



POLITECNICO DI TORINO
FACOLTÀ DI INGEGNERIA INFORMATICA

Corso di Laurea in Grafica e Multimedia

Tesi di Laurea Magistrale

Definizione e sviluppo di un manichino in ambiente VR per il design ergonomico in xG

Relatore

prof. Andrea Sanna

Candidato

Alessandro PASTORINO

Supervisore aziendale
Thales Alenia Space Italia
ing. Valter Basso

ANNO ACCADEMICO 2018/2019

Sommario

La realtà virtuale è divenuta oggi una tecnologia sempre più presente ed utilizzata anche in ambiti professionali; nello specifico contesto all'interno del quale si è svolto il progetto di tesi, la realtà virtuale viene impiegata per valutare l'ergonomia dei moduli in costruzione o in progettazione presso l'azienda Thales Alenia Space Italia. I test effettuati necessitano dei modelli tridimensionali che devono rappresentare il più fedelmente possibile le misure antropometriche dei veri astronauti e devono essere animati per ottenere movimenti che siano naturali e fluidi; per la modellazione di questi manichini si è fatto riferimento principalmente a due documenti: il documento NASA-STD-3000 [12] e il documento Human Integration Handbook [13]. Oltre a fare ciò, il team TRO-1 del Collaborative System Engineering di Thales Alenia Space Italia è al lavoro sul plugin VERITAS 4U, plugin per Unreal Engine, che si pone l'obiettivo, utilizzando un visore ottico HTC Vive, di permettere all'utente di impersonare un astronauta a bordo di un modulo spaziale e di simulare l'ambiente a microgravità, in modo tale da non solo effettuare verifiche progettuali o di ergonomia, ma anche di eseguire training per gli astronauti, che possono in questo modo prendere un primo contatto con l'ambiente con cui dovranno interagire e con le operazioni che dovranno effettuare. All'interno del lavoro di tesi assume molta importanza il concetto di ergonomia e come questo possa essere applicato ai vari contesti lavorativi; nello specifico ci si è concentrati sull'analisi dell'ergonomia in ambiente di microgravità in relazione al cambiamento che questo provoca al corpo e alle prestazioni dell'essere umano; rimanere in ambiente di microgravità, infatti, comporta numerosi disagi per l'astronauta quali: la ridistribuzione dei liquidi corporei verso la parte alta del corpo, la perdita di orientamento per il bilanciamento del corpo, incapacità, in ambienti ristretti, di mettere a fuoco gli oggetti lontani, riduzione della massa ossea e atrofia muscolare nonché una diminuzione del tempo di reazione. Successivamente i manichini sviluppati sono stati utilizzati all'interno di un nuovo ambiente virtuale, rappresentante un prototipo del modulo cis-lunare in progettazione presso Thales Alenia Space Italia, ed è stato sviluppato un sistema di controllo per riprodurre i movimenti che un astronauta deve compiere all'interno di un modulo utilizzando l'HTC Vive; grazie a questi test è possibile prendere visione del modulo prima di avviare la fase di costruzione e permettere ai futuri membri dell'equipaggio di bordo di provare determinate funzionalità per modificare e migliorare l'ergonomia del modulo.

Per lo sviluppo e la realizzazione dei manichini sono stati utilizzati principalmente 2 software: Make Human e Blender. Il primo software permette di creare modelli umanoidi attraverso la tecnica del morphing, controllata attraverso un'interfaccia che permette all'utente di modificare tutti i parametri antropometrici del modello che si vuole realizzare; il software è inoltre fornito di una libreria di modelli di vestiario e di acconciature piuttosto basilare, ma sufficiente per gli scopi della tesi. Infine, sempre attraverso una libreria dedicata e accessibile attraverso l'interfaccia è possibile stabilire la posa con cui esportare il manichino e impostare un'armatura per la successiva movimentazione dello stesso.

Blender è invece un software che permette di gestire ogni step della pipeline di creazione di grafica 3D, dalla modellazione al video editing, permettendo anche la creazione di script personali in Python. Il programma permette di avere diverse modalità di visualizzazione e di editing della

scena 3D; tra queste le più utilizzate sono state:

- 3D view, ovvero l'interfaccia principale utilizzata per la maggior parte delle operazioni di modellazione e animazione;
- Node editor, per la creazione e la modifica dei materiali;
- UV/image editor, per la creazione e la modifica delle texture;

I manichini sono stati sviluppati facendo riferimento al documento NASA [12], all'interno del quale la popolazione viene suddivisa in tre percentili per ogni sesso: 5°, 50° e 95° percentile, considerando per il sesso femminile la popolazione giapponese e per il sesso maschile quella americana; la scelta è legata al fatto che, mediamente, le donne giapponesi siano le più minute al mondo mentre i maschi americani siano i più robusti. I manichini creati rispecchiano il 50° percentile maschile e femminile, il 5° percentile femminile e il 95° percentile maschile.

I manichini così sviluppati sono stati poi importati all'interno di Unreal Engine, il motore grafico utilizzato per lo sviluppo del progetto. Il motore grafico mette a disposizione dell'utente due metodi di programmazione, il linguaggio C++ e un linguaggio di visual scripting, ovvero il blueprint; per lo sviluppo del progetto si è utilizzato quest'ultimo metodo.

Questo particolare linguaggio di visual scripting permette di scrivere, mediante la definizione di grafi composti da nodi rappresentanti le funzioni e archi che collegano i vari nodi, codice eseguibile, andando a considerare come flusso d'esecuzione il concetto di flusso del grafo stesso. Esistono tre tipi principali di blueprint: il Class blueprint, il Level blueprint e il Data only blueprint.

Il primo definisce lo script per uno specifico oggetto, viene inserito nel content browser e definisce una nuova classe che può essere poi utilizzata all'interno del progetto; prima di creare una nuova classe è possibile selezionare una Parent Class, in modo da ereditare diverse proprietà utilizzabili dalla nuova classe. Il Level blueprint definisce invece il flusso che gestisce e controlla uno specifico livello di gioco, mentre il Data Only blueprint è un blueprint che contiene solo codice al suo interno e che svolge specifiche funzioni; è un data Only blueprint, ad esempio, l'Animation graph, che ha il compito di gestire le animazioni legate ad un oggetto.

La prima parte del progetto di tesi ha lo scopo di rendere interattivo l'editor di Unreal Engine, dando all'utente la possibilità di selezionare il manichino interessato e di poter svolgere diverse operazioni quali:

- Movimentazione degli arti tramite cinematica inversa;
- La selezione di una posa per le mani;
- La selezione di un target e la successiva rotazione della testa nella direzione del target stesso;
- Il salvataggio di una serie di pose successivamente replicabili;
- L'aggiunta di una camera che mostri in basso a destra la visuale del personaggio e il cui FOV sia regolabile;
- La possibilità di attivare/disattivare il controllo del manichino attraverso la cinematica inversa

quali la movimentazione degli arti tramite cinematica inversa, la selezione di una posa per le mani, la selezione di un target e la successiva rotazione della testa nella direzione del target stesso, il salvataggio di una serie di pose successivamente replicabili e l'aggiunta di una camera che mostri in basso a destra la visuale del personaggio e il cui FOV sia regolabile.

Per la risoluzione del problema della cinematica inversa esistono numerosissimi algoritmi, molti dei quali, tuttavia, hanno costi computazionali troppo elevati o falliscono nel riprodurre una posa

complessa in modo naturale; in Unreal Engine è presente, proprio per risolvere questi problemi, il nodo FABRIK, che esegue un algoritmo di risoluzione della cinematica inversa molto semplice ed economico dal punto di vista del costo computazionale, che permette di risolvere la cinematica inversa in real-time con movimenti fluidi e naturali. Il movimento effettivo del manichino viene calcolato all'interno del file di animazione, denominato Animation blueprint, all'interno del quale vengono impostati e utilizzati i noti FABRIK; per utilizzare correttamente un nodo FABRIK bisogna, dopo aver impostato la precisione con cui si vuole utilizzare il nodo- precisione che determinerà il numero di iterazioni compiute dall'algoritmo e il costo che avrà il calcolo della cinematica inversa- impostare le catene cinematiche che bisogna movimentare; tale operazione consiste nell'identificare root, end effector e joints della catena interessata. Nel caso del progetto in esame sono state definite in totale otto catene, due per ogni arto; prendendo in considerazione una delle braccia abbiamo una catena che ha come end effector la mano e come root la spalla ed è quindi composta da due ossa; un'altra che è invece composta da un solo osso e ha come end effector il gomito e come root la spalla. La presenza della doppia catena permette di ottenere pose molto più naturali rispetto alla catena singola, aggiungendo però la necessità di risolvere due volte il problema della cinematica inversa; in questo modo, utilizzando la prima catena è possibile spostare e ruotare la mano nella posizione desiderata per poi utilizzare la seconda catena per regolare precisamente l'angolo formato dal gomito. In modo del tutto analogo sono stati aggiunti i quattro punti di controllo per gli arti inferiori.

Successivamente i manichini sono stati utilizzati per proseguire lo sviluppo del plugin VERITAS 4U, un plugin per Unreal Engine progettato e sviluppato da Thales Alenia Space Italia che permette, attraverso l'utilizzo di un visore HTC Vive, di visualizzare il modulo spaziale che si sta progettando per eseguire test, verifiche o per eseguire il training dei futuri operatori.

L'obiettivo di questa parte del lavoro di tesi è stato quello di migliorare il sistema di movimento e di rotazione e di sviluppare un sistema di gestione delle collisioni che risultasse computazionalmente leggero e di facile utilizzo per l'utente. Il sistema di movimento e di rotazione che sono stati sviluppati si basano sull'idea di voler simulare ciò che un operatore deve fare per muoversi da un punto all'altro in un ambiente a microgravità: l'utente deve quindi trovarsi sempre vicino ad una handrail da poter afferrare; una volta afferrata tramite la pressione del grip button del controller, il manichino è vincolato alla suddetta handrail fino a quando non viene rilasciato il grip button e si muove seguendo i movimenti della mano dell'utente, simulando ciò che accade ad un astronauta. L'utente può inoltre spostarsi da un punto all'altro simulando di darsi una spinta nella direzione desiderata; spostando il controller da un punto A ad un punto B, infatti, vengono calcolate la direzione e velocità con cui viene eseguito tale spostamento e, rilasciando il grip button, il manichino si stacca dalla maniglia e "fluttua" lungo la direzione calcolata. Il sistema di rotazione si basa su un meccanismo analogo: dopo aver afferrato l'handrail, l'utente può attivare la rotazione mantenendo premuto il grilletto del controller; a questo punto l'utente, ruotando il polso, fa ruotare il manichino in accordo al vincolo cui è sottoposto.

La tesi è strutturata in otto capitoli; nel primo si trova un'introduzione alla realtà virtuale, all'ergonomia e dei motivi che hanno dato origine a tale lavoro, mettendo in luce obiettivi e requisiti del progetto.

Nel secondo capitolo si trova una breve descrizione dell'azienda presso cui è stato svolto il progetto e di come sia organizzato il laboratorio di realtà virtuale; il terzo capitolo contiene una più approfondita descrizione della realtà virtuale, della sua storia e dei campi in cui è utilizzata oggi, terminando con una panoramica sullo stato dell'arte in TASI, analizzando i progetti ancora in fase di sviluppo e quelli già completati.

Il quarto e il quinto capitolo sono dedicati alla descrizione della prima parte del lavoro di tesi, dalla modellazione del manichino alla sua importazione all'interno del motore grafico utilizzato, fino alla realizzazione di tutte le caratteristiche richieste per questa prima fase, concentrandosi

maggiormente sull'algoritmo utilizzato per la risoluzione della cinematica inversa; nel sesto capitolo vengono presentati i risultati ottenuti in questa prima fase; i test in questo caso sono di natura qualitativa e sono stati svolti replicando posture e attività svolte dagli astronauti attraverso l'utilizzo dei manichini sviluppati.

All'interno del settimo capitolo viene invece presentata la seconda parte del progetto, incentrata sul miglioramento del plugin VERITAS 4U e sullo sviluppo del nuovo sistema di movimento e di rotazione e di gestione delle collisioni. Infine, nell'ottavo capitolo vengono presentati i risultati di quest'ultima parte di progetto, provata anche da due astronauti, Mathias Maurer e Samantha Cristoforetti e i possibili sviluppi futuri richiesti da Thales Alenia Space Italia.

Ringraziamenti

Prima di lasciare spazio alla trattazione, vorrei ringraziare tutti coloro che mi hanno sostenuto e mi hanno aiutato durante il mio percorso universitario.

Un primo e sentito ringraziamento va al relatore di questa tesi, il professor Andrea Sanna, per gli aiuti, i consigli, la disponibilità e la precisione dimostratemi durante tutto il periodo di stesura; non posso poi non ringraziare le meravigliose persone conosciute in Thales Alenia durante il mio periodo all'interno dell'azienda; in particolare ringrazio Lorenzo Rocci, Eugenio Sorbellini, Giuseppe Buffa e Valter Basso per avermi sempre gentilmente assistito durante tutto lo sviluppo del progetto; includo nei ringraziamenti di chi ha reso leggero e piacevole il soggiorno presso Thales anche Manuela, Mikael, Mauro e Beatrix (controllare!!!).

Ringrazio di cuore i compagni che ho conosciuto durante questo percorso e con i quali ho condiviso questa esperienza; un grazie quindi a Lugi, Martino, Dave, Manna, Enrico Isabeau, Jeanpierre e a tutti coloro che mi hanno regalato ben più di una risata. Un ringraziamento particolare va a Giovanni Patruno, con il quale ho portato a termine numerosi progetti e che ha sopportato le mie ansie durante tutto il periodo della magistrale. Non posso dimenticare Jacopo, Alessia, Cecilia e Alessandro, inseparabili amici tra i banchi di scuola e nella vita; ringrazio poi tutti i membri del gruppo storico "La Medioevale" (siete davvero troppi per citarvi tutti), la mia seconda famiglia con cui da nove anni condivido avventure, risate e "sassellate" e alla quale non posso che dedicare un solo grido: ORDO!

Infine un grazie ad Alex, Nadia, Marco, ai miei genitori Andrea e Francesca. alla mia preziosa sorella Margherita e a tutta la mia famiglia per avermi sempre sostenuto, incitato e incoraggiato nei momenti più difficili e per aver sorriso con me in quelli felici e senza i quali non sarei qui oggi. Non so se potrò mai sdebitarmi per tutto quello che fate e avete fatto per me, ma so che potrò sempre contare sul vostro incondizionato amore come voi potrete contare sempre sul mio. Grazie dal più profondo del cuore.

Indice

Elenco delle tabelle	IX
Elenco delle figure	X
I Prima Parte	1
1 Introduzione	3
1.1 Realtà virtuale e ergonomia	3
1.2 Ergonomia	4
1.3 Applicazione del concetto di ergonomia	4
1.4 Requisiti, obiettivi e strumenti utilizzati	5
2 Thales Alenia Space Italia	7
2.1 Introduzione	7
2.2 COSE Centre	8
2.2.1 VR Lab	9
3 Stato dell'arte	11
3.1 Introduzione alla realtà virtuale	11
3.2 Storia della realtà virtuale	12
3.2.1 Stereoscopia	12
3.2.2 Gli albori della VR: il primo simulatore di volo e il Sensorama	12
3.2.3 I primi visori e i primi dispositivi aptici	13
3.2.4 La Realtà Virtuale oggi	13
3.3 Architettura di un sistema di realtà virtuale	14
3.3.1 Set di elaboratori	14
3.3.2 Set di dispositivi I/O	15
3.4 Applicazioni	18
3.5 VR per l'ergonomia nel settore aerospaziale	19
3.5.1 Stato dell'arte in TASI	19
II Seconda Parte	23
4 Sviluppo dei manichini	25
4.1 Make Human	25
4.2 Blender	25
4.3 Creazione del personaggio	26

4.3.1	Creazione della mesh	26
4.3.2	Pulizia della topologia, creazione dello scheletro e associazione con la mesh	26
4.3.3	Applicazione dei modificatori	27
4.4	Definizione dei percentili NASA	27
4.4.1	Calcolo dei percentili	28
4.4.2	Percentili femminili	28
4.4.3	Percentili maschili	32
4.5	Definizione dei manichini	36
4.6	Confronto misure NASA con percentili definiti	39
4.7	Postura in microgravità	39
5	Sviluppo della cinematica inversa	41
5.1	Unreal Engine	41
5.1.1	Unreal editor	41
5.2	Blueprint e C++	43
5.2.1	Blueprint	43
5.2.2	C++ e plugin	45
5.3	Struttura del progetto	45
5.4	Analisi del menù creato	46
5.5	Definizione della cinematica inversa	46
5.5.1	Basi matematiche della cinematica inversa	47
5.5.2	Algoritmo di cinematica inversa: FABRIK	48
5.5.3	Impostazione nodi FABRIK e animazione del personaggio	49
6	Test d'uso sui manichini	53
6.1	Manichini in TASI	53
6.1.1	Replica delle posture realistiche degli astronauti con i manichini definiti nella tesi	54
6.1.2	Verifica dell'ergonomia del hardware a bordo di Columbus	55
6.1.3	Simulazione del lavoro di team	58
III	Terza Parte	59
7	Sviluppo del sistema di movimento e rotazione su HTC Vive	61
7.1	LOP-G e VERITAS 4U	61
7.2	Sostituzione del manichino Unreal Engine	62
7.3	Sviluppo del sistema di movimento e di rotazione	63
7.3.1	Sistema di movimento	63
7.3.2	Sistema di rotazione	65
7.4	Gestione delle collisioni	65
8	Test d'uso sul plugin VERITAS 4U	69
8.1	Test e collaboratori esterni	69
8.2	Futuri sviluppi	83
	Bibliografia	85

Elenco delle tabelle

4.1	Percentili femminili standard NASA relativi alla figura 4.1	29
4.2	Percentili femminili standard NASA relativi alla figura 4.2	30
4.3	Percentili femminili standard NASA relativi alla figura 4.3	30
4.4	Percentili femminili standard NASA relativi alla figura 4.4	31
4.5	Percentili femminili standard NASA relativi alla figura 4.5	32
4.6	Percentili femminili standard NASA relativi alla figura 4.6	32
4.7	Percentili maschili standard NASA relativi alla figura 4.7	33
4.8	Percentili maschili standard NASA relativi alla figura 4.8	34
4.9	Percentili maschili standard NASA relativi alla figura 4.9	34
4.10	Percentili maschili standard NASA relativi alla figura 4.10	35
4.11	Percentili maschili standard NASA relativi alla figura 4.11	36
4.12	Percentili maschili standard NASA relativi alla figura 4.12	36
4.13	Percentili maschili ottenuti	37
4.14	Percentili femminili ottenuti	38

Elenco delle figure

2.1	"Logo Thales Alenia Space"	7
2.2	"Logo COSE Centre"	9
3.1	Edward Link e la sua invenzione	12
3.3	"Ricostruzione 3D del modulo LOP-G"	20
3.4	HTC Vive PRO	20
4.1	Figura esplicativa dei percentili femminili riguardanti altezze e ampiezze	29
4.2	Figura esplicativa dei percentili femminili riguardanti mani e seduta	29
4.3	Figura esplicativa dei percentili femminili riguardanti ulteriori altezze e ampiezze	30
4.4	Figura esplicativa dei percentili femminili riguardanti le dimensioni del cranio	31
4.5	Figura esplicativa dei percentili femminili riguardanti varie circonferenze	31
4.6	Figura esplicativa dei percentili femminili riguardanti ulteriori dimensioni	32
4.7	Figura esplicativa dei percentili maschili riguardanti altezze e ampiezze	33
4.8	Figura esplicativa dei percentili maschili riguardanti mani e seduta	33
4.9	Figura esplicativa dei percentili maschili riguardanti ulteriori altezze e ampiezze	34
4.10	Figura esplicativa dei percentili maschili riguardanti le dimensioni del cranio	35
4.11	Figura esplicativa dei percentili femminili riguardanti varie circonferenze	35
4.12	Figura esplicativa dei percentili femminili riguardanti ulteriori dimensioni	36
4.13	95° maschile in misurazione e in ambiente Unreal	38
4.14	Posizione di microgravità realizzata	40
5.1	Interfaccia utente	42
5.2	Interfaccia ottenuta per la prima parte del progetto	45
5.3	Esempio di cinematica diretta e cinematica inversa	47
5.4	Schema di funzionamento dell'algoritmo FABRIK	50
5.5	Esempio di nodi utilizzati	52
6.1	Replica della foto rappresentante Chris Cassid e Luca Parmitano	54
6.2	Replica della foto rappresentante Dan Burbank	55
6.3	Replica della foto rappresentante Peggy Whitson	55
6.4	Test di prensilità e accessibilità e usabilità computer di bordo	56
6.5	Manichino all'interno del work envelope definito per il computer di bordo	57
6.6	Test lavoro di gruppo	58
7.1	50° maschile confrontato con il manichino Unreal Engine	62
7.3	Modello fisico presente per il manichino Unreal Engine	66
7.4	Immagine raffigurante un manichino il cui avambraccio è in collisione con la parte del modulo	67
8.1	Risultati test rilasciati da Andrea Casini(1)	71
8.2	Risultati test rilasciati da Andrea Casini(2)	72
8.3	Risultati test rilasciati da Andrea Casini(3)	73
8.4	Risultati test rilasciati da Lionel Ferra(1)	74
8.5	Risultati test rilasciati da Lionel Ferra(2)	75

8.6	Risultati test rilasciati da Mathias Maurer(1)	76
8.7	Risultati test rilasciati da Mathias Maurer(2)	77
8.8	Risultati test rilasciati da Samantha Cristoforetti(1)	79
8.9	Risultati test rilasciati da Samantha Cristoforetti(2)	80
8.10	Risultati test rilasciati da Samantha Cristoforetti(3)	81
8.11	Risultati test rilasciati da Samantha Cristoforetti(4)	82

Acronimi

ASI Agenzia Spaziale Internazionale.

COSE Collaborative System Engineering.

CSS Chinese Space Station.

ESA European Space Agency.

FABRIK Forward And Backward Inverse Kinematics.

HUD Heads-Up Display.

IEA International Ergonomics Association.

ISS International Space Station.

LOP-G Lunar Orbital Platform-Gateway.

NASA National Aeronautics and Space Administration.

TASI Thales Alenia Space Italia.

TRO Technology Research Office.

UE Unreal Engine.

VERITAS 4U Virtual Environment Reseach In Thales Alenia Space 4U.

VR Virtual Reality.

Parte I

Prima Parte

Capitolo 1

Introduzione

1.1 Realtà virtuale e ergonomia

Il progetto di tesi è stato sviluppato all' interno di Thales Alenia Space, azienda molto attiva nel settore spaziale; nello specifico, il contesto all'interno del quale è stato svolto il progetto riguarda lo studio dell'ergonomia di un modulo spaziale, attraverso lo sviluppo di manichini tridimensionali virtuali e sfruttando la tecnologia della realtà virtuale.

In precedenza, la fase di progettazione di un modulo spaziale prevedeva lo studio attento di dati in formato cartaceo, che riportavano misure e volumi occupati di ogni componente del modulo e di ogni oggetto in esso presente, in modo che si potesse valutare l'ergonomia e l'usabilità dell'oggetto specifico; in questo modo, ogni componente del progetto deve essere studiato attentamente, analizzando non solo il suo funzionamento, ma considerando anche in quali condizioni dovrà essere attivato e quali equipaggiamenti l'operatore dovrà utilizzare per operare sullo specifico oggetto, tenendo anche in considerazione che quest'ultimo opererà in un ambiente a microgravità. Per via di tutte queste complicazioni, questo metodo di lavoro risulta spesso poco efficace, dal momento che è estremamente complicato sviluppare l'intero progetto basandosi solo sui disegni riportanti le misure e i volumi occupati, con conseguenti errori nel progetto finale, che risultano spesso di difficile risoluzione nel momento in cui questo viene scoperto una volta lanciato il modulo. Per evitare ciò è pratica comune costruire un modulo che viene verificato preventivamente da un operatore reale, in modo da identificare possibili criticità all'interno del modulo stesso, che possono quindi essere corrette all'interno del progetto; tutto questo processo risulta dispendioso sia in termini di tempo che di denaro a causa della costruzione del rig fisico da verificare; l'operatore, inoltre, si trova ad operare in un ambiente totalmente differente da quello spaziale, data la difficoltà di replicare le condizioni di microgravità sulla terra; proprio per porre rimedio a queste criticità, già da tempo, l'attenzione di molte aziende si è posata sulla realtà virtuale e su come questa nuova tecnologia possa essere applicata allo sviluppo e alla progettazione di moduli spaziali.

Con questo presupposto, Thales Alenia Space Italia, impiega attivamente da anni l'utilizzo della realtà virtuale per studiare l'ergonomia dei suoi nuovi progetti, cercando di migliorare sempre più il realismo e l'efficacia della simulazione. Per effettuare tutti i test necessari all'interno dell'ambiente virtuale sono ovviamente necessari, oltre al modello dell'ambiente studiato, i modelli 3D rappresentanti gli astronauti, che devono essere modellati e animati in modo da rappresentare il più fedelmente possibile le misure antropometriche e i movimenti reali di un astronauta. Uno dei vantaggi dati dalla realtà virtuale consiste nella facilità con cui è possibile simulare un ambiente di microgravità e dalla possibilità di effettuare test e replicare le procedure che saranno eseguite in orbita, permettendo di simulare quindi anche questo aspetto, rendendo di fatto i test estremamente completi e affidabili; in aggiunta a ciò, il team TRO-1 del Collaborative System Engineering di

Thales Alenia Space Italia è al lavoro sul plugin VERITAS 4U, plugin per Unreal Engine, che si pone l'obiettivo, utilizzando un visore ottico HTC Vive, di permettere all'utente di impersonare un astronauta a bordo di un modulo spaziale e di simulare l'ambiente a microgravità, in modo tale da non solo effettuare verifiche progettuali o di ergonomia, ma anche di eseguire training per gli astronauti, che possono in questo modo prendere un primo contatto con l'ambiente con cui dovranno interagire e con le operazioni che dovranno effettuare; lo sviluppo del plugin sopracitato sarà oggetto della seconda parte del lavoro di tesi.

1.2 Ergonomia

L'International Ergonomics Association (IEA) definisce l'ergonomia come: *L'ergonomia 'è quella disciplina scientifica che si occupa di capire le interazioni tra gli umani e gli altri elementi del sistema. E' una professione che applica le teorie, i principi, i dati e i metodi scientifici in modo da ottimizzare il benessere dell'uomo e le performance del sistema*[10] L'ergonomia, dal greco ergon (lavoro) e nomos (regola), è quindi la disciplina che si pone l'obiettivo di modificare e adattare le condizioni e l'ambiente di lavoro alle capacità e alle caratteristiche del lavoratore.

In particolare si concentra sullo studio dell'antropometria, per definire le misurazioni dell'essere umano e il movimento di quest'ultimo, della biomeccanica, per analizzare le conseguenze del lavoro sull'apparato muscolo-scheletrico, del sistema nervoso e dell'ambiente, per valutarne tutti i fattori che influenzano l'attività lavorativa, quali il microclima, l'illuminazione il rumore e le vibrazioni.

