da impiegare (per esempio usare le manichette rosse lungo l'aspirazione, dove l'eccessiva pressione spaccerebbe il tubo).

A questo proposito è stato realizzato lo script ListTubes che tiene traccia di questi eventi (Figura 4.7). Quando un'apertura viene accostata a una boccia, viene chiamato il metodo AddTube alla catena relativa e quella manichetta/tubo viene aggiunta alla lista (e rimossa tramite RemoveTube quando separate). Se vengono agganciate si salvano anche queste informazioni e poi ListTubes le fornisce allo script di valutazione il quale segna se ci sono stati eventuali errori durante la costruzione della catena.

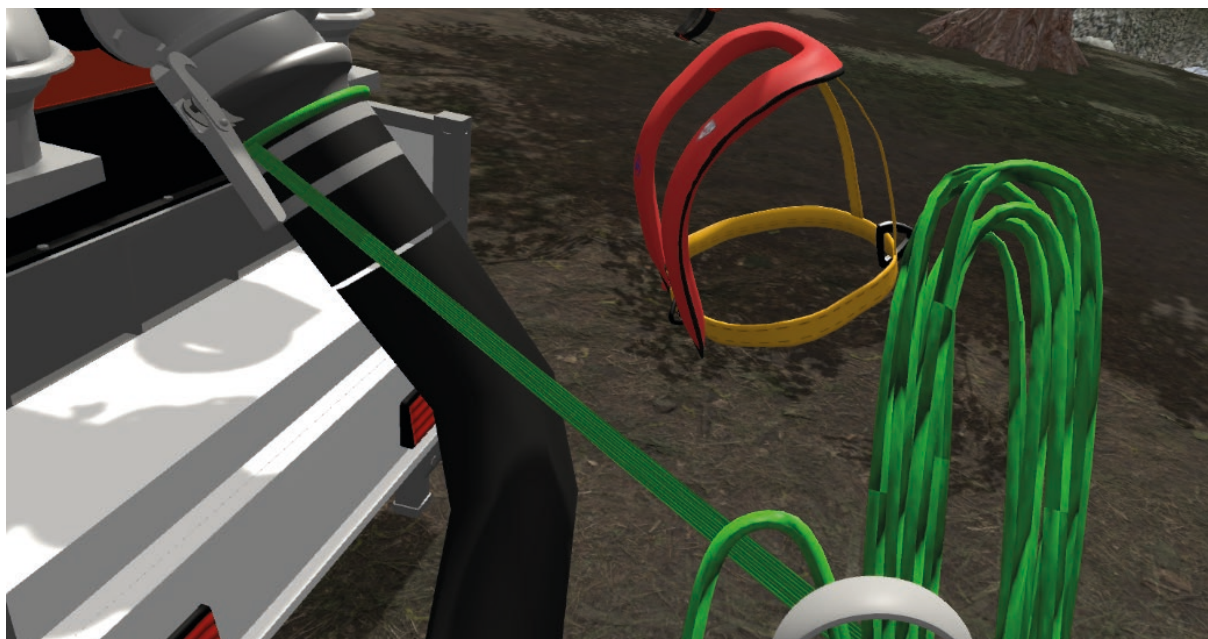


4.7 List Tubes con la catena di aspirazione formata dalle due manichette M1 e M5

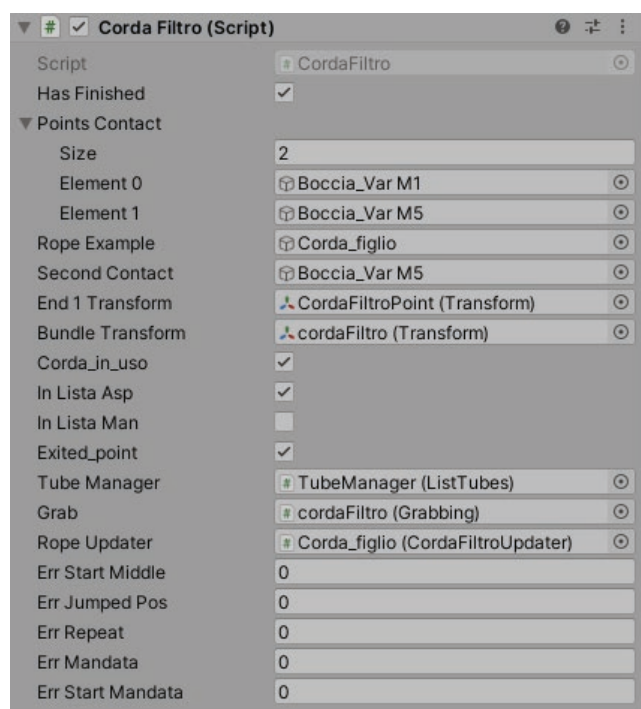
4.1.5 Corda del filtro

Una delle ultime funzionalità aggiunte è stata quella di poter fissare una corda alle manichette della catena di aspirazione in modo che una volta che il filtro è stato inserito in acqua sia possibile recuperarlo da parte degli operatori. Per evitare di appesantire ulteriormente la simulazione con altre istanziazioni di ObiRope che avrebbero dovuto essere generate a run-time, si è deciso di usare il component Line Renderer di Unity, come è stato fatto anche per le altre corde presenti nella simulazione, quella di sicurezza e il nastro dei paletti di delimitazione. A tutti i tubi è stato aggiunto un anello vicino all'estremità femmina e un piccolo collider al di sotto della leva dove può essere passata questa corda con il relativo script CordaFiltro (Figura 4.8). Non appena rileva questo collider, genera un Line Renderer tra esso e la corda, e ogni nuovo punto che incontra viene aggiunto ai punti di controllo del Line Renderer. Se si ripassa sull'ultimo punto si considererà tolto e la corda verrà aggiornata di conseguenza. L'unico modo per chiudere la corda è collegare all'interno dei

punti la prima apertura dell'aspirazione e quello del filtro pescante (o l'ultimo elemento della catena di aspirazione, se questo non fosse il filtro). La corda tiene traccia dell'ordine con cui si percorre la catena (si appoggia a ListTubes), se si saltano dei capi o se si include un capo della catena di mandata (Figura 4.9).



