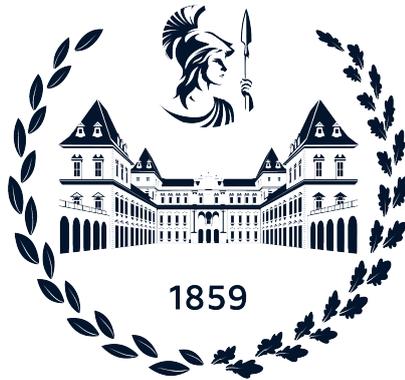


# POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Aerospaziale



## Politecnico di Torino

**Tesi di Laurea Magistrale**

### **Digitalizzazione di procedure operative in realtà virtuale per il training dei piloti tramite l'integrazione di Pacelab WEAVR con il simulatore di volo X-Plane**

**Relatore**

Prof. Giorgio GUGLIERI

**Correlatore**

Dott. Stefano PRIMATESTA

**Correlatore aziendale TXT Group**

Ing. Angelo GINO

**Candidato**

Francesco Salvatore TARTAGLIA

*Dicembre 2021*



# Sommario

Il progetto presentato in questo elaborato di tesi è il risultato della collaborazione tra Politecnico di Torino e la società TXT Group e si inserisce nello sviluppo del settore aziendale del *Training and Simulation* in realtà virtuale, al fine di consentire al pilota di familiarizzare con l'ambiente della cabina di pilotaggio dell'Airbus A320.

Il progetto prevede l'implementazione di procedure operative in grado di migliorare l'efficacia dell'apprendimento del pilota. Inizialmente verranno implementate le procedure operative che il pilota esegue **a terra**, *Engine Start* e *On Ground Emergency Evacuation*, attraverso l'utilizzo del software PACELAB WEAVR sviluppato da TXT; questa fase del lavoro consente di ampliare i progetti di tesi [1], [2], [3] e [4] precedentemente realizzati con TXT. Successivamente si procede allo sviluppo e all'implementazione di procedure operative che il pilota esegue **in volo**: *Takeoff* e *Landing*. Il fulcro della tesi consiste nell'integrare, in WEAVR, la simulazione di volo generata da X-PLANE, tramite l'utilizzo di *SimulationHub*, un altro strumento in fase di sviluppo da parte di TXT.

Nel corso del seguente elaborato si affronteranno argomenti letterari relativi alle caratteristiche della realtà virtuale e al suo stato dell'arte, ponendo l'attenzione sul *Training* in VR, oltre ad una panoramica sulla società TXT. Verranno presentate le strumentazioni software e hardware e spiegate le loro modalità di utilizzo nelle specifiche fasi del lavoro. L'elaborato riporta anche le procedure operative sviluppate, con la descrizione della loro implementazione, digitalizzazione e modalità d'uso da parte dell'utente.

Questo progetto di tesi è sviluppato in coppia con il collega Giovanni Sgarra, con cui ho collaborato nella realizzazione di tutte le fasi operative. Il risultato di questo lavoro consiste nella realizzazione di uno strumento innovativo basato sulla realtà virtuale per il training di procedure operative. Le procedure statiche e dinamiche sviluppate consentono una rapida formazione dei piloti attraverso un metodo di formazione immersivo ed interattivo.



# Indice

<b>Elenco delle tabelle</b>	VII
<b>Elenco delle figure</b>	IX
<b>Acronimi</b>	XIII
<b>1 Introduzione</b>	1
1.1 Obiettivo della tesi . . . . .	3
1.2 TXT Group . . . . .	4
1.3 Outline della tesi . . . . .	6
<b>2 Stato dell'arte</b>	9
2.1 Realtà virtuale . . . . .	10
2.2 Realtà virtuale per addestramento piloti . . . . .	13
2.3 Realtà virtuale per altri tipi di training . . . . .	19
2.3.1 Addestramento vigili del fuoco . . . . .	19
2.3.2 Training nel settore edile . . . . .	21
2.3.3 Addestramento nel settore militare . . . . .	22
2.3.4 Addestramento astronauti . . . . .	23
2.3.5 Realtà virtuale e medicina . . . . .	25
<b>3 Software e hardware</b>	27
3.1 Software . . . . .	27
3.1.1 SteamVR . . . . .	27
3.1.2 Unity . . . . .	28
3.1.3 Pacelab WEAVR . . . . .	30
3.1.4 WEAVR Simulation Hub . . . . .	33
3.1.5 Visual Studio . . . . .	34
3.1.6 X-Plane . . . . .	35
3.1.7 X-Plane Connect . . . . .	36
3.2 Hardware . . . . .	37

3.2.1	HTC Vive Pro . . . . .	37
3.2.2	Simulatore A320 . . . . .	39
<b>4</b>	<b>Procedure a terra</b>	<b>43</b>
4.1	Engine Start . . . . .	44
4.2	On Ground Emergency Evacuation . . . . .	55
<b>5</b>	<b>Procedure in volo</b>	<b>61</b>
5.1	Piano di volo LIMC - LIMF . . . . .	61
5.2	Struttura del progetto . . . . .	67
5.3	SimulationHub scripts . . . . .	70
5.4	Takeoff . . . . .	74
5.5	Approach . . . . .	89
5.6	Landing . . . . .	92
<b>6</b>	<b>Conclusioni</b>	<b>95</b>
<b>A</b>	<b>Cockpit A320</b>	<b>101</b>
A.1	Control pedestal . . . . .	101
A.2	Overhead panel . . . . .	103
A.3	Instrument panel . . . . .	105
	<b>Bibliografia</b>	<b>113</b>

# Elenco delle tabelle

2.1	Procedura di <i>Preflight cockpit</i> descritta in [23] . . . . .	18
4.1	Engine Start procedure: nodi in WEAVR . . . . .	56
4.2	On Ground Emergency Evacuation procedure: nodi in WEAVR . . . . .	59
5.1	Step della procedura di decollo . . . . .	90
5.2	Step della procedura di atterraggio . . . . .	93



# Elenco delle figure

1.1	Simulatore di volo VR: esempio di collaborazione [9]	2
1.2	TXT Group logo	4
2.1	Classificazione degli ambienti XR. Fonte didattica: [22]	9
2.2	Architettura di un sistema VR [22]	10
2.3	Previsione di vendita dei dispositivi VR e AR [25]	11
2.4	Cone of experience di Edgar Dale	12
2.5	Training in XR. Fonte: PACE - TXT [21]	13
2.6	Confronto VRFS - PFS [29]	14
2.7	SBT con visore VR e guanti tattili - Leonardo Company [34]	16
2.8	Differenza nella percezione della distanza	17
2.9	Simulazioni VR per il training dei vigili del fuoco	20
2.10	Formazione operativa in VR in settore edile	22
2.11	VR per uso militare	23
2.12	Training VR per un astronauta dell'ESA	24
2.13	Sensazioni visive e aptiche combinate al training VR [53]	26
3.1	Schermata Unity	29
3.2	Cockpit Element	32
3.3	SimulationHub: overview [57]	33
3.4	Confronto A320 - A321 in X-PLANE	36
3.5	HTC VIVE Pro	38
3.6	Equipaggiamenti del flight deck presenti nel Simulatore A320 [22]	39
3.7	Architettura del Simulatore A320	40
4.1	Startup procedure	44
4.2	Welcome billboard	45
4.3	iPad First Screen Procedure	46
4.4	Procedure	47
4.5	Welcome Engine Start	47
4.6	Step 1: Engine mode selector	48

4.7	Step 2: Engine key . . . . .	49
4.8	Modifica dell'ECAM . . . . .	50
4.9	Step 3-4: Check ECAM display . . . . .	51
4.10	Step 5: Engine Manual Starter . . . . .	51
4.11	Step 6: Engine Master switch . . . . .	52
4.12	Step 7: Engine Manual Starter OFF . . . . .	53
4.13	Engine start completed . . . . .	54
4.14	Engine Start: Checklist . . . . .	54
4.15	Procedure starter . . . . .	55
4.16	On Ground Emergency Evacuation checklist . . . . .	56
4.17	Engine 2 Fire . . . . .	57
5.1	MCDU A320 . . . . .	62
5.2	INIT . . . . .	63
5.3	F-PLN . . . . .	65
5.4	PERF . . . . .	66
5.5	Pseudo-codice: funzione <i>main</i> . . . . .	71
5.6	File ICD utilizzati . . . . .	72
5.7	Procedura di decollo . . . . .	74
5.8	Welcome billboard . . . . .	75
5.9	Welcome takeoff procedure . . . . .	76
5.10	Step 0: Throttle 1/4 . . . . .	77
5.11	Step 1: Sidestick 1/2 forward . . . . .	78
5.12	Step 2: Parking Brake OFF . . . . .	79
5.13	Step 3: Throttle FLX . . . . .	80
5.14	Step 4: Sidestick Neutral . . . . .	82
5.15	Step 5: Rotate . . . . .	84
5.16	Step 6: Gear UP . . . . .	85
5.17	Step 7: AP1 ON . . . . .	86
5.18	Step 8: Throttle CLB . . . . .	87
5.19	Step 9: Remove Flaps . . . . .	87
5.20	Step 10: Ground Spoiler Disarm . . . . .	88
5.21	End procedure . . . . .	89
5.22	Pseudo-codice: funzione <i>takeoff</i> . . . . .	91
5.23	Procedura di atterraggio . . . . .	92
A.1	Control pedestal panel 3D . . . . .	101
A.2	Control pedestal panel . . . . .	108
A.3	Throttle A320 . . . . .	109
A.4	Sidestick A320 . . . . .	109
A.5	Overhead panel . . . . .	110

A.6 Instrument panel . . . . .	111
--------------------------------	-----



# Acronimi

**VR**

Virtual Reality

**AR**

Augmented Reality

**XR**

Extended Reality

**MR**

Mixed Reality

**PFD**

Primary Flight Display

**ND**

Navigation Display

**ECAM**

Electronic Centralized Aircraft Monitor

**MCDU**

Multi-function Control and Display Unit

**FMS**

Flight Management System



# Capitolo 1

## Introduzione

La simulazione ha sempre sostenuto la formazione [5]; essa si basa sul principio per cui è necessario fornire all'utente un'esperienza surrogata in anticipo rispetto ad un determinato evento. Per lo scenario futuro questo non è più sufficiente: gli esperti della simulazione dovranno sempre maggiormente lavorare ad un approccio più mirato alle competenze specifiche che devono essere addestrate. Come sottolineato da Kantowitz nel 1992 [6] le future simulazioni devono superare il vincolo dell'aspetto delle innovazioni tecnologiche per focalizzarsi sulle variabili psicologiche necessarie per il supporto delle competenze. In questo modo è possibile ottenere un paradosso: una simulazione meno fedele, ma più utile, basata su una comprensione di ciò che migliora più facilmente l'apprendimento delle abilità per trasferirle in un'ipotetica situazione reale [7]. La simulazione non viene utilizzata solo per la formazione [8]: un ruolo importante può essere ricoperto dalla progettazione. Le caratteristiche operative fisiche e cognitive di ogni sistema possono presentarsi prima nella simulazione, per poi essere modificate per adattare la progettazione a un livello ottimale.

I moderni cockpit degli aerei commerciali sono prodotti critici per la sicurezza, con interfacce utente avanzate e ben studiate [9]. A causa di severi standard di qualità hardware e software, i costi finali sono molto elevati e l'aggiunta di modifiche può portare a compromessi che si traducono in interfacce utente non impeccabili, quindi il loro sviluppo richiede un feedback su fattori umani, ergonomia, usabilità ed aspetti cognitivi in modo da non vincolare aspetti cruciali [10]. È necessario selezionare il giusto livello di realismo di simulazione e il metodo adeguato per i fattori umani, basati su questionari soggettivi (ad esempio NASA-TLX) o fisiologici (ad esempio frequenza cardiaca, comportamento dello sguardo, tasso di errore). La ricerca del simulatore adatto può avvenire valutando l'esigenza di usufruire di bassa fedeltà o di maggiore completezza e fedeltà della situazione reale; questa scelta influenza anche i costi.

La simulazione in realtà estesa rappresenta uno dei metodi di formazione più in tendenza degli ultimi anni. L'addestramento in XR<sup>1</sup> esprime un risultato diverso rispetto all'addestramento in ambiente di controllo non simulato [11]. Il valore della XR risiede nel fornire formazione in circostanze, caratterizzate ad esempio da situazioni di pericolo o di elevato costo, che escludono i metodi tradizionali. La riduzione dei costi è un evidente vantaggio per gli addestramenti che presuppongono un campo di informazione spaziale, come l'aviazione [12], unitamente all'assenza del pericolo nella simulazione. Il simulatore di volo in realtà virtuale presentato da Oberhauser [9] consiste in una cabina di pilotaggio virtuale funzionale collegata ad un software di simulazione disponibile in commercio (ad esempio X-PLANE). Per l'interazione con il *mock-up* virtuale sono necessari: un sistema di tracciamento, un dispositivo di output per la visualizzazione e degli hardware per l'interazione con la simulazione (controlli di volo base).



**Figura 1.1:** Simulatore di volo VR: esempio di collaborazione [9]

La Figura 1.1 evidenzia la presenza di due piloti in questo esempio di simulatore: il pilota di volo e il primo ufficiale, che ha il compito di gestione delle risorse dell'equipaggio [13].

La realtà virtuale ed aumentata hanno iniziato recentemente ad influenzare la vita quotidiana, in ambiti quali la programmazione di progetti, monitoraggio di progressi, formazione dei lavoratori, gestione di sicurezza, tempi e costi [14]. La VR<sup>2</sup>, in particolare, contiene aspetti del mondo fisico che non possono essere riprodotti agli studenti nelle tradizionali aule [15]: si presenta come la simulazione generata da un computer di un'immagine tridimensionale o di uno scenario che può interagire

---

<sup>1</sup>XR: Extended Reality

<sup>2</sup>VR: Virtual Reality

in modo reale o fisico con l'utente tramite un sistema digitale dedicato, composto da casco e guanti con sensori [16]. Essa permette l'addestramento tramite una simulazione che viene aggiornata in maniera efficiente e rapida, in base alla frequenza con cui si rendono disponibili nuove informazioni. Tuttavia, l'addestramento in XR non risolve tutti i problemi della formazione classica. Uno degli avvertimenti è il tasso di innovazione tecnologica [17]; la variabilità tra le tecnologie in uso rende necessario un continuo aggiornamento sull'efficacia dell'addestramento al simulatore basato su XR. Quando l'ambiente di apprendimento è sufficientemente diverso dall'ambiente in cui l'apprendimento viene misurato, la prestazione ne risente [18]. Godden e Baddeley [19] hanno sperimentato che i subacquei che memorizzavano liste di parole sulla terraferma, le ricordavano meglio sopra che sotto l'acqua; al contrario, imparare a gestire in sicurezza un serbatoio d'acqua subacqueo sulla terraferma anziché in profondità del mare potrebbe rivelarsi dannoso. Questa preoccupazione potrebbe essere estesa all'ambiente XR: quando le situazioni di addestramento basate sulla simulazione portano estremo beneficio hanno anche il più alto rischio di fallimento qualora lo stesso addestramento si rivelasse inadeguato [11].