L'ergonomia può essere suddivisa in tre campi principali :

- Ergonomia fisica: si occupa di analizzare i fattori anatomici, fisiologici e biomeccanici dell'interazione tra uomo e ambiente di lavoro; particolare attenzione viene posta sulla figura del lavoratore e sugli strumenti da lui utilizzati, ponendosi l'obiettivo di ottimizzare questi ultimi;
- Ergonomia cognitiva: si occupa di analizzare le risposte motorie dovute alla percezione e alla interpretazioni di segnali e stimoli cui è soggetto il lavoratore; l'attenzione si sposta in questo caso sull'ambiente di lavoro, in modo da garantire libertà di movimento e sicurezza dell'operatore, nonché chiarezza e immediatezza di segnali e avvertenze per il suddetto operatore.
- Ergonomia organizzativa: si occupa di analizzare e migliorare le politiche e le strategie di organizzazione dell'attività lavorativa; particolare interesse assumono quindi i tempi, i metodi e i ritmi delle attività lavorative.

1.3 Applicazione del concetto di ergonomia

Durante lo sviluppo del progetto di tesi è stata analizzata e studiata l'ergonomia definita dalle posture dei manichini, includendo lo studio dell'antropometria per la creazione dei manichini stessi. Nello specifico, il contesto in cui si è svolta la prima parte del lavoro di tesi, riguarda l'analisi dell'ergonomia in ambiente di microgravità in relazione al cambiamento che questo provoca al corpo e alle prestazioni dell'essere umano; rimanere in ambiente di microgravità, infatti, comporta numerosi disagi per l'astronauta quali: la ridistribuzione dei liquidi corporei verso la parte alta del corpo, la perdita di orientamento per il bilanciamento del corpo, incapacità, in ambienti ristretti, di mettere a fuoco gli oggetti lontani, riduzione della massa ossea e atrofia muscolare nonché una diminuzione del tempo di reazione.

Per effettuare queste analisi è stata richiesta la modellazione di manichini virtuali, modellati con precisione facendo riferimento principalmente a due documenti: il documento NASA-STD-3000 [12] e il documento Human Integration Handbook [13].

Tali documenti stabiliscono l'antropometria, la biomeccanica e la massa muscolare dell'astronauta in ambiente extraterrestre; definiscono inoltre l'architettura e la struttura delle workstation e dell'hardware di bordo, nonché le condizioni ambientali ottimali delle cabine pressurizzate, analizzando umidità, temperatura, ricircolo dell'aria e limiti di contaminazione delle tossine presenti; i dati presenti in questi documenti sono relativi non solo all'ambiente definito in microgravità, ma anche a quelli definiti in fase di lancio, orbita e spazio extraterrestre e sono pertanto utilizzati da ingegneri progettisti, ingegneri dei sistemi, ingegneri della manutenzione e da tutte quelle figure professionali impegnate nello sviluppo di programmi spaziali.

I manichini tridimensionali sviluppati nella prima fase di questa tesi sono stati quindi realizzati in modo da sottostare a determinati requisiti ergonomici e antropometrici perché fossero in grado di replicare correttamente le pose degli astronauti durante le manovre in assenza di gravità. I requisiti ergonomici cui si è fatto riferimento nello specifico sono flessibilità, riconfigurabilità, usabilità e accuratezza; tutte queste caratteristiche possono essere definite qualitativamente e rappresentano la flessibilità nel definire una posa, la possibilità di riconfigurare tale posa, l'usabilità della stessa in una situazione reale nonché la valutazione della sua accuratezza.

Durante la seconda parte del progetto, invece, sono stati utilizzati i manichini sviluppati all'interno di un nuovo ambiente virtuale, rappresentante un prototipo del modulo cis-lunare in progettazione presso Thales Alenia Space Italia, ed è stato sviluppato un sistema di controllo per riprodurre i movimenti che un astronauta deve compiere all'interno di un modulo utilizzando l'HTC Vive; grazie a questi test è possibile prendere visione del modulo prima di avviare la fase di costruzione e permettere ai futuri membri dell'equipaggio di bordo di verificare determinate funzionalità per modificare e migliorare l'ergonomia del modulo.

Per questa fase di progetto, a differenza di quanto avvenuto per la prima, Thales Alenia Space Italia ha effettuato alcuni test con altri enti spaziali e aziende per verificare la validità del prodotto, sfruttando anche la partecipazione di due astronauti, ovvero Samantha Cristoforetti e Matthias Maures. I test si sono basati su una serie di procedure che sono state eseguite dai soggetti, le cui impressioni sono poi state raccolte; i requisiti cui si è fatto riferimento in questi test sono stati accuratezza e precisione nella riproduzione dei movimenti, fedeltà nella simulazione del movimento in microgravità e nella sensazione provata dall'utente e, infine, intuitività del sistema di movimento e dei comandi sviluppati.

1.4 Requisiti, obiettivi e strumenti utilizzati

I requisiti richiesti da Thales Alenia Space Italia sono i seguenti:

- Creazione dei manichini rappresentanti i seguenti percentili:
 - 95° maschile;
 - 50° maschile;
 - 50° femminile;
 - 5° femminile;
- Creazione di un'applicazione tramite Unreal Engine che permetta:
 - posizionamento e selezione del percentile desiderato;
 - visualizzazione del punto di vista del percentile selezionato;

- movimentazione del percentile e salvataggio della posa creata;
- Miglioramento del plugin VERITAS 4U, in particolare per quanto riguarda:
 - sistema di movimento;
 - sistema di rotazione;

Gli obiettivi che si vogliono raggiungere attraverso lo sviluppo di questa tesi sono quindi i seguenti:

- Realizzazione dei modelli 3D rappresentati i manichini, seguendo le misure antropometriche presenti nei documenti NASA;
- Posizionamento dei manichini nella posa neutra in microgravità;
- Realizzazione di una posa attraverso la cinematica inversa mediante i widget di UE;
- Salvataggio e caricamento delle pose;
- Selezione del target da osservare e movimento verso lo stesso;
- Importazione del manichino nell'ambiente del modulo cislunare LOP-G;
- Sviluppo del sistema di movimento in microgravità
- Sviluppo del sistema di rotazione in microgravità

Per la realizzazione dei passi descritti al punto precedente sono stati utilizzati i seguenti programmi:

- Make Human: programma open source per la realizzazione di prototipi umanoidi in computer grafica 3D. Nel contesto della tesi è stato utilizzato per creare la base dei manichini, del loro vestiario ed altri elementi come capelli e sopracciglia e successivamente esportati e completati in Blender;
- Blender: programma open source che supporta tutta la pipeline di lavoro per la grafica 3D, dalla modellazione al video editing. Nel contesto della tesi, è stato principalmente utilizzato per completare i manichini forniti da Make Human nonché la loro animazione e messa in posa;
- Krita: programma open source per il disegno e la colorazione digitale. Nel contesto della tesi è stato utilizzato per ritoccare e modificare dove necessario le texture provenienti da MakeHuman;
- Unreal Engine: motore grafico che permette la creazione di contenuti digitali complessi pur mantenendo una buona efficienza nell'esecuzione della scena. In Unreal Engine si è sviluppata tutta la parte di programmazione;

Capitolo 2

Thales Alenia Space Italia

2.1 Introduzione



Figura 2.1: "Logo Thales Alenia Space"

Nata come joint venture tra Thales (67%) e Leonardo (33%), Thales Alenia Space Italia (TASI) è una tra le maggiori industrie spaziali in Italia e in Europa; in Italia ha sedi a Roma, Torino, Milano e L'Aquila per un totale di 2800 dipendenti. Oltre all'Italia è presente in altri territori come Francia, Germania, Stati Uniti, Spagna, Inghilterra e Belgio.

I settori in cui opera sono:

- Aerospazio;
- Trasporto terrestre;
- Spazio: l'azienda partecipa ai programmi con l'Agenzia Spaziale Internazionale (ASI), con la European Space Agency (ESA) e collabora con la National Aeronautics and Space Administration (NASA). Sviluppa moduli per la International Space Station (ISS), satelliti scientifici e si occupa anche dell'osservazione terrestre, ricerca, esplorazione e navigazione dello spazio;
- Difesa;
- Sicurezza dello cyberspazio: aiuta le città a diventare più sicure e più smart, fornisce ottimi livelli di sicurezza nei mercati finanziari, protezione dei dati delle missioni critiche e integrità delle tecnologie che abbiamo nel quotidiano;
- Altri mercati come le telecomunicazioni, industria e simulazioni;

All'interno di questi settori Thales Alenia Space svolge svariate attività, tra le quali:

- Telecomunicazioni;
- Sistemi di navigazione;
- Monitoraggio ambientale e climatico;
- Scienza ed esplorazione;
- Infrastrutture orbitali e trasporti spaziali;
- Sistemi di elettronica avanzata;

Le peculiarità dei prodotti Thales Alenia Space sono:

- Alta tecnologia;
- Grandi investimenti e alta frammentazione organizzativa;
- Bassissimo numero di modelli prodotti;
- Parti perlopiù progettate e costruite per la prima volta;
- Progetti dalla durata complessiva fino a dieci anni;
- Multidisciplinarietà delle tematiche affrontate;
- Vita operativa garantita da pochi a dieci anni in ambiente ostile;
- Difficoltà ad effettuare la manutenzione in fase operativa;

Per quanto riguarda il settore spaziale, i requisiti dei prodotti Thales Alenia Space devono essere:

- Sicurezza: bisogna evitare di mettere in pericolo la vita dell'equipaggio;
- Difettosità zero: massima riduzione possibile del difetto di produzione e dei guasti del prodotto
- Abitabilità: i moduli abitativi devono essere sottoposti ad una serie di test il cui scopo è quello di verificare che i moduli siano adatti ad ospitare esseri umani;
- Utilizzabilità del prodotto: il prodotto deve essere utilizzabile nell'ambiente per il quale è stato progettato.

[15]

2.2 COSE Centre

Il Collaborative System Engineering (COSE), attivo da Giugno 2003, è un laboratorio di ricerca per la realtà virtuale composto da diversi ambienti:

- Due stanze collaborative: contengono le infrastrutture per videoconferenze, meeting e la condivisione di informazioni;
- Technology Research Office TRO: diviso in 2 settori, TRO-1, che ospita il team per lo sviluppo delle applicazioni di realtà virtuale e TRO-2, che ospita il team di sviluppo delle applicazioni ingegneristiche;
- VR-Lab: il laboratorio che ospita tutti gli strumenti e i device necessari a sviluppare applicazioni in realtà virtuale;



Figura 2.2: "Logo COSE Centre"

Il team TRO-1 dal 2003 ha portato a termine venti progetti, collaborando con importanti enti quali ASE, ESA e NASA e ha sviluppato quattro applicazioni in Veritas4U, una piattaforma software per supportare diverse applicazioni, soprattutto in realtà virtuale. Per il futuro sono sette i progetti su cui lavorerà il team del COSE Centre, tra i quali si annoverano Exomars, in collaborazione con ESA e che partirà nel 2020, Space Rider, sempre in collaborazione con ESA e Space Home, progetto volto all'integrazione della realtà virtuale per il training degli studenti.

Inoltre, il COSE Centre viene utilizzato per molte altre attività, tra le quali:

- Simulazioni di missioni con il rover: simulatore di guida su superficie extraterrestre, con simulazione del movimento e ambientale (illuminazione, vento, polvere);
- Revisione del design: misurazioni e test di raggiungibilità sul prodotto;
- Simulazioni AIT: simulazioni di interazione dell'utente con il prodotto per mezzo di un device aptico e il tracking dell'utente stesso;
- Simulazioni di missioni con il lander;

[4]

2.2.1 VR Lab

Oggi, in TASI e in particolare nel VR Lab del COSE Centre, per simulare la realtà virtuale, vengono principalmente utilizzati gli occhiali HTC Vive, un dispositivo di realtà virtuale progettato da Valve in collaborazione con HTC, disponibile sul mercato a partire dal 2016 [17]; in precedenza, invece, veniva utilizzato il sistema CAVE, un sistema hardware e software integrato che permette all'utente di vivere un'esperienza di realtà virtuale immersiva. [5].

Il dispositivo CAVE utilizza un ambiente definito da 3 schermi retro-proiettati che utilizzano la tecnologia della stereoscopia passiva: le immagini vengono proiettate con un leggero sfasamento, in modo che, utilizzando occhialini polarizzati, il nostro occhio possa ricostruire l'ambiente 3D. Per tracciare i movimenti dell'utente vengono utilizzate delle telecamere in infrarosso, mentre per l'interazione si possono utilizzare diversi dispositivi, come un braccio robotico, guanti aptici o una tuta per il tracciamento esatto della posizione.

Il dispositivo HTC Vive, invece, utilizza un caschetto su cui sono montate due lenti, una per occhio, per visualizzare la scena tridimensionale, un giroscopio, un accelerometro e dei sensori laser per tracciare con precisione la posizione e la rotazione del caschetto nell'ambiente virtuale. Questo dispositivo adotta una nuova tecnologia chiamata room scale, che permette di trasformare l'ambiente che circonda l'utente in uno spazio virtuale in cui può muoversi; per fare ciò vengono posizionate ai due angoli opposti della stanza due telecamere, che definiscono la posizione dell'utente facendo il tracking del caschetto. Per l'interazione vengono utilizzati due controller, anch'essi tracciati dalle telecamere, che permettono di simulare il movimento e il comportamento delle mani dell'utente; l'unione di ciò alla tecnologia room scale ha permesso di rendere maggiormente interattivo l'ambiente virtuale e donando a tutta l'esperienza un grado di immersività nettamente maggiore.

Attualmente il dispositivo in dotazione al VR Lab è l'HTCV Vive Pro, che permette una risoluzione maggiore e una miglior gestione del suono.

Capitolo 3

Stato dell'arte

3.1 Introduzione alla realtà virtuale

Con il termine realtà virtuale si definisce la realizzazione di una simulazione computerizzata che vuole portare l'utente a credere di trovarsi e muoversi in un ambiente reale. Per fare ciò, la simulazione dovrebbe stimolare la totalità dei sensi dell'utente e simulare tutti gli stimoli che vengono percepiti nel mondo reale; tuttavia, oggi, non disponiamo della tecnologia necessaria per simulare e stimolare tutti i sensi, ragion per cui ci si concentra solo su tre di essi: vista, udito e tatto, quest'ultimo realizzato, solitamente in maniera approssimativa, attraverso l'utilizzo di controller supplementari.

Quando si sviluppa un software per la realtà virtuale sono tre i principali fattori da tenere in considerazione:

- Immersione: rappresenta il grado di coinvolgimento dell'utente; strettamente legata alla qualità della simulazione e dalla tecnologia utilizzata, può essere di tre tipi:
 1. Totalmente immersivo: per ottenere una simulazione più realistica, l'utente viene completamente isolato dal mondo esterno; l'esempio più esplicativo di questo tipo di tecnologia è quello associato all'utilizzo di un Head Mounted Display, come nel caso dell'HTC Vive. Questo tipo di tecnologia, tuttavia, è la più propensa a causare cybersickness, che comporta per l'utente un senso di nausea, problemi visivi e di orientamento.
 2. Semi immersivo: in questo caso l'utente non ha bisogno di indossare dispositivi particolari ed è ancora conscio del mondo che lo circonda. Solitamente si utilizzano tre schermi concavi e la riproduzione di ciò che si vuole simulare. Se prendiamo ad esempio un simulatore di volo semi immersivo, esso sarà composto da tre schermi e dalla riproduzione della cabina di pilotaggio con cui dovrà interagire l'utente.
 3. Non immersivo: sono le tecnologie più economiche e diffuse, ma offrono un livello minimo di immersione. Rientrano in questa categoria i videogiochi.
- Presenza: rappresenta la sensazione psicologica di trovarsi all'interno dell'ambiente virtuale; si dice che l'utente ha sensazione di presenza quando agisce all'interno della simulazione come se fosse in un ambiente reale, ovvero il livello di realismo che l'utente sperimenta nel rapporto con l'ambiente e nella coerenza della sua evoluzione; volendo sviluppare una simulazione di realtà virtuale, bisogna considerare il grado di presenza che si vuole raggiungere e scegliere in base a ciò il grado e le modalità di interazione con l'ambiente.
- Interazione: rappresenta la possibilità di muoversi nella scena 3D e di interagire con gli oggetti contenuti al suo interno usando appositi controller.

3.2 Storia della realtà virtuale

3.2.1 Stereoscopia

La tecnologia della realtà virtuale si basa su un concetto oggi noto, dimostrato nel 1838 da Charles Wheatstone: il nostro cervello è in grado di elaborare le immagini bidimensionali catturate da ciascuno dei nostri occhi per creare un'unica immagine tridimensionale; Wheatstone dimostrò ciò utilizzando uno stereoscopio, un dispositivo che permette di visualizzare due immagini stereoscopiche affiancate, dando così l'effetto di tridimensionalità all'utente. La stereoscopia rimarrà in auge fino alla fine dell'Ottocento, quando lo sviluppo e la diffusione del cinema prenderà il sopravvento, ma, con l'attuale diffusione e miglioramento della Realtà Virtuale, i principi della stereoscopia sono stati ripresi e sfruttati da numerosi dispositivi low budget, utilizzati principalmente attraverso lo smartphone.

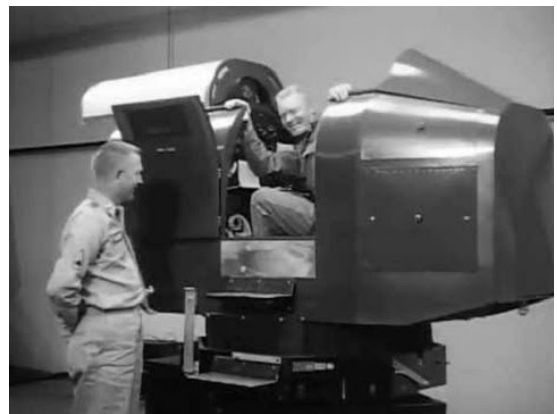
Il dispositivo stereoscopico ideato da Wheatstone era basato su un sistema di specchi e prismi che proietta le immagini corrette all'occhio sinistro e destro, in modo da permettere di avere l'illusione della tridimensionalità guardando immagini bidimensionali; successivamente, anche grazie agli sviluppi della fotografia Wheatstone intravede nuove possibilità per la sua ricerca, dando così il via ai primi esperimenti sulla "stereofotografia". Tuttavia, lo stereoscopio realizzato da Wheatstone non riscontrò successo a causa delle sue ingombranti dimensioni e bisognerà aspettare lo stereoscopio di David Brewster, presentato nel 1851 all'Esposizione Universale di Londra perché la stereoscopia possa conquistare il pubblico e cominciare a diffondersi.

3.2.2 Gli albori della VR: il primo simulatore di volo e il Sensorama

Il primo vero progresso nel campo della realtà virtuale è rappresentato dal "Link Trainer" sviluppato da Edward Link nel 1929, che rappresenta il primo esempio di simulatore di volo; il dispositivo era completamente elettromeccanico, controllato da motori collegati ad un timone per determinare beccheggio e rollio. Un motore era stato aggiunto per simulare turbolenze ed eventuali disturbi di volo. Nel corso della seconda guerra mondiale, i piloti statunitensi utilizzarono proprio il "Link Trainer" per il loro addestramento.



(a) Edward Link



(b) Il Link Trainer

Figura 3.1: Edward Link e la sua invenzione

Lo sviluppo della realtà virtuale continua poi verso la fine degli anni '50, quando il regista Morton Heilig mette a punto il Sensorama, una cabina al cui interno l'utente doveva infilare la testa per guardare la pellicola proiettata e ricevere feedback acustici, tattili e olfattivi.



(a) "Sensorama"



(b) "Virtuality 100CS"

Tuttavia, a causa dell'elevato costo di produzione e della mancanza di finanziamenti, il Sensorama non ebbe successo e il progetto fu abortito dopo soli cinque film prodotti. La tecnica su cui si basava la riproduzione era quella del "Cinerama", tecnica che si basa sulla riproduzione in contemporanea su tre schermi della stessa scena ripresa da tre punti di vista differenti; l'elevato costo di produzione sopracitato era soprattutto dovuto all'impiego di questa tecnica, che richiedeva la creazione di tre pellicole per ogni film prodotto.

3.2.3 I primi visori e i primi dispositivi aptici

Una prima svolta significativa si ha nel 1961, quando due ingegneri della Philco progettano l'Headsight, il primo visore della storia. Nato per la videosorveglianza e per controllare in tempo reale immagini a circuito chiuso provenienti da una telecamera, non è esattamente definibile come visore per la realtà virtuale, ma ha sicuramente ispirato gli sviluppi in questo campo che contraddistinsero gli anni a venire. Bisogna aspettare fino al 1968 per avere un primo prototipo di realtà aumentata ad opera di Ivan Sutherland; ribattezzato la Spada di Damocle, era costituito da due tubi catodici che proiettavano davanti agli occhi immagini 3D. Inoltre il software poteva suggerire la prospettiva a seconda della posizione della testa.

Le industrie videoludiche cominciano quindi a fare ricerca nel campo della realtà virtuale applicata ai videogiochi. Da questa ricerca nasce il progetto del DataGlove, un guanto in grado di convertire i movimenti della mano

in segnali elettrici da trasmettere su fibra ottica. Il DataGlove fu anche adottato dalla NASA per esperimenti nel campo della realtà virtuale e la Nintendo partì proprio da questo progetto per lanciare sul mercato il PowerGlove, un guanto in grado di riconoscere 256 posizioni per dito. E' in questo stesso periodo, precisamente 2 anni prima del lancio del PowerGlove, che viene coniato il termine realtà virtuale grazie a Jaron Lanier, fondatore della Visual Programming Lab, nonché sviluppatore e inventore di numerosi dispositivi aptici e visori.

Si arriva, nel 1991, all'uscita del più influente e iconico dispositivo di realtà virtuale degli anni '90, il Virtuality 100CS; sfruttando il gran parte l'Amiga 3000, il Virtuality 100CS usava un visore per audio e video mentre l'utente poteva sfruttare una pedana per muoversi e un joystick 3D per l'interazione. Come prevedibile, il progetto fallì a causa del costo estremamente proibitivo (circa 60.000 dollari), ma nonostante questo gettò le basi per i visori e per le pedane di movimento di cui disponiamo oggi. [3]

3.2.4 La Realtà Virtuale oggi

La tecnologia informatica di cui disponiamo oggi ha permesso in questi primi anni del XXI secolo un rapido miglioramento della realtà virtuale; la diffusione sempre più massiccia di smartphones sempre più potenti e dotati di schermi ad alta definizione ha portato alla nascita di numerosi

dispositivi VR da utilizzarsi con il telefonino. Ne è un esempio Google Cardboard, un visore "fai-da-te" che si basa sull'utilizzo di specifiche applicazioni dello smartphone. Inoltre, dal 2016 sono stati rilasciati numerosi visori che riprendono i lavori svolti negli anni '90, migliorandoli e sviluppandoli, portando la tecnologia della realtà virtuale su un piano totalmente nuovo. Esempio di questa famiglia di visori sono il pionieristico Oculus Rift, seguito poi dal visore HTC Vive, dagli HoloLens di Microsoft e dal PlayStation VR di Sony.

Parallelamente alla realtà virtuale, durante gli anni '90, si sviluppa una nuova tecnologia, la realtà aumentata, che, pur avendo caratteristiche comuni alla realtà virtuale, si differenzia da questa poiché si pone l'obiettivo di permettere all'utente di visualizzare all'interno del mondo reale oggetti appartenenti a quello virtuale. Inizialmente adottata in pubblicità e video musicali, oggi questa tecnologia viene largamente utilizzata in tutti quei casi in cui l'utente possa aver bisogno di ampliare la percezione che l'utente ha dell'ambiente che lo circonda: per questo sono nate applicazioni che permettono di arredare virtualmente una camera, cataloghi in grado di proiettare oggetti in 3D fino ad arrivare a camerini e specchi virtuali.

Sia il mercato della realtà virtuale che quello della realtà aumentata sono in forte sviluppo e per questo sempre più aziende si avvicinano a questa tecnologia cercando di integrarla e utilizzarla; questo interesse sempre crescente ha favorito enormemente lo sviluppo tecnologico e oggi, sia la realtà aumentata che la realtà virtuale, sono in grado, con i giusti strumenti, di fornire simulazioni estremamente vicine alla realtà. Il principale verso cui si stanno muovendo le ricerche in questo campo è rappresentato dal miglioramento dell'hardware in termini di costi, dimensioni e comodità: lo scopo è renderne più immediato e facile l'utilizzo e aumentare il bacino di utenza abbattendone il principale difetto, ovvero il costo. Altro problema, riscontrato questa volta dagli sviluppatori per la realtà virtuale sembrava essere, fino a poco tempo fa il senso di nausea causato negli utenti, problema che tuttavia non sembra essere più così presente come in precedenza; tuttavia, ora, la preoccupazione maggiore è strettamente legata all'industria della realtà virtuale, ancora poco sviluppata e diffusa presso il pubblico.([19]).

3.3 Architettura di un sistema di realtà virtuale

Qualunque sistema di realtà virtuale si basa su tre elementi principali: un set di elaboratori, un set di dispositivi per le operazioni di I/O e un sistema software.

3.3.1 Set di elaboratori

Il set di elaboratori di un sistema VR si occupa di creare e aggiornare in tempo reale lo stato dell'ambiente virtuale rappresentato; rappresenta quindi il centro di controllo del sistema, che deve essere in grado di ricevere i segnali di input, elaborarli e generare segnali di output. Questi compiti sono svolti grazie a CPU e GPU di alto livello e che si occupano, rispettivamente, dell'esecuzione di calcoli e operazioni matematiche per l'elaborazione dei dati e di effettuare il rendering e di gestire la parte grafica dell'applicazione.

Le applicazioni in realtà virtuale sono estremamente pesanti dal punto di vista computazionale e, per questo, necessitano di CPU, GPU e, in generale, di macchine con un'elevata capacità di calcolo; se prendiamo per esempio un normale HMD con una risoluzione di 2160*1200 divisa su schermi, si ha bisogno di riprodurre le immagini a circa 90 fps, una frequenza molto più alta di quella di un normale monitor, ma necessaria a causa della vicinanza con i nostri occhi; viste queste premesse, un'applicazione in realtà virtuale ha un costo computazionale pari circa al triplo che avrebbe se fosse una normale applicazione. Ciò ovviamente rappresenta uno dei più grandi limiti che questa tecnologia ha oggi, dal momento che le macchine in grado di fornire una tale potenza di calcolo hanno costi molto alti, ai quali si devono aggiungere i costi per i dispositivi I/O VR.

3.3.2 Set di dispositivi I/O

Dispositivi di input

I dispositivi di input utilizzati da un'applicazione di realtà virtuale nascono con l'intenzione di sfruttare tutti i canali utilizzati dall'utente per emettere segnali e informazioni e per interagire con l'ambiente; comunemente, un essere umano interagisce con l'ambiente muovendosi al suo interno e utilizzando gli oggetti in esso contenuti, mentre come principali canali di comunicazione utilizza voce, gesti ed espressioni facciali([16]). Questi due aspetti hanno fatto sì che ci si concentrasse sullo sviluppo di sistemi per il tracking di tutto il corpo e sulla creazione di interfacce che permettessero di interagire tramite comandi vocali.

Per ragioni pratiche, tuttavia, si è dimostrato necessario limitare i possibili segnali di input inviabili dall'utente; ad esempi, dal momento che la mano concentra in una zona ristretta molti gradi di libertà rispetto alle altre parti del corpo, per molte applicazioni è sufficiente sfruttare un sistema di hand tracking; allo stesso modo, di notevole importanza sono i sistemi di head tracking e di eye tracking, sistemi che permettono di aggiornare la finestra di visualizzazione in coordinazione con i movimenti dell'utente.