4.8 Tiraggio della corda del filtro



4.9 Script della corda del filtro con i parametri di valutazione

4.2 Sincronizzazione tra utenti

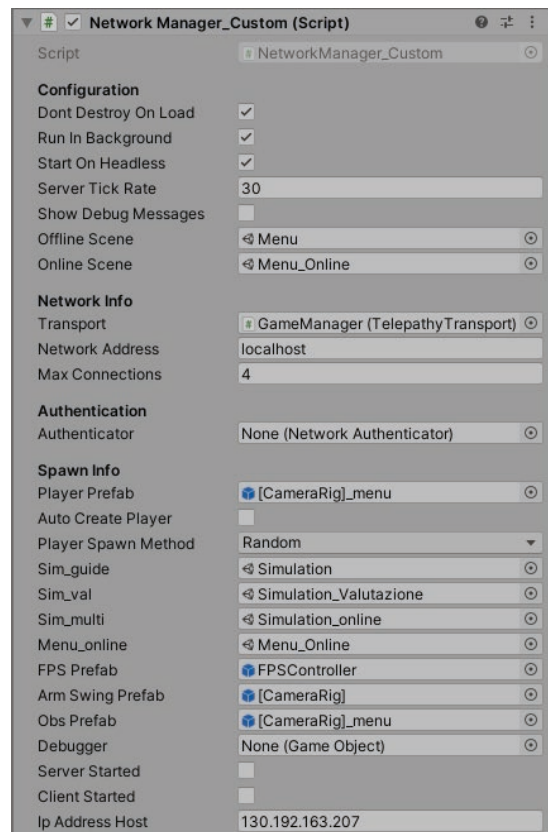
Usando le funzionalità di Mirror si è deciso di realizzare la simulazione multiutente a due utenti, avendo un utente che ospita sia il server che una propria istanza di client (ovvero avviando il server in modalità host). Durante la realizzazione della versione precedente dell'applicazione ci fu un abbozzo di implementazione di questa funzionalità, ma solo in questa versione si è completato l'obiettivo. Ha richiesto diverse modifiche, in quanto molti degli oggetti e sistemi sottostanti non erano stati concepiti per essere network-compliant. Di seguito si illustrano suddette modifiche.

4.2.1 NetworkManager_Custom

Per questa versione personalizzata del Network Manager sono stati reimplementate alcune callback fornite dal sistema, oltre ad aggiungere nuovi metodi (Figura 4.10).

1. `OnStartServer()`: quando il server viene avviato, oltre alla normale funzionalità si registra un handler per una funzione di lettura di messaggi personalizzati, chiamati `CustomNetworkMessage`, che contengono un solo campo, una public string `msg`.
2. `OnClientConnect(NetworkConnection)`: quando un client si collega al server (anche l'host si connette alla propria istanza) esegue le normali funzionalità e poi invia un messaggio lungo la connessione che indica i seguenti parametri: che tipo di controller ha selezionato l'utente (se FPS o cameraRig VR), la scena che deve caricare e il ruolo nella costruzione di quale catena (se mandata o aspirazione). Una volta inviato il messaggio il client viene messo in stato di Ready, ovvero vengono abilitate tutte i suoi oggetti dotati di `NetworkIdentity`.
3. `ReadMessage(NetworkConnection, CustomNetworkMessage)`: funzione che fa parsing del messaggio e istanzia i player prefab che i client dovranno utilizzare, e se trova una `StartPosition`, li posiziona in quel punto. Si salva anche le informazioni riguardo la connessione e il loro id appena istanziato. Se inoltre ha controllato che gli utenti connessi sono due, abilita alcune funzionalità del Menu Online.
4. `OnServerDisconnect(NetworkConnection)`: callback invocata quando la connessione di un client viene terminata e la notifica arriva al server. In questo caso, si effettuano alcune funzionalità di aggiornamento se ci si trova nella scena del Menu Online che resettano le scelte fatte fino a quel punto, mentre se ci si disconnette dalla simulazione si attiva il component `Pagella` del `GameManager` per segnalare gli errori compiuti fino a quel momento. Dopodiché si rimuove dalla lista delle connessioni accettate la connessione appena abbandonata e poi ferma il client del server, effettivamente fermando l'host. Questo perché si è deciso che se uno dei due utenti si fosse disconnesso, anche l'altro avrebbe dovuto terminare la propria connessione. Sebbene questo sia il comportamento di default per il client se l'host si disconnette, il contrario non era implementato.

5. OnStopServer(): Quando il server si ferma, si resettano tutti i parametri della connessione.
6. OnClientDisconnect(): callback invocata sul client quando si disconnette dal server. Come funzionalità extra si salvano i progressi ed errori fatti fino a quel momento sulla pagella del Game-Manager e poi viene fermato il client e in secondo luogo anche il server.