## 1.1 Obiettivo della tesi

L'obiettivo del seguente lavoro di tesi consiste nel realizzare delle procedure operative per il *training* dei piloti. In particolare, si fa riferimento all'addestramento in realtà virtuale. Le procedure vengono realizzate tramite la piattaforma PACELAB WEAVR, integrata in ambiente *Unity*. L'obiettivo dell'implementazione delle procedure VR è di studiare l'efficienza con cui i piloti apprendono gli step da eseguire: questo metodo, in forte concorrenza con l'approccio didattico classico che precede l'addestramento al simulatore fisico, consente al pilota di vivere un ambiente immersivo in cui può ricevere feedback sensoriali utili ad un apprendimento più rapido.

Verranno implementate due tipologie di procedure.

1. **Procedure a terra.** Consentono al pilota di familiarizzare con la strumentazione di bordo del cockpit dell'Airbus A320, eseguendo operazioni *standard* o *abnormal*. In questo progetto vengono implementate le procedure di *Engine Start* ed *On Ground Emergency Evacuation*, che si aggiungono alle procedure, già implementate in precedenti lavori di tesi [1] [2] [3] [4], di *Before Start*, *After Start*, *After Landing*, *De-Icing*, *Parking* e *Securing the aircraft*.
2. **Procedure in volo.** Sono delle operazioni dinamiche che presuppongono un collegamento con un simulatore di volo. In questo progetto viene utilizzato X-PLANE, acquistato appositamente per il Computer principale del Laboratorio di Simulazione del Volo del Politecnico di Torino. Queste procedure prevedono

una comunicazione tra due *client*, ossia WEAVR e X-PLANE, che fanno riferimento al *server* di *SimulationHub*. Per permettere la connessione del client X-PLANE con il server e con l'altro client è stato sfruttato il toolbox open-source *X-Plane Connect*. L'obiettivo di queste procedure è quello di formare il pilota a compiere delle operazioni dinamiche, valutandone successivamente il grado di apprendimento. Le procedure in volo implementate sono *Decollo* e *Atterraggio*.

L'intero progetto di tesi è volto alla formazione del pilota. Infatti, tramite la realtà virtuale, è possibile ottimizzare i costi e i tempi dell'apprendimento. TXT Group sta percorrendo questa direzione per fornire ai propri clienti alcuni strumenti efficaci per la formazione dei piloti. Il training in VR verrà descritto per fornire una panoramica sulla modalità di utilizzo, sui vantaggi e gli svantaggi di questo metodo.

A conclusione del progetto di tesi, si intende effettuare un'analisi sui risultati ottenuti.

## 1.2 TXT Group

TXT Group è una società internazionale specializzata nella consulenza riguardante soluzioni digitali improntate all'innovazione [20]. Le linee guida della società riguardano sostenibilità (green aviation), cybersecurity e innovazione.

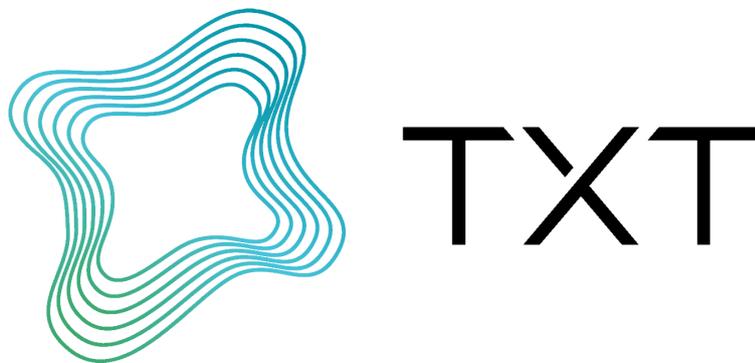


Figura 1.2: TXT Group logo

TXT nasce nel 1989 ed è quotata in borsa a partire dal luglio 2000. Il mercato dell'azienda è volto, tra gli altri, al settore *Aerospace & Defense*. L'azienda fornisce soluzioni software personalizzate ai propri clienti appartenenti all'industria aerospaziale, che supportano i processi ingegneristici durante il ciclo di vita del velivolo. In questi anni TXT ha intrapreso un percorso di crescita che portato all'inclusione di alcune aziende. Dal 2016, infatti, TXT ha incluso il gruppo PACE,

fondato nel 1995, che fornisce soluzioni software e strategie digitali per soddisfare le richieste dell'industria aerospaziale e aeronautica, mediante una vasta gamma di prodotti commerciali, i cui campi di applicazione sono molteplici. Per citare un esempio, la piattaforma PACELAB WEAVR, utilizzata in questo progetto di tesi, è stata adottata da *Leonardo Company* per le applicazioni di addestramento aeronautico e integrata nei simulatori della compagnia, per il dipartimento *Training and Simulation* della Divisione Velivoli. PACE e TXT si impegnano ad offrire al cliente una piattaforma di creazione di contenuti interattivi, con pacchetti che includono scenari di formazione realistici e completamente immersivi.

- **Progettazione preliminare** di aeromobili e sistemi. Le soluzioni di progettazione PACE si basano su una piattaforma software model-based che consente la scelta delle tecnologie opportune nelle fasi preliminari, fino al dimensionamento dei sottosistemi del velivolo e alla progettazione preliminare di componenti e superfici.
- **Configurazione** del prodotto. Il software consente una scelta degli interni degli aerei e delle cabine per ridurre al minimo l'errore e la latenza, consentendo al cliente di raggiungere un livello di gradimento opportuno.
- **Analisi e valutazione** del velivolo. Scegliere il giusto aeromobile per la giusta rotta richiede un compromesso tra autonomia di volo, dimensioni del velivolo e layout del carico dei passeggeri. Gli strumenti PACE di valutazione degli aeromobili forniscono proiezioni delle prestazioni tecniche ed economiche di vari velivoli, offrendo alternative ai pianificatori di flotte aeree e ai pianificatori di volo riguardo gli scenari operativi. Anche i costi dei velivoli dovranno essere inclusi nella valutazione delle alternative.
- **Training & Simulation.** Rappresenta il settore a cui vuole collegarsi il seguente elaborato di tesi. TXT fornisce ai propri clienti delle soluzioni software avanzate per la formazione dei piloti con i simulatori di volo. La realtà estesa viene utilizzata a tal proposito. PACE, infatti, è leader di mercato nelle soluzioni di realtà estesa per applicazioni industriali [21]; il suo prodotto di punta, PACELAB WEAVR, offre una soluzione aziendale completa per la formazione XR che consente ai clienti di creare un ambiente immersivo. Si tratta di una piattaforma progettata per gestire l'ambiente di realtà estesa, basata su *Unity*, piattaforma leader per la creazione e la gestione di contenuti 3D interattivi in real-time.
- **Flight operations.** Le soluzioni software per le operazioni di volo sfruttano le più recenti tecnologie aeronautiche per ottimizzare la pianificazione della missione e l'analisi post volo delle attività di manutenzione.

- **On-board software.** TXT è in grado di fornire ai clienti dei software che consentono di progettare e gestire i sistemi di bordo del velivolo.

### 1.3 Outline della tesi

Il seguente elaborato di tesi si divide in una parte bibliografica sulla realtà virtuale, una parte tecnico/sperimentale sulla trattazione del progetto condotto e una parte conclusiva sulla discussione e l'analisi dei risultati ottenuti con la simulazione VR.

Inizialmente si fornisce una panoramica sullo stato dell'arte della **realtà virtuale**. Si riporta un quadro generico su questo argomento, includendo la differenza con le altre tipologie di realtà estesa, eventuali scenari di sviluppi futuri, vantaggi e svantaggi di questa tecnologia. La realtà virtuale viene poi analizzata nell'ambito applicativo della formazione: dapprima si fornisce una descrizione sull'utilizzo del VR per il training dei piloti, analizzandone alcuni dati numerici di riferimento e successivamente si riportano esempi di training VR per altri settori, di cui vengono mostrati alcuni esempi di applicazione degli ultimi anni.

La descrizione di **software e hardware** utilizzati conclude la *literature review* di questo elaborato e costituisce uno strumento di collegamento con la parte tecnica affrontata nel progetto. Tra gli hardware descritti vengono inclusi gli strumenti del VR HTC *Vive Pro* e il simulatore dell'A320, presenti nel Laboratorio di Simulazione del Volo del Dipartimenti di Ingegneria Meccanica e Aerospaziale del Politecnico di Torino, sede dei lavori di questo elaborato. La descrizione dei software invece copre tutti gli strumenti informatici utilizzati in questo progetto; sono inclusi i software di realtà virtuale, il simulatore con il relativo connettore e l'ambiente di programmazione in linguaggio C++. Sia i software che gli hardware sono presentati inizialmente in chiave bibliografica e successivamente si descrive il modo in cui un determinato strumento viene utilizzato nelle fasi del lavoro.

La parte tecnico/sperimentale del progetto riguarda l'implementazione delle **procedure operative**. Queste ultime sono state implementate in collaborazione con il mio collega Giovanni Sgarra, con cui ho contribuito alla realizzazione del progetto di tesi, frutto dell'accordo di TXT Group con il Politecnico di Torino.

Le procedure **a terra** sono due:

- *Engine Start* descritta nel dettaglio nel seguente elaborato
- *On Ground Emergency Evacuation* descritta in dettaglio nell'elaborato di tesi di Giovanni Sgarra.

Le procedure **in volo** sono:

- *Decollo* trattato in questo elaborato
- *Atterraggio* trattato in dettaglio nell'elaborato di tesi di Giovanni Sgarra.

Le procedure in volo presuppongono anche la descrizione dell'integrazione in WEAVR degli output della simulazione del volo generata da X-PLANE, che avviene tramite il software *SimulationHub* fornito direttamente da TXT Group. La comunicazione tra il simulatore di volo e l'ambiente di realtà virtuale è un componente portante dell'intero progetto di tesi.

Vengono inoltre riportati in appendice i comandi del cockpit dell'A320 utilizzati durante l'esecuzione delle procedure da parte dell'utente, al fine di informare il lettore riguardo le funzionalità e il collocamento dei tasti all'interno dei pannelli della cabina di pilotaggio.

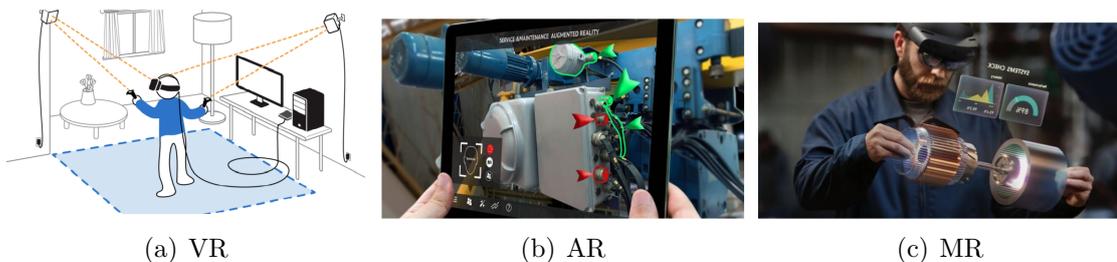


## Capitolo 2

# Stato dell'arte

Il mondo della simulazione si avvale di strumenti per la resa visuale molto più evoluti degli strumenti proiettivi. La realtà estesa diventa complementare nello sviluppo del simulatore moderno; essa include le seguenti definizioni [22].

- Realtà virtuale **VR**: operatore immerso in ambiente virtuale o artificiale che esclude la parte visiva del mondo fisico.
- Realtà aumentata **AR**: all'ambiente fisico vengono sovrapposti oggetti digitali per fornire agli utenti informazioni contestuali aggiuntive.
- Realtà mista **MR**: all'ambiente fisico vengono sovrapposti oggetti digitali o virtuali non passivi, ossia interagenti con l'utente e controllati direttamente da un operatore.



**Figura 2.1:** Classificazione degli ambienti XR. Fonte didattica: [22]

In Figura 2.1 si riporta un confronto visivo sulla classificazione dei diversi ambienti artificiali.

Di principale interesse per questo elaborato è l'ambiente di realtà virtuale, utilizzato per il training dei piloti. Si decide quindi di riportare una panoramica

generale sul VR: usi e principali vantaggi, cenni di architettura software e campi di applicazione, con un'attenzione particolare sulla formazione. Quest'ultima viene approfondita in maggiore dettaglio sia per il training dei piloti che per altre applicazioni; si descriveranno i principali vantaggi sui metodi di apprendimento riportati dalle fonti bibliografiche utilizzate. Tra gli utilizzatori degli ambienti immersivi spiccano alcuni dati interessanti [23]. La maggior parte degli utenti (circa il 69%) sfrutta il VR per intrattenimento, mentre solo il 30% ne conosce le potenzialità nel contesto dell'istruzione o del lavoro. La percezione degli utenti sulla possibilità di utilizzare la realtà virtuale per scopi didattici conferma un'efficacia di questo strumento all'84%; il 40% degli intervistati dichiara di aver avuto contatti con il VR non destinati a scopi di intrattenimento, mentre il 60% ha indicato la possibilità di piena divulgazione di realtà virtuale in futuro.

## 2.1 Realtà virtuale

Il concetto di realtà virtuale prevede l'immersione in un ambiente creato artificialmente da un computer che consente all'utente di ricevere delle sensazioni visive, sonore e tattili quanto più possibili affini alla realtà. Si può dunque definire come uno scenario generato dal computer in cui l'utente può interagire con la scena [24]. L'ambiente VR viene presentato all'utente attraverso degli appositi hardware: visore e cuffie per la vista e l'ascolto, guanti o controller per l'azione.