Tra i principali sistemi di input disponibili oggi possiamo citare:

- Mouse e tastiera: rappresentano i dispositivi di input più comuni per i PC, ma non consentono una forte immersione all'interno dell'ambiente virtuale;
- Joystick e gamepad: rappresentano i dispositivi di input più comuni per l'interazione con avatar virtuali nei videogiochi, ma, analogamente a quanto avviene per mouse e tastiera, non permettono una forte sensazione di immersione;
- Speech recognizer: sistema di riconoscimento audio (es: microfono) che consente all'utente di interagire con l'ambiente virtuale tramite l'uso della voce;

In particolare, negli ultimi tempi hanno assunto sempre maggior importanza i sistemi di tracking, che sono diventati il sistema per eccellenza per aumentare il senso di immersione all'interno di un sistema VR e che per questo meritano un discorso a parte; nello specifico, i sistemi di tracking possono essere:

- Meccanici: molto accurati e con un basso ritardo, ma inadatti a tracciare grossi volumi a causa del loro ingombro;
- Magnetici: costituiti da un dispositivo in grado di generare campi magnetici e da un ricevitore in grado di determinare forza e angolazione del campo; sono caratterizzati da una buona accuratezza, da un basso costo e possono essere utilizzati per tracciare volumi mediamente estesi; tuttavia il campo generato dal dispositivo è, per sua natura, soggetto a interferenze elettromagnetiche, che possono influire pesantemente sulla precisione e sull'affidabilità del sistema;
- Ottici: utilizzano telecamere per riconoscere la posizione del tracker e calcolarne le coordinate all'interno dello spazio; sono veloci, efficienti e permettono di lavorare anche in grandi ambienti; i principali difetti di questi dispositivi sono il costo spesso elevato e il loro peso eccessivo;
- Acustici: utilizzano tre emettitori e tre microfoni per calcolare la distanza tra sorgente e destinazione attraverso la triangolazione del segnale; sono molto economici e leggeri, ma estremamente sensibili al rumore e hanno una bassa precisione, a causa della variazione della velocità del suono attraverso l'aria e a causa di possibili eco generati;

- Inerziali: utilizzano accelerometri e giroscopi per effettuare il calcolo della variazione di velocità angolare e di posizione lungo ogni asse di movimento; sono veloci e non limitati da un particolare range, ma sono imprecisi per piccoli movimenti.

Come già detto, notevole importanza rivestono oggi i sistemi di eye tracking, ovvero quei sistemi che vanno a misurare la direzione verso la quale l'utente è rivolto, allo scopo di aggiornare la scena di conseguenza; questi dispositivi possono basarsi su tre differenti tecnologie:

- Ottici: utilizzano una fonte di luce, solitamente un LED a infrarossi, per generare un riflesso sulla superficie oculare, utilizzato per determinare la direzione verso cui è rivolto lo sguardo;
- Elettrooculari: utilizzano un elettrooculogramma per misurare il potenziale generato all'interno dell'occhio dalle attività metaboliche della retina; questo dispositivo ha il vantaggio di essere non invasivo, ma è più instabile e, pertanto, necessita di essere calibrato spesso.
- Elettromagnetico: si basano sulla misurazione del voltaggio indotto su un induttore collegato ad una lente;

Il problema principale di queste tecnologie è rappresentato dal costo molto elevato, motivo per il quale vengono impiegate solo nei settori in cui tale tecnologia è fondamentale.

Recentemente sono, infine, stati sviluppati treadmill e piattaforme a pressione, ovvero sistemi di tracking che consentono all'utente di muoversi liberamente all'interno dell'ambiente virtuale; il sistema è generalmente costituita da una pedana omnidirezionale sopra la quale l'utente può camminare e correre, sempre rimanendo fermo sul posto. Questa tecnologia, unita ad un efficace sistema di tracking per testa e mani dell'utente fornisce un elevato senso di immersione e di interazione con l'ambiente; nuovamente, il difetto principale di questi dispositivi risulta essere il costo, sebbene sia una tecnologia sempre più diffusa ed utilizzata.

Dispositivi di output

I dispositivi di output sono invece quei dispositivi che forniscono un feedback all'utente andando a stimolare uno specifico senso dello stesso; tuttavia alcuni sensi sono difficili da stimolare attraverso una simulazione di realtà virtuale, e per questo spesso questo aspetto viene ignorato, come accade per esempio con i sensi di gusto e olfatto. Di seguito vengono analizzati i principali sistemi per dare un feedback sensoriale all'utente:

- Feedback visivo: i display utilizzati nel contesto della realtà virtuale devono garantire la visione stereoscopica e devono garantire la possibilità di aggiornare le immagini mostrate riproducendo il movimento dell'utente; inoltre devono garantire la più alta risoluzione possibile, con il giusto livello di luminosità, contrasto e colore, in modo da rappresentare l'ambiente virtuale nel modo più gradevole possibile per l'utente. Oggi esistono molteplici tecniche per realizzare la visione stereoscopica: gli head mounted displays(HMD), per esempio, utilizzano un elmetto provvisto di una lente per ogni occhio; esistono poi due tipologie di lenti per riprodurre visione stereoscopica, ovvero lenti passivi e lenti attivi. Le prime sono dotate di un filtro polarizzatore, mentre lo schermo riproduce sia l'immagine destra che quella sinistra; le due lenti filtreranno l'immagine facendo vedere solo l'immagine corretta all'occhio sinistro e a quello destro; le lenti attive, viceversa, sono dotate di un otturatore sincronizzato con lo schermo, che alterna immagine sinistra e immagine destra, mentre l'otturatore oscura la vista dell'occhio opposto. I dispositivi per la realtà virtuale maggiormente diffusi oggi utilizzano HMDs;
- Feedback tattile: nonostante il tatto sia un senso di primaria importanza nell'esplorazione dell'ambiente, oggi è ancora molto difficile fornire un feedback realistico all'utente; per

quanto sia comune oggi riuscire a utilizzare le mani dell'utente per interagire con un oggetto, ancora non è possibile replicare la sensazione di aver afferrato un oggetto o di aver toccato un elemento dell'ambiente e, nella maggior parte dei casi, il feedback tattile si limita ad una vibrazione del controller al verificarsi di determinati eventi. Tuttavia, recentemente, Microsoft avrebbe brevettato una nuova tecnologia che, grazie a stimoli mandati alla mano dell'utente attraverso una serie di elettrodi, riuscirebbe a simulare la presa di un oggetto, impedendo di chiudere la mano oltre un certo limite ([18]);

- Feedback sonoro: settore fortemente sviluppato, analogamente a quanto detto per il feedback visivo, esistono oggi numerose tecnologie che permettono di riprodurre un segnale sonoro, dalle più fedeli e costose a quelle più commerciali e meno sofisticate; negli ultimi anni, in particolare, è nata una tecnica di registrazione che permette di registrare e riprodurre audio 3D. Queste tecniche prendono il nome di tecniche di registrazione e riproduzione binaurale e permettono di riprodurre fedelmente l'apparato uditivo umano e permettono all'ascoltatore di riconoscere precisamente la direzione da cui proviene il suono e a che distanza la sorgente si trovi. Utilizzando efficacemente questa tecnica ed integrandola con precisione all'interno della propria applicazione, è possibile creare un effetto di immersione nettamente superiore a quanto avveniva in precedenza, dal momento che l'utente si trova completamente immerso in un ambiente dal quale riceve stimoli sonori fedeli e reali e dei quali riesce a percepire origine e distanza.
- Feedback olfattivo: esistono determinati campi in cui anche la simulazione degli specifici odori che si registrano nella vita reale sono importanti e meritano di essere riprodotti; ciò avviene, per esempio, all'interno delle simulazioni chirurgiche che devono :

"provide the proper olfactory stimuli at the appropriate moments during the procedure. Similarly, the training of emergency medical personnel operating in the field should bring them into contact with the odors that would make the simulated environment seem more real and which might provide diagnostic information about the injuries that simulated casualty is supposed to have incurred" ([1])

Per questo motivo sono stati sviluppati dispositivi che permettono l'emissione di odori percepibili dall'utente; il sistema più diffuso per emettere odori è quello di creare dei punti di immagazzinamento di odori sotto forma di liquido, gel o cere, che possono quindi essere emessi al momento opportuno. Un altro meccanismo diffusamente adottato consiste nel diluire tali sostanze in un gas che viene emesso verso l'utente. Recentemente è stato annunciato il progetto di FEELREAL([8]) una periferica compatibile con diversi HMDs, come ad esempio Oculus Rift, HTC Vive e Playstation VR, che permette, sfruttando una cartuccia contenente fino a 9 odori su un totale di 255, di inviare un feedback olfattivo specifico a seconda dell'applicazione, lasciando comunque libertà all'utente di scegliere quali odori simulare;

Software

Il software è lo strumento che ha il compito di mettere in comunicazione e sincronizzare le periferiche, sia di input che di output, con l'elaboratore; all'interno del software si trovano:

- Un gestore degli eventi: gestisce il flusso di eventi generati durante l'esecuzione dell'applicazione; gli eventi possono essere generati dall'utente attraverso l'interazione con l'ambiente, da un trigger relativo ad un particolare stato del sistema oppure da pacchetti ricevuti tramite un protocollo di rete;

- Un motore grafico: si occupa di comunicare alla GPU quali elementi grafici devono essere renderizzati e trasmessi al sistema di visualizzazione;
- Un motore fisico: effettua i calcoli relativi all'aggiornamento delle geometrie, delle posizioni, delle velocità in base allo stato del sistema ed in base agli input dell'utente;
- Un motore audio: si occupa di sincronizzare, modulare ed eseguire il suono in base alla posizione e gli eventi relativi all'utente;
- Un data loader: l'entità che invia al sistema i dati necessari all'ambiente virtuale e all'applicazioni; tali dati possono essere modelli 3D, audio, texture, immagini e possono provenire da diverse fonti, come HDD, databases o internet;

3.4 Applicazioni

Per quanto oggi il termine realtà virtuale sia familiare per la maggior parte delle persone, spesso non si comprendono le reali potenzialità di questa tecnologia e sui suoi campi di applicazioni, limitandola spesso alla sola applicazione in campo videoludico, spesso viziati anche dalla forte spinta che questa industria ha dato allo sviluppo della realtà virtuale stessa. A differenza di quanto si pensi, infatti, la realtà virtuale trova oggi impiego in numerosi campi, molto diversi dal gaming e dall'intrattenimento, grazie alle enormi potenzialità che tale tecnologia ha dimostrato di avere in termini di sviluppo di nuovi modelli, metodi di apprendimento e di comunicazione.

Oggi, i settori che principalmente stanno investendo nella realtà virtuale sono [14]:

- Gaming: come già accennato, è l'industria che più si concentra sullo sviluppo della realtà virtuale, cercando di proiettare i giocatori all'interno del mondo di gioco attraverso simulazioni sempre più accurate e immersive; al momento, tutti i colossi del mondo videoludico stanno investendo su questa tecnologia, sia sviluppando nuovi giochi, sia sviluppando nuove periferiche hardware e dispositivi;
- Militare: fin dalla comparsa del primo simulatore di volo, il settore militare ha cominciato ad interessarsi alle simulazioni virtuali a scopo di addestramento, dal momento che è possibile replicare situazioni di pericolo in cui è necessario reagire prontamente, rimanendo però all'interno di un ambiente sicuro. Inoltre la realtà virtuale si è dimostrata essere anche meno costosa rispetto ai metodi di addestramento tradizionale;
- Educazione: attraverso la realtà virtuale è possibile permettere l'interazione di un grande numero di partecipanti all'interno dello stesso ambiente virtuale, rendendola un ottimo strumento in contesti didattici; all'interno dell'ambiente virtuale è possibile rappresentare comodamente e in modo più semplice una gran quantità di dati complessi e permettendo agli studenti di interagire con essi all'interno dell'ambiente.
- Medicina e salute: dopo il settore dei videogiochi, sono quelli che maggiormente sfruttano questa tecnologia, per via della possibilità di simulare operazioni chirurgiche, simulare l'anatomia umana a scopo di training per infermieri e medici fino anche all'assistenza al trattamento di fobie, autismo e disabilità;
- Intrattenimento: altro esempio di settore notevolmente attivo nell'utilizzo della realtà virtuale; troviamo in questo contesto numerose opere cinematografiche, televisive, musicali e artistiche, fino ad arrivare ad attrazioni all'interno di parchi a tema che fanno uso di visori per la realtà virtuale;
- Cultura e turismo: l'utilizzo delle tecnologie immersive della VR rappresenta un modo nuovo e stimolante di comunicare informazioni al pubblico di musei, gallerie d'arte, monumenti, centri storici e visite virtuali;

- Ingegneria: utilizzata soprattutto in fase di progettazione, l’utilizzo di tale tecnologia permette di visualizzare e valutare un progetto, permettendo di localizzare eventuali errori o malfunzionamenti prima che venga avviata la fase di costruzione;
- Visualizzazione scientifica: l’interesse della ricerca scientifica nella simulazione virtuale sta crescendo sempre più, grazie allo sviluppo tecnologico e informatico che permette di visualizzare dati sempre più complessi e massicci in maniera sempre più semplice e intuitiva, aiutando gli studiosi durante le fasi di analisi e comprensione dei dati raccolti.

Il presente lavoro di tesi, come già accennato, nasce e si svolge in un ambito particolare, quello dell’ergonomia per il settore aerospaziale.

3.5 VR per l’ergonomia nel settore aerospaziale

Lo studio e lo sviluppo in realtà virtuale di un modulo ancora in fase di costruzione permette di studiare e eventualmente ridefinire il progetto per renderlo più adatto ai suoi scopi e di rendere i suoi spazi più comodi per l’equipaggio. L’ambiente che si vuole studiare è quello di microgravità, riproducibile solo per pochi secondi, attraverso un complesso procedimento che prevede l’impiego del moto parabolico di un aereo (il velivolo C9 della NASA effettua voli parabolici producendo assenza di peso per non più di 25 secondi). Viceversa, all’interno dell’ambiente virtuale, non solo è più semplice definire un ambiente di microgravità, ma è anche più sicuro svolgere test utilizzando meno risorse; dall’altro lato, la difficoltà nell’utilizzare l’ambiente virtuale è quella di dare la giusta sensazione di presenza all’utilizzatore e di farlo sentire come se stesse operando in ambiente reale.

3.5.1 Stato dell’arte in TASI

Attualmente il team COSE Centre lavora sul progetto Virtual Environment Research In Thales Alenia Space 4U, un plugin per lo sviluppo di applicazioni in realtà virtuale tramite Unreal Engine (UE); tale plugin è stato utilizzato per lo sviluppo del modulo spaziale Columbus, della Chinese Space Station (CSS) e per lo sviluppo del modulo lunare Lunar Orbital Platform-Gateway (LOP-G), modulo su cui si è concentrato lo sviluppo di questa tesi.

Columbus

Interamente sviluppato da Thales Alenia Space Italia, sia come prodotto fisico che come prodotto virtuale, Columbus è un modulo di ricerca della International Space Station lanciato nel 2008. Al suo interno dispone di vari rack standard per eseguire esperimenti in microgravità; con il termine rack si intende un telaio a struttura modulare per permettere la raccolta di strumenti elettronici. I vari esperimenti conducibili in questo modulo comprendono esperimenti sul comportamento dei fluidi in microgravità, sugli effetti che il volo spaziale ha sul corpo umano ed esperimenti biologici su microrganismi. [7]

CSS

Il lancio del primo modulo della Chinese Space Station è programmato per il 2020 e sarà seguito dal lancio di altri due moduli sperimentali, che saranno utilizzati per vari esperimenti, come accade sul modulo Columbus.

La Cina ha usato la CSS per favorire la cooperazione internazionale, annunciando l’opportunità per tutte le Nazioni Unite di essere coinvolte in missioni scientifiche a bordo della CSS; a conferma di ciò, lo scorso Agosto, i due astronauti, Samantha Cristoforetti e Matthias Maurer, si sono uniti a sedici colleghi cinesi per una sessione di addestramento facente parte di una missione d’addestramento a bordo della CSS. [9]

LOP-G

Il modulo LOP-G, acronimo di Lunar Orbital Platform-Gateway, precedentemente noto come Deep Space Gateway, è una nuova infrastruttura cis-lunare la cui costruzione è prevista a partire dal 2020. All'interno del progetto, Thales Alenia guiderà, in qualità di prime contractor, uno degli studi sia per ESPRIT (European System Providing Refuelling, Infrastructure and Telecommunications) che per I-HAB (International-Habitat).

La stazione, in particolare, rappresenta un test per il sistema ambientale a lungo termine che manterrà in vita gli astronauti durante i lunghi viaggi, nonché come punto di partenza per le prossime missioni di atterraggio sulla superficie lunare, studio di asteroidi o per le prossime missioni su Marte.

Dopo il posizionamento del primo elemento in orbita cislunare, i seguenti saranno lanciati come parte dello SLS, in comanifest della capsula Orion dotata di equipaggio. [11]



Figura 3.3: "Ricostruzione 3D del modulo LOP-G"

In questo contesto, il lavoro di tesi, svolto nel tema TRO-1 di COSE Centre, è stato suddiviso in due parti: nella prima ci si è concentrati sullo sviluppo di manichini realistici e le cui pose potessero essere modificate tramite la cinematica inversa, per poterle quindi salvare e caricare a piacimento; in questa fase si riprende e amplia il lavoro svolto in precedenza dalla tesista Paola Minai per migliorare la qualità dei manichini e renderli più realistici e fedeli nella riproduzione delle pose desiderate; inoltre, si pone come obiettivo quello di operare direttamente attraverso l'editor di Unreal Engine, senza avere quindi la necessità di passare alla modalità "simulate" offerta dal motore grafico. In questo modo l'utente può interagire in maniera più semplice e immediata con l'ambiente virtuale considerato, modificandolo e adattandolo al manichino rimanendo sempre all'interno dell'editor e senza cambiare modalità.

Oltre alla resa grafica dei manichini, si vuole migliorare il loro movimento e la loro capacità di replicare fedelmente le pose assunte dagli operatori a bordo dell'ISS, ottenendo un risultato che sia utilizzabile per la valutazione delle condizioni ergonomiche di un modulo.



Nella seconda parte si è utilizzato il visore di realtà virtuale HTC Vive per proseguire lo sviluppo del plugin VERITAS 4U, provato sul progetto del modulo cislunare

Figura 3.4: HTC Vive PRO

LOP-G, per il miglioramento del sistema di movimento in microgravità. Lo scopo è quello di sviluppare un nuovo sistema di movimento che possa dare un grado maggiore di realismo rispetto al sistema attuale, che risulta essere poco reale e difficile da utilizzare per eseguire determinati compiti.

Il sistema di movimento presente prevede che l'utente, tenendo premuto un tasto del controller, possa allontanare o avvicinare a sé il controller stesso, muovendo il proprio alter ego virtuale di conseguenza, senza curarsi di simulare il movimento in microgravità e permettendo di muoversi anche senza bisogno di aggrapparsi alle handrail presenti. Il sistema di rotazione, invece, prevede, alla pressione di un tasto, di ruotare tutto l'ambiente virtuale intorno a noi, lasciando fermo il giocatore, cosa che provoca un forte senso di nausea e di disorientamento.

Si vogliono quindi progettare un sistema di movimento e un sistema di rotazione che permettano non solo di migliorare la qualità della simulazione, ma che limitino anche il senso di disorientamento e di nausea, provato soprattutto dagli utenti meno avvezzi ad utilizzare visori ottici per la realtà virtuale come l'HTC Vive.

Parte II

Seconda Parte

Capitolo 4

Sviluppo dei manichini

4.1 Make Human

Make Human è un'applicazione open source per la creazione di modelli umanoidi; è disponibile su tutti i sistemi operativi, essendo sviluppato in Python usando OpenGL e QT, utilizzando un'architettura a plugin.

Il software utilizza la tecnologia di morphing: si parte da una umano base standard che può quindi essere trasformato in una gran varietà di personaggi ottenuti per interpolazione lineare. Per la manipolazione del morphing si utilizza una GUI molto semplice ed intuitiva che permette di modificare tutti i parametri che caratterizzano la corporatura di un essere umano, dall'altezza alla circonferenza delle caviglie. Make Human dispone anche di una libreria di modelli di vestiario piuttosto basilare, ma sufficiente per gli scopi della tesi e di acconciature.

Inoltre il programma presenta un semplice sistema di pose per simulare i movimenti muscolari, effettuata tramite una tecnica di mesh relaxing, e di verificare eventuali errori nel modello.

A differenza dei programmi di grafica maggiormente utilizzati, Make Human è sicuramente più facile da utilizzare, con un'interfaccia veloce ed intuitiva, permettendo di ottenere rapidamente un modello 3D umanoide, utilizzabile direttamente o come base per altri modelli.

4.2 Blender

Blender è un software che dispone di numerosissime caratteristiche ed è in grado di gestire qualunque step della pipeline di creazione di grafica 3D. Permette infatti di gestire la modellazione, sia attraverso modificatori sia attraverso lo sculpting, la creazione di materiali e texture, il rigging, l'animazione e la gestione di particellari e flocks fino alla creazione di script personali in Python.

L'interfaccia utente di Blender permette di avere molteplici modalità di visualizzazione della scena e di editing dei contenuti di quest'ultima; le interfacce maggiormente utilizzate sono state 3D view, node editor e UV/image editor. La prima rappresenta l'interfaccia principale del programma e viene utilizzata per la maggior parte delle operazioni che riguardano la modellazione o l'animazione; node editor e UV/image editor, invece, sono le due interfacce principali per la creazione di materiali e textures: il node editor permette di creare un materiale collegando diversi nodi tra loro fino a raggiungere il risultato desiderato, mentre il secondo esegue l'unwrap del modello, proiettandolo sul piano 2D in modo da poter applicare la texture al modello.

All'interno della 3D view è inoltre possibile impostare la modalità di interazione con il modello, a seconda di quali operazioni si desiderano effettuare sul modello; le modalità di interazioni principali sono:

- Object Mode: permette di ruotare, scalare o spostare i modelli presenti nella scena, andando a calcolare la posizione di tale oggetto rispetto alle coordinate del mondo;
- Edit Mode: permette di modificare la struttura del modello, andando a modificare, aggiungere o eliminare vertici, spigoli e facce;
- Sculpt Mode: permette sempre la modifica della struttura del modello, ma andandolo, in questo caso, a "sculpire" usando differenti pennelli;
- Weight Mode: permette di assegnare i pesi allo scheletro associato alla mesh, ovvero di definire come il movimento dello scheletro modifica la posizione della mesh.

4.3 Creazione del personaggio

4.3.1 Creazione della mesh

La mesh è l'insieme di vertici, spigoli e facce che definisce la struttura esterna di un modello 3D, esprimibile sia in forma poligonale che in forma parametrica.

Le prime sono delle superficie costituite da un numero più o meno elevato di facce poligonali, che in Blender sono rappresentate dalle facce, ovvero da un insieme ordinato e chiuso di vertici: più questo numero è elevato e più è elevata la qualità della rappresentazione della mesh. Tuttavia, nella creazione della mesh bisogna prestare attenzione al numero di lati del poligono, in quanto poligoni con più di quattro facce possono creare artefatti ed effetti indesiderati sulla superficie del modello. Per modellare una mesh poligonale si deve agire direttamente sul singolo elemento (vertice, spigolo o faccia), andandolo a modificare manualmente o tramite modificatori; questa tecnica approssima molto bene gli oggetti meccanici, ma risulta poco efficace quando è necessario realizzare superfici più complesse, per le quali è necessario passare ad una rappresentazione parametrica. Le mesh che utilizzano questa rappresentazione sfruttano determinate superfici e curve, come le Spline o le NURBS (Non Uniform Rational Basis-Splines) per rappresentare la propria superficie; le superfici così ottenute possono essere modificate andando ad agire sulle maniglie di controllo delle curve stesse, ottenendo superfici estremamente realistiche, ma andando ad aumentare notevolmente il costo computazionale; proprio per questo motivo è bene limitare il loro utilizzo ai soli casi in cui si rende necessario.

Come già detto, Make Human semplifica molto il processo di creazione di una mesh umanoide di base; grazie alla possibilità di modificare le varie dimensioni del corpo umano è possibile creare una prima bozza che rispecchi fedelmente le dimensioni stabilite per i vari percentili. Successivamente si sono scelti dei vestiti di cui sono state poi modificate le texture. Una volta terminata la mesh, questa può essere salvata ed esportata in formato fbx per poter essere successivamente importata in Blender.

4.3.2 Pulizia della topologia, creazione dello scheletro e associazione con la mesh

Una volta importata la mesh su Blender bisogna controllare che tutto sia stato esportato ed importato correttamente ed eventualmente modificare la topologia delle varie meshes (corpo, capelli, occhi e vestiti). In questo caso è stato sufficiente agire sulle varie parti del corpo, formato solo da testa e braccia non connesse tra loro; allo scopo di alleggerire il conteggio dei vertici è infatti tecnica comune eliminare tutte le parti non necessarie o coperte da una mesh. Tuttavia nell'esportazione spesso si creano artefatti e/o compenetrazione con altre meshes nelle zone di confine della topologia; in questo caso basterà passare alla edit mode di Blender e, dopo aver selezionato il o i vertici interessati, spostarli o eliminarli per ricreare la mesh. Dopo queste piccole modifiche sono

state controllate ed eventualmente corrette tutte le misure antropometriche, in modo da avere la certezza che i percentili definiti fossero corretti.

Terminato questo passaggio è necessario creare uno scheletro da poter associare alla mesh per poterla muovere e animare. La fase di creazione di uno scheletro si divide in due passaggi: rigging e skinning. Il rigging riguarda la creazione delle ossa e del loro collegamento. In questa fase si crea quindi la gerarchia dello scheletro, andando ad impostare vincoli sulle singole ossa; in particolare in Blender si possono creare quattro tipologie di osso differente:

- Osso di controllo: sposta la mesh cui è associato all'interno della scena; solitamente si ha un nodo controllo che rappresenta la radice dello scheletro;
- Osso di deformazione: influenza la parte di mesh cui viene associato in fase di skinning, andandola a muovere, ruotare o scalare;
- Osso di funzione: chiamati anche end effector, sono le ossa che muovono la mesh seguendo la cinematica inversa, sfruttando la gerarchia di ossa dello scheletro;

Nella fase di skinning invece si va ad associare il manichino alla mesh, in modo che quest'ultima possa seguire il movimento dello scheletro. Per effettuare lo skinning si devono definire, osso per osso, le aree di mesh influenzate. Per fare ciò si segue una procedura chiamata *weighting*, che per mezzo di colori che vanno dal rosso al blu, permette di definire il grado di influenza che il singolo osso ha su una parte della mesh. In Blender il blu corrisponde al grado di influenza nulla (quindi le parti blu non verranno mosse a seguito del movimento dell'osso), mentre il rosso corrisponde al grado massimo di influenza; andando a sfruttare tutte le sfumature intermedie si arriva a simulare realisticamente l'effetto che ha il movimento di un osso su una determinata mesh. Questa procedura può essere automatizzata attraverso la procedura di *parenting*, che va a generare automaticamente dei gruppi di vertici, cui assegnare poi i pesi corrispondenti; essendo una procedura automatizzata, spesso è necessario andare a modificare i pesi assegnati per avere precisamente il comportamento voluto.

4.3.3 Applicazione dei modificatori

In Blender, i modificatori, sono operazioni automatiche che permettono di migliorare la qualità della mesh o di ottenere particolari effetti, che risulterebbero altrimenti complesse e laboriose; è possibile applicare più modificatori in sequenza, prestando però attenzione all'ordine con cui questi vengono selezionati in quanto un diverso ordine può portare ad effetti differenti; le modifiche apportate da un modificatore divengono effettive e permanenti solo dopo l'applicazione dello stesso. Per lo sviluppo dei manichini sono stati principalmente usati due modificatori, *armature* e *subdivision surface*.