4.10 Visione da edito di Network Manager Custom

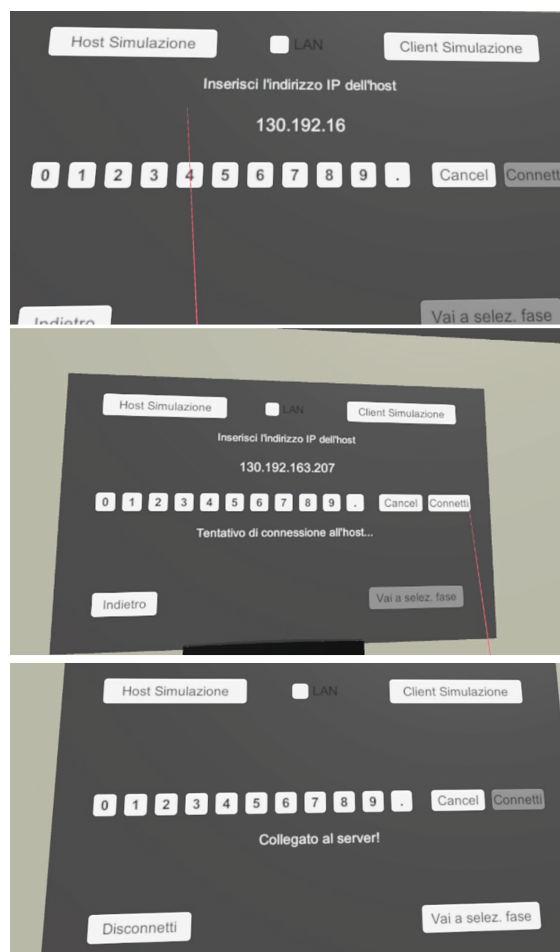
4.2.2 Menu Online

Prima di essere connesso alla simulazione online, si è deciso che i due utenti si incontrassero in una lobby in cui potessero osservare in tempo reale le proprie decisioni su che mansioni svolgere, definendo questa scena intermedia il Menu Online. Per potervi accedere si seleziona tramite puntatore laser l'ultima opzione dello splash screen iniziale, e si arriva in una schermata in cui sono disponibili due pulsanti: Host Simulation e Client Simulation. In base a quale dei due utenti vuole ospitare il server, questi premerà il pulsante Host Simulation e il sistema interrogherà la sua interfaccia di rete: tramite un apposito Toggle è possibile decidere se si vuole conoscere l'indirizzo IP esterno (WAN) oppure interno della rete locale (LAN), in base da dove si connetta il client. Dopo tre secondi, questo utente viene connesso alla scena online del menu mentre aspetta che un client si connetta (Figura 4.11).



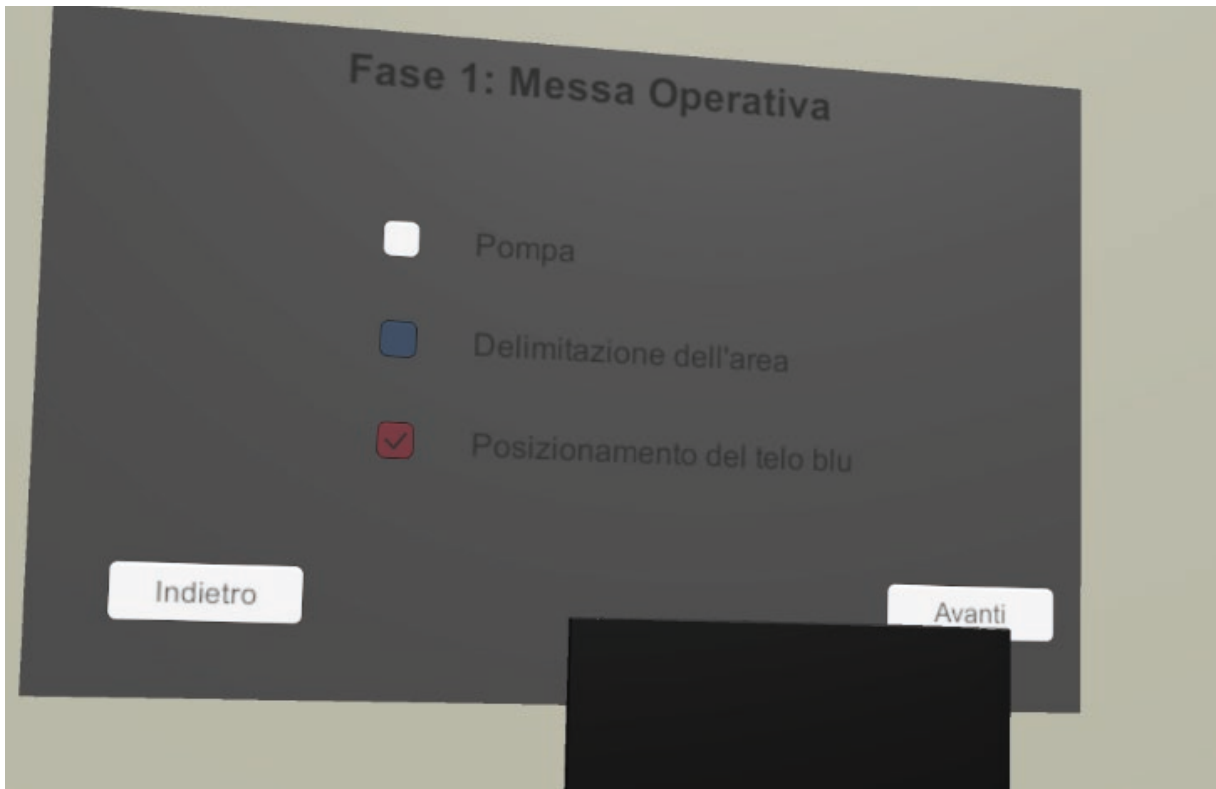
4.11 Processo di connessione dell' host al menu online

Se invece si preme il pulsante Client Simulation, apparirà un tastierino numerico che permetterà al client di inserire l'indirizzo IP a cui si deve connettere. La porta invece non deve essere specificata in quanto si utilizza quella di default a cui Mirror si connette, 7777. Se l'indirizzo IP è valido e la porta è aperta il client si conatterà alla stessa istanza del menu online in cui l'host è presente. In qualsiasi momento è possibile premere il pulsante di disconnessione per tornare alla schermata precedente e disconnettersi dalla stanza comune. Altrimenti, si può procedere alla selezione delle attività, il cui pulsante è disabilitato fino a che due utenti non sono connessi contemporaneamente (Figura 4.12).