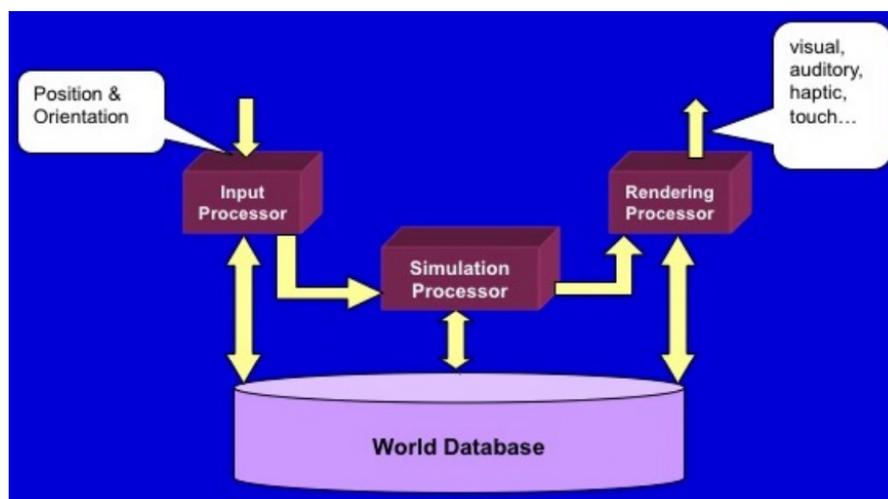


Figura 2.2: Architettura di un sistema VR [22]

La Figura 2.2 mostra l'architettura di un sistema a realtà virtuale. Sono presenti: un database esteso che rappresenta lo scenario digitale, un motore di simulazione che simula l'ambiente dal punto di vista dinamico (attingendo dal database e

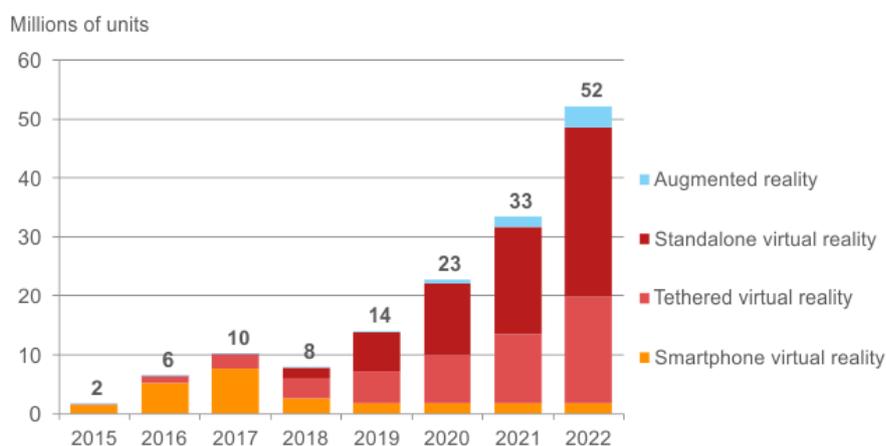
collegando il sistema di resa visuale e sonora) e, in input, le azioni che l'operatore esegue.

La realtà virtuale al giorno d'oggi prevede numerosi campi di applicazione; se ne riportano alcuni esempi.

- Campo di intrattenimento - gaming
- Controllo robotico - applicazioni tecniche
- Formazione ed educazione - simulazione di scenari operativi

Su quest'ultimo campo si concentra il seguente elaborato di tesi.

La realtà virtuale viene utilizzata principalmente per i vantaggi che ne conseguono. Principalmente, infatti, la VR consente all'utente una totale sensazione di *immersione*; il mondo reale viene temporaneamente escluso e si apre la strada di uno scenario fittizio che induce all'utente una eguale sensazione di realtà. Ovviamente, tutto questo dipende dal grado di accuratezza con cui la VR viene costruita: sono necessari opportuni requisiti di frame rate dello schermo, di latenza del sistema e di dettaglio grafico e sonoro (audio 3D). Secondo *CCS Insight* [25] la tecnologia AR e VR subirà un incremento vertiginoso, passando da poco meno di 8 milioni di dispositivi venduti nel 2018 a circa 52 milioni nel 2022. L'andamento previsto viene riportato in Figura 2.3, che offre un'istantanea della previsione globale di *CCS Insight* sulle unità di dispositivi.



**Figura 2.3:** Previsione di vendita dei dispositivi VR e AR [25]

Relativamente al campo della **formazione**, la realtà virtuale consente di offrire un miglioramento dell'insegnamento. Ad esempio, l'allievo risulterà molto più

pronto all'apprendimento se vengono stimolati tutti i suoi sensi, rispetto ad un metodo di insegnamento basato solamente su lettura e ascolto.

Secondo gli studi condotti, la realtà virtuale permette la conservazione della conoscenza fino al 400% in più. In questo modo si ottimizza il tempo che i piloti impiegano nello studio e si migliora l'apprendimento.

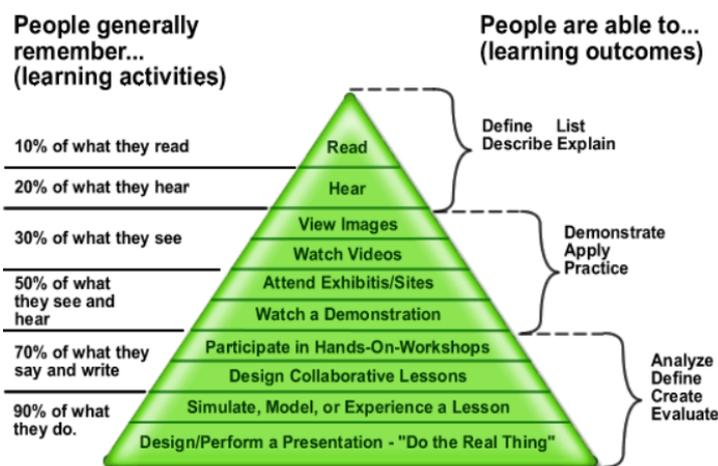


Figura 2.4: Cone of experience di Edgar Dale

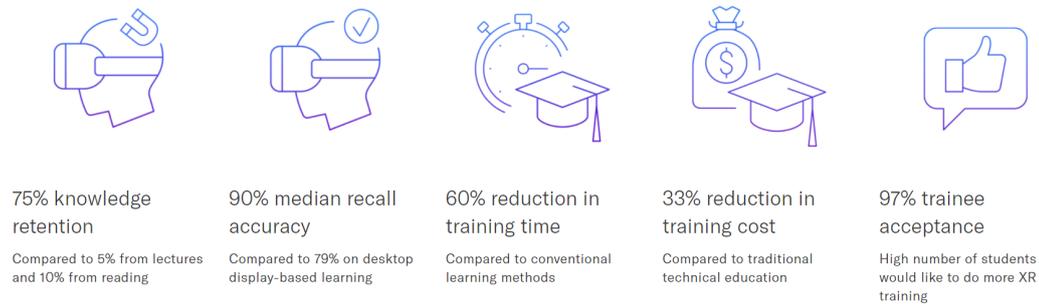
La Figura 2.4 si riferisce al Cone of experience di Edgar Dale [26]: esso evidenzia come, durante il processo di apprendimento, gli studenti riescano a ricordare generalmente il 10% di ciò che leggono, in confronto al 90% di ciò che eseguono nella pratica.

La formazione in VR offre dunque vantaggi significativi ai settori industriali in cui è richiesto un apprendimento esperienziale. Alcune variabili vengono considerate da Kaplan [11] nella valutazione di questo processo di apprendimento:

- **Immersività.** Gli ambienti virtuali vengono comparati con condizioni non simulate. L'immersività è la variabile che stabilisce il grado di resa coinvolgente di una situazione virtuale.
- **VR e controllo.** Si stabilisce l'immersività come un fattore e si confronta un ambiente immersivo rispetto ad un ambiente di formazione non immersivo.
- **Pre e post training.** Include qualsiasi confronto tra la performance di un individuo (o di un gruppo) prima e dopo l'allenamento in XR. Questa variabile esamina gli eventuali miglioramenti post-training.

L'analisi si sviluppa con l'elaborazione di dati riguardanti compiti cognitivi, fisici e misti e i risultati sono interessanti: l'immersività non genera significativi

miglioramenti pre e post training, ma sui compiti fisici la formazione XR si dimostra più adatta [11].



**Figura 2.5:** Training in XR. Fonte: PACE - TXT [21]

La Figura 2.5 evidenzia importanti riscontri numerici sull’efficienza della formazione in ambiente immersivo. In particolare, si nota una riduzione dei costi di formazione del 33% rispetto alla formazione tecnica tradizionale e una riduzione del 60% del tempo di formazione rispetto ai metodi di apprendimento convenzionali.

Gli studi di TXT propongono una previsione di accrescimento delle tecniche di formazione XR del 70% entro il 2022.

Il ruolo centrale che l’esperienza gioca nel processo di apprendimento differenzia la formazione basata sulla simulazione dalla formazione basata sulle lezioni [27]. Il modello di valutazione della formazione ideato da Kirkpatrick [28] misura le componenti di reazione, apprendimento, comportamento e risultati di un programma di formazione. Ne emerge che la principale carenza dei metodi tradizionali consiste nel non riuscire a misurare sensazioni, emozioni, esperienze e reazioni dei tirocinanti in misura pari alla simulazione.

## 2.2 Realtà virtuale per addestramento piloti

La simulazione del volo svolge un ruolo cruciale nella formazione di un pilota professionista. La sua fruizione può avvenire attraverso l’utilizzo di un simulatore di volo fisico o virtuale. L’articolo [29] presenta un’importante comparazione tra i simulatori fisici PFS<sup>1</sup> e quelli in realtà virtuale VRFS<sup>2</sup>. Viene infatti presentato uno studio con 11 partecipanti che ha come obiettivo finale quello di stabilire se un VRFS può integrare o sostituire un PFS nella formazione del pilota e nella sua familiarizzazione con la cabina di pilotaggio di un Boeing 737-800 (Figura 2.6).

<sup>1</sup>PFS: Physical Flight Simulators

<sup>2</sup>VRFS: Virtual Reality Flight Simulators



**Figura 2.6:** Confronto VRFS - PFS [29]

I PFS certificati dalle autorità dell'aviazione presentano alcuni **svantaggi**:

- **Costi notevoli.** L'analisi costo-efficienza per l'addestramento militare di Orlansky [30] sottolinea come il costo di un PFS adeguatamente accurato può avvicinarsi all'ordine di grandezza di alcune unità di milioni di dollari, a cui sommare altre centinaia di dollari all'ora per l'addestramento del pilota con personale specializzato.
- **Inflessibilità.** I PFS riescono a simulare solo un tipo di aeromobile e a formare il pilota ad una cabina prefissata.

Per ovviare ai succitati inconvenienti, negli ultimi anni le scuole di volo e le compagnie aeree stanno effettuando oculute valutazioni sulla realtà virtuale, tramite cui è possibile sfruttare la tecnologia odierna per fornire notevole qualità per il Training dei piloti. Auer e Gerken [29] hanno presentato interessanti risultati nella loro analisi. Le task di lettura della strumentazione di bordo sono state condotte con successo (oltre il 99%) sia per i PFS che per i VRFS. Gli utenti hanno completato positivamente le procedure di check in entrambi gli ambienti, sebbene nel VRFS fosse assente il feedback aptico. Sono stati anche presentati alcuni inconvenienti del VRFS: sforzo mentale e fisico maggiore dovuto al peso della strumentazione VR e alla consapevolezza dell'utente di operare in ambiente virtuale, lievi malessere da simulazione virtuale e maggiori tempi di completamento delle operazioni.

Le task condotte avevano una crescente difficoltà:

- *Engine start* - manipolazione di 5 interruttori
- *After engine start* - manipolazione di 8 interruttori
- *Engine shutdown* - manipolazione di 14 interruttori.

In base alle risposte degli utenti nel questionario finale, volto a valutare le variabili di riferimento per il confronto tra PFS e VRFR, i risultati finali mostrano

una preferenza ancora maggiore dell'utente nell'utilizzare un simulatore fisico. Sembra tuttavia promettente un'esplorazione ancora più approfondita dell'utilizzo della realtà virtuale nell'addestramento dei piloti.

Oberhauser, nel 2018, indaga sulla fedeltà di un VRFS rispetto ad un simulatore convenzionale, valutando tempi di risposta del pilota, carico di lavoro, deviazione dal percorso di volo e malori dovuti alla simulazione [31]. I risultati evidenziano come i futuri scenari di test debbano essere creati sulla base di una maggiore fedeltà dello scenario, per consentire all'utente di avere una interazione più coinvolgente: i ritmi devono essere adattati e i tempi di esposizione limitati per giudicare accettabile un maggiore carico di lavoro dell'utente in VR. Sebbene la fedeltà di un VRFS sia inferiore a quella di un PFS, il VRFS può presentarsi come uno strumento valido per lo studio dei fattori umani.

Un altro interessante confronto tra la formazione immersiva e in *real-life* viene condotto da una ricerca dell'Università di Tolosa ISAE Supaero [32] per aiutare la formazione degli allievi piloti. 10 partecipanti hanno imparato una checklist di azioni ed ispezioni visive pre volo e di avvio motore di un velivolo leggero. Un gruppo è stato istruito in ambiente immersivo con video a 360°, mentre l'altro ha ricevuto insegnamenti di persona da un istruttore. Dopo una breve pausa, i gruppi hanno eseguito la checklist sull'aereo reale. La valutazione della performance avviene sulla base del tempo impiegato per trovare un oggetto e sulla realizzazione della task associata. I risultati preliminari sono promettenti per l'ambiente immersivo, in quanto mostrano che è possibile imparare checklist complesse usando solo video immersivi.

Dunque, mentre Auer [29] e Oberhauser [31] propongono la visualizzazione del cockpit in ambiente VR, Peysakhovich [32] utilizza procedure di visione video a 360° tramite un *head mounted display* che inibisce i benefici della memoria muscolare, apprezzabili maggiormente in ambiente di realtà virtuale.

Vengono ora presentate delle applicazioni reali sull'utilizzo della **realtà virtuale** per la formazione dei piloti.

Nel novembre 2020 la compagnia aerea di bandiera dei Paesi Bassi, **KLM**, ha introdotto lo strumento di realtà virtuale per consentire la formazione di piloti di velivoli Embraer [33]. I **vantaggi** sono molteplici:

- Maggiore accessibilità al sito
- Minor tempo di familiarizzazione con la cabina di pilotaggio, in modo da ottimizzare il tempo impiegato nella formazione al simulatore fisico
- Riduzione dei costi

Ovviamente bisogna citare delle controindicazioni da considerare nell'approccio a questa tipologia di training. Tra questi, in particolare, l'impossibilità, a costi accessibili, di riprodurre nella totalità un feedback sensoriale completo: ad esempio, feedback olfattivi, tattili (in maniera molto accurata) o forze di inerzia sono sicuramente difficilmente riproducibili nella simulazione virtuale.