Il primo è necessario per la costruzione dei sistemi scheletrici, permettendo di animare e posizionare la mesh; il modificatore viene applicato automaticamente in fase di skinning.

Il secondo, invece, viene utilizzato per suddividere le facce che compongono la mesh in facce più piccole, in modo da ottenere una maggior risoluzione per la mesh; il pannello di controllo del modificatore permette di aumentare o ridurre il numero di suddivisioni. Usare questo modificatore, invece di creare una mesh ad alta definizione, permette di evitare di salvare e memorizzare una mole considerevole di dati e di alleggerire il carico computazionale in fase di modellazione.

4.4 Definizione dei percentili NASA

All'interno del documento NASA [12] usato per la costruzione dei manichini, si fa riferimento al concetto di *percentile* per indicare la suddivisione della popolazione in gruppi di appartenenza

caratterizzati da precise misure antropometriche; inoltre, tale concetto, offre un termine di paragone per la descrizione delle misure di una specifica popolazione. Le misure del 25° indicano, per esempio, che il 25% della popolazione considerata avrà misure pari o inferiori a quelle riportate, mentre il rimanente 75% avrà misure maggiori, misure che ricadranno quindi nei percentili successivi. I percentili definiti dal documento sopra citato sono stati definiti pensando al personale di bordo e di visita della ISS, assumendo si tratti di individui in buona salute e con un'età media di 40 anni, senza considerare razza o sesso. Le misure sono state registrate nel 2000 e definiscono tre percentili per ogni sesso, ovvero 5°, 50° e 95°, considerando per il sesso femminile la popolazione giapponese e per il sesso maschile quella americana; il motivo per cui sono state scelte queste due popolazioni risiede nel fatto che, mediamente, le donne giapponesi siano le più minute al mondo, mentre i maschi americani risultano essere i più robusti. Tra i percentili definiti, quelli più interessanti e considerati per eseguire test sono il 5° femminile e il 95° maschile, che vanno ad escludere il 10% della popolazione, le cui misure risulterebbero troppo piccole o troppo grandi. Le misure definite nel documento [12] vengono utilizzate anche al di fuori del settore aerospaziale, per scopi sia militari che civili.

4.4.1 Calcolo dei percentili

Il calcolo di misure antropometriche corrette ed utilizzabili è un processo complesso, che richiede una gran quantità di informazioni che rispecchino il soggetto che si vuole rappresentare; una volta definito quest'ultimo, per definirne i percentili, si procede selezionando una parte della popolazione che rispecchi i requisiti richiesti e se ne riportano successivamente le misure su una superficie gaussiana, il cui picco identificherà il 50° percentile. Si dividono quindi i dati per classi in ordine crescente, per tutte le classi si calcola la frequenza in termini percentuali, frequenze dalle quali si ricava la percentuale cumulativa, che indica il percentile della popolazione considerata.

Nel definire i percentili richiesti, soprattutto per quelli utilizzati in campo aerospaziale, bisogna considerare diversi fattori che vanno ad incidere notevolmente sulle misure antropometriche quali:

- Età: variabile fondamentale nella definizione di qualsiasi percentile, in quanto rappresenta uno dei maggiori fattori di variazione per quanto riguarda peso, massa e altezza;
- Genere: si tiene conto del fatto che, mediamente, le donne siano più basse e pesino meno degli uomini; Vestiti: fattore da considerarsi fondamentale quando l'equipaggiamento risulta particolarmente voluminoso rispetto alle misure antropometriche definite, come accade per l'equipaggiamento di un astronauta: la tuta di pressurizzazione, ad esempio, viene gonfiata di aria per permettere all'utente di respirare, ma ciò provoca un aumento notevole del volume dello stesso, che deve essere necessariamente considerato;
- Postura: la postura che l'utente assume durante lo svolgimento di un compito può andare ad alterare le misure antropometriche definite e deve quindi essere valutata attentamente; nel caso di studio si è considerata come riferimento la posizione neutra in ambiente di microgravità, anch'essa descritta nel documento [12], caratterizzata da una ben precisa posizione degli arti superiori e inferiori;

Ai fini della tesi sono stati modellati i manichini rappresentanti del 95° e 50° percentile maschile e de 50° e 5° percentile femminile, che saranno mostrati in seguito.

4.4.2 Percentili femminili

Nelle tabelle riportate più avanti vengono riportate le misure usate per la costruzione dei manichini femminili; ogni tabella è composta da una figura su cui sono riportati numeri collegati alle voci

presenti.

La tabella 4.1 riporta, per ogni percentile femminile, altezze e ampiezze di varie parti del corpo, come ad esempio caviglie o spalle; la figura 4.1 mette in evidenza, per ogni voce della tabella, a quale grandezza si fa riferimento.

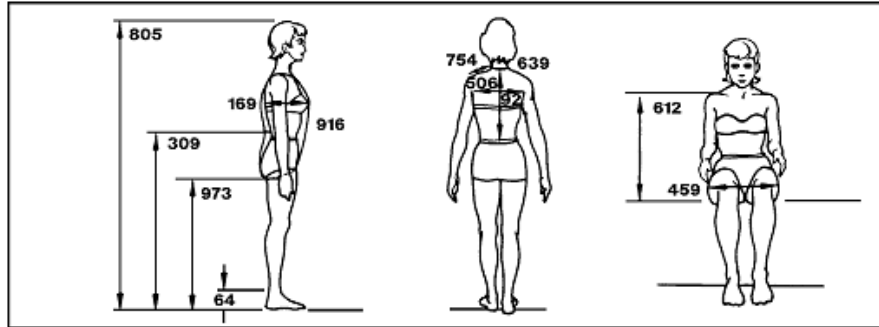


Figura 4.1: Figura esplicativa dei percentili femminili riguardanti altezze e ampiezze

N	Descrizione	5° Percentile	50° Percentile	95° Percentile
805	Altezza	148.9	157.0	165.0
64	Altezza caviglie	5.2	6.1	7.0
169	Profondità busto	17.4	20.5	23.6
754	Lunghezza spalla	11.3	13.1	14.8
639	Circonferenza collo	34.5	37.1	39.7
23	Altezza spalle	119.6	127.1	134.5
873	Altezza tibiale	35.9	39.3	42.7
122	Ampiezza deltoidi	35.6	38.9	42.1
223	Ampiezza petto	24.5	26.8	29.0
457	Ampiezza bacino	30.5	32.9	35.3

Tabella 4.1: Percentili femminili standard NASA relativi alla figura 4.1

La tabella 4.2 riporta le dimensioni della mano, le altezze e le ampiezze in posizione seduta, facendo riferimento alla figura 4.2.

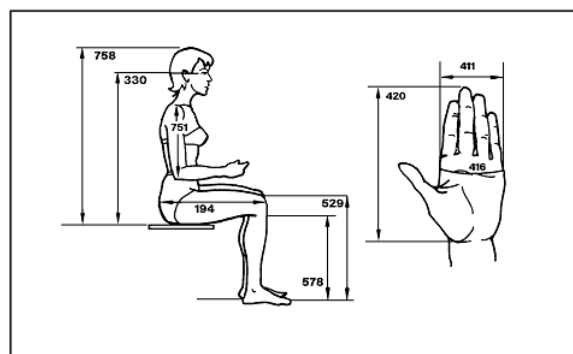


Figura 4.2: Figura esplicativa dei percentili femminili riguardanti mani e seduta

N	Descrizione	5° Percentile	50° Percentile	95° Percentile
420	Lunghezza mano	15.8	17.2	18.7
411	Ampiezza mano	6.9	7.8	8.6
416	Circonferenza mano	16.5	17.9	19.3
785	Altezza seduta	78.3	84.8	91.2
330	Altezza occhi in seduta	68.1	73.8	79.5
529	Altezza ginocchio in seduta	41.6	45.6	49.5
578	Altezza popliteo	34.7	38.3	41.9
751	Lunghezza spalla-gomito	27.2	29.8	32.4
194	Lunghezza natica-ginocchio	48.9	53.3	57.8

Tabella 4.2: Percentili femminili standard NASA relativi alla figura 4.2

Seguendo lo stesso principio, di seguito sono riportate le tabelle e le figure di riferimento per tutte le altre grandezze considerate.

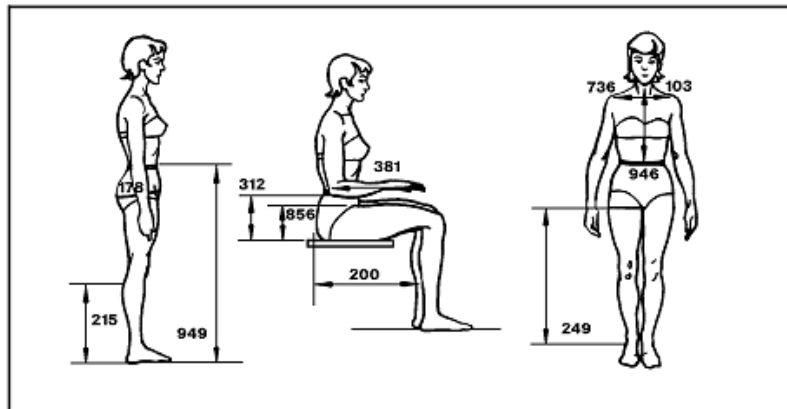


Figura 4.3: Figura esplicativa dei percentili femminili riguardanti ulteriori altezze e ampiezze

N	Descrizione	5° Percentile	50° Percentile	95° Percentile
949	Altezza vita	90.1	96.7	103.4
249	Altezza cavallo	65.2	70.6	76.1
215	Altezza polpaccio	25.5	28.9	32.3
103	Ampiezza biacromiale	32.4	35.7	39.0
178	Circonferenza natiche	79.9	87.1	94.3
312	Altezza gomito a riposo	207	25.0	29.3
856	Altezza coscia	11.2	12.9	14.5

Tabella 4.3: Percentili femminili standard NASA relativi alla figura 4.3

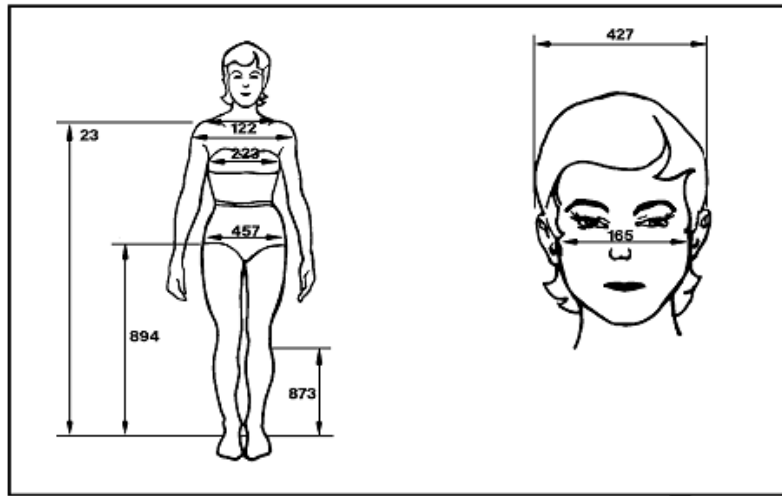


Figura 4.4: Figura esplicativa dei percentili femminili riguardanti le dimensioni del cranio

N	Descrizione	5° Percentile	50° Percentile	95° Percentile
23	Altezza acromiale	119.6	127.1	134.5
894	Altezza bacino	71.0	76.7	82.4
873	Altezza tibiale	35.9	39.3	42.7
122	Ampiezza deltoidi	35.6	38.9	42.1
233	Ampiezza del petto	24.5	26.8	29.0
457	Ampiezza bacino	30.5	32.9	35.3
165	Ampiezza viso	13.3	14.5	15.7
427	Ampiezza cranio	14.4	15.6	16.8

Tabella 4.4: Percentili femminili standard NASA relativi alla figura 4.4

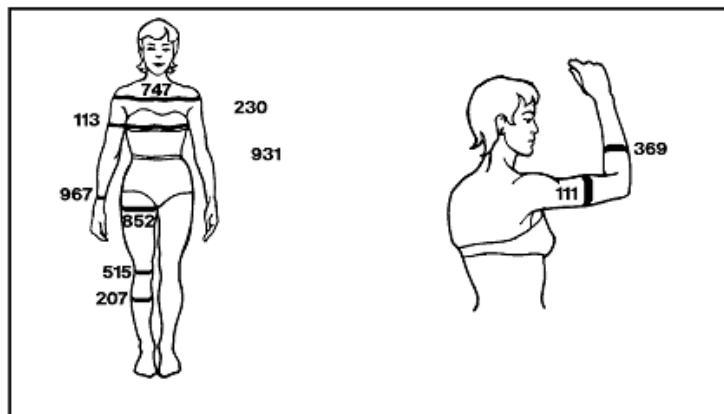


Figura 4.5: Figura esplicativa dei percentili femminili riguardanti varie circonferenze

N	Descrizione	5° Percentile	50° Percentile	95° Percentile
230	Circonferenza Petto	73.2	82.1	90.9
931	Circonferenza vita	55.3	63.2	71.2
852	Circonferenza cosce	45.6	51.6	57.7
515	Circonferenza ginocchio	31.0	34.6	38.2
207	Circonferenza polpaccio	30.3	34.1	37.8
113	Circonferenza bicipite rilassato	21.8	25.5	29.3
967	Circonferenza polso	13.7	15.0	16.2
369	Circonferenza avambraccio rilassato	19.9	22.0	24.1

Tabella 4.5: Percentili femminili standard NASA relativi alla figura 4.5

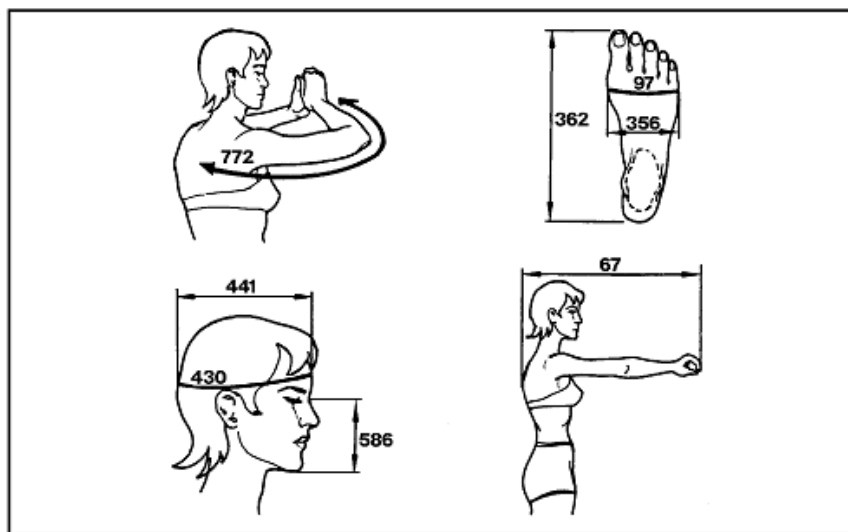


Figura 4.6: Figura esplicativa dei percentili femminili riguardanti ulteriori dimensioni

N	Descrizione	5° Percentile	50° Percentile	95° Percentile
67	Lunghezza spalla-pollice	65.2	71.6	78.0
441	Lunghezza testa	16.7	18.2	19.6
430	Circonferenza testa	53.2	55.2	57.2
586	Lunghezza Menton-sellion	9.0	10.8	12.6
362	Lunghezza piede	21.3	22.9	24.4
356	Ampiezza piede	8.6	9.3	10.0

Tabella 4.6: Percentili femminili standard NASA relativi alla figura 4.6

4.4.3 Percentili maschili

Analoghe misure antropometriche sono riportate sul documento NASA [12] per quanto riguarda i percentili maschili; di seguito vengono riportate, come per i femminili, i valori utilizzati per i percentili maschili.

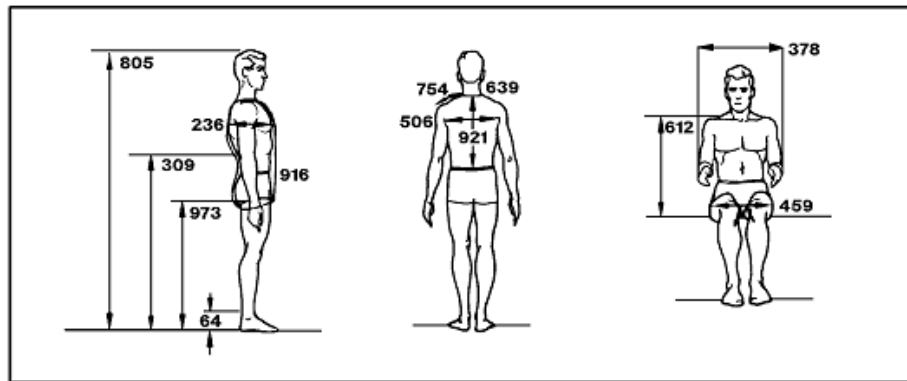


Figura 4.7: Figura esplicativa dei percentili maschili riguardanti altezze e ampiezze

N	Descrizione	5° Percentile	50° Percentile	95° Percentile
805	Altezza	169.7	179.9	190.1
64	Altezza caviglie	12.0	13.9	15.8
169	Profondità busto	21.8	25.0	28.2
916	Circonferenza tronco	158.7	170.7	182.6
612	Altezza spalle in seduta	60.8	65.4	70.0
459	Ampiezza anca in seduta	34.6	38.4	42.3
921	Girovita	43.7	47.6	51.6
639	Circonferenza collo	35.5	38.7	41.9
754	Lunghezza spalle	14.8	16.9	19.0
378	Ampiezza avambraccio-avambraccio	48.8	55.1	61.5

Tabella 4.7: Percentili maschili standard NASA relativi alla figura 4.7

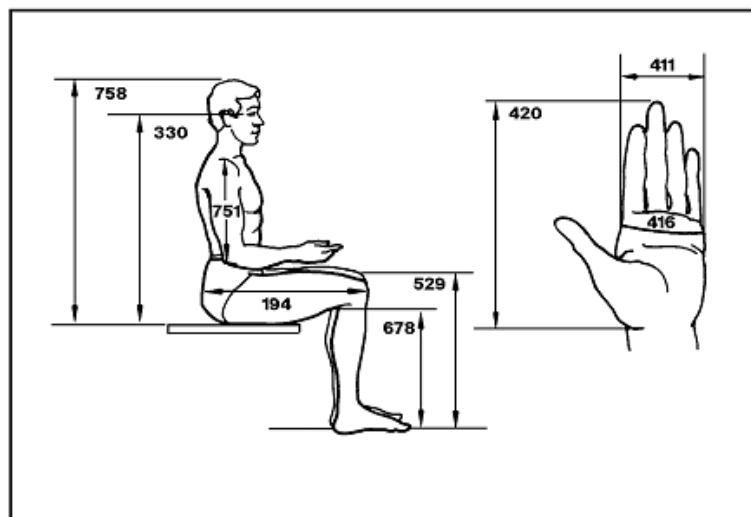


Figura 4.8: Figura esplicativa dei percentili maschili riguardanti mani e seduta

N	Descrizione	5° Percentile	50° Percentile	95° Percentile
420	Lunghezza mano	17.9	19.3	20.6
411	Ampiezza mano	8.2	8.9	9.6
416	Circonferenza mano	20.3	21.8	23.4
785	Altezza seduta	88.9	94.2	99.5
330	Altezza occhi in seduta	76.8	81.9	99.5
529	Altezza ginocchio in seduta	52.6	56.7	60.9
578	Altezza popliteo	40.6	44.4	48.1
751	Lunghezza spalla-gomito	33.7	36.6	39.4
194	Lunghezza natica-ginocchio	56.8	61.3	65.8

Tabella 4.8: Percentili maschili standard NASA relativi alla figura 4.8

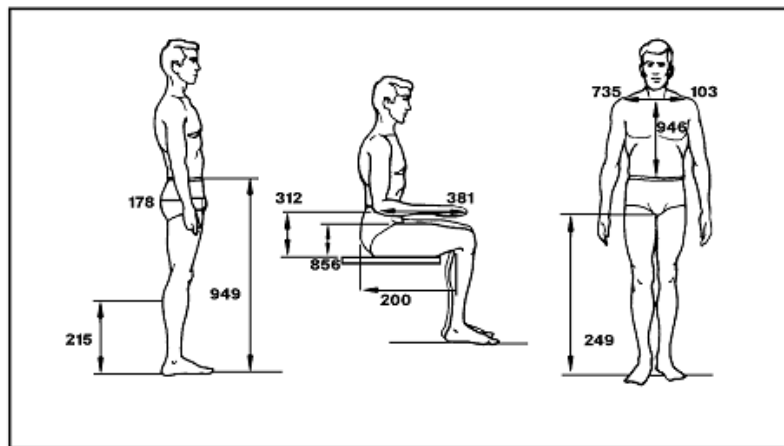


Figura 4.9: Figura esplicativa dei percentili maschili riguardanti ulteriori altezze e ampiezze

N	Descrizione	5° Percentile	50° Percentile	95° Percentile
949	Altezza vita	100.4	108.3	116.2
249	Altezza cavallo	79.4	86.4	93.3
215	Altezza polpaccio	32.5	36.2	40.0
103	Ampiezza biacromiale	37.9	41.1	44.3
178	Circonferenza natiche	91.0	100.2	109.4
312	Altezza gomito a riposo	21.1	25.4	29.7
856	Altezza coscia	14.5	16.8	19.1

Tabella 4.9: Percentili maschili standard NASA relativi alla figura 4.9

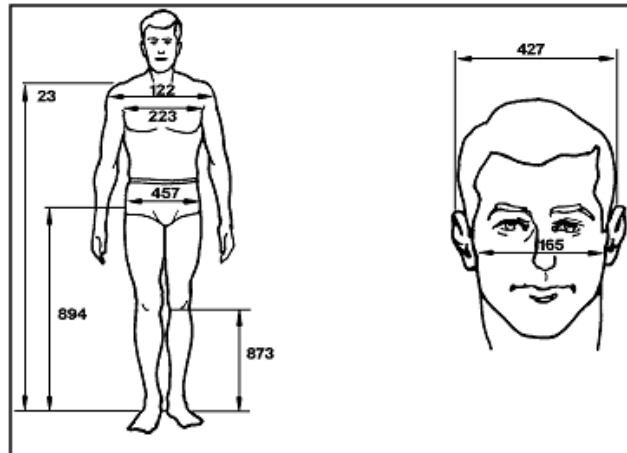


Figura 4.10: Figura esplicativa dei percentili maschili riguardanti le dimensioni del cranio

N	Descrizione	5° Percentile	50° Percentile	95° Percentile
23	Altezza acromiale	138.0	147.6	157.3
894	Altezza bacino	88.3	95.8	102.6
122	Ampiezza deltoidi	44.6	48.9	53.2
233	Ampiezza del petto	29.7	33.2	36.7
457	Ampiezza bacino	32.7	35.8	39.0
165	Ampiezza viso	13.4	14.3	15.1
427	Ampiezza cranio	14.8	15.7	16.5

Tabella 4.10: Percentili maschili standard NASA relativi alla figura 4.10

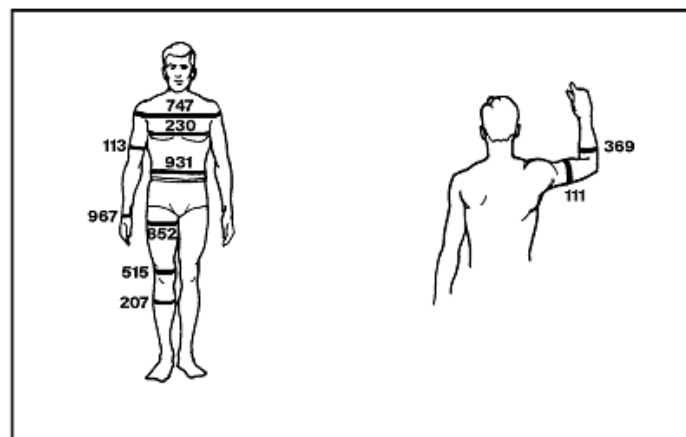


Figura 4.11: Figura esplicativa dei percentili femminili riguardanti varie circonferenze

N	Descrizione	5° Percentile	50° Percentile	95° Percentile
734	Circonferenza spalle	109.5	119.2	128.8
230	Circonferenza Petto	89.4	100.0	110.6
931	Circonferenza vita	77.1	89.5	101.9
852	Circonferenza cosce	52.5	60.0	67.4
515	Circonferenza ginocchio	35.9	39.4	42.9
207	Circonferenza polpaccio	33.9	37.6	41.4
113	Circonferenza bicipite rilassato	27.3	31.2	35.1
967	Circonferenza polso	16.2	17.7	19.3
369	Circonferenza avambraccio rilassato	27.4	30.1	32.7

Tabella 4.11: Percentili maschili standard NASA relativi alla figura 4.11

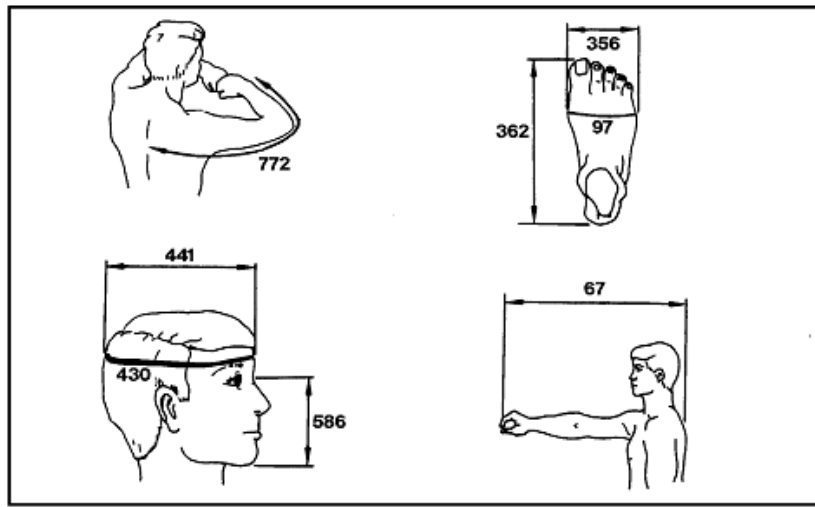


Figura 4.12: Figura esplicativa dei percentili femminili riguardanti ulteriori dimensioni

N	Descrizione	5° Percentile	50° Percentile	95° Percentile
67	Lunghezza spalla-pollice	74.9	81.6	88.2
441	Lunghezza testa	18.8	20.0	21.1
430	Circonferenza testa	55.5	57.8	60.2
586	Lunghezza Menton-sellion	11.1	12.1	13.1
362	Lunghezza piede	25.4	27.3	29.3
356	Ampiezza piede	9.0	9.9	10.7

Tabella 4.12: Percentili maschili standard NASA relativi alla figura 4.12

4.5 Definizione dei manichini

Una volta analizzate le misure riportate, si è passati alla parte di modellazione, seguendo gli step definiti precedentemente; sono state quindi create le prime bozze grazie al programma Make Human, bozze che sono state successivamente importate in Blender per effettuare piccoli aggiustamenti e

verificare le misure. Per fare ciò è stato utilizzato lo strumento righello di Blender, che permette di eseguire misure sia sul piano 3D che su quello 2D, più comodo per gli scopi di questa tesi; le misure così definite sono approssimative, ma approssimano abbastanza fedelmente le lunghezze dei vari segmenti sopra riportate.

Per quanto riguarda invece le circonferenze, sono stati creati vari oggetti di tipo "circle" che sono poi stati sovrapposti alla mesh e scalati in adeguatamente per valutarne la misura; come prima questa è ovviamente approssimata, ma comunque in linea con il documento NASA [12].