4.12 Processo di connessione del client al menu online

Qui la selezione delle attività è molto simile a quella delle simulazioni a utente singolo, e sebbene gli utenti possano scorrere le schermate liberamente (a patto che selezionino almeno una delle attività per pagina), la differenza è che ogni utente è notificato di quale scelta l'altro abbia fatto in tempo reale. Nella prima pagina, che elenca le attività della fase 1 e che quindi possono essere eseguite da un singolo utente, vige un principio di esclusività, ovvero se uno tra host e client effettua una scelta, quella non sarà disponibile per l'altro. Le attività che selezionerà l'host saranno colorate in rosso, quelle del client in blu (Figura 4.13). Nella seconda e terza schermata invece la selezione dovrà essere comune per forzare l'aspetto interattivo e collaborativo della simulazione; se ad esempio l'host dovesse selezionare che vuole lavorare alla catena di aspirazione, la scelta di lavorare sulla catena di mandata sarebbe disabilitata su entrambi i menu degli utenti. Ciò ovviamente è frutto dell'implementazione corrente, ma potrebbe essere modificato se necessario.



4.13 Esempio della simulazione esclusiva

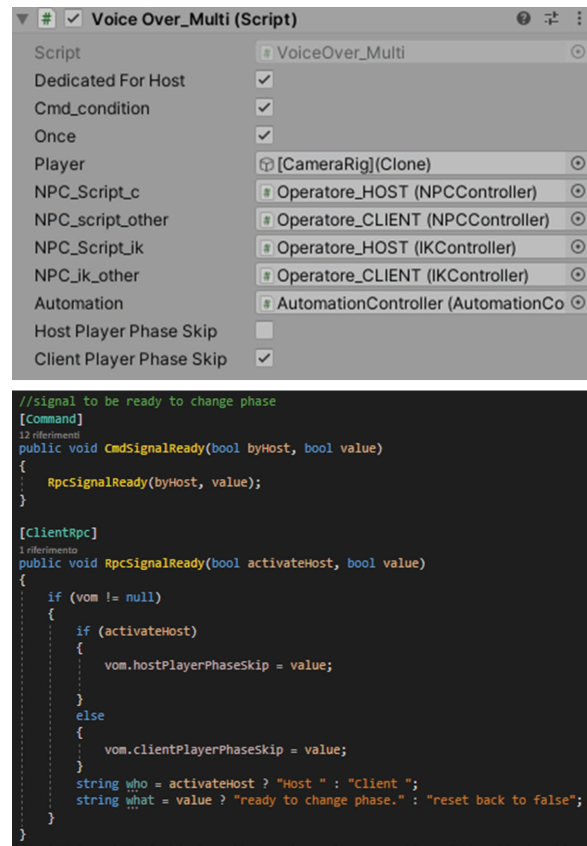
Una volta che si sono effettuate anche le scelte per la terza fase, la schermata finale si presenta differente per client e host. L'host ha accesso al pulsante che avvia la simulazione, mentre il client no. Tuttavia, il pulsante è disabilitato fino a che il client non ha finito di selezionare le proprie attività, e solo una volta che entrambi hanno confermato la simulazione multiutente potrà essere avviata.

4.2.3 Modifiche per la simulazione online

Per apportare le modifiche al comportamento della simulazione in accordo con gli obiettivi del documento di design per la parte multiutente (promuovere la cooperazione tra gli utenti e farli interagire), si è disabilitata la FSM degli NPC associati al singolo utente, che ora hanno come unico compito quello di fornire agli operatori giocanti i propri DPI e non li aiutano più a trasportare i tubi. Per favorire l'ulteriore grado di cooperazione, si è richiesto che uno degli utenti sorregga il tubo mentre lo si sta agganciando, come l'NPC di supporto faceva durante le simulazioni singolo utente.

Si è lavorato su una versione alternativa del voice-over, chiamata VoiceOverMulti, che agisce in maniera differente riguardo lo svolgimento delle attività. È un componente offline, ma l'aspetto di sincronizzazione è garantito dal fatto che per passare da una fase all'altra sia il client che il server devono acconsentire tramite una chiamata RPC, eseguita premendo contemporaneamente i pad di entrambi i controller di un singolo utente verso l'alto. Tuttavia, non è necessario che questa con-

ferma sia fatta nello stesso istante da ambo le parti (Figura 4.14). Molti dei passaggi tra una fase e l'altra sono stati modificati per tenere traccia delle modifiche online.



4.14 Esempio in editor e codice del passaggio di fase sincronizzato

Un'altra modifica è stata quella concernente il sistema di interazione: la versione singolo utente prevedeva il parenting alla mano dell'utente che raccoglieva l'oggetto. Dato che non è possibile spostare dalla cima della gerarchia un elemento dotato di Network Identity, si è scelto di mantenere l'offset di posizione e rotazione mediante chiamate a Command e Client RPC fino a che permane il trasporto dell'oggetto. Per identificare in maniera univoca chi lo ha afferrato, il sistema salva il numero di connessione di chi effettua il grabbing. Per i sistemi in cui l'interazione consiste di un mero utilizzo del trigger per attivare un'animazione, come nel caso delle porte, la chiave e i rubinetti di svuotamento dei serbatoi della girante e del filtro, si è invece fatto ereditare tutte le classi che fungevano da controllore da uno script comune con il metodo virtuale HandleInteraction, che poi ognuna di queste ha reimplementato secondo le proprie necessità.