Il programma di addestramento della compagnia prevede più applicazioni. Nella cabina di pilotaggio il pilota può immergersi in maniera interattiva nell'immagine 3D generata dai computer, può monitorare i pannelli e memorizzare la posizione dei comandi; successivamente un video di istruzioni consente al pilota, posizionato sul sedile di pilotaggio, di assistere ad un volo a 360°. Infine il pilota può effettuare una vera e propria ispezione del cockpit e della cabina passeggeri, camminando nella stanza in cui viene proiettata la simulazione in VR. Attualmente KLM sta valutando la possibilità di ottenere una certificazione EASA per la formazione in VR dei piloti da sostituire all'istruzione teorica in classe con libri e fotografie.

Nell'aprile 2021 EASA ha rilasciato la prima certificazione di un FSTD (Flight Simulation Training Device) in VR.

La realtà virtuale per l'addestramento dei piloti viene utilizzata anche da **Leonardo Company**, che recentemente ha introdotto l'utilizzo del velivolo da addestramento Aermacchi M-345 [34].



**Figura 2.7:** SBT con visore VR e guanti tattili - Leonardo Company [34]

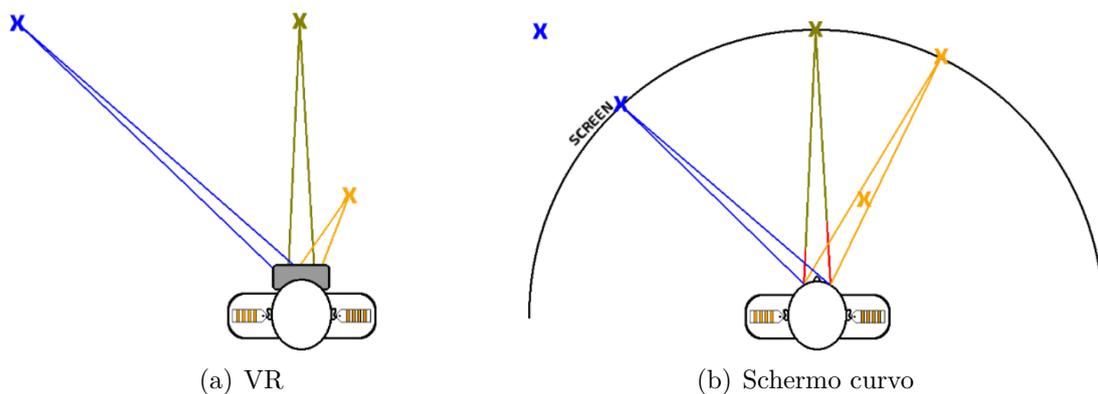
Il Simulation Based Training (SBT, Figura 2.7) di tale velivolo è dotato di visore e guanti tattili per la realtà virtuale immersiva. I sistemi installati simulano le caratteristiche aeromeccaniche, avioniche e prestazionali del velivolo; la realtà

virtuale fornisce stimoli sensoriali (visivi, sonori e tattili) al pilota in maniera fedele al volo reale. Leonardo porta un dato interessante sul risparmio dei costi relativo alla formazione VR: nel triennio 2018-2020 l'impiego di simulatori VR per il training dei piloti ha permesso il risparmio di circa 116.000 tonnellate di CO2.

Nel 2019 la **US Air Force** ha formato una classe di 30 piloti addestrati con visori VR e biometria AI in sostituzione ai simulatori di volo tradizionali [35]. 13 piloti sono stati certificati in soli quattro mesi, con un risparmio di tempo di circa otto mesi rispetto alla procedura di certificazione classica. Il costo dell'addestramento VR per unità è stato di circa 1000\$, a scapito di circa \$4.5 milioni per un simulatore full-flight. La biometria AI permette di monitorare la frequenza cardiaca e la dilatazione delle pupille, consentendo di stabilire il grado di immersione dei piloti nell'esperienza di apprendimento. Inoltre, in soli 10 secondi, il pilota è in grado di cambiare la cabina di pilotaggio su cui sta effettuando il training. Si evidenziano quindi i vantaggi dal punto di vista dei costi e dei tempi.

**TXT Group**, mediante la piattaforma di produzione propria PACELAB WEAVR con tecnologia XR, propone delle soluzioni di training in realtà virtuale per piloti e manutentori [36]. L'obiettivo rimane quello di fornire contenuti interattivi da sostituire alle modalità di formazione classica, grazie ad un elevato dettaglio grafico presente negli scenari simulativi.

Tra gli obiettivi dell'addestramento al volo per i piloti vi è quello di riuscire a valutare correttamente le distanze. Nei simulatori di volo fisici tale valutazione risulta più difficoltosa rispetto al simulatore VR a causa della presenza di uno schermo posto a distanza fissa dallo sguardo dell'allievo e inadatto a simulare correttamente lo scenario 3D.



**Figura 2.8:** Differenza nella percezione della distanza

In Figura 2.8 si evidenzia la differenza nella percezione della distanza tra un ambiente VR (in cui le distanze risultano correttamente percepibili) e uno schermo curvo, tipicamente utilizzato per i simulatori di volo tradizionali, in cui i 3 bersagli sono alla stessa distanza, poiché proiettati sullo schermo [26].

Anche il campo visivo deve essere correttamente simulato. Nei simulatori di volo tradizionali il campo visivo non supera i 180°, mentre l'applicazione di un visore VR consente di allargare tale campo a 360° consentendo, ad esempio, di simulare correttamente l'ambiente durante una virata.

Il cockpit non è fisico ma è ricostruito mediante immagini 3D del sistema digitale. Il pilota si muove nell'ambiente e usa dispositivi di interazione per controllare i componenti: può agire sugli switch dei pannelli, sulla barra o sulla manetta. Queste azioni sono eseguite con interfacce completamente digitali.

L'articolo [23] riporta una applicazione di addestramento in ambiente VR per gli allievi piloti del velivolo Socata TB-9 "Tampico". Il formatore supervisiona il processo di formazione, modera o modifica lo scenario in corso e gli studenti seguono delle procedure indipendenti durante le varie fasi del training. La procedura presentata, rinominata *Preflight - cockpit* consiste in 6 step, riportati in Tabella 2.1.

<i>Cockpit element</i>	<i>Status</i>
Ignition Switch	OFF
Control lock	REMOVED
Master switch	ON
Voltmeter	CHECKED
Fuel quantity	CHECKED
Trimmer	SET

**Tabella 2.1:** Procedura di *Preflight cockpit* descritta in [23]

Ogni volta che un pulsante viene premuto, si attiva la riproduzione sonora che ne imita l'azionamento. La checklist completa viene mostrata frontalmente al pilota, con dei *tick* che indicano che l'attività è stata completata con successo. Gli hardware VR della procedura presentata in [23] comprendono occhiali e controller; questi ultimi vengono visualizzati in realtà virtuale come guanti. Gli step della procedura devono essere eseguiti in maniera corretta. L'analisi di questo training viene illustrata in un questionario finale proposto a diciassette utenti, di età compresa tra 21 e 23 anni, che eseguivano la procedura una volta in VR per una durata massima di 5 minuti. Alla richiesta del livello di difficoltà di movimento e manipolazione del controller il risultato è stato di 2.26 su una scala 1÷5; un valore così basso è attribuibile a un livello di conoscenza poco adeguato della strumentazione utilizzata.

La totalità degli utenti ha stabilito che le istruzioni visualizzate sono risultate sufficienti all'esecuzione della procedura guidata, 11 utenti hanno affermato di essere in grado di riprodurre la procedura su un aereo reale e 15 utilizzatori hanno dichiarato l'efficienza dell'apprendimento in VR. Conclude il questionario la richiesta di eventuali effetti fisiologici indesiderati attribuibili al VR, a cui 3 utenti hanno risposto affermativamente.

L'avvento della simulazione in VR ha dunque portato numerosi benefici al settore della formazione e, in particolare, del training dei piloti. Tuttavia, secondo uno studio del 2017 [9] la fedeltà di un simulatore di volo VR rimane inferiore rispetto ad un simulatore tradizionale. Tutti i controlli del VR sono renderizzati digitalmente, dunque incapaci di fornire al pilota le necessarie skills tecniche e l'abilità di gestione delle sensazioni di stress durante il volo. Di conseguenza la simulazione VR si configura come un ottimo strumento di formazione dei piloti aeronautici, ma non ancora capace di sostituire in toto l'esperienza di un simulatore full-flight.

Il futuro del training VR per i piloti può essere dunque vincolato ad un successivo step tecnologico che mira al raggiungimento della *Fused Reality* [35]. Si tratta di un progetto a cui stanno lavorando l'*Armstrong Flight Research Center* della NASA e *System Technology* che consente ai piloti di effettuare training in volo mediante dei visori *head mounted*. L'esperienza di volo VR infatti è in grado di convincere maggiormente il cervello umano della realtà della situazione rispetto al posizionamento sul sedile.

## 2.3 Realtà virtuale per altri tipi di training

Le tecnologie di apprendimento immersivo presentano efficacemente un ambiente esperienziale, interattivo e coinvolgente per l'apprendimento tecnico e non solo [27].

L'uso di simulatori che utilizzano sistemi di realtà virtuale per formare professionisti in diversi settori è stato menzionato da diversi decenni [37]. Il tema del training in VR coinvolge molteplici settori lavorativi, di cui verranno citati alcuni esempi nei successivi paragrafi.

### 2.3.1 Addestramento vigili del fuoco

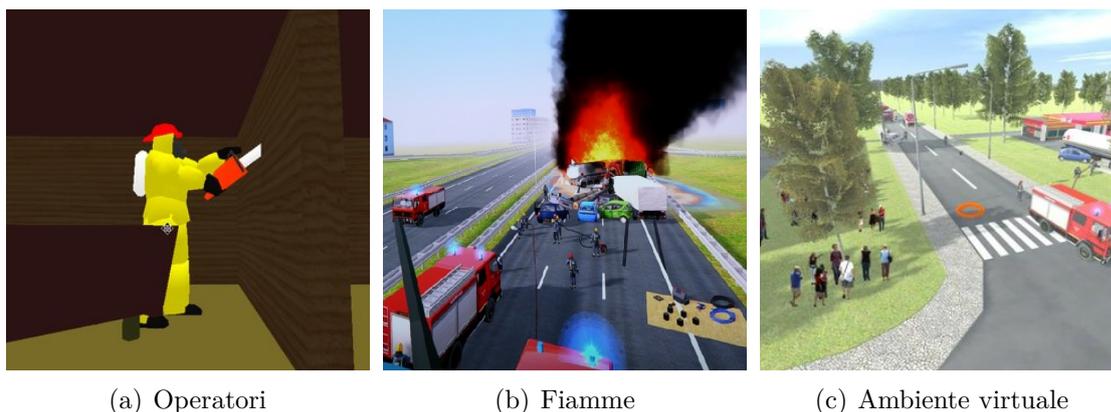
Con l'utilizzo della realtà virtuale è possibile simulare situazioni di addestramento dei vigili del fuoco in ambienti pericolosi: acqua, aria, vicinanza a fiamme e ghiaccio.

A partire dal secolo scorso hanno iniziato a verificarsi dei casi di addestramento di questo tipo; recentemente è solito effettuare dei test di addestramento in ambiente fisico e virtuale, confrontando i risultati ottenuti, che spesso evidenziano la maggiore efficacia del training VR.

I vantaggi inoltre sono numerosi: riduzione dei costi, maggiore sicurezza, eterogeneità delle situazioni da simulare, maggiore velocità di apprendimento, maggiore autonomia, ripetizione delle procedure di addestramento [38].

L'applicazione del training VR per i vigili del fuoco esplicitata nella fonte bibliografica citata prevede una fase iniziale di tutorial, in cui si comprendono le interazioni con gli oggetti, una fase di apprendimento, per imparare le procedure e una fase di valutazione finale per avere un riscontro sull'efficienza dell'apprendimento. I test di valutazione vengono effettuati su un numero sufficiente di persone per avere un campionamento vasto dei risultati. La simulazione in VR effettuata prevede l'immersione in un edificio in cui viene simulato un incendio; l'operatore deve localizzare l'aula in cui si verificano le fiamme utilizzando la mappa dell'edificio, raggiungere l'aula, permettere l'evacuazione all'esterno degli alunni, tornare in aula per spegnere l'incendio con l'estintore e raggiungere in sicurezza l'esterno dell'edificio.

Al Georgia Tech di Atlanta viene sviluppato l'ambiente di realtà virtuale di addestramento al comando dei vigili del fuoco [39]. L'utente dell'ambiente virtuale è un apprendista comandante che istruisce squadre di vigili del fuoco per eseguire azioni volte allo spegnimento di incendi virtuali. La corretta sequenza della procedura estinguerà con successo le fiamme con la minor quantità di pericolo per gli operatori e per la casa. Questo sistema consiste di tre principali componenti: ambiente di realtà virtuale, interfaccia di comando e il simulatore dinamico della fiamma. All'interno dell'ambiente virtuale sono presenti: una casa arredata, un camion di pompieri e vari strumenti anti-incendio dei vigili del fuoco (Figura 2.9). Per spegnere le fiamme, l'utente impartisce dei comandi ai vigili del fuoco virtuali tramite un operatore che traduce i comandi verbali dell'apprendista in comandi sull'interfaccia grafica di comando.



**Figura 2.9:** Simulazioni VR per il training dei vigili del fuoco

### 2.3.2 Training nel settore edile

Nell'ottobre del 2019 l'Università di Siena ha presentato il progetto pilota per il training in VR nel settore edile, grazie alla collaborazione tra l'Ente Senese Scuola Edile e il Laboratorio VR dell'Università [40]. Il progetto prevede la formazione, in materia di sicurezza nei luoghi di lavoro, per i training abilitanti ai lavori in quota nel settore edile. Attraverso il training in VR è possibile replicare una serie di contesti particolarmente rischiosi da riprodurre nella vita reale, restituire un feedback realistico che stimoli i sensi dello staff per avere una percezione non uguagliabile dai soli studi teorici. Le simulazioni in realtà virtuale rappresentano esperienze interattive, consentono all'utente di testare e riconoscere le sue reazioni in situazioni potenzialmente pericolose ed insegnano a gestire il rischio in totale sicurezza. Le situazioni di lavoro in quota vengono simulate attraverso una guida indicativa per la gestione degli eventi e la sperimentazione, soltanto virtuale, di conseguenze di particolari errori.

Il progetto prevede un approccio psicologico, finalizzato ad assumere il controllo emotivo delle lavorazioni in quota, unitamente ad un approccio tecnico, utile a comprendere le procedure di sicurezza.

Le simulazioni in VR sono effettuate nel Laboratorio di Realtà Virtuale, fornito di visori a caschetto e camera immersiva gestita dal computer che proietta immagini e scenari 3D.