Nella tabella sottostante vengono riportate le misure ottenute per i manichini rappresentati il 95° percentile maschile e per il 5° percentile femminile.

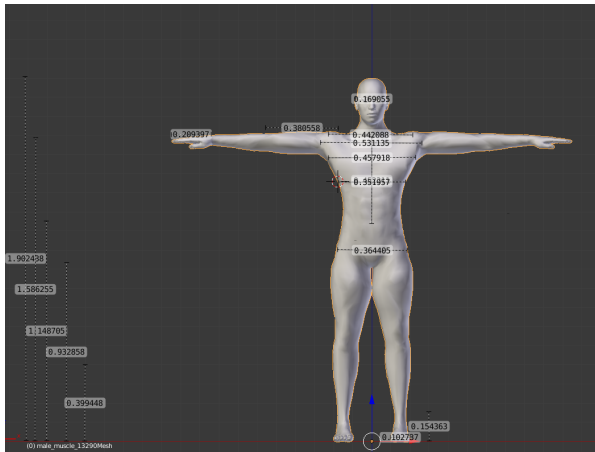
N	Descrizione	95° Percentile	Valore tabulato
805	Altezza	190.0	190.1
64	Altezza caviglie	15.4	15.8
169	Profondità busto	27.1	28.2
751	Lunghezza spalla-gomito	38.0	39.4
420	Lunghezza mano	20.9	20.6
411	Ampiezza mano	9.0	9.6
949	Altezza vita	114.8	116.2
249	Altezza cavallo	93.2	93.3
215	Altezza polpaccio	39.9	40.0
103	Ampiezza biacromiale	44.2	44.3
23	Altezza spalle	158.6	157.3
894	Altezza bacino	102.5	102.6
122	Ampiezza deltoide	53.1	53.2
457	Ampiezza bacino	36.4	39.0
427	Ampiezza cranio	16.6	16.5
178	Circonferenza natiche	108.9	109.4
639	Circonferenza collo	40.1	41.6
230	Circonferenza petto	111.2	110.6
931	Circonferenza vita	102.0	101.9
852	Circonferenza coscia	65.5	67.4
515	Circonferenza ginocchio	42.7	42.9
207	Circonferenza polpaccio	40.2	41.4
113	Circonferenza bicipite rilassato	36.3	35.1
967	Circonferenza polso	20.2	19.3
369	Circonferenza avambraccio rilassato	33.1	32.7
362	Lunghezza piede	27.7	29.3
356	Ampiezza piede	10.2	10.7

Tabella 4.13: Percentili maschili ottenuti

N	Descrizione	5° Percentile	Valore tabulato
805	Altezza	148.4	148.9
64	Altezza caviglie	5.1	5.2
169	Profondità busto	17.7	17.4
751	Lunghezza spalla-gomito	27.0	27.2
420	Lunghezza mano	15.8	15.8
411	Ampiezza mano	6.8	6.9
949	Altezza vita	90.2	90.1
249	Altezza cavallo	67.0	65.2
215	Altezza polpaccio	26.1	25.5
23	Altezza spalle	120.6	119.6
894	Altezza bacino	67.0	71.0
457	Ampiezza bacino	41.3	30.5
427	Ampiezza cranio	14.8	14.4
178	Circonferenza natiche	85.3	79.9
639	Circonferenza collo	36.1.1	34.5
230	Circonferenza petto	86.5	73.2
931	Circonferenza vita	64.2	55.3
852	Circonferenza coscia	60.5	57.7
515	Circonferenza ginocchio	42.7	38.2
207	Circonferenza polpaccio	35.2	30.3
113	Circonferenza bicipite rilassato	23.3	21.8
967	Circonferenza polso	15.2	13.7
369	Circonferenza avambraccio rilassato	24.1	19.9
362	Lunghezza piede	20.2	21.3
356	Ampiezza piede	8.59	8.6

Tabella 4.14: Percentili femminili ottenuti

Nelle figure 4.13a e 4.13b viene mostrato il manichino in fase di misurazione da una parte e completo di texture e materiali dall'altra.



(a) 95° maschile in fase di misurazione



(b) 95° maschile in ambiente Unreal

Figura 4.13: 95° maschile in misurazione e in ambiente Unreal

4.6 Confronto misure NASA con percentili definiti

Analoghe misure sono state prese per gli altri percentili sviluppati, ovvero per il 50° femminile e per il 50° maschile, ottenendo risultati simili a quelli riportati. In generale si può notare che le misure non discostano mai per più di pochi centimetri dal valore tabulato, con alcune eccezioni che riguardano i percentili femminili; per questi ultimi, infatti, la circonferenza di bacino, vita e petto risultano molto piccoli e difficili da rappresentare con precisione; per eliminare problemi dovuti a circonferenze troppo piccole si è deciso, mantenendo comunque le stesse ampiezze, di approssimare le circonferenze più piccole con valori più elevati. Per tutte le altre misurazioni non si riscontrano altre significative variazioni, rendendo i manichini sviluppati molto fedeli ai percentili riportati nei documenti considerati.

4.7 Postura in microgravità

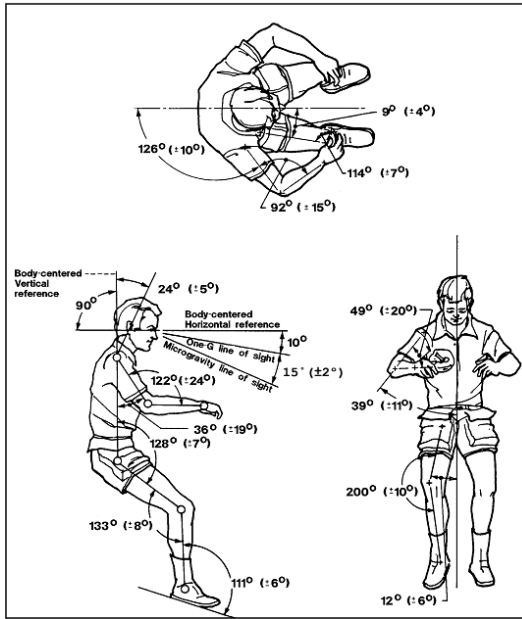
Terminata la fase di modellazione occorre impostare la rest pose dei manichini, ovvero la posa che il personaggio assumerà una volta importato all'interno di Unreal Engine, il motore grafico utilizzato in questa tesi; la rest pose utilizzata è la postura neutra in ambiente di microgravità, anch'essa caratterizzata da parametri ben precisi e tabulati nel documento [12]. Per definire tale pose si è utilizzato la pose mode di Blender, che permette la manipolazione delle ossa associate alla mesh, sia tramite cinematica diretta che indiretta; oltre alla postura di microgravità sono state create tre posizioni per le mani: chiuse in fase di presa di un oggetto, completamente aperte e con l'indice alzato per puntare un oggetto.

Come per i percentili, anche la postura neutra in microgravità è caratterizzata da parametri ben precisi e tabulati all'interno del documento NASA [12]; nello specifico la posa è definita mediante gli angoli che si formano tra le varie parti del corpo:

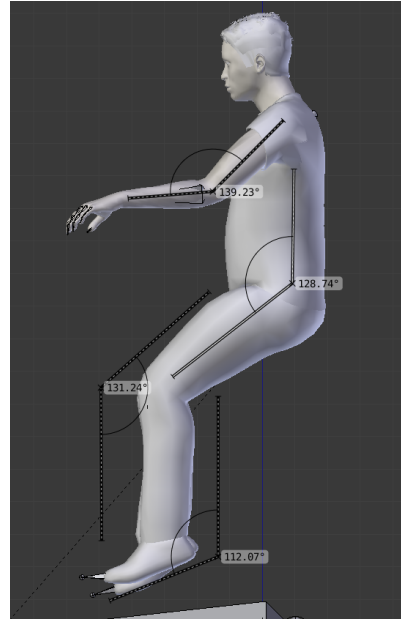
- 122° tra braccio e avambraccio;
- 32° tra braccio e busto;
- 128° tra busto e coscia;
- 133° tra coscia e polpaccio;
- 111° tra polpaccio e piede;

Tutti gli angoli riportati hanno un'incertezza di circa 10°; gli angoli formati dalla posizione realizzata all'interno di Blender, invece, sono stati misurati con il tool Ruler interno a Blender, come per le lunghezze delle altre parti del corpo.

Terminata la fase di modellazione si è effettuato l'export dei manichini prodotti in ambiente Unreal Engine; il manichino viene esportato in formato FBX, formato sviluppato da Kayadara per Autodesk e oggi usato da tutti i tool per lo sviluppo 3D. E' tuttavia un formato ancora in via di sviluppo, disponibile sia in versione ASCII che binaria; al momento Blender, però, dispone della sola versione binaria.



(a) Posizione definita sul documento [12]



(b) Posizione definita dal manichino

Figura 4.14: Posizione di microgravità realizzata

Capitolo 5

Sviluppo della cinematica inversa

5.1 Unreal Engine

Unreal Engine [6], grazie alla sua potenza di calcolo e alla possibilità di essere utilizzato gratuitamente, è oggi uno dei motori grafici di maggior successo; sviluppato da Epic Games, la sua prima versione è stata rilasciata nel 1998 ed è ora alla sua quarta versione. Per la generazione e la compilazione del codice in C++, il motore grafico si appoggia a Microsoft Visual Studio; per lo sviluppo di questo progetto è stata utilizzata la versione 4.21 di Unreal Engine con Microsoft Visual Studio 2017. Il launcher di Unreal Engine, ovvero l'Epic Games launcher, permette di gestire non solo tutti i progetti presenti sul dispositivo e il proprio account, ma anche tutte le versioni disponibili del motore grafico, permettendo all'utente di mantenere i vecchi progetti semplicemente cambiando la versione con cui si intende aprire il progetto, permettendo anche di aprire vecchi progetti con versioni più recenti per verificarne la compatibilità.

5.1.1 Unreal editor

L'estensione utilizzata per salvare un progetto Unreal è .uproject; una volta aperto, si presenta all'utente l'interfaccia di progettazione, definita dall'Unreal Editor (5.1). Un normale progetto di gaming è composto da un insieme di scene interscambiabili e definite per mezzo di livelli, definibili nel Level Editor; quest'ultimo è lo strumento che permette di creare una scena, modificare, inserire, spostare o eliminare un oggetto. In Unreal Engine il livello è sempre un ambiente tridimensionale dove vengono piazzati gli altri oggetti; ogni oggetto è definito attraverso una classe programmabile interna ad Unreal Engine chiamata actor, che permette di controllarne posizione, rotazione e scala e di scrivere codice specifico per quell'oggetto.

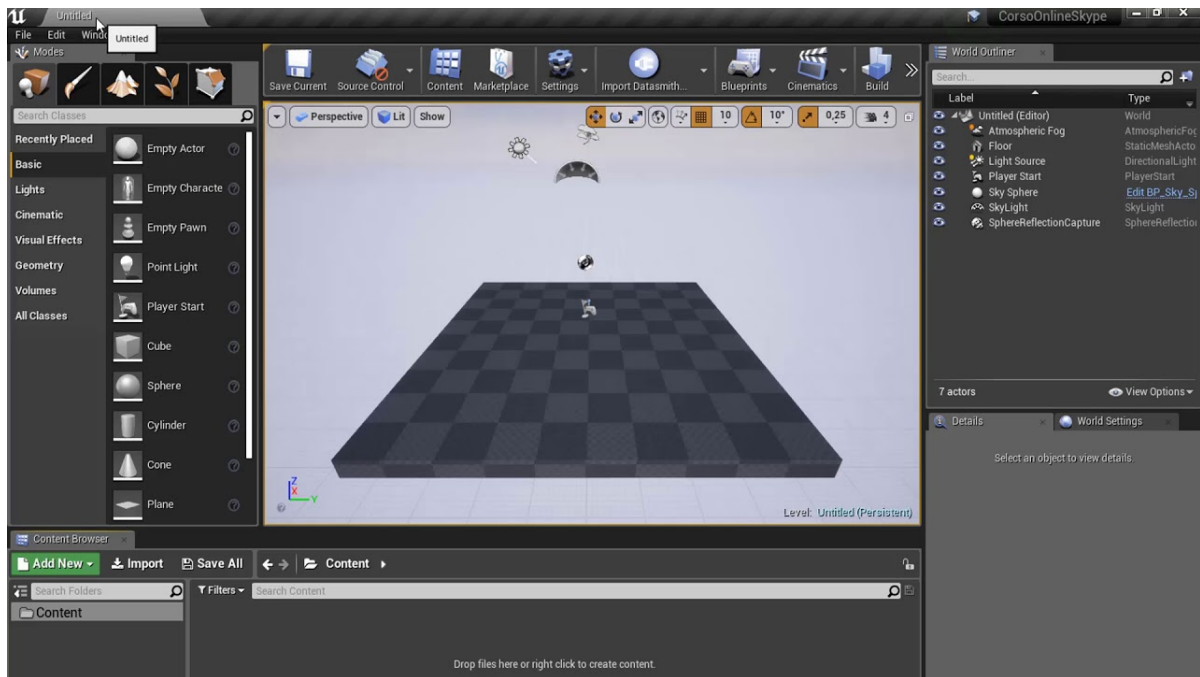


Figura 5.1: Interfaccia utente

L'interfaccia utente è composta da sette elementi principali:

- Menu bar: la barra dei menù in Unreal;
- Toolbar: raggruppa una serie di comandi molto utilizzati per facilitare l'utilizzo del motore grafico; da qui è inoltre possibile selezionare la modalità per lanciare il progetto: oltre alla modalità standard che si limita a compilare ed eseguire il progetto, esistono altre modalità che risultano utili in fase di debug; tra queste, le più usate sono state: selected viewport, che permette di eseguire solo il livello corrente direttamente dall'editor e la modalità simulate, che fa partire il progetto lasciando però la possibilità di avere accesso agli strumenti dell'editor per modificare la scena inserendo nuovi actors o spostando quelli presenti;
- Viewport: occupa la parte centrale dello schermo e permette di visualizzare il livello e manipolare gli oggetti al suo interno;
- Details: contiene tutte le informazioni relative all'oggetto selezionato, le sue coordinate, il materiale a lui associato e varie altre impostazioni;
- World outliner: riporta tutti gli attori presenti sulla scena;
- Modes: permette di aggiungere attori alla scena, di creare particellari e di modificare la geometria degli oggetti stessi;
- Content browser: è il cuore del progetto Unreal, serve per organizzare, creare ed eliminare assets, ovvero i file facenti parte del progetto e definibili sia internamente che esternamente all'ambiente Unreal: come assets interni abbiamo particellari, luci, scripts, blueprints e algoritmi, mentre come esterni troviamo texture, meshes statiche e animazioni.

5.2 Blueprint e C++

Unreal Engine mette a disposizione dell'utente due metodi con i quali definire il codice associato ad un particolare attore, il linguaggio di programmazione C++ e un linguaggio di visual scripting, ovvero il blueprint. E' possibile definire un progetto solo in termini di blueprint o solo in termini di C++ ed è possibile passare dalla versione di programmazione in C++ a quella con blueprint e viceversa.

5.2.1 Blueprint

Questo particolare linguaggio di visual scripting permette di scrivere, mediante la definizione di grafi composti da nodi rappresentanti le funzioni e archi che collegano i vari nodi, codice eseguibile, andando a considerare come flusso d'esecuzione il concetto di flusso del grafo stesso.

Attraverso i blueprint è possibile creare prefabs, ovvero oggetti le cui funzionalità vengono definite nel construction script, un tipo di grafo eseguito non durante il gameplay, ma solo quando l'editor viene aggiornato, creare personaggi, gestire l'input, definire le regole di gioco e costruire HUD(Heads-Up Display).

Esistono tre tipi principali di blueprint: il Class blueprint, il Level blueprint e il Data only blueprint.

Il primo definisce lo script per uno specifico oggetto, viene inserito nel content browser e definisce una nuova classe che può essere poi utilizzata all'interno del progetto; prima di creare una nuova classe è possibile selezionare una Parent Class, in modo da ereditare diverse proprietà utilizzabili dalla nuova classe. Le Parent Classes più utilizzate sono:

- Actor: classe generica che identifica tutti gli oggetti che possono essere dinamicamente piazzati all'interno del mondo di gioco;
- Pawn: particolare Actor controllabile dall'utente;
- Character: particolare Pawn che definisce un oggetto in grado di simulare le caratteristiche di un umanoide;
- Player Controller: particolare classe che si occupa della gestione degli input e delle interazioni e che può essere collegato a un Pawn o a un Character; ogni progetto deve avere un Player Controller correttamente configurato;
- Game Mode: classe fondamentale in ogni progetto che permette di specificare le regole del progetto, come ad esempio il numero di giocatori, il fatto che si possa o meno mettere in pausa il gioco o le condizioni di transizione da un livello ad un altro;

Il Level blueprint definisce invece il flusso che gestisce e controlla uno specifico livello di gioco, mentre il Data Only blueprint è un blueprint che contiene solo codice al suo interno e che svolge specifiche funzioni; è un data Only blueprint, ad esempio, l'Animation graph, che ha il compito di gestire le animazioni legate ad un oggetto.

Gli archi che collegano i vari nodi possono invece essere di due tipi: archi di collegamento tra variabili e archi che definiscono il flusso di esecuzione.

I primi possono avere diversi colori a seconda del tipo di variabile che collegano (rosso per i boolean, turchese per gli interi, verde per i float, giallo per i vettori...) e non influenzano il flusso di esecuzione del programma; i secondi, invece, sono di colore bianco e collegano i nodi stabilendo in quale ordine verranno eseguiti.

Event graph e Animation graph

Event graph e Animation graph sono due particolari tipi di grafo che sono sempre presenti all'interno di un progetto e che rivestono un ruolo particolare.

Un Event graph viene creato per ogni Class blueprint che viene istanziato e utilizza eventi e funzioni per rispondere a ciò che avviene all'interno del mondo di gioco; al suo interno è possibile definire gli eventi che fungono da punti di ingresso, che si collegano poi ad altri nodi e/o funzioni che hanno il compito di gestire la risposta allo specifico evento. Le tipologie di evento cui un Class blueprint può rispondere sono molto numerose e vanno dall'aggiornamento frame per frame, a condizioni attivabili una sola volta in uno specifico punto del gioco, fino ad arrivare a eventi che lanciano eventi in altri Class blueprint. Se prendiamo come esempio una lampadina definita come Class blueprint, all'interno del suo Event graph potremo definire le condizioni per le quali è accesa o spenta e cambiare i suoi parametri di conseguenza, o stabilire delle condizioni per le quali questa non possa più essere accesa e, in questo caso, cambiare la sua mesh con una lampadina danneggiata. Si noti che anche il Level blueprint dispone di un Event graph, che viene solitamente usato per coordinare tra di loro le varie entità presenti all'interno del livello.

L'Animation graph è un particolare grafo presente solo all'interno dell'Animation blueprint e gestisce le animazioni dei personaggi e ne definisce il movimento all'interno del gioco; per ogni oggetto si possono definire diversi Animation graphs, ognuno dei quali può definire diversi comportamenti, che però non possono essere combinati tra loro per generare un unico grafo.

Al contrario dell'Event graph, ogni Animation graph ha una serie di nodi posizionati al suo interno che servono per campionare le sequenze di animazioni, combinare le animazioni e controllare posizione, rotazione e scalamento delle ossa del personaggio. Ogni nodo dell'Animation graph definisce una posa e al termine dell'esecuzione lo scheletro assume la posa risultante; ogni Animation graph ha un proprio Event graph per decidere quale animazione si deve lanciare in un preciso momento. A differenza di un normale Event graph, però, l'Animation graph ha un flusso di esecuzione differente, che non coincide con il verificarsi di particolari eventi, ma viene invece eseguito frame per frame, valutando i nodi che compongono il grafo.

All'interno dell'Animation graph è possibile definire diversi tipi di operazione, le più comuni delle quali sono:

- Animation blending: permette di combinare più pose per ottenerne una finale, servendosi di vari tipi di nodi di blending per ottenere effetti differenti;
- Nodi di animazione: utilizzati per agire direttamente sulle pose di input manipolando direttamente le ossa dello scheletro;
- Blend Space: strumento per fondere più pose a seconda del valore assunto da determinate variabili;
- Controlli scheletrici: fanno parte di queste operazioni i nodi per la cinematica inversa e servono per guidare direttamente una catena di ossa di una determinata lunghezza;
- Macchine di stato: fondamentali per gestire un personaggio che presenta molteplici animazioni; sempre servendosi di un grafo, si vanno a stabilire tutte le condizioni per le quali debba essere eseguita una determinata animazione o per le quali si debba invece cambiare animazione, stabilendo anche con che modalità passare da un'animazione all'altra, ovvero scegliendo una certa interpolazione oppure facendo un cambio netto;
- Sync groups: utilizzati per mantenere correlate e sincronizzate le diverse animazioni di un personaggio.

5.2.2 C++ e plugin

Tutte le volte in cui si devono definire comportamenti più complessi e specifici, che risulterebbero difficili da sviluppare tramite blueprints, è possibile scrivere quella parte di codice in linguaggio C++, che può poi essere convertito in blueprint e aggiunto al progetto; i plugin esterni per Unreal Engine vengono solitamente scritti utilizzando questa tecnica e sono poi utilizzati dall'utente come normali blueprints.

I plugin realizzati per Unreal Engine possono essere catalogati a seconda di alcune loro caratteristiche; nello specifico possono essere divisi in tre macro categorie: non game module, ovvero plugin che generano un'interfaccia utilizzabile dall'utente; stand alone plugin, ovvero plugin il cui codice non può essere modificato da moduli esterni; content module, ovvero plugin che generano nuovi elementi, utilizzabili sia nell'editor di sviluppo che in game.

Tutti i plugin possono, inoltre, essere installati esclusivamente per lo scopo che si vuole raggiungere, essere già presenti internamente all'editor di Unreal Engine o essere definibili all'interno del progetto. In tutti questi casi, la scrittura del codice C++ e la sua conversione in blueprint è complicata e necessita l'esecuzione passo passo di una procedura riportata nella guida di Unreal Engine, in modo da scrivere codice che possa poi essere facilmente interpretato e successivamente convertito dal motore grafico.

5.3 Struttura del progetto

La prima parte del progetto di tesi ha lo scopo di rendere interattivo l'editor di Unreal Engine, dando all'utente la possibilità di selezionare il manichino interessato e di poter svolgere diverse operazioni, quali la movimentazione degli arti tramite cinematica inversa, la selezione di una posa per le mani, la selezione di un target e la successiva rotazione della testa nella direzione del target stesso, il salvataggio di una serie di pose successivamente replicabili e l'aggiunta di una camera che mostri in basso a destra la visuale del personaggio e il cui FOV sia regolabile. Nella 5.2 si può vedere il risultato finale e il menù con tutte le opzioni selezionabili, con l'aggiunta della possibilità di attivare e disattivare la manipolazione tramite cinematica inversa e il comportamento in microgravità, in modo da rendere più versatile il plugin, che può quindi essere usato anche al di fuori di un ambiente in 0g.

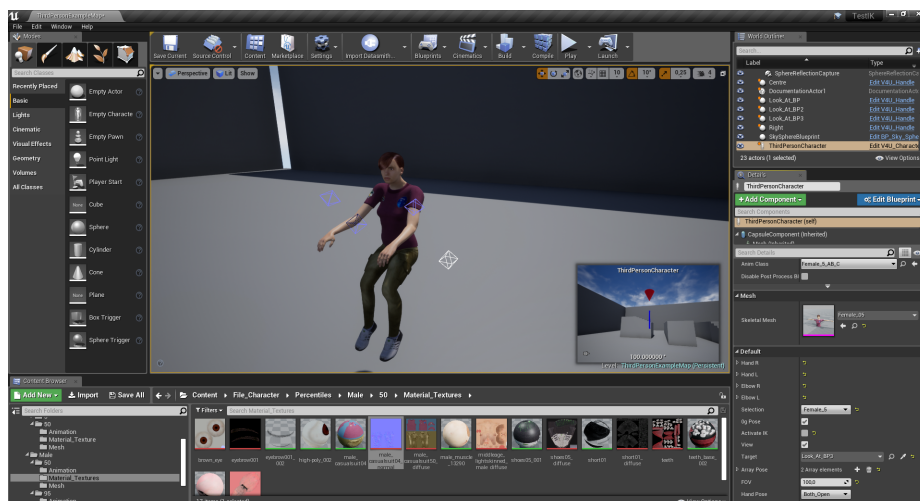


Figura 5.2: Interfaccia ottenuta per la prima parte del progetto

5.4 Analisi del menù creato

Per selezionare un personaggio giocabile all'interno di Unreal Engine bisogna settare diversi parametri legati al pawn del giocatore, quali lo scheletro da utilizzare, la mesh da visualizzare, il suo materiale con relative texture e l'Animation blueprint dello scheletro. Per poter fare ciò direttamente dall'editor è stata creata una variabile di tipo enum, ovvero una variabile che dà accesso ad una lista di oggetti definiti dall'utente e su cui si può eseguire uno switch che viene gestito nel construction script del pawn. In questo modo l'utente può scegliere il percentile desiderato dal menù a destra mentre, internamente, lo switch deciderà quali mesh, scheletro, materiale e animation blueprint caricare per il giocatore. Simile procedimento viene fatto per la modifica della posizione delle mani: dal menù l'utente seleziona la posa desiderata, che internamente viene vista come una variabile enum su cui eseguire uno switch che viene gestito internamente; lo switch interagisce quindi con l'animation blueprint e l'animation graph corrispondente per eseguire il blending della posizione in cui si trova il manichino con la posizione della mano selezionata.

Le opzioni 0g pose, activate IK e view sono invece valori booleani che l'utente può settare tramite la spunta per attivare o disattivare determinate caratteristiche: la prima attiva o disattiva il comportamento in 0g, passando da una normale posa in piedi alla posa di microgravità; la seconda, invece, attiva o disattiva la possibilità di sfruttare la cinematica inversa per stabilire la posa del personaggio, attivando o disattivando il nodo FABRIK (Forward And Backward Inverse Kinematics) - di cui si parlerà nelle prossime sezioni - nell'animation graph; l'attivazione e la disattivazione sono state sviluppate andando ad agire sul canale alpha del nodo di blending: attivando la cinematica inversa, l'alpha viene settato ad 1, facendo in modo che la posa sia determinata solo dal nodo FABRIK; viceversa, disattivandola, l'alpha è settato a 0 e la posa viene determinata solo dalla posa di partenza, con cui è stato importato il modello. L'ultima opzione, infine, attiva o disattiva la camera che rappresenta la visuale del giocatore, andando a disattivare l'oggetto camera associato alla visuale; per regolarne il FOV è sufficiente agire sullo slider FOV, che agisce direttamente sul parametro della camera.

Il menù a tendina target, invece, include tutti i vari target che possono essere selezionati e verso cui il manichino può rivolgere lo sguardo; a titolo di esempio, come target, è stato utilizzato il modello della maniglia realmente presente sul modulo Columbus. Una volta selezionato il target desiderato, se ne prendono le coordinate spaziali e, nell'animation blueprint, si va a stabilire la rotazione che deve eseguire la testa del manichino per essere rivolta verso il target grazie al nodo Look at rotation.

I primi quattro parametri del menù, infine, regolano posizione e rotazione delle mani e dei gomiti per realizzare la cinematica inversa, di cui si parlerà più ampiamente nelle prossime sezioni; tali parametri possono essere modificati inserendo il valore numerico oppure agendo sugli handle a schermo, per un controllo più preciso; una volta realizzata la posa desiderata è possibile, attraverso il tasto "+" nella sezione array pose, salvare la posa all'interno dell'array di pose, che sarà possibile scorrere una volta lanciata l'applicazione attraverso il tasto "spacebar".