Si è inoltre deciso di evitare il posizionamento procedurale dei tubi e della delimitazione dell'area operativa in base alla posizione della pompa e avere invece un'area specifica dove manichette e oggetti vengono rivelati quando si passa dalla fase di sopralluogo alla fase 1, piuttosto che aspettare la comparsa della pompa.

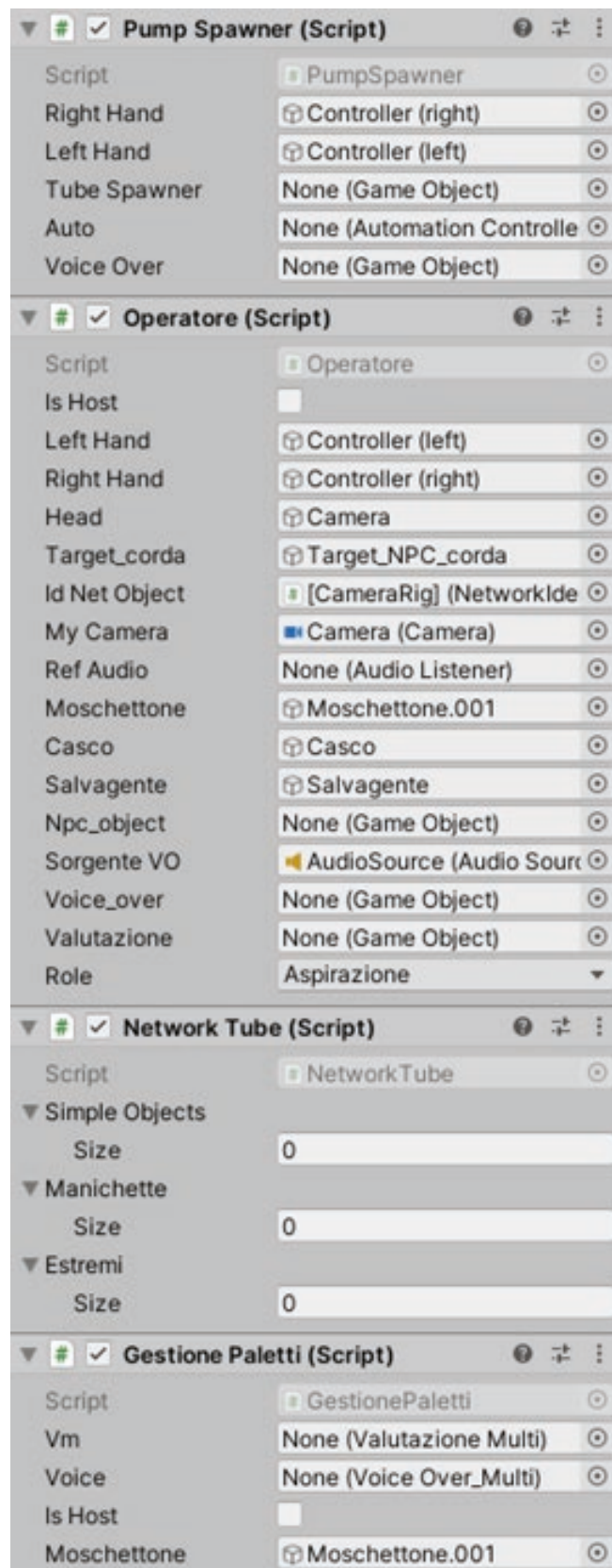
4.2.4 Script per la comunicazione client-server

Si sono utilizzati diversi script sul prefab del player per gestire le comunicazioni di rete durante la simulazione: PumpSpawner, Operatore, GestionePaletti e NetworkTube, mentre quando è mostrato il menu online viene impiegato lo script MenuVRToggle. Il primo è dedicato a calcolare la posizione e rotazione del puntatore della pompa e poi spostarne il modello per entrambi i giocatori nel luogo scelto.

Lo script Operatore invece si occupa di gestire le interazioni tra utente e oggetti (interazioni e deinterazioni, grabbing e ungrabbing) oltre ai metodi Cmd/RpcLockingApertura, Cmd/RpcLockedApertura, Cmd/RpcLockingBoccia e Cmd/RpcLockedBoccia che cambiano in maniera network compliant lo stato degli script Apertura e Boccia, facendoli passare dallo stato scollegato a locking, da locking a locked e viceversa. Anche lo script di aggiornamento della rotazione dello strumento usato per chiudere/aprire la leva.

GestionePaletti serve principalmente a far funzionare alcune attività della fase 1, come l'inserimento dei paletti nel terreno e i picchetti negli occhielli del telo blu. Permette inoltre di sincronizzare tutte le azioni che prevedono l'uso di un LineRenderer, ovvero il nastro della delimitazione, la corda di sicurezza per il posizionamento del filtro in acqua, la corda per assicurare la catena di aspirazione e quella per fissare l'ultimo elemento della catena di mandata al telo blu.

Lo script NetworkTube si occupa di rivelare gli oggetti con i quali i giocatori dovranno andare a interagire e impostare i PinConstraint delle manichette semirigide e appiattibili. Per quanto riguarda lo script MenuVRToggle, lo scopo è segnalare le scelte effettuate da client e server quando è mostrato il menu online a entrambi (Figura 4.15).



4.15 Elenco degli script Network Behaviour sul player prefab

4.2.5 Elementi di valutazione specifici

La precedente simulazione valutata era concepita per analizzare il comportamento e le azioni di un singolo utente, quindi sebbene parte del codice potesse essere replicato, andava espanso con ulteriori casi che tenessero conto delle criticità che si presentano durante una simulazione multiutente. Questi consistono di ulteriori controlli che si attivino in caso un utente porti a termine un'attività che non gli compete al posto dell'altro, magari rischiando di compromettere l'esecuzione dell'attività (ad esempio fissare il telo nel luogo sbagliato o delimitare una zona non consona). Per quanto riguarda le attività in comune, queste vengono valutate complessivamente; pertanto un errore da parte di un utente se non corretto dall'altro prima della valutazione sarà segnata negativamente per entrambi.