Le tecnologie AR e VR potrebbero consentire l'accelerazione della realizzazione del cantiere e un accrescimento del livello di sicurezza [41]: nel settore edile viene utilizzato un modello 3D da consulenti, appaltatori e consumatori tramite il quale si interagisce con i progetti del mondo reale, valutandone difetti, rischi e potenziali conseguenze prima del loro effettivo manifestarsi.

Il lavoro di ricerca [14] si concentra sull'opportunità di ricavare dei vantaggi dalla tecnologia VR. La qualità della costruzione e la sicurezza dei lavoratori dipendono principalmente dalla corretta formazione dei lavoratori [42].

La Figura 2.10 mostra un lavoratore addestrato in VR all'utilizzo di un escavatore pesante; la formazione in VR volta all'apprendimento dell'utilizzo della gru, dell'escavatore e di altre attrezzature da costruzione è una tecnologia collaudata ed efficace [43]. Infatti, poiché il settore edile possiede un elevato tasso di rischio, è necessario garantire la sicurezza assoluta; le tecnologie di formazione in VR giocano un ruolo cruciale nel fornire maggiore consapevolezza durante l'apprendimento, per consentire la riduzione del numero di incidenti, dei tempi di costruzione e dei costi associati. La realtà virtuale contribuisce anche ad una migliore gestione dei difetti.

Anche se VR e AR si presentano come strumenti di formazione quasi essenziali nel mondo industriale, ci sono degli svantaggi da considerare per la fruizione di queste tecnologie. Infatti, esse sono caratterizzate da costi difficilmente affrontabili dalla totalità delle aziende; le limitazioni abbracciano anche l'eventuale indisponibilità



**Figura 2.10:** Formazione operativa in VR in settore edile

delle tecnologie e la loro manutenzione. Molti ricercatori ritengono, tuttavia, che i principali svantaggi saranno superati nel corso dei prossimi anni [14].

### 2.3.3 Addestramento nel settore militare

Le sessioni in realtà virtuale e aumentata addestrano alla battaglia in maniera efficace in quanto riproducono fedelmente scenari di guerra, con stress e costi ridotti (risparmio su carburante e munizioni) e sicurezza elevata, data dall'assenza di pericoli fisici. L'esercito britannico sta conducendo esperimenti in VR in collaborazione con la PLEXTEX [44] specializzata nell'addestramento del personale medico in situazioni belliche; tuttavia il Paese con l'addestramento VR più avanzato è l'USA, che impiega tecniche di *Dismounted Soldier Training System* [45] per offrire armi realistiche ed effetti sensoriali per trasmettere i movimenti delle reclute, permettendo il monitoraggio degli istruttori. La Marina Militare americana utilizza anche la realtà aumentata e la realtà mista, con i progetti *Augmented Immersive Team Training* e *Battlefield Exploitation of Mixed Reality*, per sfruttare tecnologie non immersive, basate sulla visualizzazione parziale dell'ambiente esterno.

In Italia il training VR ad uso militare viene sfruttato dalla Marina Militare per sviluppare le capacità di apprendimento delle procedure e degli strumenti comunicativi. Inoltre, il Corpo Forestale sfrutta la simulazione in VR per la formazione di personale esperto nel contrastare gli eco-reati.

Il Politecnico di Torino, nell'ottobre del 2019 [46] in collaborazione con Fondazione Links e Aeronautica Militare, ha realizzato un simulatore VR che riproduce fedelmente la base militare di Verona per consentire l'addestramento individuale o

in team e con la possibilità di comunicare con i colleghi e interagire con l'ambiente. Il progetto è stato presentato a *Toxic Trip*, evento internazionale organizzato dalla NATO presso l'Accademia Militare di Villafranca di Verona, con l'obiettivo di rafforzare la prontezza alle situazioni di emergenza relative a rischi biologici, chimici e nucleari.

Le esercitazioni militari tradizionali sono lunghe e costose. Se addestrati con sistemi militari virtuali, le esercitazioni di guerra possono essere condotte dai militari in costi e tempi ridotti [47].



(a) Addestramento

(b) Scenario

**Figura 2.11:** VR per uso militare

Attraverso la creazione di un campo di battaglia virtuale (Figura 2.11) le parti in guerra sono coinvolte in un esercizio di confronto reale basato su situazioni e cambiamenti interni alla realtà virtuale. Nella fase di combattimento, il sistema di simulazione VR può eseguire simulazioni e analisi ripetute in base al piano decisionale stabilito dal comandante al fine di fornire la migliore scelta del piano di guerra. Inoltre, la simulazione VR offre spazio per intraprendere delle manovre di battaglia difficilmente replicabili con attrezzature reali [48].

### 2.3.4 Addestramento astronauti

La realtà virtuale per l'addestramento degli astronauti ricopre un ruolo cruciale. Le massime agenzie spaziali internazionali (NASA, SpaceX, ESA ed Australian Space Agency) investono molto in questa tecnologia, abbinandola a tecniche di simulazione subacquea [49] per sfruttare l'effetto di una spinta gravitazionale

inferiore. La simulazione si svolge con l'utente che, indossando un visore VR, visualizza correttamente gli ambienti spaziali, mentre il tutto viene immerso in una piscina, consentendo la sperimentazione di assenza di peso. Le procedure disponibili sono relative a sessioni di training individuali o di gruppo e riguardano molteplici operazioni per consentire una formazione completa del personale.

Durante il periodo di lockdown internazionale causato dall'emergenza sanitaria del *Covid-19* del 2020, gli astronauti hanno potuto continuare la propria formazione in remoto, utilizzando la realtà virtuale. Attraverso i visori VR è stato possibile ricostruire in 3D la cabina di pilotaggio in qualità 4K per effettuare operazioni di pre-lancio, attracco, sganciamento e atterraggio.

A partire dal settembre 1993 più di 100 membri del team di controllo di volo della NASA sono stati addestrati in ambiente immersivo per valutarne le potenzialità al fine di effettuare manutenzione e riparazione del telescopio *Hubble* [50]. Sono stati creati 6 diversi scenari di addestramento usando due software sviluppati dalla NASA, un'applicazione audio e un programma di modellazione, quest'ultimo utilizzato per costruire oggetti 3D per animazioni e simulazioni concettuali, con un livello di dettaglio grafico sempre crescente unitamente alla complessità degli oggetti. Gli scenari di addestramento contenevano anche un modello della baia del payload dello Space Shuttle e le procedure di manutenzione pianificate per step; sono stati sviluppati 6 scenari di Extra Vehicular Activities (EVA) previste per la missione, i cui step procedurali specifici sono stati determinati in base alle procedure utilizzate per l'addestramento nella Weightless Environment Training Facility.



**Figura 2.12:** Training VR per un astronauta dell'ESA

In Figura 2.12 è illustrato un istante di addestramento di Alexander Gerst,

astronauta dell'ESA, a Houston, in Texas, per la preparazione alla sua missione sulla ISS del 2014.

### 2.3.5 Realtà virtuale e medicina

In ambito medico la realtà virtuale viene utilizzata principalmente per riabilitazione motoria e/o cognitiva, terapia di disturbi e apprendimento mediante simulazione [51].

La *cyberterapia* annovera tra i suoi vantaggi la capacità di stimolare la multisensorialità del paziente, l'adattamento delle difficoltà degli esercizi in maniera proporzionale alle abilità raggiunte e il monitoraggio degli indici prestazionali. L'Istituto Auxologico Italiano (IRCCS) dispone di due stanze virtuali immersive in cui si sperimenta la *Telepresenza Immersiva virtuale - TIV* per simulare disturbi cognitivi, motori o psicologici per promuovere l'esercizio delle funzionalità compromesse. La Casa di Cura *Villa Santa Chiara* di Verona utilizza la realtà virtuale per il trattamento dell'anoressia nervosa, trattata con protocolli riabilitativi.

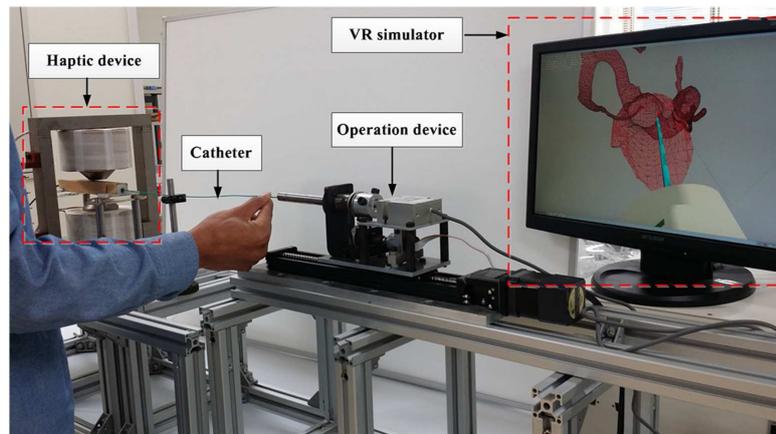
Anche la terapia del dolore costituisce un campo di applicazione della VR. Secondo alcuni studi la realtà virtuale può avere effetti analoghi ad antidolorifici oppiacei, riducendo di circa il 24% il dolore percepito.

In ambito medico il campo della formazione si avvale degli strumenti della realtà virtuale. In chirurgia l'allievo può essere addestrato con gli strumenti in VR prima di procedere all'operazione. In campo radiologico si può riscontrare l'utilizzo della realtà virtuale per elaborazioni di immagini TAC e RMN.

Le simulazioni mediche in realtà virtuale offrono un'esperienza di apprendimento su misura che può essere standardizzata o personalizzata secondo uno stile differente dal metodo di apprendimento classico [52]. Queste simulazioni facilitano l'apprendimento auto-diretto, permettendo agli studenti di sviluppare le competenze al proprio ritmo e di replicare scenari specifici per colmare le carenze in un ambiente sicuro. Tuttavia, mentre il realismo grafico è elevato, molti simulatori non sono in grado di fornire feedback aptici e tessuti interattivi sufficientemente adeguati.

Nella telechirurgia, poiché i chirurghi non possono sentire direttamente la resistenza all'inserimento degli strumenti, vengono sviluppati dei fornitori di forza aptica [53]. Il sistema di formazione basato sulla realtà virtuale permette di sfruttare la ripetibilità delle pratiche, di costruire modelli di vascolatura e di mostrare immagini visive dell'area chirurgica in simultanea. Le sensazioni visive e tattili combinate con il sistema di addestramento in VR mirano a migliorare la coordinazione occhio-mano dell'utente.

In Figura 2.13 viene riportato un esempio di sistema di addestramento VR con attenuazione del trauma da collisione per il funzionamento sicuro del cateterismo. Quattro utenti sono stati invitati a testare una sessione di formazione per 5 giorni. I



**Figura 2.13:** Sensazioni visive e aptiche combinate al training VR [53]

risultati sperimentali hanno evidenziato una riduzione della distanza di *impingement* di  $0.6mm$  ed un aumento fino al 50% della percentuale di punta del catetere in area sicura. Il sistema di addestramento VR in questo caso garantisce una performance più elevata degli utenti, accompagnata da un'acquisizione delle competenze veloci e stabili.

# Capitolo 3

## Software e hardware

L'ambiente di realtà virtuale utilizzato in questo lavoro di tesi simula il cockpit del velivolo Airbus A320, in cui l'utente può immergersi per apprendere le procedure operative *standard* o *abnormal*.

La descrizione della strumentazione utilizzata in questo progetto di tesi include una componentistica hardware e software.

### 3.1 Software

Nella componentistica software vengono descritti i principali strumenti informatici utilizzati per la finalizzazione del lavoro. Sono stati utilizzati diversi software, poiché il progetto verte sulla comunicazione dell'ambiente di realtà virtuale con la simulazione di volo. Di conseguenza vengono trattati, in ordine, i software relativi alla fruizione della realtà virtuale, alla simulazione del volo e al collegamento dei *client* utilizzati.

Tutti i software elencati sono installati sul PC utilizzato nel Laboratorio di Simulazione del Volo del Politecnico di Torino; alcuni sono *open source* e quindi direttamente installati tramite download dal sito ufficiale, altri sono stati forniti direttamente da TXT Group con la relativa licenza di utilizzo, altri ancora sono stati acquistati appositamente per il progetto di tesi da store ufficiali.

#### 3.1.1 SteamVR

SteamVR [54] è un software di realtà virtuale, sviluppato da *Valve* e HTC, che consente la gestione dell'hardware VR. Si tratta di una piattaforma che viene installata dal sito ufficiale. L'hardware utilizzato, HTC *VIVE Pro*, è perfettamente compatibile con SteamVR.

Tale software consente di configurare la stanza utilizzata per usufruire della realtà virtuale: mediante le stazioni ad infrarossi, opportunamente posizionate, si permette al software di individuare lo spazio disponibile e i componenti dell'hardware. L'area di movimento viene calibrata e delimitata dal movimento dell'utente intorno alla stanza, puntando il controller lungo il perimetro della superficie disponibile. Il software ne mostra i limiti di gioco attraverso una griglia virtuale.

SteamVR si attiva automaticamente con il collegamento via cavo dell'hardware al computer su cui è installato e lavora in background con *Unity* e *WEAVR*, a cui consente la corretta fruizione degli hardware.

SteamVR include dei componenti di realtà virtuale utilizzati per permettere l'immersività in una scena. Alcuni *prefabs* sono sfruttati in *Unity* per questo progetto, in particolare:

- *Teleport Point*: è un punto di teletrasporto verso una zona della scena che permette questa funzionalità. In questo progetto sono utilizzati due *Teleport Point* per permettere lo spostamento dell'utente tra il sedile del pilota e del copilota attraverso l'utilizzo del trackpad del controller.
- *Teleport Area*: rappresenta un'area di teletrasporto. In questo progetto viene utilizzata una *Back Teleport Area*, posizionata dietro i sedili dei piloti, che consente all'utente di muoversi nell'area delimitata mediante il puntamento del telecomando.
- *Hand Poses*: permette la visualizzazione della posa della mano al momento di interazione con un oggetto della scena, a seconda del comportamento di quest'ultimo.

L'utilizzo dei *prefabs* consente di aumentare l'esperienza di immersività nell'ambiente virtuale della cabina di pilotaggio.

### 3.1.2 Unity

*Unity* è un software gratuito, sviluppato da *Unity technologies*, di tipologia *game engine*, capace di creare applicazioni per diverse piattaforme con sistemi operativi differenti. *Unity* permette di creare interfacce grafiche 2D e 3D, sia in VR che in AR, al cui interno è possibile integrare modelli 3D, audio, video, immagini e script. *Unity* replica elementi fisici come i centri di gravità degli oggetti [55] e li rende compatibili con le applicazioni *game* che è in grado di sviluppare.