5.5 Definizione della cinematica inversa

Definita una catena di ossa formata da un osso radice "root", una serie di ossa intermedie "joint" e un osso finale "end effector", il problema della cinematica inversa consiste nel definire la corretta posizione e rotazione delle ossa intermedie partendo dalla posizione e dalla rotazione dell'end effector, differenziandosi dal problema della cinematica diretta, nel quale si definisce la posizione dell'end effector andando a definire le coordinate di tutte le ossa intermedie.

considerando la figura 5.3 in cui l'end effector è la sfera più a destra si può vedere la differenza tra i due problemi: nella figura di sinistra, rappresentate la cinematica diretta, si vede come il

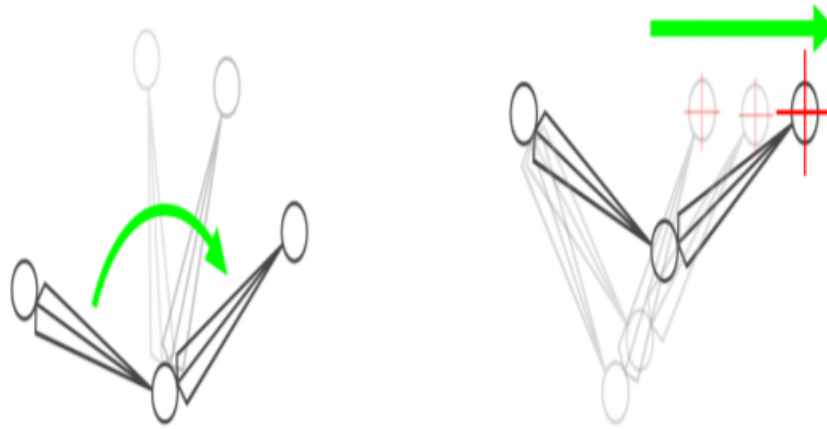


Figura 5.3: Esempio di cinematica diretta e cinematica inversa

movimento dell'end effector produca il movimento del solo osso interessato mentre, nella cinematica inversa rappresentata nella figura a destra, il movimento dell'end effector comporti il movimento di tutta la catena cinematica, a partire dall'osso "root". Una delle difficoltà principali nella risoluzione della cinematica inversa consiste nel selezionare gli angoli giusti per ottenere la posa desiderata, dal momento che possono esistere più soluzioni analitiche del problema; per risolvere questo problema bisogna porre dei vincoli fisici alle articolazioni che formano la catena cinematica, vincoli che spesso si traducono in angoli limite che possono essere assunti; considerando per esempio il braccio umano si devono escludere tutte le soluzioni che comprendono angoli maggiori di 180° , dal momento che la nostra articolazione impedisce di andare oltre quel limite.

Oggi giorno, sono molteplici i campi in cui viene utilizzata la cinematica inversa e comprendono, oltre ovviamente al campo videoludico, quello dell'industria, della robotica, del design e di tutti quei campi che, per un qualunque motivo, hanno necessità di simulare il comportamento di una catena cinematica di qualunque tipo. Per risolvere questo problema sono stati creati numerosissimi algoritmi, molti dei quali, tuttavia, hanno costi computazionali troppo elevati o falliscono nel riprodurre una posa complessa in modo naturale; in Unreal Engine è presente, proprio per risolvere questi problemi, il nodo FABRIK, che fornisce un algoritmo di risoluzione della cinematica inversa molto semplice ed economico dal punto di vista del costo computazionale, che permette di risolvere la cinematica inversa in real-time con movimenti fluidi e naturali.

5.5.1 Basi matematiche della cinematica inversa

Le equazioni cinematiche su cui si basa la cinematica inversa vengono utilizzate per definire i vincoli non lineari che intervengono sul sistema; all'interno di queste equazioni, i parametri indipendenti prendono il nome di gradi di libertà del sistema. Per rappresentare e risolvere il problema della cinematica inversa esistono numerosi metodi, tra i quali il più comune si basa sulla ricerca iterativa della miglior soluzione approssimata, a causa della difficoltà di invertire l'equazione della cinematica diretta e vista la possibilità di ottenere uno spazio di soluzioni vuoto. L'idea alla base di molti di questi metodi è quella di rappresentare l'equazione della cinematica diretta utilizzando un'espansione in serie di Taylor, che risulta semplice da invertire e da risolvere.

Tecnica della Jacobiana inversa

La tecnica qui analizzata rappresenta un modo semplice ed efficace per risolvere il problema della cinematica inversa. Si considerano m variabili che governano l'equazione della cinematica diretta, che possono rappresentare angoli tra le giunture, lunghezze o altri valori reali. Se il sistema di cui si vuole calcolare la cinematica inversa si trova in uno spazio 3D, la posizione può essere vista come: $p(x) : R^m \rightarrow R^3$. Si prenda $p_0 = p(x_0)$ come punto iniziale del sistema e $p_1 = p(x_0 + \Delta x)$ come punto di arrivo del sistema.

Il metodo della Jacobiana inversa consiste nel calcolare iterativamente una stima di Δx che minimizzi l'errore, calcolato come $\|p(x_0 + \Delta x_{\text{stimato}}) - p_1\|$. Per piccoli vettori Δx , l'espansione in serie fornisce:

$$p_1 \approx p(x_0) + J_p(x_0)\Delta x$$

dove $J_p(x_0)$ rappresenta la matrice Jacobiana nella posizione x_0 . Prendendo la pseudoinversa della Jacobiana e riposizionando i membri dell'equazione si arriva a definire

$$\Delta x \approx J_p^+(x_0) \Delta p$$

Dove $\Delta p = p(x_0 + \Delta x) - p(x_0)$.

Applicando il metodo una sola volta produce un valore fortemente approssimato di del vettore Δx , motivo per il quale è necessario applicare iterativamente il metodo sopra descritto fino ad ottenere un risultato soddisfacente.

5.5.2 Algoritmo di cinematica inversa: FABRIK

L'efficienza di tale algoritmo e il fatto che converga in poche interazione è principalmente dovuto al fatto che, a differenza di tutti gli altri algoritmi per la risoluzione della cinematica inversa, non vengano utilizzate le matrici per ottenere la posizione di un joint, ma si utilizzi invece la localizzazione di un punto su una linea; per poter meglio comprendere il funzionamento di questo algoritmo si farà riferimento alla figura 5.4 e allo pseudo-codice del nodo FABRIK sottostante [2]:

```

1 Input: The joint positions pi
2   for i = 1, . . . , n, the target position t and the distances between each joint
   di = |pi+1 - pi|
3 |   for i=1, . . . , n-1
4
5 Output: The new joint positions pi for i = 1, . . . , n.
6 % The distance between root and target
7   dist = |p1 - t|
8 % Checkt whether the target is within reach
9   if dist > d1 + d2 + . . . + dn-1 then
10 % The target is unreachable
11   for i = 1, . . . , n-1 do
12     % Find the distance ri between the target t and the joint position pi
13     ri = |t - pi|
14     i = di/ri
15     %Find the new joint positions pi
16     pi+1 = (1 - i)pi + it
17   end
18 else
19   %The target is reachable; thus, set as b the initial position of the joint
20   p1 = p1
21   b = p1
22   %Check whether the distance between the effector pn and the target t is
23   greater than a tolerance
24   dif A = |pn - t|
25   while dif A > tol do
26     %STAGE 1: FORWARD REACHING
27     %Set the end effector pn as target t
28     pn = t

```

```

29     for i= n 1 ,...,1 do
30         %Find the distance ri between the new joint position pi+1 and the
31         joint pi
32         ri = |pi+1      pi
33         i = di/ri
34         %Find the new joint positions pi
35         pi = (1      i )pi+1 + pi
36     end
37     %STAGE 2: BACKWARD REACHING
38     %Set the root p1 its initial position
39     p1 = b
40     for i=1,..., n 1 do
41         %Find the distance ri between the new joint position pi and the
42         joint pi
43         ri = |pi+1      pi
44         i = di/ri
45         %Find the new joint positions pi
46         pi+1 = (1      i )pi + ipi +1
47     end
48     dif A = |pn      t|
49 end
50 end

```

L'algoritmo si basa su due cicli, un ciclo forward e un ciclo backward; inoltre, invece di utilizzare gli angoli di rotazione, va a individuare un punto su una linea che corrisponda alla locazione del joint; i joint vengono indicati con p_1, \dots, p_n , dove p_1 è il root, mentre p_n è l'end effector; infine il target da raggiungere è indicato con la lettera t .

Come prima cosa, l'algoritmo calcola le distanze con $i=1, \dots, n-1$, andando poi a verificare se un target sia raggiungibile o meno; per eseguire questa verifica si prende la distanza tra target e root, si controlla se $dist < d_1 + \dots + d_n$ e, se il target è raggiungibile, ha inizio la fase di forward: dell'end effector p_n al root p_1 , si considera la nuova posizione dell'effettore $p'_n = t$ e si crea la linea l_{n-1} , che passa tra p_{n-1} e p'_n ; questo procedimento viene effettuato dall'end effector al root incluso. Dal momento che viene considerato anche il movimento della radice, si tiene conto del fatto che p'_1 , calcolato nella fase di forward, non dovrebbe aver assunto una nuova posizione rispetto a quella iniziale, dal momento che il root deve sempre mantenere tale posizione.

Terminata la fase di forward, ha inizio quella di backward, nella quale si ripete il procedimento, andando dal primo nodo verso l'ultimo e considerando come posizione iniziale p''_1 . Al termine di questa fase, il nodo finale sarà in una posizione differente da quella desiderata e si può quindi ripetere l'algoritmo fino al raggiungimento della posa desiderata; si può quindi dire che, dal momento che l'algoritmo si esaurisce in un numero finito di iterazioni e che verifica che una posa sia raggiungibile o meno, l'algoritmo FABRIK converge.

L'algoritmo, inoltre, è stato fortemente adottato negli ultimi anni da molti utenti dal momento che risolve alcuni problemi, che si creavano con gli algoritmi usati in precedenza; l'algoritmo, infatti, riesce a convergere anche in presenza di *leaf point* (ovvero catene in cui l'end effector non è l'ultimo nodo della catena), di end effector multipli e di catene cinematiche chiuse.

5.5.3 Impostazione nodi FABRIK e animazione del personaggio

Il movimento effettivo del manichino viene calcolato all'interno del file di animazione, denominato Animation blueprint, all'interno del quale vengono impostati e utilizzati i noti FABRIK; per utilizzare correttamente un nodo FABRIK bisogna, dopo aver impostato la precisione con cui si vuole utilizzare il nodo- precisione che determinerà il numero di iterazioni compiute dall'algoritmo e il costo che avrà il calcolo della cinematica inversa- impostare le catene cinematiche che bisogna movimentare.

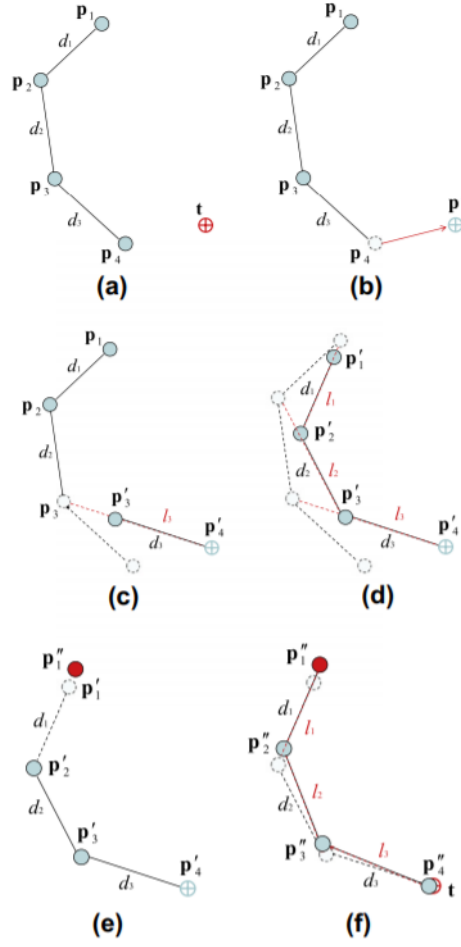


Figura 5.4: Schema di funzionamento dell'algoritmo FABRIK

Impostazione catene cinematiche

L'impostazione delle catene cinematiche è un'operazione fondamentale per utilizzare un qualsiasi algoritmo per la risoluzione della cinematica inversa e consiste nell'identificare root, end effector e joints della catena interessata. Nel caso del progetto in esame sono state definite in totale otto catene, due per ogni braccia: prendendo in considerazione una delle braccia abbiamo una catena che ha come end effector la mano e come root la spalla ed è quindi composta da due ossa; un'altra che è invece composta da un solo osso e ha come end effector il gomito e come root la spalla.

L'utilizzo di una doppia catena permette di ottenere pose molto più naturali rispetto alla catena singola, aggiungendo però la necessità di risolvere due volte il problema della cinematica inversa; in questo modo, utilizzando la prima catena è possibile spostare e ruotare la mano nella posizione desiderata per poi utilizzare la seconda catena per regolare precisamente l'angolo formato dal gomito.

Con lo stesso principio sono stati aggiunti i punti di controllo per la mobilitazione degli arti inferiori, aggiungendo, per ognuno di essi, un punto di controllo sulla caviglia, per modificare la posizione generale della gamba e la rotazione del piede e uno sul ginocchio, per ritoccare la posa finale e per

meglio definire gli angoli formati dall'articolazione.

Event graph

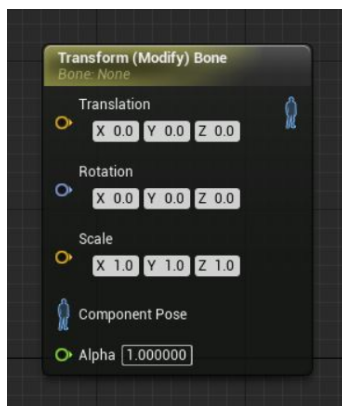
Per fare in modo che il manichino sia modificabile direttamente dall'editor bisogna agire sull'Event graph, e più nello specifico sul Construction script, in modo che la modifica di determinati parametri influenzi l'Animation blueprint e cambi la posa del manichino. Per fare ciò vengono realizzati i wrapper per il passaggio di valori attraverso il blueprint, in particolare dall'Event graph all'Animation graph e che vengono inizializzati con i parametri imposti dalla catena cinematica.

Per il movimento delle braccia del manichino sono state create quattro variabili (*Hand R*, *Hand L*, *Elbow L* e *Elbow R*), che vengono visualizzate direttamente sull'editor grazie all'opzione "show 3D widget", che visualizza, per le variabili di tipo vettore, un ottaedro in corrispondenza del valore della variabile; le prime due variabili servono per movimentare la catena cinematica che ha come end effector la mano, mentre le ultime due movimentano la seconda catena cinematica. Una volta modificato il valore della variabile, inserendo manualmente il valore all'interno del menù creato o muovendo il 3D widget, il wrapper corrispondente provvede a mandare il valore all'Animation blueprint, che utilizza il valore ricevuto per risolvere la cinematica inversa, produrre la nuova posa e muovere il manichino in modo che assuma la posa risultante. Disattivando la cinematica inversa attraverso l'opzione descritta in precedenza è possibile disattivare l'aggiornamento dell'Animation blueprint da parte dei wrapper e disattivare l'opzione per la visualizzazione dei 3D widget.

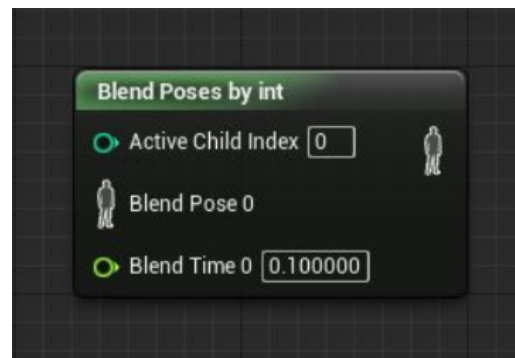
Animation graph

All'interno dell'Animation graph viene costruito il flusso di esecuzione che porta alla modifica della posa del manichino; come posa base il manichino assume la posa in microgravità se l'opzione 0g pose è attiva, una normale posa in piedi viceversa; la manipolazione della posa viene ottenuta grazie all'aggiunta di particolari nodi che possono modificare direttamente la posizione delle ossa del manichino o che possono effettuare operazioni di blending sulle pose passate come input.

I nodi maggiormente usati, oltre al nodo FABRIK già descritto in precedenza, sono stati: Transform (modify) bone, Layered blend per bone e Blend poses by int. Il primo nodo riceve come input una posa e permette di selezionare un osso dello scheletro di cui modificare posizione, rotazione o scalamento; nello specifico il nodo è stato utilizzato per modificare la rotazione dell'osso head in relazione alla posizione del target selezionato. Gli altri due nodi sono nodi di blending: il primo esegue il blend di due pose limitando gli effetti del blending ad un sottoinsieme di ossa ed è stato utilizzato per gestire la posizione della mano, andando ad eseguire il blend tra la posa del manichino e la posa della mano desiderata, limitandone gli effetti alle sole ossa della mano. Il secondo, viceversa esegue il blend di n pose, basandosi sul valore assunto da una variabile di tipo int. Il valore della variabile determina la posa da utilizzare e, cambiando il valore, si esegue un blend tra la posa di partenza e quella selezionata dal nuovo valore della variabile; il blending avviene in un intervallo di tempo che può essere specificato per ogni posa.



(a) Transform (modify) bone node



(b) Blend pose by int node

Figura 5.5: Esempio di nodi utilizzati

Capitolo 6

Test d'uso sui manichini

6.1 Manichini in TASI

Per svolgere la fase di test della prima parte del progetto descritta in questo capitolo sono stati utilizzati i manichini sviluppati in precedenza e il modello del modulo Columbus. Il test si divide in tre punti:

- Confronto tra le posture definite dai manichini e le posture realistiche definite dagli astronauti, allo scopo di verificare il realismo delle pose e valutare le differenze tra le pose di un essere umano e quelle del manichino;
- Test ergonomici sull'hardware presente a bordo del modulo Columbus;
- Test di rappresentazione di una squadra di manichini all'interno del modulo.

Per tutti i test, i parametri presi in considerazione sono stati:

- Accuratezza della posa;
- Flessibilità della posa;
- Riconfigurabilità della posa;
- Usabilità della posa,

Tutti i test sono stati svolti operando esclusivamente attraverso l'editor del motore grafico, cosa che rappresenta un grosso vantaggio rispetto al progetto precedente, che prevedeva il passaggio attraverso la modalità "simulate" e che quindi richiedeva di eseguire il progetto in una modalità specifica; così facendo, invece, è possibile configurare le pose in fase di progettazione, senza bisogno di eseguire il progetto e operando comodamente attraverso il menù creato. Per alcuni test è stato inoltre utilizzato il punto di vista del manichino, per poter meglio comprendere gli spazi occupati da un futuro operatore e migliorare l'ambiente per renderlo più comodo e di facile utilizzo.

I parametri valutati per i test ergonomici riguardanti l'hardware di bordo sono stati invece:

- Visibilità;
- Accessibilità;
- Operatività.

Tutte le misure sono di tipo qualitativo e basate su una valutazione visiva di quanto ottenuto, considerando i criteri ergonomici definiti in precedenza.

6.1.1 Replica delle posture realistiche degli astronauti con i manichini definiti nella tesi

In questa prima fase di test, utilizzando fotografie che mostrano alcune posizioni assunte dagli astronauti all'interno del modulo, sono stati posizionati i manichini in modo da riprodurre tali posizioni per verificare la qualità con la quale vengono riprodotte le posture umane.

L'immagine ?? mostra due membri dell'equipaggio ISS, Chris Cassid e Luca Parmitano mentre eseguono un test spinale ad ultrasuoni, test facente parte di una missione di ricerca per valutare il potenziale danno al disco intervertebrale causato dalla prolungata permanenza in ambiente di microgravità, dalla forza di lancio o atterraggio o da altre cause quali radiazioni, alimentazione, metabolismo e così via. La ricerca permise di ottenere dati in tempo reale per valutare la progressione delle alterazioni del disco intervertebrale durante il volo spaziale, assistere nell'individuazione delle cause di danno del disco e valutare il sistema muscolo-scheletrico dell'equipaggio e le lesioni che esso può subire durante il volo spaziale. Per rappresentare la scena si è usato il 50° percentile e, come si vede in figura la posa risulta molto naturale e fedele rispetto all'originale; anche le gambe, che risultavano essere il punto debole del progetto precedente, ora riproducono la posa fedelmente grazie ai due punti di controllo, che permettono di impostare un range maggiore di pose.



Figura 6.1: Replica della foto rappresentante Chris Cassid e Luca Parmitano

Nella fotografia ?? viene ritratto l'astronauta Dan Burbank mentre prende parte all'esperimento PASSAGES all'interno del laboratorio Columbus, utilizzando l'hardware Neurospat; tale esperimento vuole valutare come l'assenza di peso e la permanenza in ambiente in microgravità influiscano e modifichino il campo visivo degli astronauti. Utilizzando sempre il 50° percentile è possibile replicare la posa desiderata e modificare il campo visivo come si ritiene necessario, ottenendo sempre un risultato estremamente fedele.

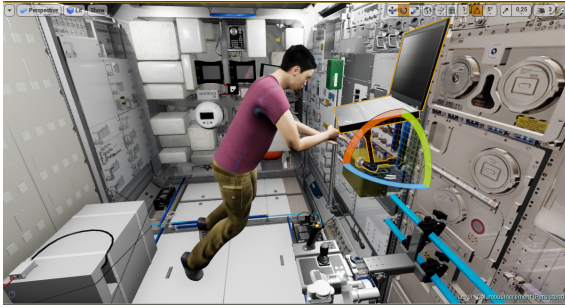


Figura 6.2: Replica della foto rappresentante Dan Burbank

Nell'immagine ??, invece, si vede l'astronauta Peggy Whitson all'interno del modulo mentre fluttua attraverso un groviglio di cavi; in questo caso si è utilizzato il 5° percentile femminile per riprodurre la posa e, nonostante la posa difficile e non standard dell'astronauta, anche in questo caso, il manichino riesce a riprodurre la posa senza troppe difficoltà, dando come risultato una posa realistica e fedele.

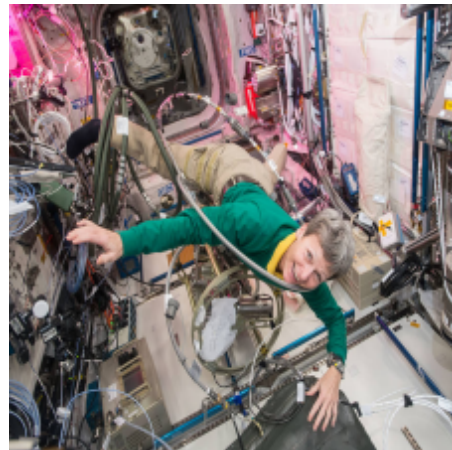


Figura 6.3: Replica della foto rappresentante Peggy Whitson

In conclusione si nota che i manichini mantengono una buona flessibilità e accuratezza, anche in presenza di pose complesse assunte realmente dall'equipaggio del modulo durante varie operazioni ed esperimenti e possono quindi fornire uno strumento per lo studio ergonomico non solo degli oggetti che sono presenti o saranno presenti nel modulo Columbus, ma anche di nuovi e futuri moduli spaziali.

6.1.2 Verifica dell'ergonomia del hardware a bordo di Columbus

I test riguardanti l'hardware di bordo prendono in considerazione la visibilità e l'accessibilità, ma non possono verificare l'operabilità, dal momento che nessuno dei modelli dei macchinari presenti

dispone di un algoritmo che ne simuli il comportamento. Per il primo test si è utilizzato il 5° percentile femminile per verificare l'accessibilità e, tramite il punto di vista del manichino, la visibilità di un computer di bordo (figura 6.4a); tramite il braccio meccanico è possibile spostare e ruotare il computer di bordo per trovare le migliori posizioni per ogni percentile e garantire che chiunque possa utilizzarlo mantenendo una buona visibilità e accessibilità.

Successivamente è stata verificata la presa del manichino facendo afferrare al 95° percentile prima un dosimetro PDP per la misura di radiazioni presenti nel modulo e poi le handrail presenti all'interno del modulo, usate dagli astronauti per fissarsi in una determinata posizione e per muoversi da un punto all'altro (figura 6.4b).

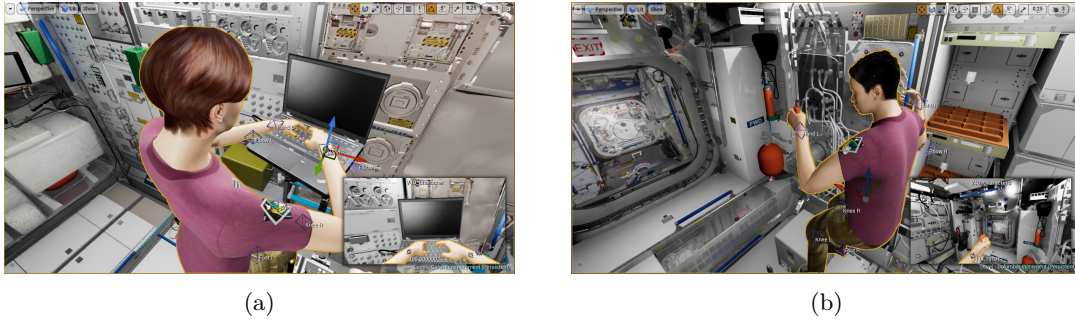


Figura 6.4: Test di prensilità e accessibilità e usabilità computer di bordo

Nonostante il manichino simuli bene la posizione di presa, si possono riscontrare problemi a causa delle dimensioni degli oggetti: oggetti più grandi richiedono infatti una presa più aperta, mentre i manichini possono assumere solo una presa molto stretta, che può causare compenetrazioni con oggetti spessi. Tale problema si può risolvere o aggiungendo dei punti di controllo sulle dita del manichino, rischiando però di rendere complessa l'interfaccia, o creando un algoritmo che regoli l'apertura delle dita andando a verificare la collisione tra l'oggetto e la mesh del manichino.

Per verificare inoltre l'accessibilità del computer di bordo e il volume occupato dal manichino è stato messo in evidenza il work envelope definito nei documenti di progettazione del modulo Columbus attraverso un cubo con un materiale semi-trasparente (figura 6.5); in questo modo è possibile visualizzare il volume occupato da un astronauta mentre opera al computer. Attraverso test analoghi e sviluppando il work envelope per tutte le strumentazioni presenti a bordo è possibile accertare l'accessibilità di una determinata zona. In figura (immagine) si vede come il 95° percentile maschile riesca a rientrare all'interno del volume previsto per l'utilizzo del computer nella posizione stabilita dal braccio meccanico.