4.3 Altri sistemi

4.3.1 Sistema di interazione

All'interno della simulazione, usando una combinazione del plugin OpenVR, ArmSwinging e del Raycasting è stato possibile realizzare tutte le possibili azioni che gli utenti sono in grado di compiere nella simulazione, come:

- sollevamento, trasporto, rotazione e rilascio degli oggetti mediante il pulsante Trigger del controller;
- interazioni per l'apertura porte, valvole e rotazione chiave tramite pulsante Trigger;
- interazione con i menu: raycasting per posizione e pulsante Trigger per conferma;
- spawn pompa: raycasting per posizionamento, Trackpad destro per rotazione, Trigger per conferma;
- segnalazione spegnimento: pulsante Trigger con sollevamento del braccio al di sopra della testa;
- locomozione mediante ArmSwinging.

4.3.2 Logica di automazione degli NPC

Durante tutte le tipologie di simulazioni è fondamentale il ruolo svolto dagli NPC: si occupano delle mansioni non svolte dagli utenti per quanto riguarda la coppia di NPC automatizzati, mentre durante le simulazioni singolo utente si aggiunge la funzionalità dell'NPC di supporto alle azioni dell'utente. Essi sono stati realizzati da un prototipo umanoide creato con MakeHuman[36], gestiti mediante Animator e Final IK per quanto riguarda le animazioni dei loro movimenti, Interaction Point e Navmesh per navigare nell'ambiente e raggiungere punti di interesse e una Finite State Machine per gestirne gli stati dell'intelligenza artificiale. In particolare, gli stati che sono stati codificati includono:

- Idle: stato di riposo dell'NPC;
- Follow_player: stato in cui l'NPC segue l'utente;
- Reach_target: stato in cui l'NPC raggiunge un punto specifico nella scena;
- Grab_tube: stato in cui l'NPC deve raggiungere il capo del tubo da sollevare;

- Carry_tube: stato in cui l’NPC deve trasportare il tubo insieme all’utente;
- Drop_tube: stato in cui l’NPC deve posizionare il capo a terra prima di sostenere l’altro capo per la fase di aggancio;
- Hold_tube: stato in cui l’NPC deve sostenere il capo del tubo durante la fase di aggancio.

4.3.3 Ambiente di gioco

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 strumenti e asset, riportati di seguito.

- Terrain[37]: tool incorporato a Unity che permette di creare e poi modificare uno scenario naturale. Lo strumento partendo da un piano crea un ambiente tridimensionale sul quale il modellatore può andare a compiere un’operazione di sculpting utilizzando un pennello che crea rilievi e valli. Tramite il pennello è anche possibile disporre una o più texture sovrapposte sul terreno oltre che alberi e rocce.
- Cascade[38]: l’asset 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 i parametri quali numero di particelle, velocità di movimento, frequenza e velocità di generazione.

4.3.4 Voice over e valutazione

Dal momento che gli operatori della PC operano scambiandosi informazioni a voce o utilizzando un walkie talkie con altri gruppi di volontari, si è deciso di utilizzare un voice-over con effetti radio, per poter rendere più immediata la comunicazione tra utente e sistema e per mantenere un grado di realismo nella simulazione. Il voice-over è stato progettato a partire da un testo appositamente scritto per la simulazione costituito da una serie di frasi raccordate tra di loro, le quali sono state generate da un programma di sintesi vocale online. Il voice-over per la versione guidata indica all’utente con cosa interagire non solo con stimoli audio ma anche icone e suggerimenti visivi. Ciascuna frase riporta una serie di informazioni oppure un’indicazione su come procedere durante le operazioni ed è stata convertita in clip audio trasmessa attraverso una sorgente sonora disposta nella scena virtuale. Nelle modalità valutate poi 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.

5. Conclusione e sviluppi futuri

Questo capitolo è dedicato all'analisi delle difficoltà, i punti di pregio e i futuri aggiornamenti che questo lavoro ha raccolto durante la sua realizzazione.

5.1 Conclusioni

Questo lavoro di tesi è stato particolarmente complesso ed elaborato in successive iterazioni. Una prima versione è stata completata nel luglio 2020, e presentava una versione preliminare dell'applicazione VR in cui erano presenti due simulazioni completabili da un solo utente. Data la situazione globale della pandemia del Covid-19, la usuale fase di sperimentazione, raccolta e analisi dati non era stata effettuata. Durante questa tesi è stato possibile effettuare delle sperimentazioni su quella versione, ma i risultati ottenuti non sono lo scopo di questo documento.

La versione corrente dell'applicazione, completata a luglio 2021, è stata la culminazione dell'idea originale, che mirava ad offrire la possibilità di eseguire una versione valutata dell'applicazione in modalità multiutente.

Alcuni problemi riscontrati nella versione precedente sono stati parzialmente risolti.

- Basso framerate: nella vecchia versione si verificavano a volte bruschi picchi di tempo di calcolo che portavano la simulazione a scendere ben al di sotto della soglia dei 90 fps al secondo, necessari per avere un'esperienza VR fluida. Sebbene su PC di vecchia generazione in modalità Editor si presentino ancora framerate bassi, questi migliorano notevolmente in modalità eseguibile e su macchine più recenti anche la versione editor è fluida. Questo si è ottenuto analizzando meglio gli script presenti ed evitando chiamate a callback di sistema inutili, limitando l'aspetto grafico laddove possibile (l'enorme scenario impatta molto sulle prestazioni di rendering) e cercando di contenere l'impatto che la simulazione fisica di Obi richiedeva.
- Mancanza dello scenario multiutente, che era parte del progetto originale, ma non implementato con i precedenti lavori di tesi.
- Miglior gestione delle manichette: dover rendere il meccanismo di chiusura network-compliant ha portato a una ristrutturazione del suo funzionamento che ne ha migliorato l'affidabilità e stabilità.