I linguaggi di programmazione supportati sono *C++* e *Java*, ma al suo interno viene utilizzato il *C#*.

Durante i lavori del progetto di tesi vengono utilizzate due versioni di *Unity*:

- *Unity 2019.2.18f1* utilizzata per le **procedure a terra**.
- *Unity 2019.4.31f1* utilizzata per le **procedure in volo**.

Il motivo per cui sono state utilizzate due versioni differenti riguarda l'aggiornamento delle procedure. Le procedure a terra realizzate in questo progetto di tesi sono l'ampliamento di un progetto precedentemente esistente, che rende necessario l'utilizzo di versioni già installate di *Unity* e *WEAVR* affinché sia scongiurata la perdita dell'intero progetto dovuta ad un upgrade della versione di *Unity*. Nell'avanzamento del lavoro verso la realizzazione delle procedure in volo, la versione di *Unity* già presente non si è rivelata adeguata per la lettura dei dati attraverso i vari elementi. Si è dunque deciso, su indicazione di TXT, di scaricare l'aggiornamento del software ad una versione più recente che permettesse di sfruttare nuove funzionalità abbinate ad una più recente versione di *WEAVR*.



**Figura 3.1:** Schermata Unity

In Figura 3.1 viene illustrata la schermata di *Unity* con i suoi principali pannelli e comprensiva del modello del cockpit utilizzato per la realizzazione delle procedure dinamiche.

- *Hierarchy*: si tratta della gerarchia in cui vengono definite le relazioni tra i vari oggetti a seconda della loro interdipendenza. Più oggetti possono essere raggruppati tra loro e connessi ad un componente di gerarchia più alta.

- *Scene*: è il pannello in cui vengono mostrati gli oggetti contenuti in una determinata scena.
- *Game*: permette la visualizzazione della scena in modalità definitiva nella versione utilizzabile per l'applicazione. In questo progetto questo si traduce nella visualizzazione del cockpit dell'A320 in realtà virtuale attraverso l'utilizzo del visore. In modalità *game* il pilota interagisce con gli oggetti ed esegue le procedure.
- *Project*: contiene un menu di navigazione inclusivo delle directory presenti nella cartella principale del progetto. In particolare, la directory *Assets* rappresenta la destinazione in cui viene incluso il pacchetto WEAVR; essa contiene alcuni elementi cruciali per l'esecuzione del videogioco come scene, script, componenti audio e video, texture, modelli 3D e materiali.
- *Console*: viene utilizzato per testare il corretto funzionamento del progetto in quanto mostra una stampa di errori ed avvisi all'utente.

### 3.1.3 Pacelab WEAVR

WEAVR è una piattaforma software di realtà estesa sviluppata da PACE, società inclusa nel gruppo TXT a partire dal 2016. WEAVR consente la formazione in realtà virtuale se abbinata a *Unity* (paragrafo 3.1.2), in cui viene integrata come *plugin*. L'utilizzo di WEAVR si basa sull'interfaccia del programmatore con un editor visuale, che permette di evitare la scrittura di codici. TXT rilascia una guida di utilizzo di WEAVR al momento dell'acquisto della licenza [56].

**Configurazione** Per poter integrare WEAVR in *Unity* è necessario importarne il pacchetto in formato *.unitypackage*, senza il quale non è possibile visualizzare la finestra di WEAVR sul Toolbar Menu. In alternativa, è possibile usufruire di WEAVR in *Unity* importando l'apposita cartella all'interno della directory *Assets*. Con l'aggiornamento alla versione più recente di *Unity*, l'installazione di WEAVR si conclude con l'importazione della URP (*Universal Render Pipeline*), che offre un'ottimizzazione della grafica.

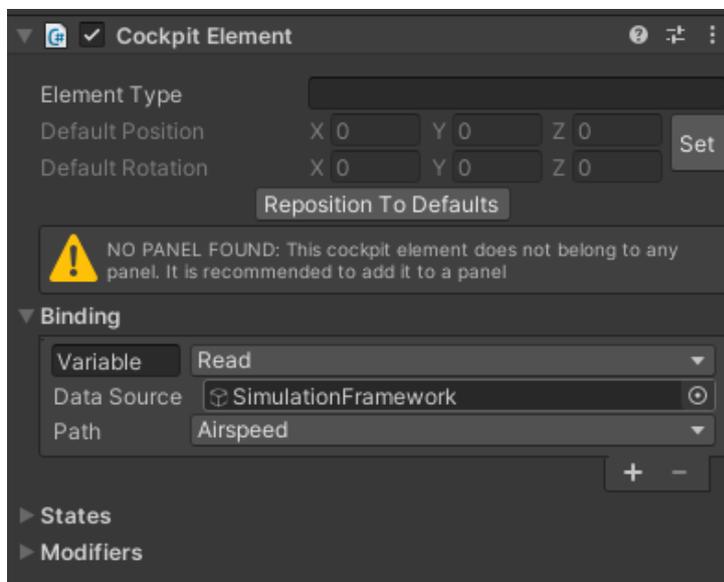
In seguito all'importazione è necessario configurare le impostazioni per l'abbinamento con SteamVR (paragrafo 3.1.1) in *Player settings* e l'abilitazione alla realtà virtuale in *Manage Extensions*. Per il *setup* della scena bisogna anche abilitare le voci *cockpit* e *simulation*. Al termine delle operazioni di setting, nella finestra *Hierarchy* è possibile visualizzare gli oggetti necessari alla creazione delle procedure di tipo *Virtual training*.

**Importazione del cockpit** Selezionando la finestra *Project* e ponendosi in *Assets*, è possibile importare il nuovo modello di cockpit dell'Airbus A320 fornito da TXT. Si tratta di un file in formato *.unitypackage* che contiene il modello 3D della cabina di pilotaggio. Inserendo successivamente l'oggetto importato all'interno della *Hierarchy* è possibile visualizzare il cockpit nella scena virtuale.

**Inspector** Ogni oggetto a cui è associata una determinata funzione, utile per la formazione reale in VR e non solo per fini grafici, è chiamato, in *Unity*, *Game Object*. L'*Inspector* è un pannello di *Unity* che permette l'associazione, ad un determinato *Object*, di differenti componenti associabili con WEAVR. Nel nostro lavoro di tesi sono state usate principalmente le seguenti:

- *VR Object*. Si utilizza per creare l'oggetto in VR. Se non fosse impostata questa componente, l'oggetto non sarebbe riconosciuto dal controller VR come tale. Questo *component* include anche la scelta dei modi di interazione, ossia le proprietà che il software assegna alla mano virtuale quando è in procinto di avvenire il contatto con l'oggetto. Spesso, nel nostro progetto, viene utilizzato il modo di interazione di tipo *Blend To Final Pose*, ovvero una funzione che permette alla mano di cambiare la sua posizione nel momento in cui si raggiunge il punto finale di interazione con l'oggetto.
- *Interaction Controller*. Rende l'oggetto interagibile. Quando il controller si avvicina all'oggetto si attiva l'interazione, evidenziata dal contorno illuminato dell'oggetto stesso. È un *component* creato automaticamente quando si aggiunge un determinato comportamento all'oggetto. I comportamenti controllati dall'*Interaction Controller* sono abilitati quando la mano interagisce con il *Box Collider* dell'elemento. Vengono elencati i comportamenti associati ad un determinato elemento.
- *Behaviours*. Permette di stabilire un comportamento preciso dell'oggetto, in conformità con il comportamento che assume nella realtà. I comportamenti principalmente utilizzati nelle procedure implementate sono:
  - *Executable*: eseguibile, ad esempio un tasto.
  - *Hinge door*: porta che ruota attorno ad una cerniera fissa.
  - *Two way switch*: interruttore a due vie.
  - *Three way switch*: interruttore a tre vie.
- *Cockpit Element*. Permette la comunicazione con la simulazione del volo tramite il *binding* necessario per la lettura e la scrittura. In Figura 3.2 viene illustrato il *Cockpit Element* che permette all'utente la visualizzazione del dato di *airspeed*.

- *Text*. È un *Game Object* necessario per la stampa dei dati di interesse in WEAVR.



**Figura 3.2:** Cockpit Element

**Procedure Editor** La realizzazione delle procedure in WEAVR avviene nel *Procedure Editor*, in cui è possibile creare una nuova procedura o importarne una esistente. Le procedure realizzate in questo lavoro di tesi sfruttano un collegamento tramite nodi dei vari step, che si considerano superati al soddisfacimento di una determinata condizione.

Le modalità di esecuzione di una procedura ne definiscono la tipologia.

- *Automatic Execution Mode*. Viene utilizzata per essere gestita ed eseguita internamente dal player WEAVR con una interazione minima dell'utente con gli oggetti e movimenti pre-determinati della camera. È una modalità utilizzata per mostrare all'utente come l'intera procedura dovrebbe essere eseguita.
- *Guided Execution Mode*. È la tipologia utilizzata in questo progetto. L'utente è guidato nell'esecuzione della procedura e interagisce con gli oggetti in maniera coerente con l'avanzamento degli step. Questa modalità non consente la navigazione tra i vari step, ma obbliga l'utente all'esecuzione cronologicamente corretta degli step procedurali.
- *Feedback Execution Mode*. È una modalità progettata per testare l'apprendimento dell'utente che non ha alcun tipo di guida e deve sapere autonomamente

quali step procedurali eseguire e in quale ordine. L'utente può comunque chiedere un *help* definito in altre modalità di esecuzione.

Il *Procedure Editor* mostra i vari nodi e il loro collegamento nel *Graph View*; durante l'esecuzione della procedura in VR da parte dell'utente, essi si coloreranno in verde per indicare l'esecuzione di una determinata azione o il verificarsi di una condizione. Ogni procedura viene salvata in formato *.asset* nel percorso *Assets* → *Procedures*, indicato nel *Procedure file* presente nella barra in basso. Una *Minimap* può aiutare l'utente ad avere una preview dell'intera procedura.

**Procedure Inspector** Permette di ispezionare i dettagli di ogni elemento della procedura e di realizzare i passaggi interni a ciascun nodo. Il *Procedure Inspector* consente l'implementazione, per ogni nodo, di:

- *Enter Actions*: azioni che si realizzano in ingresso al nodo stesso.
- *Exit Conditions*: condizione di uscita che, se soddisfatta, permette di accedere al nodo successivo.
- *Exit Actions*: non sono obbligatorie, poiché alcuni nodi prevedono l'uscita solamente dopo il soddisfacimento della condizione. Le azioni di uscita si realizzano immediatamente dopo il soddisfacimento delle *Exit Conditions*, prima di passare all'azione di ingresso del nodo successivo, se presente.

### 3.1.4 WEAVR Simulation Hub

WEAVR *SimulationHub* è un modulo realizzato per stabilire delle connessioni tra diversi punti di simulazione, permettere lo scambio e il mantenimento di informazioni tra loro e in memoria condivisa. Sfrutta il linguaggio di programmazione C++.

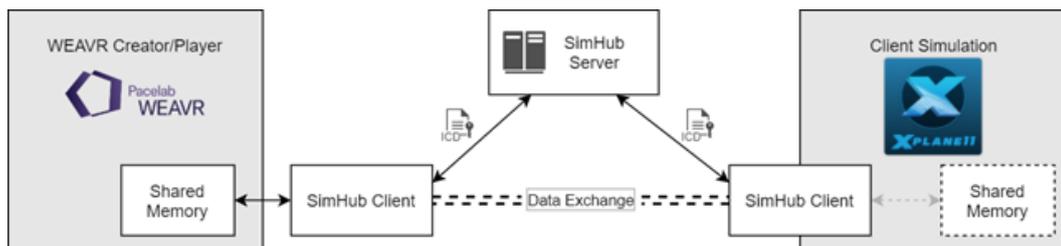


Figura 3.3: SimulationHub: overview [57]

Si compone di diversi elementi, come mostrato in Figura 3.3:

- **SimHub Server**: stabilisce e gestisce qualsiasi connessione o disconnessione tra i vari *client*. Si tratta di una applicazione autonoma, implementabile su

qualsiasi host. Quando un nuovo *client* viene connesso, il server riceve il file ICD<sup>1</sup> tramite il quale comunica al *client* con quali altri clienti deve connettersi. Il client è disponibile come versione autonoma, se immediatamente pronta all'utilizzo, o come SDK da integrare nella simulazione. Il server viene fornito da TXT come eseguibile da lanciare mediante un file *.bat*.

- **SimHub Client:** scambia dati con altri *client* usando il protocollo stabilito dal framework di SimulationHub. Ogni client fornisce un ICD al server per ottenere una connessione con altri client.
- **Shared Memory:** permette la gestione della memoria condivisa. SimulationHub contiene una libreria per l'accesso alla memoria condivisa che permette anche ad altri sviluppatori, che non utilizzano il linguaggio C++, di accedere alla memoria condivisa. Ad esempio WEAVR, sebbene scritto in C#, utilizza questa libreria per accedere alla *Shared Memory*.

### 3.1.5 Visual Studio

Microsoft Visual Studio è un ambiente di sviluppo integrato, multilinguaggio e capace di garantire la programmazione per dispositivi mobili, desktop, giochi in AR/VR, machine learning e intelligenza artificiale.

In questo lavoro di tesi viene utilizzata la versione Visual Studio 2019, attualmente la più aggiornata sul mercato. Le applicazioni in cui è stato necessario ricorrere alla programmazione in Visual Studio sono le seguenti.

- **Client X-Plane.** La realizzazione del codice ha richiesto l'applicazione di conoscenze di programmazione in linguaggio C++ per stabilire una compatibilità di linguaggio con SimulationHub.
- **Script C#.** Per realizzare alcuni script in WEAVR è stato necessario utilizzare il linguaggio C# per la scrittura del codice in Visual Studio.

Il *Client X-Plane* si connette a SimulationHub. È dunque necessario impostare le proprietà di configurazione del progetto per includere le librerie e le funzionalità di SimulationHub descritte nel paragrafo 3.1.4. Nella "Directory di inclusione aggiuntiva" della scheda "C/C++" si specifica il percorso della cartella contenente i file *.h* delle librerie e, nella voce "Input" della scheda "Linker", si imposta il percorso dei file *.lib*. Per l'esecuzione del codice è necessario che i file *.dll* siano nella stessa cartella dell'eseguibile. Le cartelle di SimulationHub vengono inserite all'interno della cartella principale del *client* al fine di rendere più immediato e versatile il progetto attraverso l'utilizzo dei percorsi relativi in fase di configurazione.