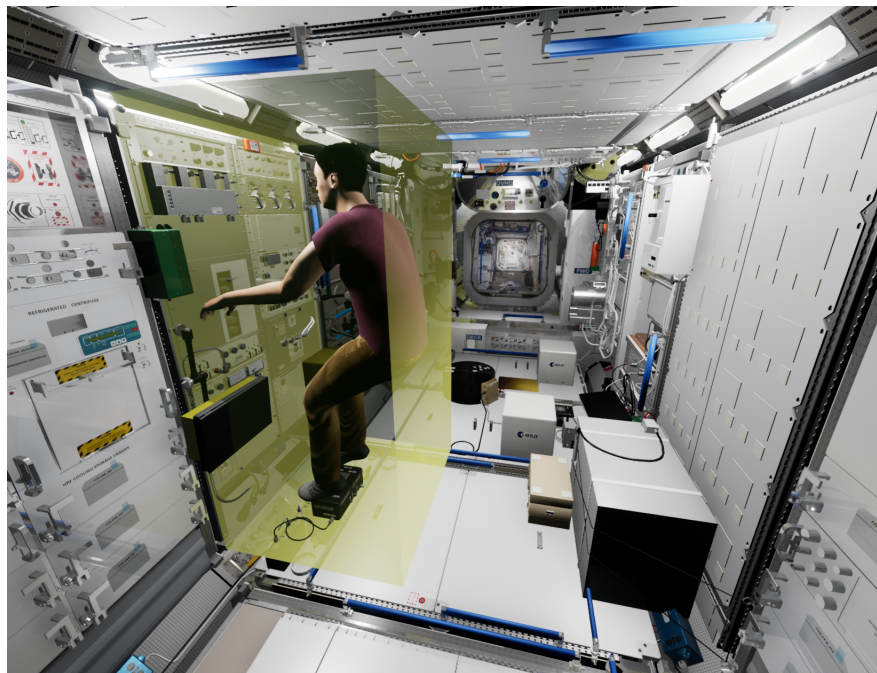
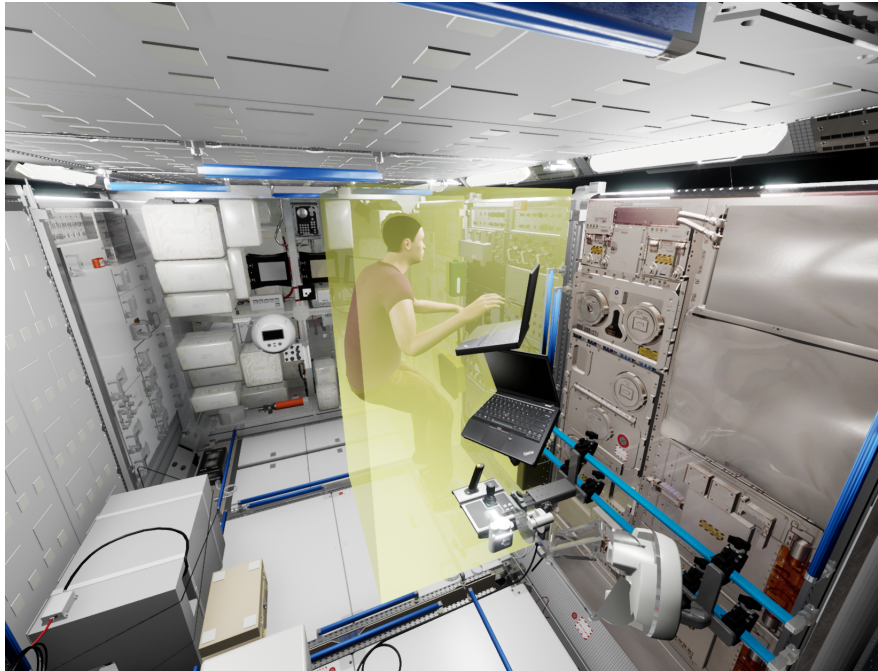


Figura 6.5: Manichino all'interno del work envelope definito per il computer di bordo

6.1.3 Simulazione del lavoro di team

Infine i quattro manichini sviluppati sono stati inseriti contemporaneamente all'interno del modulo per simulare una sessione di lavoro di team (figura 6.6). A questo scopo sono stati rappresentati due tipi di scenario: nel primo troviamo i due manichini maschili nella simulazione del test medico spinale, mentre il 95° femminile opera sul terminale di bordo e il 50° femminile rimane in sospensione aggrappata all'handrail.

Come si può vedere, il modulo è sufficientemente ampio da ospitare comodamente i quattro astronauti, addirittura altri due manichini potrebbero essere aggiunti, anch'essi impegnati nell'utilizzare una delle strumentazioni presenti a bordo.



Figura 6.6: Test lavoro di gruppo

Parte III

Terza Parte

Capitolo 7

Sviluppo del sistema di movimento e rotazione su HTC Vive

7.1 LOP-G e VERITAS 4U

Il modulo LOP-G (Lunar Orbital Platform-Gateway) è un modulo spaziale cislunare pianificato e voluto dalla NASA e da Roscomos, la cui costruzione è prevista a partire dal 2020 e i cui lanci dovrebbero avvenire tra il 2022 e il 2023. Il progetto prevede 7 moduli più un braccio robotico, con una copertura totale di 125 m^3 ; i moduli previsti sono:

- Power and Propulsion Element (PPE), atto alla produzione di energia fotovoltaica;
- European System Providing Refuelling, Infrastructure and Telecommunications (ESPRIT), che costituirà il punto di attracco per i carichi in arrivo;
- Utilization Module, che costituirà il primo modulo abitativo;
- International Habitation Module (I-HAB), contenente spazi notte, di cucina e di esercizio fisico;
- U.S. Habitation Module (U.S.HAB), ovvero la parte statunitense del modulo abitativo della stazione
- Gateway Logistics Modules, usati come moduli di rifornimento;
- Gateway Airlock Module, che costituirà l'airlock principale per le passeggiate spaziali svolte dagli astronauti;

All'interno del vasto progetto che rappresenta la progettazione e la costruzione di un nuovo modulo spaziale cislunare, Thales Alenia Space Italia, aiutata da Airbus, si occupa della progettazione di due dei moduli sopra descritti, ovvero del modulo ESPRIT e del modulo I-HAB; all'interno della rappresentazione 3D di quest'ultimo finora sviluppata è stata svolta la seconda parte del lavoro di tesi, incentrata sullo sviluppo di nuove meccaniche per il plugin per Unreal Engine sviluppato all'interno di TASI, VERITAS 4U.

Il plugin sviluppato dal team TRO-1 del COSE ha come obiettivo quello di sviluppare una serie di funzionalità che permettano di simulare tramite visore ottico HTC Vive un utente in ambiente di microgravità, in modo da valutare anticipatamente il progetto per il quale viene impiegato il plugin, sfruttando anche l'opinione degli astronauti, che possono, immergendosi all'interno dell'ambiente

virtuale, provare e verificare i vari elementi che compongono il modulo. Inoltre, raggiunto un grado di simulazione e di immersione sufficiente, il plugin può essere utilizzato per l'addestramento degli astronauti stessi attraverso lo svolgimento sequenziale di compiti prestabiliti.

Per questa parte del progetto di tesi è stato richiesto come prima cosa di sostituire il manichino standard dell'Unreal Engine utilizzato (figura 7.1) con uno di quelli sviluppati nella prima parte. E' stato scelto il 50° percentile maschile perché quello più somigliante alle dimensioni date al manichino all'interno del plugin, in modo da non aver problemi per determinate variabili che presentano valori di default non ancora modificabili.

Oltre a ciò è stato richiesto di sviluppare un nuovo e più intuitivo sistema di movimento, che includesse un sistema per muoversi all'interno della scena e uno per ruotare il personaggio utilizzato, cercando di simulare il più fedelmente possibile il movimento compiuto dagli astronauti; infine è stato richiesto di ideare e sviluppare un modo per gestire le collisioni del personaggio con le pareti del modulo in modo tale che non risultasse frustrante per l'utente, date le ridotte dimensioni dell'ambiente, ma che fosse comunque utilizzabile per valutare l'ergonomia e la dimensioni degli spazi, per facilitare la fase di progettazione.



Figura 7.1: 50° maschile confrontato con il manichino Unreal Engine

7.2 Sostituzione del manichino Unreal Engine

La sostituzione del manichino è stata richiesta per migliorare la sensazione di simulazione e per dare maggior realismo alla scena. Infatti, da un lato il manichino rappresentante un androide sviluppato da Unreal è completo di numerose animazioni e il suo scheletro, favorito anche dall'aspetto meccanico delle giunzioni e delle varie parti del corpo, rende molto più efficace l'algoritmo di cinematica inversa, che non deve tener conto di artefatti creati sulla mesh dalle rotazioni applicate; dall'altro lato si però riduce notevolmente il realismo della simulazione.

Come già detto, la sostituzione è stata fatta con il manichino rappresentante il 50° maschile per semplicità, ma il progetto finale prevede che l'utente possa scegliere il sesso del manichino e il

percentile che meglio lo rappresenta; per eseguire la sostituzione con un manichino fisso è necessario, come prima operazione, importare la mesh desiderata e includerla nel plugin; bisogna quindi creare un animation blueprint che sia una copia di quello utilizzato sul manichino precedente e ricreare tutte le pose e tutte le animazioni. Fortunatamente Unreal permette di automatizzare questo processo nel caso in cui gli scheletri dei modelli siano sufficientemente simili, come in questo caso, attraverso la procedura di retargeting. Tale procedura deve essere effettuata con attenzione, seguendo vari passaggi e facendo attenzione che le pose iniziali dei due modelli siano le stesse, pena l'ottenimento di pose e animazioni completamente innaturali e inutilizzabili.

I passi da seguire per eseguire correttamente il retargeting sono i seguenti:

- Importare la mesh per la quale si vuole eseguire il retargeting in formato FBX all'interno di Unreal Engine;
- Aprire lo scheletro della mesh appena importata e quello della mesh da cui si vuole eseguire il retargeting;
- Selezionare il pannello Target Manager e, sotto la voce *set up rig* -> *select rig*, selezionare *select humanoid rig* ed assegnare ad ogni osso del manichino di partenza il corrispettivo osso del nostro manichino;
- Posizionare lo scheletro da cui si vogliono copiare animazioni e pose nella stessa posa del secondo scheletro, facendo attenzione che tutte le ossa siano nella corretta posizione;
- Dal menù *skeleton tree* cliccare su *option* -> *show retargeting options*, selezionare la root bone con il click destro e selezionare *recursively set translation retargeting skeleton* per impostare tutte le ossa in *skeleton mode*; successivamente settare l'osso pelvis in *selected mode* e la root bone in *animation mode*;
- Selezionare l'Animation Blueprint o la singola animazione di cui si vuole eseguire il retarget con il click destro e selezionare *retarget anim blueprint* -> *duplicate anim blueprint* o, nel caso di una singola animazione *retarget anim asset* -> *duplicate anim asset*, selezionare lo scheletro bersaglio e cliccare su *retarget*;

Nel caso in cui si creassero artefatti o imprecisioni nel risultato si può riprovare ad eseguire il retargeting solo per le animazioni errate, oppure importarle su un software che ne permetta la modifica.

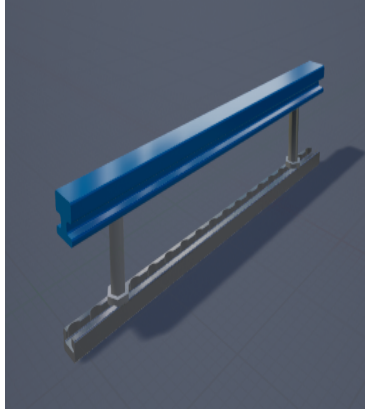
Terminata quindi la fase di retargeting, è stato possibile assegnare al manichino l'animation blueprint completo di animazioni e pose, in modo da sostituire il manichino precedente senza perdere il lavoro svolto per quanto riguarda animazioni, pose e gestione di queste ultime, permettendo all'utente di muovere il proprio avatar attraverso i controller dell'HTC Vive.

7.3 Sviluppo del sistema di movimento e di rotazione

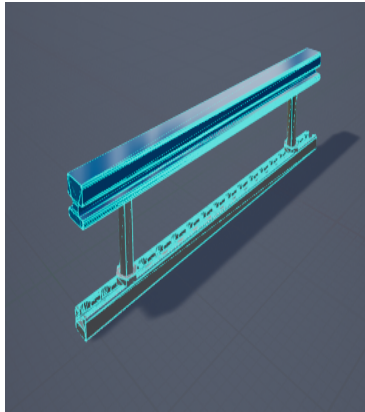
7.3.1 Sistema di movimento

Il sistema di movimento precedente non si preoccupava di simulare il movimento reale che si deve compiere in orbita per spostarsi da un punto del modulo all'altro, né di dare la sensazione di fluttuare in ambiente di microgravità, ma aveva come unico obiettivo quello di permettere all'utente di muoversi all'interno dell'ambiente; si è quindi progettato un modo che permettesse di simulare il più fedelmente possibile il movimento di un astronauta in orbita e che desse la sensazione di trovarsi in ambiente a microgravità. Per fare ciò, la soluzione proposta prevede che l'utente debba sempre avere a disposizione un handrail da afferrare o una parete cui appoggiarsi per darsi una

spinta verso il punto successivo e che, durante il tragitto, venga costantemente applicata al pawn una forza direttamente proporzionale alla spinta impressa che lo faccia fluttuare nel vuoto, come accadrebbe in una situazione reale.



(a) Mesh di una hand rail



(b) collision box dell'hand rail

Per sviluppare questo algoritmo, come prima cosa è stata preparata la collision box della mesh dell'handrail, in modo che si potesse registrare la sovrapposizione tra la mano del manichino e la mesh desiderata; a sovrapposizione avvenuta, la mano del manichino cambia posizione per segnalare all'utente che sia possibile afferrare l'handrail attraverso la pressione del grip button del controller.

In seguito alla pressione, la mano viene chiusa e l'utente è vincolato alla maniglia selezionata; mantenendo premuto il grip button e muovendo il controller il manichino si muove di conseguenza, simulando ciò che accade ad un astronauta all'interno della ISS; se ci si vuole spostare verso un altro punto del modulo, l'utente deve quindi simulare di volersi dare una spinta nella direzione desiderata; dopo aver calcolato l'intensità della spinta basandosi sulla velocità con cui il controller passa da una posizione A ad una posizione B, viene impressa una forza al manichino nella direzione selezionata, permettendo all'utente di "fluttuare" verso il prossimo punto di aggancio. L'algoritmo è stato sviluppato per entrambe le mani, che possono agire indipendentemente e contemporaneamente; l'utente può quindi afferrare due handrail e agire di conseguenza, ottenendo anche più velocità in seguito ad un'ipotetica spinta, oppure può alternare l'utilizzo delle due mani passando da una maniglia all'altra, senza darsi spinte e con movimenti controllati.

Utilizzando questo tipo di movimento l'utente è in grado di raggiungere ogni punto del modulo, dal momento che, come accade nella realtà, ogni superficie del modulo stesso, sia interna che esterna, dispone di numerosi handrail, dal momento che rappresentano l'unica modalità con cui gli astronauti possono comodamente muoversi e orientarsi nello spazio. Nel caso di un modulo in progettazione, inoltre, questo tipo di movimento può essere utilizzato per valutare numero e posizione delle handrail che è necessario inserire all'interno o all'esterno per garantire che tutti gli spazi siano comodamente raggiungibili; per fare ciò è stata aggiunta la possibilità di spostare e aggiungere handrail dove si ritiene possano essere utili; questa caratteristica, tuttavia non è stata oggetto di test in quanto ancora sperimentale e in fase di valutazione da parte di Thales Alenia Space Italia.

7.3.2 Sistema di rotazione

Analogamente al vecchio sistema di movimento, quello di rotazione non simulava realmente il movimento di un astronauta, ma si limitava, alla pressione di uno specifico tasto, a ruotare l'ambiente virtuale lungo uno specifico asse. La scelta di ruotare l'ambiente e non il pawn viene fatta poiché la posizione del pawn non coincide con la posizione della mesh; la mesh, infatti, viene spostata in base alla posizione che assume nello spazio reale il visore HTC Vive lasciando inalterata la posizione dell'oggetto pawn; ciò causa molte criticità, dal momento che la rotazione della mesh viene applicata considerando come centro la posizione pawn che, già dopo pochi secondi, risulta diversa da quella della mesh, causando delle rotazioni errate. Oltre a risolvere questo problema, come per il sistema di movimento, è stato richiesto di sviluppare una meccanica che risultasse realistica e precisa, limitando il senso di nausea provocato agli utenti meno esperti.

Per rendere più immersiva la simulazione si è adottata la stessa tecnica usata per lo spostamento: l'utente deve quindi aggrapparsi a una maniglia e premere un pulsante del controller per poter iniziare la rotazione; mantenendo premuto il pulsante e ruotando il controller lungo un asse, il manichino, vincolato alla maniglia selezionata, ruota lungo l'asse desiderato, simulando il comportamento di un astronauta che desidera effettuare una rotazione analogica. La difficoltà maggiore in questa fase di progetto è stata rappresentata dalla diversità dei sistemi di riferimento di pawn e mesh, risolvibile solo attraverso vari calcoli sui vettori e sui rotori rappresentanti la posizione e gli assi di pawn e mesh. Premendo il pulsante, inoltre, viene ripristinata la corrispondenza tra posizione della mesh e del pawn, in modo da risolvere il problema sopracitato; tuttavia, dal momento che l'utente può afferrare l'handrail orientando la mano in diversi modi, si creano problemi per quanto riguarda la scelta degli assi lungo cui ruotare: l'utente infatti, muovendo il suo controller in un determinato modo si aspetta una certa rotazione della mesh ma, essendo il controller in un sistema di coordinate potenzialmente differente da quello della mesh è necessario introdurre un cambiamento di coordinate e che questo venga ricalcolato per ogni tick di gioco, dal momento che ad ogni Δ di rotazione corrisponde un cambiamento nel sistema di coordinate sia della mesh che della mano dell'utente. Per poter fare ciò bisogna agire all'interno del nodo event tick di Unreal, il quale viene rilanciato all'inizio di ogni tick, andando a prendere in considerazione il sistema di coordinate della mano e della mesh e andando ad operare sui rotori identificati dal Δ di rotazione effettuato dalla mano e applicare tali effetti alla mesh, aggiornando di conseguenza i due sistemi di riferimento.

Vista la complessità del sistema di rotazione, quest'ultimo è stato sviluppato come ultima feature, grazie anche all'aiuto dell'ing. Eugenio Sorbellini ed è momentaneamente disponibile solo sulla mano destra; l'attivazione su entrambe le mani risulta infatti problematica nel momento in cui l'utente decide di ruotare contemporaneamente usando entrambe le mani, causando rotazioni molto rapide e innaturali, producendo non solo glitch grafici, ma anche forte disturbo per l'utente. Inoltre, essendo stato sviluppato tardivamente, il nuovo sistema di rotazione non è stato collaudato durante la prima fase di testing, svolta dall'astronauta Mathias Bauer, mentre è stata provata e valutata da Samantha Cristoforetti durante una seconda sessione.

7.4 Gestione delle collisioni

La gestione delle collisioni risulta di primaria importanza nel momento in cui si voglia utilizzare il plugin per valutare gli spazi e l'ergonomia del modulo, in quanto è necessario registrare, per ogni percentile, i problemi riscontrati muovendosi all'interno di un determinato ambiente.

L'idea iniziale per sviluppare tale feature consisteva nello sfruttare una tecnica basata sulle animazioni fisiche.

Le animazioni fisiche sono un particolare tipo di animazione che si basano sul modello fisico creato per la mesh (figura 7.3); il modello fisico è un particolare tipo di scheletro, composto da

sfere, capsule e giunti che simulano la fisica reale dell'oggetto in questione in seguito all'applicazione di una forza e genera, attraverso una simulazione fisica, animazioni estremamente naturali e realistiche, difficilmente ottenibili con altri metodi. Per esempio, un personaggio che, muovendosi, scontra un muro con la spalla, automaticamente curva il busto a causa dello scontro, accompagnandone il movimento e adattando l'animazione di movimento di conseguenza. Tale approccio è stato scartato per vari motivi:

- Difficoltà nella creazione di un modello fisico: la creazione di un modello fisico è un processo estremamente complicato, che non può essere automatizzato come avviene invece per un normale scheletro e che richiede un'ottima conoscenza non solo del motore grafico, ma anche dell'anatomia umana e di come questa si comporta in relazione ad una sollecitazione; inoltre, sarebbe stato necessario creare un modello fisico per ogni percentile, aumentando notevolmente la complessità del lavoro;
- Pesantezza della simulazione: a fronte di animazioni naturali e realistiche, questa tecnica consuma notevoli risorse, soprattutto in un ambiente piccolo, dove ogni collisione può creare una potenziale catena di collisioni, fino a diventare ingestibile per un PC non particolarmente sofisticato;
- Perdita del controllo del manichino: l'utente, non muovendosi nello stesso ambiente del manichino, non risentirebbe dell'impulso generato sul manichino, percependo solo la fastidiosa e frustrante sensazione di aver perso il controllo del proprio avatar;



Figura 7.3: Modello fisico presente per il manichino Unreal Engine

Per questi motivi si è optato per un metodo più efficace e di più semplice realizzazione: alla mesh del manichino vengono aggiunte sfere e capsule per ricoprire le diverse zone del corpo (testa, torace, busto, braccio, avambraccio, mano, coscia, polpaccio e piede) che non sono visibili dalla camera attraverso cui osserva l'utente; premendo un pulsante dedicato, si attivano due camere che riprendono frontalmente il manichino; impostando una delle due come camera di profondità è possibile isolare solo il manichino, cancellando di fatto lo sfondo e facendo in modo che l'utente veda in sovrapposizione questa ripresa sulla parte laterale dello schermo. Nel momento in cui una delle sfere o delle capsule aggiuntive si sovrappone ad un oggetto del modulo, questa diventa visibile solo nell'immagine aggiuntiva, con un materiale lampeggiante rosso, in modo che l'utente veda quale zona del corpo è in collisione con l'ambiente. In questo modo si ha un impatto minimo sulle prestazioni del programma e l'utente ha la possibilità di attivare il controllo delle collisioni solo quando necessario; inoltre non bisogna eseguire nessuna modifica specifica per un determinato percentile, in quanto le due camere aggiuntive sono facenti parte dell'oggetto pawn e, quindi, indipendenti dalla mesh e dall'animation blueprint utilizzati. Nella figura sottostante si vede un esempio in cui l'avambraccio destro del manichino si sta sovrapponendo con il modulo spaziale. Per motivi tecnici la foto è stata scattata ponendo il manichino al di fuori del modulo e facendolo collidere con le pareti esterne; è stato necessario fare ciò poiché la schermata proiettata sullo schermo e quella vista dall'utente che indossa l'elmetto sono diverse in termini di luminosità e contrasto, cosa che rende poco chiara una foto scattata all'interno del modulo.

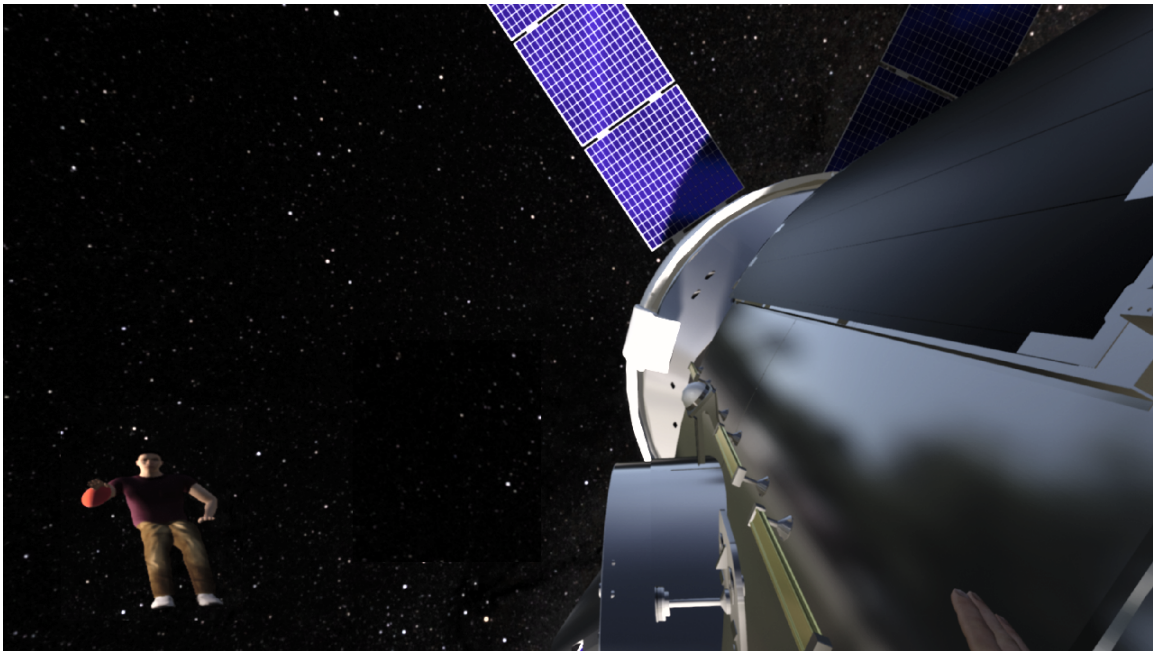


Figura 7.4: Immagine raffigurante un manichino il cui avambraccio è in collisione con la parte del modulo

Capitolo 8

Test d'uso sul plugin VERITAS 4U

8.1 Test e collaboratori esterni

Per il plugin VERITAS 4U, oggetto di studio in questa seconda parte di progetto, sono stati svolti due test, con l'aiuto di collaboratori esterni, nonché di due astronauti, ovvero Samantha Cristoforetti e Mathias Maurer. I test sono stati svolti in tempi diversi, quindi alcune features presenti nell'ultimo test, svolto con l'aiuto di Samantha Cristoforetti, non erano presenti nei test precedenti, che non tengono quindi in considerazione determinati aspetti e caratteristiche presenti nella versione provata dall'astronauta italiana.

In entrambi i casi i test sono stati svolti in remoto, con un avatar comandato in locale dall'ing. Eugenio Sorbellini e uno comandato da un centro esterno dal tester; i test erano strutturati come segue:

- Prima fase: familiarizzazione con i comandi e con l'ambiente virtuale; in questa fase l'ing. Eugenio Sorbellini ha spiegato tutti i comandi ai tester e ha permesso loro di provare a esplorare liberamente l'ambiente, in modo da poter svolgere i test presenti nella seconda fase;
- Seconda fase: in questa fase ai candidati è stato richiesto di svolgere tre procedure specifiche:
 - Window: in questo primo test, svolto per valutare la posizione della finestra all'interno del modulo e per valutare il sistema di movimento sviluppato. Il candidato deve avvicinarsi alla finestra utilizzando le handrails presenti, ancorarsi alle due poste vicino alla finestra e valutare la visibilità offerta dalla finestra;
 - Crew Quarter: in questo test il soggetto deve aprire il crew quarter utilizzando i comandi presenti a bordo del modulo, posizionarsi al suo interno e valutare gli spazi e le eventuali collisioni;
 - Medical Care: infine al soggetto viene chiesto di aprire lo sportello contenente il kit di primo soccorso, ruotarsi e passarlo all'avatar comandato in locale e posizionato vicino ad un terzo manichino, facente le veci di un astronauta ferito;

La prima sessione di test è stata svolta dall'astronauta Mathias Maurer e da due appartenenti all'ESA, Andrea Casini e Lionel Ferra, quando l'applicazione ancora non disponeva della nuova versione per la rotazione; tra questi soggetti solo Mathias Maurer ha una scarsa conoscenza della realtà virtuale, mentre gli altri due candidati hanno dichiarato di possedere un buon livello di

esperienza con questa tecnologia. Tutti i candidati hanno trovato positiva l'esperienza in termini di usabilità della simulazione, valutando quindi positivamente la facilità di utilizzo dello strumento e di movimento nell'ambiente, ritenuto sufficientemente immersivo. Pur valutando positiva l'interazione con l'altro avatar presente, impersonato dall'ing. Eugenio Sorbellini, la mancanza di un sistema di rotazione ha causato problemi nell'orientamento del personaggio all'interno dell'ambiente, senza però causare motion sickness o frustrazione negli utenti. In generale, due dei tre candidati hanno dichiarato di aver trovato un prodotto sopra le proprie aspettative, mentre le aspettative dell'istruttore Lionel Ferra sono state confermate. I commenti rilasciati ricoprono vari aspetti, pur concordando sulla buona riuscita della simulazione; in particolare l'astronauta Mathias Maurer, terminato il test, si è mostrato entusiasta della simulazione, affermando che tale esercitazione dovrebbe essere obbligatoria ("it should be mandatory"). I commenti, che si possono trovare nei test riportati di seguito, concordano sulla possibilità di migliorare ulteriormente il realismo e il senso di presenza che dà la simulazione, pur lamentando la mancanza di un sistema di rotazione migliore.