Tuttavia, uno studio più in dettaglio delle meccaniche ha portato alla luce altri problemi che prima non erano stati notati.

- Bassa stabilità e mantenibilità del codice: purtroppo lavorando a più mani, in tempi diversi su componenti differenti dell'applicazione che però dovevano interagire tra loro, e avendo avuto poco tempo per provare il tutto con l'unico visore VR che si aveva a disposizione data la pandemia, ha portato ad alcuni casi di mancata separazione delle funzionalità e scopi degli script (anche dovuti ad inesperienza sulle buone pratiche nel campo), causando enormi sforzi e perdite di tempo in un secondo momento quando si sono rimaneggiati quegli stessi script.
- Informazioni a volte poco esplicative: durante la simulazione guidata alcune linee audio o indizi visivi non scendevano nel dettaglio sulle effettive motivazioni del perché si dovesse effettuare quella determinata azione, la quale non era stata inclusa per non allungare ulteriormente la durata della simulazione che risulta sempre molto provante nella sua forma più lenta (la versione guidata appunto). Sarebbe ideale correggere queste linee ed esplicitare meglio questi punti oscuri, ascoltando i feedback dei volontari della PC.
- Scarso playtesting: Questa nota è purtroppo particolarmente dolente perché è alla base della maggior parte dei problemi dell'applicazione. Senza un ciclo di feedback adeguato, purtroppo, molte funzionalità non sono state testate in casi limite volti a compromettere il corretto funzionamento dell'applicazione, e di conseguenza si è adottato un approccio in cui si dava per assodato il caso migliore possibile di esecuzione.

5.2 Sviluppi futuri

Ci sono alcuni punti in cui l'applicazione può essere migliorata.

1. Aggiunta di chat di comunicazione real-time: usando un plugin come Dissonance e integrandolo con il servizio di rete di Mirror si potrebbe creare un canale di comunicazione tra i due operatori umani e permettere loro di coordinarsi meglio anche nel caso in cui si trovassero da remoto. Sfortunatamente, non è stato possibile aggiungere questa funzionalità a causa di tempi stringenti in quanto la risoluzione dei problemi dell'applicazione era più importante.
2. Aumento del numero dei partecipanti umani: Al momento la versione multiutente presenta interattività per solo due operatori non-NPC in contemporanea, ma con opportune modifiche sarebbe possibile estendere il numero da 2 a 4, permettendo così di simulare una squadra al completo. Inoltre sarebbe possibile permettere a gruppi di utenti di scegliere mansioni diverse per quanto riguarda le fasi 2 e 3 dell'applicazione.
3. Aggiungere nuove ambientazioni e casi d'uso: Sarebbe possibile replicare le attività considerate anche in scenari urbani, come sottopassaggi allagati, e personalizzare la simulazione per essi. Inoltre, sarebbe interessante aggiungere anche nuove attività, come l'utilizzo della gru per depositare le manichette appiattibili al terreno.
4. Miglior feedback: è possibile migliorare il feedback audio/visivo di alcuni componenti, come indicare la direzione della pompa durante il suo posizionamento, oppure migliorare le informazioni fornite durante il voiceover.

Bibliografia

- [1] «PITEM RISK – Progetto RISK - FOR | Regione Piemonte». <https://www.regione.piemonte.it/web/temi/fondi-progetti-europei/programmi-progetti-europei/cooperazione-territoriale-europea-piemonte/pitem-risk-progetto-risk-for> (consultato feb. 23, 2021).
- [2] «RISK - Resilienza, Informazione, Sensibilizzazione e Comunicazione ai Cittadini | Alcotra». <https://www.interreg-alcotra.eu/it/decouvrir-alcotra/les-projets-finances/risk-resilienza-informazione-sensibilizzazione-e> (consultato feb. 23, 2021).
- [3] «Progetto Risk-Act - La forza dell'azione», www.pitem-risk.eu. <https://www.pitem-risk.eu/progetti/risk-act> (consultato lug. 14, 2021).
- [4] «Servizio Nazionale», Dipartimento della Protezione Civile. <http://www.protezionecivile.gov.it/servizio-nazionale> (consultato feb. 23, 2021).
- [5] «Colonna Mobile Regionale», Regione Piemonte. <https://www.regione.piemonte.it/web/temi/protezione-civile-difesa-suolo-opere-pubbliche/protezione-civile/logistica/colonna-mobile-regionale> (consultato feb. 23, 2021).
- [6] «COORDINAMENTO OdV PROTEZIONE CIVILE VERCELLI: Contenuto / ARTICOLI / PROTEZIONE CIVILE EUROPEA - OPERATIVO IL MODULO HCP DELLA REGIONE PIEMONTE». http://protezionecivile-coordinamentovc.it/e107_plugins/content/content.php?content.28 (consultato feb. 23, 2021).
- [7] «Apell-Euromodex». <http://www.apell-euromodex.eu/> (consultato feb. 23, 2021).
- [8] «COORDINAMENTO OdV PROTEZIONE CIVILE VERCELLI: Contenuto / ARTICOLI / ESERCITAZIONE INTERNAZIONALE H.C.P. IN MONTENEGRO». http://protezionecivile-coordinamentovc.it/e107_plugins/content/content.php?content.96 (consultato feb. 23, 2021).
- [9] «Rischio meteo-idrogeologico ed idraulico», Dipartimento della Protezione Civile. <http://www.protezionecivile.gov.it/attivita-rischi/meteo-idro> (consultato feb. 23, 2021).