---

<sup>1</sup>ICD: Interface Control Document

### 3.1.6 X-Plane

X-PLANE è un software di simulazione aeronautica sviluppato per diverse piattaforme e sistemi operativi e considerato, analogamente a *Microsoft Flight Simulator*, uno dei più realistici e completi presenti nel mercato internazionale.

Il modello dinamico del velivolo di X-PLANE fa fede alla Teoria dell'Elemento di Pala<sup>2</sup>, secondo cui ogni componente del velivolo viene diviso in sotto-parti di cui viene modellato il comportamento. L'intero velivolo si ottiene integrando tutti i contributi di forze e accelerazioni. X-PLANE adotta questa teoria per modellare in maniera più accurata differenti tipologie di aeromobili.

Tale simulatore consente di effettuare sia il training per neo-piloti, sia modalità di volo apposite per piloti esperti. Il database di aeroporti incluso nel software è completo; essi vengono modellati in 3D. Le simulazioni possono essere effettuate in VR.

La versione acquistata per il Laboratorio di Simulazione del Volo del Politecnico di Torino è X-PLANE 11, per Windows. Tra i velivoli disponibili in X-PLANE 11 figurano [58]: Cessna 172SP, Boeing 737-800, Beechcraft Baron 58, Cirrus Vision SF50, Schleicher ASK 21, Stinson L-5 Sentinel, Aero-works Aerolite 103, Beechcraft King Air C90, Boeing 747-400, McDonnell Douglas MD-80, Sikorsky S-76.

Come si nota, non figura il modello *Airbus A320* su cui si incentra il progetto di tesi.

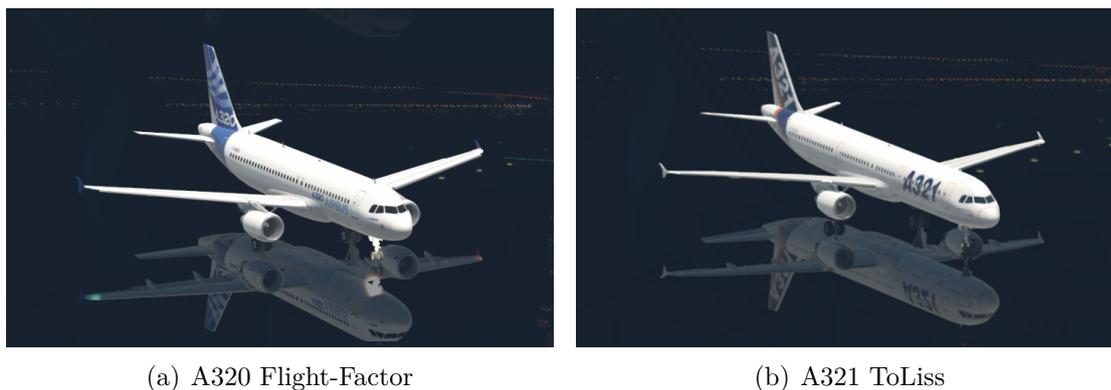
Vengono dunque acquistati separatamente due modelli (Figura 3.4):

- *Airbus A320 Ultimate Flight-Factor*. Contiene un modello di volo ad alta definizione che fornisce un'esperienza molto realistica all'utente, che include: modello aerodinamico affinato e inclusivo anche di effetti transonici, modello di motore specifico, grafica dei display di qualità 4K e modello di autopilota completo.
- *Airbus A321 ToLiss*. Fornito di licenza ufficiale Airbus, questo modello ha un modello di cockpit 3D con texture di elevata qualità, 2 MCDU e autopiloti indipendenti, sistemi Fly-By-Wire di notevole precisione e modellazione dettagliata dei sistemi di bordo.

La scelta di acquistare due modelli è stata dettata da esigenze emerse durante i lavori. Sebbene il *Flight-Factor* mostrava un realismo e un dettaglio grafico difficilmente replicabili, risultava un modello pesante e difficilmente gestibile in previsione di un'implementazione delle procedure di volo. Tra gli svantaggi, infatti,

---

<sup>2</sup>Blade Element Theory



(a) A320 Flight-Factor

(b) A321 ToLiss

**Figura 3.4:** Confronto A320 - A321 in X-PLANE

emergono la limitata scelta dei *datarefs*<sup>3</sup> e l'impossibilità di salvare e caricare accuratamente specifiche situazioni di volo.

L'acquisto del modello *ToLiss* ha permesso il superamento di queste limitazioni. Quest'ultimo, infatti, è fornito di numerosi *datarefs* in grado di controllare ogni aspetto della simulazione. Inoltre il *ToLiss* dispone di un plugin integrato, chiamato *ToLiss Interactive Simulation Control System*, che consente il salvataggio e il caricamento corretto delle situazioni. L'acquisto di tale modello garantisce anche la compatibilità totale con il simulatore di volo presente nel Laboratorio, che fa fede allo stesso modello. Sebbene si tratti di un modello A321, anziché di un A320, non si rilevano differenze nella cabina di pilotaggio, che risulta essere la medesima; i piloti, infatti, con l'abilitazione per la famiglia A320, sono in grado di pilotare entrambi i modelli reali, a meno di una breve integrazione.

### 3.1.7 X-Plane Connect

X-Plane Connect (XPC) è un toolbox di ricerca *open source* sviluppato dalla NASA per permettere l'interazione con X-PLANE. XPC consente agli utenti di controllare e interagire col velivolo tramite un flusso di dati e utilizza delle funzioni sviluppate in diversi linguaggi (C, C++, Java, MATLAB, Python) in tempo reale. Viene utilizzato per testare algoritmi di controllo attivi sulla simulazione di volo di X-PLANE; l'interazione con il simulatore avviene tramite UDP<sup>4</sup>.

---

<sup>3</sup>Informazioni pubblicate da X-PLANE, accessibili tramite stringhe che permettono di controllare la simulazione di volo da codice

<sup>4</sup>UDP: User Datagram Protocol

XPC include un plugin da integrare in X-PLANE e un *client* da abbinare direttamente al simulatore. Nel nostro caso sono state utilizzate le librerie di XPC per riuscire a stabilire un collegamento con SimulationHub (paragrafo 3.1.4) al fine di creare il *Client X-Plane*. Questo toolbox consente all'utente di manipolare lo stato del velivolo nella simulazione mediante l'utilizzo di *datarefs* e comandi, ossia stringhe che il codice invia al simulatore affinché il velivolo esegua una determinata azione o imposti un determinato parametro su un valore desiderato.

**Utilizzo di XPC** Il download di *X-Plane Connect* viene effettuato dalla piattaforma *Github* [59]; la sua cartella *.zip* viene estratta e copiata nella directory dei *plugin* di X-PLANE, mentre all'interno della soluzione C++ del *Client X-Plane* viene inserito il file *xplaneConnect.c* alla voce "File di origine".

Nel codice sono utilizzate le seguenti funzioni proprie di XPC:

- *openUDP*: crea un socket UDP che permette di stabilire la connessione tramite porta all'indirizzo IP del computer su cui X-PLANE è in esecuzione.
- *getDREF*: legge uno o più valori del *dataref* inviato come stringa tramite il socket.
- *sendDREF*: invia uno o più valori del *dataref* inviato come stringa tramite il socket; è necessario che il *dataref* sia di tipo *writable*.
- *sendCOMM*: invia un comando come stringa tramite il socket.

I *dataref* e i comandi posso essere propri di X-PLANE o tipici del velivolo.

## 3.2 Hardware

La componentistica hardware utilizzata per questo lavoro di tesi comprende la strumentazione VR e fisica per la simulazione.

### 3.2.1 HTC Vive Pro

L'HTC *Vive Pro* [60], raffigurato in Figura 3.5, è l'hardware utilizzato per la realtà virtuale.

Esso include le seguenti componenti.

**Visore** È un *headset* di tipologia HMD - *Head Mounted Display* che permette all'utente di visualizzare l'ambiente artificiale con un display AMOLED da 3.5", con risoluzione totale di 2800 x 1600 pixel e un field-of-view di 110°. Al visore è collegato un impianto audio certificato Hi-Res composto da due microfoni e cuffie ad



**Figura 3.5:** HTC VIVE Pro

alta impedenza che, mediante la riproduzione di un audio 3D, permette all'utente un'immersione completa nella realtà. Il visore consente all'utente anche di muoversi liberamente nello spazio, in maniera coerente con le dimensioni della stanza e dell'ambiente creato; questo è possibile mediante la presenza di sensori, compatibili con il software SteamVR (paragrafo 3.1.1), che permettono un tracciamento a sei gradi di libertà del movimento. Il *motion tracking* è di tipo *outside-in tracking*: rileva la posizione e il movimento del visore, affidandosi alle stazioni ad infrarossi per guidare l'utente nei limiti dell'area di movimento. Questa tipologia di tracciamento garantisce maggiore accuratezza e minore latenza, sebbene vincolata alla presenza di cavi fisici che potrebbero intralciare il movimento dell'utente.

**Telecomandi** I *controller* sono dei mezzi di interfaccia tra l'utilizzatore e l'ambiente VR, che consentono di effettuare azioni fisiche come la selezione dei tasti del pannello, l'attivazione dei pulsanti o l'impugnatura di oggetti. Mediante i controller, nel caso applicativo del training dell'A320 in WEAVR, si può anche effettuare un teletrasporto tra i vari *teleport point* del cockpit. I controller sono di tipo wireless, ricaricabili e sono inoltre dotati di numerosi sensori per il tracciamento, di un trackpad multifunzionale che consente il teletrasporto, di un grilletto posteriore che serve per l'interazione con i tasti, di due tasti frontali per l'accensione o l'accesso al menù e di due pulsanti laterali. I telecomandi dell'HTC *Vive Pro* non includono un feedback aptico nella simulazione in realtà virtuale. Per provare una simile esperienza è necessario ricorrere a dispositivi più aggiornati sul mercato come i *VR Gloves*, ossia guanti virtuali capaci di simulare l'intero movimento della mano e delle dita e di trasmettere la percezione dei pesi all'utente.

**Base-stations** Sono emettitori a luce infrarossa e costituiscono un sistema di riferimento fisso esterno in cui il visore, con gli appositi sensori, si orienta. Le

base-stations sono in grado di ricoprire un'area di  $5 \times 5 \text{ m}$  ciascuna con un angolo di copertura complessivo di  $120^\circ$  per la fruizione dell'ambiente VR, dunque esse devono essere poste a distanza superiore ai 5 metri l'una dall'altra. Nel Laboratorio di Simulazione del Volo del Politecnico di Torino sono presenti due base-stations con cavi.

### 3.2.2 Simulatore A320

Nel Dipartimento di Ingegneria Meccanica ed Aerospaziale (DIMEAS) del Politecnico di Torino è presente il simulatore del velivolo Airbus A320, basato interamente su X-PLANE, che riproduce le principali componenti del cockpit (riportati in Figura 3.6) in scala 1:1.



**Figura 3.6:** Equipaggiamenti del flight deck presenti nel Simulatore A320 [22]

**Architettura** L'architettura del simulatore è riportata in Figura 3.7. Diversi computer interagiscono per l'esecuzione della simulazione.

Il **computer primario** è dotato di un processore i7 e scheda grafica NVIDIA GTX 1080 ed è collegato a un sistema di proiezione, costituito da due **proiettori**. Una terza uscita video permette di generare i PFD per il primo ufficiale. Questo computer ha prestazioni grafiche elevate per fronteggiare le esigenze di una

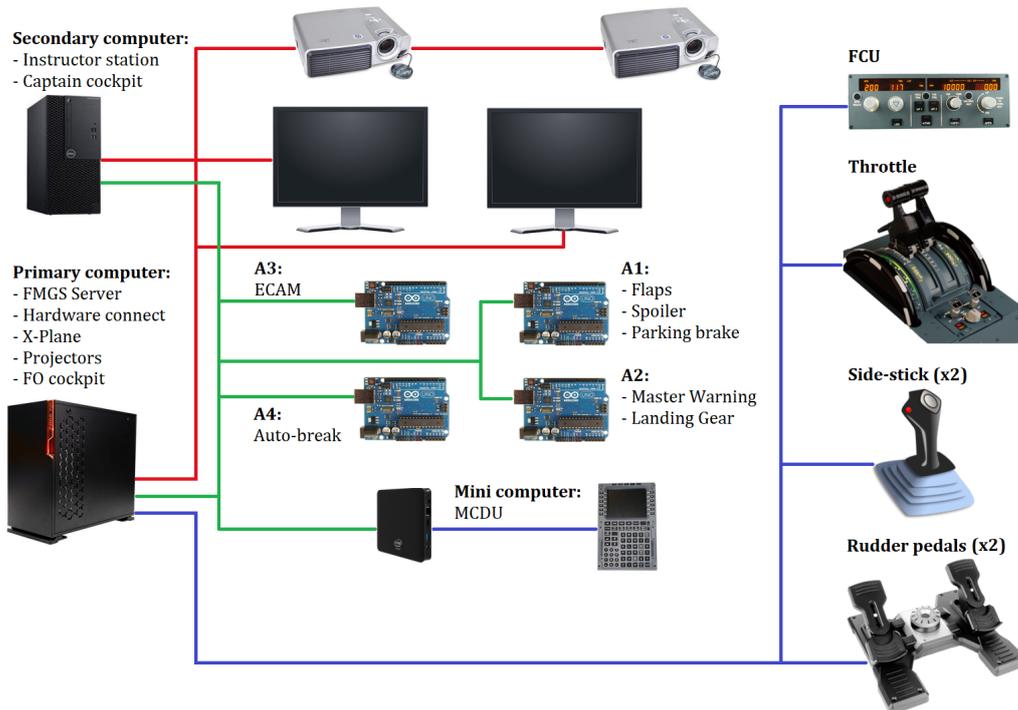


Figura 3.7: Architettura del Simulatore A320

simulazione realistica. È inoltre collegato al FCU, alla manetta, alla pedaliera e al sidestick.

Il **computer secondario** ha il compito di generare la grafica del display del capitano e gli schermi dell'ECAM, permette di definire la missione tramite l'inserimento dei dati di volo durante la programmazione dell'MCDU. Rappresenta la piattaforma dell'istruttore, che può inserire delle *failure* nel sistema di bordo per valutare la capacità del pilota di gestire delle situazioni di emergenza.

Sono presenti 4 **microcontrollori**, che permettono il controllo degli *switch*, dei potenziometri di interfaccia con le leve del *pedestal panel*, delle luci, del carrello e dei warnings. Un **mini computer** gestisce il funzionamento dell'MCDU del capitano.

Tutti i computer sono collegati ad una rete locale che ne permette la sincronizzazione dei dati e la comunicazione. Tra i computer e i proiettori vi è un collegamento video, mentre tra i computer e gli hardware del pilota vi è un collegamento che permette di raccogliere i dati di input, poiché nel computer principale è installato il modello di volo dell'A320.