Lionel Ferra, inoltre, ha richiesto di studiare un modo per tracciare le gambe dell'utente; tale feature era già stata richiesta, ma mai introdotta da Thales a causa della natura stessa del plugin: durante l'utilizzo, infatti, l'utente rimane in piedi, di conseguenza ha le gambe in una posizione sicuramente diversa da quella del manichino e, anche compensando questa differenza, risulterebbe innaturale per l'utente vedere le gambe del proprio manichino muoversi diversamente da come sono mosse le proprie nell'ambiente reale. Tutti i tester si sono inoltre lamentati del vecchio sistema di gestione delle collisioni, basato sulla vibrazione dei controller in presenza di una collisione; il sistema si è rivelato fastidioso e poco utile, dal momento che la vibrazione risulta pressoché continua a causa dello spazio ristretto del modulo e senza dare indicazioni sulla parte del corpo che si trova a collidere; in particolare Lionel Ferra, non solo lamenta l'assenza di un sistema che segnali la collisione, ma addirittura chiede, all'interno dei commenti, per quale motivo il controller continuasse a vibrare, segnalando l'effettiva inefficienza di questo sistema di gestione delle collisioni. Per queste motivazioni, quest'ultimo è stato aggiornato, prendendo in considerazione anche le idee proposte dai tester, portando allo sviluppo del sistema descritto nel capitolo precedente, che non solo segnala la parte del corpo che si trova a collidere, ma è anche disattivabile da parte dell'utente, che può quindi decidere quando tale informazione possa risultare utile.

COLLABORATIVE VR – USER QUESTIONNAIRE

Please compile this user questionnaire about your experience with the collaborative Vr System-Gateway4U (t3)

NAME: Andrea

SURNAME: Casini

Age: 30

Left/Right handed: Right

Role: Research Fellow

Have you ever used a VR system before? Yes

Please indicate your expertise with VR applications of similar type, circling the appropriate point on the scale:

Low	Moderate	Medium	Considerable	High
-----	----------	--------	--------------	------

Please answer the following questions about the collaborative vr system you just used. Please indicate how much you agree with each of the statements by checking the appropriate box on the scale.

Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Usability Questions					
Audio voice chat allows clear communication	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found it easy to use the virtual reality system	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I did not notice any disturbing lag or delay between my movements of the controls and the response in the virtual environment to my actions	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
My general sense of presence (feeling of "being there") in the virtual environment was high.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I could move easily through the virtual environment	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The avatar body orientation control causes excessive sickness	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
The body in the virtual environment faithfully reproduces my movements	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 8.1: Risultati test rilasciati da Andrea Casini(1)

The body of the other player faithfully reproduces human movements	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
I can interact correctly with other players	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Input Device Usability Questions					
I found it frustrating to use the Collaborative VR System	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
I found it easy to understand how to use the controllers input device to interact with the virtual environment	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Input VR Usability (Functional Allocation Volume)					
I found it easy the perception of the field of view looking through the window	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found it easy to understand the allocated volume for the Crew Quarter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found it easy to understand the allocated volume for the Medical Care	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

How did your actual experience of the Virtual Environment compare with your initial expectations?

<i>Much worse than I expected</i>	<i>Much better than I expected</i>
1 2 3 4 5 6	

Please , add comments if any, here:

The assessment of the volumes was easier that t2 because there were no "colored transparent" volume highlighted (all the time).

The mannequin was better in terms of external looking, but can still be improved/humanized a little bit more (maybe add some physics/more natural behaviours in clothes and less "robotic" movement of body/upper body/arms). The upper body bending shall be implemented as well as a revised form of rotation (not only using the user's centre of gravity. The leg movements were not at all realistic and shall be improved (most of the time I saw the other player/user floating around in the microgravity environment with his/her legs "waving as a flag pushed by wind"). Grabbing handles was fine, but the rotation while being attached to them (using the directional pad of the left controller) was too artificial and not really mimicking human movements.

The assessment for the virtual window was fine (for the purpose of the demo), even if the actual purpose and potential movements (e.g. folding) of the blue walls were a bit confusing. I was indeed not sure if the actual available volume was the one confined between the walls or it could had been more. However, the task to be performed in scenario 2 of t3 was clearly explained and understood (i.e. crew restrain positions).


The polygon appearing behind the virtual users' mannequins and object was a bit confusing to see since it should be not part of the simulation (it is not a part of any of the objects the users can interact with).

Even if vibration was implemented as a collision/interpenetration avoidance/awareness response to help the user, in the end it turned out to be most of the time active and a bit confusing. While feeling it all the time is a bit meaningless, it could be useful to implement a sort of proportional vibration (the more you are "inside" any object, the more the controller vibrate) plus small visual aids (e.g. if the user is in the proximity to touch anything with his/her


Figura 8.2: Risultati test rilasciati da Andrea Casini(2)


left body part, then a small visual indicator should be added on his/her left field of view or a small human body replica can be added similar to what is done in videogames with the mini-map displayed somewhere in the screen). The feeling of microgravity (even if I never experienced in first person) was enhanced by the inertia implemented while pushing myself away and ungrasping the handles, but this was the only "real" feeling of microgravity (apart from using/throwing the medical bag in the scenario 2). The extensive use of the "free floating" translation to move from one location to another and the unnatural "artificial" rotation were compromising a bit the global feeling of the entire experience. Moreover, the "grasp by hands" command was not the best button combination: apart from the initial mistakes I did in pressing the wrong part of the controllers, from time to time the command was not working and "squeezing" the controller required a not negligible effort (both physical if the position should had been held for quite some time and mental for performing other tasks that required pressing other buttons). As desired features for future improvements, it is strongly suggest to upgrade/ameliorate the entire body "suite" (collision/physics and general microgravity motion in particular). Finally, related to the scenario 2, the mannequin was positioned too close to the walls, so it is suggested to move it to the centre of the compartment for a better accessibility (regardless if the activity is performed by 1 or more crewmembers).

Figura 8.3: Risultati test rilasciati da Andrea Casini(3)



COLLABORATIVE VR – USER QUESTIONNAIRE



Please complete this user questionnaire about your experience with the collaborative  System-Gateway4U (t3)

NAME: Lionel

SURNAME: Ferra

Age: 43


Left/Right handed: Right

Role: Astronaut Instructor/VR lead

Have you ever used a VR system before? yes

Please indicate your expertise with VR applications of similar type, circling the appropriate point on the scale:

Low	Moderate	Medium	Considerable	High
-----	----------	--------	---------------------	------

Please answer the following questions about the collaborative  system you just used. Please indicate how much you agree with each of the statements by checking the appropriate box on the scale.

Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Usability Questions					
Audio voice chat allows clear communication	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found it easy to use the virtual reality system	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I did not notice any disturbing lag or delay between my movements of the controls and the response in the virtual environment to my actions	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
My general sense of presence (feeling of "being there") in the virtual environment was high.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I could move easily through the virtual environment	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The avatar body orientation control causes excessive sickness	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
The body in the virtual environment faithfully reproduces my movements	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 8.4: Risultati test rilasciati da Lionel Ferra(1)

The body of the other player faithfully reproduces human movements	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
I can interact correctly with other players	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Input Device Usability Questions					
I found it frustrating to use the Collaborative VR System	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
I found it easy to understand how to use the controllers input device to interact with the virtual environment	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Input VR Usability (Functional Allocation Volume)					
I found it easy the perception of the field of view looking through the window	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
I found it easy to understand the allocated volume for the Crew Quarter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found it easy to understand the allocated volume for the Medical Care	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did your actual experience of the Virtual Environment compare with your initial expectations?

<i>Much worse than I expected</i>	<i>Much better than I expected</i>
1 2 3 4 5 6	

Please , add comments if any, here:

Overall, you did a good job.


My points for improvements and ensure to get proper feedback from astronauts and other users later on:

- 1- Make the user move only using grabable/interactive surfaces and handrails. E.g I can push myself away using a wall, or a corner.
- 2- Always implement inertia: if I grab a handrail, the quicker I release it and push me away the faster I go.
- 3- Body orientation shall be achieved only when rotating my hands around handrails or similar. It shall not be possible to use the left touch pad on the controller to do so.
- 4- By doing so, enable the grabbing of handrails...etc... maybe done with the trigger instead of the squeeze button. To be investigated.
- 5- The handrails squeezing was not very easy.
- 6- Add annunciation (vibration, mini HUD or else) indicating if my avatar is in collision with any object/surface.
- 7- What were all those vibrations this time?
- 8- Investigate also the tracking of the feet.

I believe, before involving more astronaut review, points 1- to 7- shall be implemented.

Figura 8.5: Risultati test rilasciati da Lionel Ferra(2)

COLLABORATIVE VR – USER QUESTIONNAIRE



Please compile this user questionnaire about your experience with the collaborative Vr System-Gateway4U (t3)

NAME: Maurer

SURNAME: Matthias

Age: 49

Left/Right handed: right

Role: Astronaut

Have you ever used a VR system before? yes

Please indicate your expertise with VR applications of similar type, circling the appropriate point on the scale:

Low	Moderate	Medium	Considerable	High
-----	----------	--------	--------------	------

Please answer the following questions about the collaborative vr system you just used. Please indicate how much you agree with each of the statements by checking the appropriate box on the scale.

Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Usability Questions					
Audio voice chat allows clear communication	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found it easy to use the virtual reality system	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I did not notice any disturbing lag or delay between my movements of the controls and the response in the virtual environment to my actions	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
My general sense of presence (feeling of "being there") in the virtual environment was high.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I could move easily through the virtual environment	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
The avatar body orientation control causes excessive sickness	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
The body in the virtual environment faithfully reproduces my movements	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 8.6: Risultati test rilasciati da Mathias Maurer(1)

The body of the other player faithfully reproduces human movements	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I can interact correctly with other players	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Input Device Usability Questions					
I found it frustrating to use the Collaborative VR System	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
I found it easy to understand how to use the controllers input device to interact with the virtual environment	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Input VR Usability (Functional Allocation Volume)					
I found it easy the perception of the field of view looking through the window	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found it easy to understand the allocated volume for the Crew Quarter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found it easy to understand the allocated volume for the Medical Care	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did your actual experience of the Virtual Environment compare with your initial expectations?

<i>Much worse than I expected</i>	<i>Much better than I expected</i>
1 2 3 4 5 6	

Please , add comments if any, here:

1. The VR is VERY helpful to personally visualize and experience a future module. This is tremendously helpful to give proper crew feedback on design or functional features before any expensive hardware is built.
2. The VR feeling is already quite natural but still offers room for further improvements: I should be able to use also the walls and objects to grasp hold of or to apply force to push myself away. A HR should allow to apply rotation forces with my hand.
3. The question you askes "where should the HR in the crew quarter be?" is the wrong question. Crew would prefer a foot restraint to be able to work with both hands on the computer.
4. When designing the module and the windows, please consider that the astronaut in 0g adopts a slightly more fetal like body position. This should be reflected in the VR body position but also in the architectural arrangement. In this respect VR is a perfect too to optimize this.
5. To get an even more useful feedback I'd recommend to offer more training time for the evaluator before starting the assessment; I feel that I was still low in my training curve while already providing feedback to you

Figura 8.7: Risultati test rilasciati da Mathias Maurer(2)

Successivamente è stato svolto un test dall'astronauta italiana Samantha Cristoforetti, che ha potuto provare il progetto completo di sistema di rotazione.

Nonostante anche in questo caso l'esperienza sia stata tendenzialmente positiva, l'attenzione ricade sul voto negativo relativo al sentimento di presenza; ciò è probabilmente dovuto ai controller disponibili nel centro dal quale operava, dal momento che il controller sinistro risultava spesso non funzionare in fase di presa, rendendo di fatto impossibile provare la presa con due mani. Tra i commenti rilasciati invece, quello forse più interessante riguarda la difficoltà di rappresentare la posizione dell'utente all'interno dell'ambiente virtuale e di farlo interagire con l'ambiente stesso, ponendo l'attenzione sulla difficoltà nel valutare le distanze.

Samantha Cristoforetti, oltre agli stessi test basati su procedure ben definite svolti dai candidati precedenti, ha svolto anche test riguardanti la struttura del modulo I-HAB (International Habitation Module) e l'abitabilità del Crew Quarter in esso contenuto; l'importanza del suo parere su questi precisi argomenti è strettamente legata al suo periodo di soggiorno a bordo dell'ISS, che le permette di avere ben presente quali possano essere le criticità che si riscontrano una volta in orbita. Grazie ai suoi commenti è possibile quindi migliorare determinati aspetti dell'esperienza di simulazione e prendere coscienza di quali siano gli aspetti più difficili, o addirittura impossibili, da riprodurre in un ambiente di realtà virtuale.

Come si legge nei test sotto riportati, l'attenzione viene posta sulla difficoltà di valutare la posizione del sacco a pelo utilizzato per dormire a causa della difficoltà nel valutare l'illuminazione e l'isolamento audio all'interno del Crew Quarter, come è altresì difficile valutare quale possa essere un buon sostegno per il sacco a pelo stesso.

Si può inoltre notare come il progetto sia migliorato durante il tempo intercorso tra i due test, dal momento che, mentre i soggetti del primo test lamentavano una scarsa naturalezza nella riproduzione dei movimenti da parte dei manichini, Samantha Cristoforetti ha valutato positivamente le animazioni e i movimenti degli stessi, rimanendo neutrale sul giudizio del proprio avatar in quanto difficile da giudicare in prima persona (si rimanda al commento 4 contenuto all'interno del test sotto riportato).

COLLABORATIVE VR – USER QUESTIONNAIRE

Please compile this user questionnaire about your experience with the collaborative Vr System-**Gateway4U (t4)**

NAME: Samantha

SURNAME: Cristoforetti

Age: 42

Left/Right handed: Right

Role: Astronaut

Have you ever used a VR system before? Y

Please indicate your expertise with VR applications of similar type, circling the appropriate point on the scale:

Low	Moderate	Medium	Considerable	High
-----	----------	--------	--------------	------

Please answer the following questions about the collaborative vr system you just used. Please indicate how much you agree with each of the statements by checking the appropriate box on the scale.

Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
Usability Questions					
Audio voice chat allows clear communication ¹⁾	<input type="checkbox"/>	<input type="checkbox"/>	x	<input type="checkbox"/>	<input type="checkbox"/>
I found it easy to use the virtual reality system	<input type="checkbox"/>	x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I did not notice any disturbing lag or delay between my movements of the controls and the response in the virtual environment to my actions	x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
My general sense of presence (feeling of "being there") in the virtual environment was high. ²⁾	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x	<input type="checkbox"/>
I could move easily through the virtual environment ³⁾	<input type="checkbox"/>	x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The avatar body orientation control causes excessive sickness	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x	<input type="checkbox"/>
The body in the virtual environment faithfully reproduces my movements ⁴⁾	<input type="checkbox"/>	<input type="checkbox"/>	x	<input type="checkbox"/>	<input type="checkbox"/>
The body of the other player faithfully reproduces human movements	<input type="checkbox"/>	x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I can interact correctly with other players ⁵⁾	<input type="checkbox"/>	<input type="checkbox"/>	x	<input type="checkbox"/>	<input type="checkbox"/>

Figura 8.8: Risultati test rilasciati da Samantha Cristoforetti(1)

Input Device Usability Questions					
I found it frustrating to use the Collaborative VR System	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x
I found it easy to understand how to use the controllers input device to interact with the virtual environment	<input type="checkbox"/>	x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

How did your actual experience of the Virtual Environment compare with your initial expectations?

Much worse than I expected *Much better than I expected*

1 2 3 **4** 5 6

Please, add comments if any, here below:

- 1) There was an echo issue, so that I heard my voice as I spoke, with a small delay. This was fixed in the subsequent session with TAS-F. Please make sure you are aware of the correct settings to avoid the echo. Otherwise audio communication was good.
- 2) This is still one of the major issues. The VR environment provides little cues about one's position in the three-dimensional space and one's interactions with the habitat geometry. It's also difficult to estimate distances: if I reach out my hand to reach a handrail, how far am I from it?
- 3) In general yes, but it would be easier if it was clear what objects can actually be grabbed and used to translate. Also, it seems to me that it was impossible to grab handrails with both hands at the same time and pivot in a more natural way, i.e. by generating a torque with one's hands.
- 4) Hard to judge, better seen from the outside, I guess.
- 5) I am not sure what interactions were possible. Interactions in the test were limited to seeing each other and talking. Both those functions worked fine.

Figura 8.9: Risultati test rilasciati da Samantha Cristoforetti(2)

SESSION A - Crew Quarter Questions					
Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I found that the overall functions provided fulfill the user needs ¹⁾	<input type="checkbox"/>	<input type="checkbox"/>	X	<input type="checkbox"/>	<input type="checkbox"/>
I found that a flexible/foldable rack is suitable for habitat volume use ²⁾	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X	<input type="checkbox"/>
I found that the removable light is enough ³⁾	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found that the functions location (USB, laptop, ventilation etc....) is suitable for the user ³⁾	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found that the stowage area is easily accessible	<input type="checkbox"/>	X	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I found that the stowage area is enough to accommodate personal items/utilities	<input type="checkbox"/>	X	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I need a dedicated support to sustain and fix the body during work or personal activities (e.g. jacket, seat with a belt) ⁴⁾	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X
I found the small door side of the multifunctional panel a good support for the sleeping bag ⁵⁾	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Please , add comments if any, here:					
<p>1) Hard to say. The "sleeping" experience really depends a lot on the quality of the light and auditory insulation. This will always be hard to judge in a VR environment, I'm afraid.</p> <p>2) It seems like a hybrid solution, intermediate between a makeshift sleeping arrangement and a permanent CQ. Unfortunately, at a first assessment it seems to lack the benefits of one of the other, while I am not sure of what added benefits it would bring. Maybe if one was extremely mass constrained, but wanted to offer at least some type of CQ, this could be an option.</p> <p>3) Did not assess</p> <p>4) There is no need for it, foot restraints and hooks for bungees like on ISS is enough. One could however assess the delta benefits of such solutions.</p> <p>5) Impossible to judge in the VR environment.</p>					

Figura 8.10: Risultati test rilasciati da Samantha Cristoforetti(3)

SESSION B - IHAB (4.2m dia+ 3m dia) Functional Allocation Volume Questions					
Statement	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree
I found it easy to understand the allocated volume for the CQ ¹⁾	<input type="checkbox"/>	x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<p>Which are the main guidelines you consider necessary to address the "temporary" CQ design?</p> <p>"Temporary" means that there shall be no volume allocated exclusively to sleeping, but rather CrewMembers will attach sleeping bags in the cabin. Besides providing attachment points in suitable places, in consideration of available volume, simple, deployable solutions to provide some measure of auditory and visual isolation might be considered. The temporary nature of CQs in iHAB is based on the assumption that iHAB will be the first habitat, that it will host a toilet and therefore permanent CQs shall be in the second habitat, that CrewMembers can accept a temporary solution for one mission.</p>					
I found it easy to understand the allocated volume for the Hygiene	<input type="checkbox"/>	x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<p>Which are the main guidelines you consider necessary to address the "temporary" Hygiene?</p> <p>A location/volume in the cabin that is not exclusively dedicated to hygiene, but can be reconfigured temporarily (for example with a curtain) during hygiene functions and provides restraint aids and the possibility to store hygiene items, or at least temporarily stow them during hygiene ops.</p>					
I found it easy to understand the allocated volume for the Toilet	<input type="checkbox"/>	x	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<p>Since the limited volume available, the cross-contamination risk, noise produced and ECLSS accommodation could be a criticalities from configuration and human factor point of view?</p> <p>Yes. Therefore the current tentative functional allocation foresees that sleep and food preparation/consumption function be only temporarily allocated to iHAB and then moved to the second habitat. Proximity of toilet with food preparation/consumption and sleep as a permanent solution is extremely undesirable.</p>					

Please , add further comments if any, here:

- 1) I think it's OK, but there are definite limitations in the VR environment, in terms of perceiving one's relationship to the surrounding space, that inevitably leave an element of doubt.

Overall, I found it a pity that the internal volume was only filled in small part, since it would have been interesting to see all the volume-intensive functions, like exercise, accommodated in the cabin.

Figura 8.11: Risultati test rilasciati da Samantha Cristoforetti(4)

8.2 Futuri sviluppi

I manichini sviluppati e i miglioramenti apportati al plugin VERITAS 4U sono già stati utilizzati su altri scenari, per esempio all'interno dei test per il modulo ESPRIT presso la sede francese di Thales Alenia Space; è inoltre possibile utilizzarli per altri scenari che richiedano l'utilizzo di manichini con misure antropometriche precise e con ambiente non necessariamente in microgravità. I manichini potranno essere utilizzati quindi in molti dei futuri progetti che saranno avviati in TASI; per esempio potrebbero essere utilizzati per i test ergonomici che verranno svolti in relazione al progetto della Chinese Space Station (CSS).

Esistono ancora numerosi miglioramenti che possono essere fatti per rendere sempre più realistica la simulazione e sempre più utile e utilizzabile il progetto presentato; per i manichini, i miglioramenti attuabili possono essere:

- Creazione dei percentili mancanti: nello specifico non sono stati creati il 5° percentile maschile e il 95° percentile femminile; i percentili fondamentali per la maggior parte dei test sono infatti rappresentati dagli estremi, ovvero dal 5° percentile femminile e dal 95° maschile;
- Espressioni facciali: al momento i manichini non dispongono di alcuna espressione facciale; per quanto queste non siano strettamente necessarie per le finalità dell'applicazione, potrebbe risultare utile inserire una variazione nell'espressione del manichino al solo scopo di rendere più realistica e piacevole la simulazione; con le stesse finalità potrebbe essere aggiunta l'animazione di palpebre e sopracciglia;
- Miglioramento dei movimenti: per ampliare il range di posizioni realizzabili dal manichino e per aumentarne la naturalezza; una possibile soluzione potrebbe essere l'aggiunta di catene cinematiche che comprendano bacino, busto e collo del manichino;

Il plugin VERITAS 4U è invece ancora in fase di sviluppo e numerosi miglioramenti sono al momento in valutazione da parte di Thales Alenia per rendere più immersivo ed utilizzabile il plugin; tra questi troviamo:

- Scelta del manichino: al momento è possibile utilizzare solo il 50° percentile maschile, nonostante all'avvio dell'applicazione venga chiesto all'utente di inserire la propria altezza; in futuro, a seconda dell'altezza inserita verrà assegnato il percentile corretto; inoltre, l'utente potrà specificare il proprio sesso, in modo tale da variare di conseguenza anche quello del manichino.
- Comunicazione: la comunicazione avviene attraverso il microfono incorporato nel visore HTC Vive, ma al momento la comunicazione soffre, talvolta, di disturbi di vario genere; nonostante sia stato risolto il problema di eco messo in evidenza da Samantha Cristoforetti, la qualità audio è ancora migliorabile, così come il ritardo nella ricezione da parte degli altri utenti.
- Miglioramento della cinematica inversa: il movimento improvviso dell'utente può causare un errore nel tracking dei due controlli, dando origine per pochi secondi a pose innaturali e a una risoluzione errata della cinematica inversa del manichino; inoltre la rotazione della mano da parte dell'utente si ripercuote solo sul polso del manichino e non anche sull'avambraccio come accade nella realtà; per risolvere questi problemi bisogna agire direttamente sull'algoritmo risolutore della cinematica inversa, aumentandone la precisione ed eseguendo controlli sugli angoli raggiunti a run time.

Bibliografia

- [1] Interactive technology and the new paradigm for healthcare, 1995. pag 180-181, [devono fornire i giusti stimoli olfattivi nel momento appropriato durante l' operazione. Allo stesso modo, la formazione del personale medico di emergenza che opera sul campo dovrebbe metterli in contatto con gli odori che renderebbero l' ambiente virtuale più reale e che potrebbero fornire informazioni diagnostiche sulle ferite che si suppone la vittima abbia riportato].
- [2] J.Lasenby A.Aristou. Fabrik: A fast, uterative solver for inverse kinematics problem. Graphical Models, vol.73, pp.243-260, 2011.
- [3] Luke Bastianich, 2017. [Online; in data 29-Luglio-2019], disponibile all' indirizzo <https://www.virtualerealta.it/storia-realta-virtuale-30/>.
- [4] Mino Bibawy. Sito cose centre. [Momentaneamente disponibile solo su rete intranet aziendale], indirizzo intranet tomatedp01:8088.
- [5] Cave-Reply. [Online; in data 26-Luglio-2019], disponibile all' indirizzo <https://www.reply.com/forgereply/it/cave/>.
- [6] Unreal Engine. [Online in data 17-Agosto-2019], disponibile all'indirizzo <https://www.unrealengine.com/en-US/feed>.
- [7] Esa. [Online; in data 29-Luglio-2019], disponibile all' indirizzo https://www.esa.int/Our_Activities/Human_and_Robotic_Exploration/Columbus.
- [8] FeelReal. Feelreal sensory mask. [Online in data 17-Settembre-2019], disponibile all' indirizzo <https://feelreal.com/>.
- [9] GBTimes, 2018. [Online; in data 29-Luglio-2019], disponibile all' indirizzo <https://gbtimes.com/what-we-know-about-chinas-space-station-modules-crew-launch-plans-and-more>.
- [10] IEA. [Online; in data 2-Agosto-2019], disponibile all' indirizzo <https://www.iea.cc/>.
- [11] Il Messaggero, 2018. [Online; in data 29-Luglio-2019], disponibile all' indirizzo https://www.ilmessaggero.it/tecnologia/scienza/stazione_spaziale_luna_terra_accordo_esa_thales_alenia_space_moduli-3952226.html.
- [12] Nasa. [Online; in data 2-Agosto-2019], disponibile all' indirizzo <https://msis.jsc.nasa.gov/sections/Section03.htm>.
- [13] Nasa. Human integration design handbook, 27 Gennaio 2010.
- [14] Virtual Reality Society. Applications of virtual reality, 2015. [Online in data 12-Settembre-2019], disponibile all' indirizzo <https://www.vrs.org.uk/virtual-reality-applications>.
- [15] Thales Alenia Space. [Online; in data 25-Luglio-2019], disponibile all' indirizzo <https://www.thalesgroup.com/it/italia/global-presence-europe/italia>.
- [16] D. Thalmann and N. Magnenat-Thalmann. New trends in animation and visualization, 1992. Eds. Wiley, New York, NY, USA.
- [17] VIVE. [Online; in data 26-Luglio-2019], disponibile all' indirizzo <https://www.vive.com/eu/>.
- [18] Breaking VR. Microsoft breveta un dispositivo di feedback tattile per realtà mista, 30/05/2018. [Online in data 17-Settembre-2019], disponibile all' indirizzo <https://breakingvr.it/2018/05/30/>

[microsoft-brevetta-un-dispositivo-di-feedback-tattile-per-realta-mista/#more-2468](#).

- [19] Wired. Want to know the real future of ar/vr? ask their devs. [Online in data 17-Settembre-2019], disponibile all' indirizzo <https://www.wired.com/story/future-ar-vr-survey/>.