- [10] «Reality–virtuality continuum», Wikipedia. gen. 13, 2021. Consultato: feb. 23, 2021. [Online]. Disponibile su: https://en.wikipedia.org/w/index.php?title=Reality%E2%80%93virtuality_continuum&oldid=1000076065
- [11] «History of VR - Timeline of Events and Tech Development». <https://virtualspeech.com/blog/history-of-vr> (consultato feb. 23, 2021).
- [12] «Sensorama», Wikipedia. dic. 31, 2020. Consultato: feb. 23, 2021. [Online]. Disponibile su: <https://en.wikipedia.org/w/index.php?title=Sensorama&oldid=997501438>
- [13] «USC Hugh M. Hefner Moving Image Archive » MORTON HEILIG : INVENTOR VR». <https://www.uschefnerarchive.com/morton-heilig-inventor-vr/> (consultato feb. 23, 2021).
- [14] «The Sword of Damocles (virtual reality)», Wikipedia. gen. 19, 2021. Consultato: feb. 23, 2021. [Online]. Disponibile su: [https://en.wikipedia.org/w/index.php?title=The_Sword_of_Damocles_\(virtual_reality\)&oldid=1001415027](https://en.wikipedia.org/w/index.php?title=The_Sword_of_Damocles_(virtual_reality)&oldid=1001415027)
- [15] «History Of Virtual Reality», Virtual Reality Society. <https://www.vrs.org.uk/virtual-reality/history.html> (consultato lug. 03, 2021).
- [16] R. Purchase, «Happy Go Luckey: Meet the 20-year-old creator of Oculus Rift», Eurogamer, lug. 11, 2013. <https://www.eurogamer.net/articles/2013-07-11-happy-go-luckey-meet-the-20-year-old-creator-of-oculus-rift> (consultato lug. 04, 2021).
- [17] M. Rakver, «VR Display Comparison – LCD vs OLED, (Sub)Pixels, FOV & PPD», Smart Glasses Hub. <https://smartglasseshub.com/vr-headset-display-comparison/> (consultato lug. 09, 2021).
- [18] D. Robinson, B. Cassidy, G. Sim, e D. Gandy, «A Virtual Reality Platform for Analyzing Remote Archaeological Sites», *Interacting with Computers*, vol. 31, apr. 2019, doi: 10.1093/iwc/iwz011.
- [19] «VRChat», Wikipedia. feb. 22, 2021. Consultato: feb. 23, 2021. [Online]. Disponibile su: <https://en.wikipedia.org/w/index.php?title=VRChat&oldid=1008334199>
- [20] E. H. Tanaka et al., «Immersive virtual training for substation electricians», in *2017 IEEE Virtual Reality (VR)*, Los Angeles, CA, 2017, pagg. 451–452. doi: 10.1109/VR.2017.7892374.
- [21] 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», in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, Osaka, Japan, mar. 2019, pagg. 181–189. doi: 10.1109/VR.2019.8797889.

- [22] 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.
- [23] 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 Reality*, vol. 23, n. 3, pagg. 229–234, set. 2019, doi: 10.1007/s10055-018-0359-y.
- [24] J. Saunders, S. Davey, P. S. Bayerl, e P. Lohrmann, «Validating Virtual Reality as an Effective Training Medium in the Security Domain», in 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, mar. 2019, pagg. 1908–1911. doi: 10.1109/VR.2019.8798371.
- [25] B. Lang, «Boeing is Using Varjo Headsets to Train Astronauts for an ISS Docking Mission in 2021», *Road to VR*, giu. 11, 2020. <https://www.roadtovr.com/boeing-varjo-virtual-reality-astronaut-training-starliner/> (consultato giu. 18, 2020).
- [26] 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.
- [27] «Progetto - Prodiges». <http://www.pro-prodige.eu/index.php?id=1273> (consultato giu. 26, 2020).
- [28] «Living in a changing climate — European Environment Agency», Consultato: lug. 01, 2020. [Online]. Disponibile su: <https://www.eea.europa.eu/signals/signals-2015/articles/living-in-a-changing-climate>
- [29] «Obi - Unified Particle Physics for Unity 3D». <http://obi.virtualmethodstudio.com/> (consultato lug. 09, 2020).
- [30] «Virtual Method Studio». <http://www.virtualmethodstudio.com/products.html> (consultato lug. 11, 2021).
- [31] «Obi Physics for Unity - Big Picture». <http://obi.virtualmethodstudio.com/manual/4.X/bigpicture.html> (consultato lug. 07, 2021).
- [32] «Mirror Networking – Open Source Networking for Unity». <https://mirror-networking.com/> (consultato lug. 10, 2021).
- [33] «Showcase – Mirror Networking». <https://mirror-networking.com/showcase/> (consultato lug. 10, 2021).

- [34] «OpenVR (Steamworks Documentation)». <https://partner.steamgames.com/doc/features/steamvr/openvr> (consultato lug. 11, 2021).
- [35] «Final IK | Animation Tools | Unity Asset Store». <https://assetstore.unity.com/packages/tools/animation/final-ik-14290> (consultato lug. 11, 2021).
- [36] «www.makehumancommunity.org». <http://www.makehumancommunity.org/> (consultato lug. 14, 2021).
- [37] U. Technologies, «Unity - Manual: Creating and editing Terrains». <https://docs.unity3d.com/Manual/terrain-UsingTerrains.html> (consultato lug. 14, 2021).
- [38] «Cascade - River, Lake, Waterfall and more | Terrain | Unity Asset Store». <https://assetstore.unity.com/packages/tools/terrain/cascade-river-lake-waterfall-and-more-106072> (consultato lug. 11, 2021).