**Equipaggiamenti del simulatore** Il simulatore dell'A320 è fornito di:

- **EFIS: Electronic Flight Instrument System.** Comprende i selettori del pannello e le luci di *Master Warning* e *Caution Warning*.
- **PFD: Primary Flight Display.** Mostra le indicazioni di volo primarie, come altitudine, velocità di volo e verticale, assetto, heading.
- **ND: Navigation Display.** Fornisce le indicazioni relative alla direzione di navigazione, waypoints o eventuale presenza di aeroporti.
- **FCU: Flight Control Unit.** Rappresenta lo strumento di interfaccia con il Flight Management System ad uso simulazione e permette al pilota di impostare i modi di guida e l'autopilota.
- **MCDU: Multi-function Control and Display Unit.** È la parte del FMS che consente al pilota di impostare e modificare il piano di volo.
- **ECAM: Electronic Centralized Aircraft Monitor.** Si compone di schermi di monitoraggio dei sistemi di bordo del velivolo.
- **Throttle:** manetta per il controllo dei propulsori.
- **Speed brake:** leva per il controllo dei freni.
- **Flap lever:** leva per il controllo di flaps e slats.
- **Parking brake:** leva del freno di parcheggio.
- **Rudder pedals:** pedaliera. Sono presenti sia per il capitano che per il primo ufficiale e permettono il controllo di timone e freni.
- **Sidestick:** permettono il controllo di alettoni ed equilibratore.

Il simulatore presente nel Laboratorio di Simulazione del volo non include gli switch per l'accensione dei propulsori e l'*overhead panel* nella struttura del cockpit dell'A320. Viene però fornito agli studenti un pannello 2D con una raffigurazione stampata dell'overhead, che consente di familiarizzare con la struttura, sebbene ne impedisca l'utilizzo fisico nella simulazione. La simulazione deve dunque necessariamente essere avviata in situazione di velivolo posizionato nel piazzale di parcheggio con i motori già avviati e pronto per la procedura di *taxi*.



# Capitolo 4

## Procedure a terra

Le procedure a bordo del velivolo, a terra o in volo, permettono di descrivere le attività del personale aeronautico in ogni fase attraverso un elenco di *task* che il pilota dovrà seguire per svolgere le operazioni in sicurezza.

Poiché il velivolo su cui vengono implementate le procedure di questo lavoro di tesi è l'*Airbus A320*, si fa riferimento direttamente al manuale di addestramento della crew di bordo di *Airbus* [61].

La standardizzazione delle procedure operative consente al personale di bordo dell'A320 di:

- Gestire eventuali situazioni anomale
- Prevenire gli errori
- Portare a termine le *task* della fase in sicurezza.

Le procedure si dividono in *standard* e *abnormal*. Inizialmente il pilota porta a termine una *normal checklist* delle operazioni da seguire a terra e in volo. Le procedure *standard* vengono seguite abitualmente in ogni volo; le *abnormal* prevedono l'applicazione in situazioni di emergenza.

In questo lavoro di tesi vengono implementate due procedure: una *standard* e una *abnormal*. Tali procedure, in quanto statiche, sono portate a compimento dal pilota **a terra** ed implementate in *Unity* con il supporto di WEAVR. Successivamente, attraverso un collegamento tra *Unity* ed il simulatore X-PLANE sono implementate altrettante procedure dinamiche, con l'aggiornamento dei dati di volo in tempo reale. Verranno presentate le procedure a terra di **Engine Start** ed **On Ground Emergency Evacuation**, che vanno ad ampliare la rete di procedure implementate in progetti di tesi svolti in anni accademici precedenti ([1], [2] [3], [4]).

## 4.1 Engine Start

La *Engine start procedure* appartiene alla categoria delle *standard operations* poiché viene effettuata in ogni missione del velivolo. Si tratta di una procedura di *startup* che prevede l'accensione e la verifica del corretto funzionamento dei motori in seguito al completamento della *Before Start procedure*; è quindi necessaria per preparare il velivolo al decollo. Nella seguente procedura viene illustrato solo il metodo di accensione del Motore 2, a scopo di training; il pilota però dovrà effettuare l'operazione di accensione dei motori ripetendo i passaggi anche per il Motore 1.

STARTUP CHECKLIST	
APU .....	START
APU BLEED .....	RUN (WHEN AVAILABLE)
APU GEN .....	ON / CHECK VOLTS
<b>Request Pushback – Initiate Pushback</b>	
THRUST LEVERS .....	IDLE
ENGINE AREA .....	CLEAR
FUEL PUMP SWITCHES .....	ON
MODE SELECTOR .....	IGN / START
LEFT ENGINE .....	(ENG 1)
ENGINE MASTER 1.....	START
AT N2 > 20% FUEL FLOW .....	CHECK ON
N1 INCREASING AS N2 INCR. ....	CHECK
OIL PRESSURE .....	CHECK
GENERATOR SWITCH .....	ON
REPEAT FOR RIGHT ENGINE .....	(ENG2)
FUEL FLOW .....	CHECK
HYDRAULIC PUMP SWITCHES .....	ON
APU .....	OFF
MODE SELECTOR .....	NORM

**Figura 4.1:** Startup procedure

Un esempio di *startup procedure* è rappresentato in Figura 4.1 e riprende la checklist presentata da *Fly UK* [62].

**Benvenuto** L'utente indossa il visore, impugna i joystick ed è pronto per affrontare la procedura in realtà virtuale, in cui è posizionato sul sedile del pilota e deve impugnare il tablet per dare inizio alle operazioni.

In Figura 4.2 è riportata la schermata di benvenuto che l'utente visualizza, in cui si consiglia di impugnare il tablet e selezionare la procedura da seguire. Questa schermata è una *billboard* implementata in WEAVR, centrata sul *Pedestal control panel*; è dinamica, poiché segue eventuali movimenti del tablet qualora venisse impugnato e spostato dall'utente, dura 10 secondi ed è sincronizzata con un messaggio vocale che riproduce il contenuto del testo. Una freccia di navigazione guida l'utente nell'individuazione della billboard.



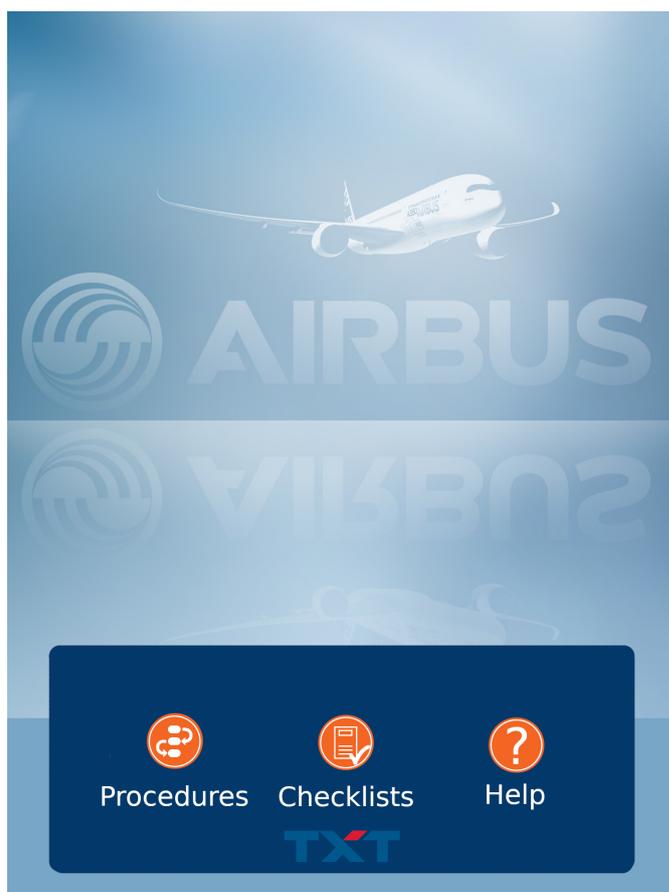
**Figura 4.2:** Welcome billboard

**Tablet** L'utente dunque afferra il tablet, che generalmente può trovare alla sua sinistra, in cui vi è riportata la schermata principale (Figura 4.3).

In essa sono presenti 3 tasti, etichettati in WEAVR come *executable*.

- **Help:** fornisce una spiegazione sulla modalità di scelta della procedura e di visualizzazione delle varie *checklist*.
- **Checklists:** contiene un elenco delle checklist implementate in WEAVR. Nella prima pagina si trovano le cosiddette *Normal Procedures*, nella seconda è presente la checklist della procedura *De-Icing*, nella terza sono presenti le checklist delle procedure di *Engine Start* e *On Ground Emergency Evacuation*. È possibile passare da una pagina all'altra mediante i tasti **Next** e **Prev**, posizionati in basso, accanto a cui si trova il pulsante **Home**, che consente il ritorno alla schermata principale.
- **Procedures:** contiene l'elenco delle procedure implementate in WEAVR ed eseguibili in VR. Si compone di due pagine (Figura 4.4) poiché si è ritenuto opportuno suddividere l'elenco delle procedure per avere una pulizia grafica maggiore; in basso sono presenti i tasti per passare da una pagina all'altra (**Pag.1** o **Pag.2**) e per tornare al menu principale (**Home**). L'utente, per avviare la procedura, deve selezionare il tasto **Play** accanto alla relativa dicitura. Se invece viene cliccato il nome della procedura, il tablet ne mostrerà la checklist relativa.

**Step 0: Preparazione della scena** L'utente decide di avviare la procedura di *Engine Start*. Dopo aver cliccato sull'apposito **Play**, viene visualizzata la schermata



**Figura 4.3:** iPad First Screen Procedure

di benvenuto nella procedura, mostrata in Figura 4.5. Si tratta di una billboard centrata sul pedestal panel, analogamente alla schermata di benvenuto iniziale, che sarà sincronizzata con un messaggio vocale che riproduce il testo.

Sul tablet verrà dunque visualizzata la checklist dell'*Engine Start*: in ogni riga viene nominata una task che, quando correttamente realizzata, sarà affiancata da un apposito *tick*.

A questo punto, in WEAVR, vengono settati i vari *Object* e la scena in maniera coerente con il termine della procedura di *Before Start*.

**Step 1: Engine mode selector IGN/Start** Dopo aver correttamente completato la preparazione del cockpit in seguito alla verifica della *Before Start checklist*, già settata in WEAVR, l'utente visualizzerà una billboard, centrata sull'*Engine mode selector*, che gli suggerisce di effettuare uno switch del tasto.



(a) Pag.1

(b) Pag.2

Figura 4.4: Procedure

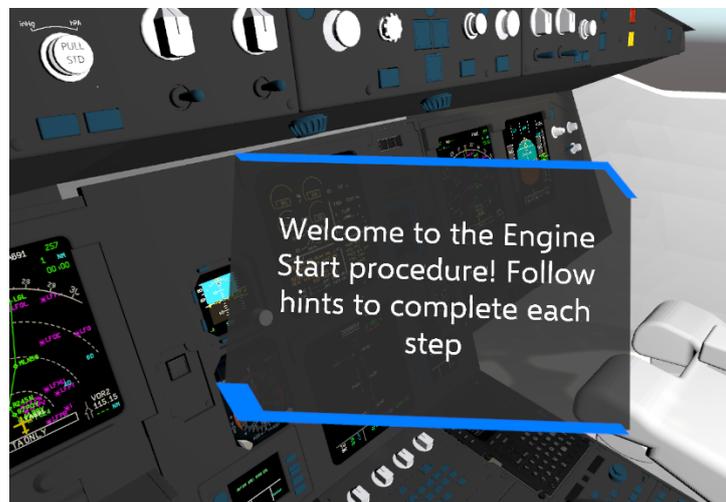


Figura 4.5: Welcome Engine Start

In Figura 4.6 è riportata la visuale dell'utente. L'interazione della mano dell'utente con il tasto avviene tramite il controller. L'**Engine mode selector** viene



**Figura 4.6:** Step 1: Engine mode selector

opportunamente settato in WEAVR: si tratta di un *Object* che ha un comportamento di tipo *Two way switch*, ossia un interruttore a due stadi. Quando l'utente aziona l'interruttore compare una billboard della durata di 1 secondo che annuncia il nuovo stato del pulsante. Lo switch di tale interruttore costituisce una *Exit condition*: quando è verificata, si passa al nodo successivo. Infine, in questo step, sono inserite, come *Exit actions*, l'accensione dell'ECAM display e la rimozione della billboard di partenza: infatti dopo aver azionato l'interruttore, l'utente non ha più bisogno della guida e può procedere allo step successivo.

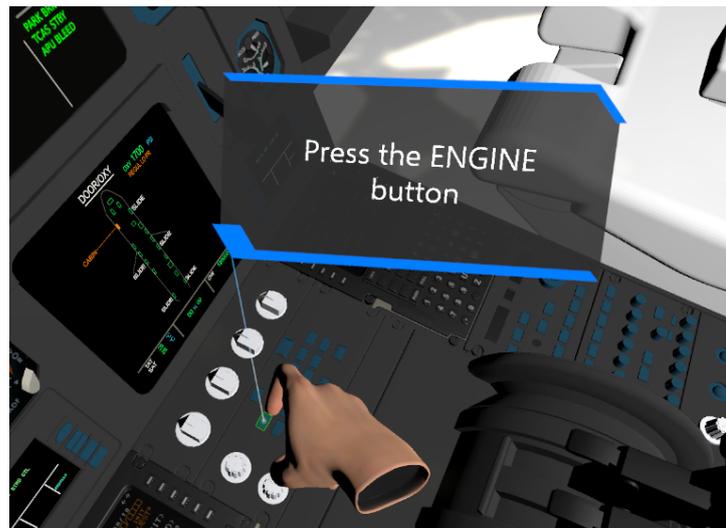
**Step 2: Engine Key PUSH** Il pilota, a questo punto, aziona il tasto *Engine key* per visualizzare lo stato dei motori sull'ECAM.

In Figura 4.7 si evidenzia l'azione dell'utente in questo step. La billboard è centrata nel target apposito e un messaggio vocale ne riproduce il contenuto. L'**Engine Key** è un *Object* di WEAVR di tipo *executable*, equivalente ad un pulsante da premere; la condizione di uscita dal nodo prevede che l'utente abbia schiacciato il pulsante, mentre l'azione di uscita cancella la billboard e attiva l'**ECAM System Display**, che riporta i parametri del motore.

**Step 3-4: ECAM check** I successivi due step sono dei check che il pilota deve effettuare per controllare che tutti i parametri del motore siano disponibili sull'**ECAM display** superiore ed inferiore.

In Figura 4.8 è riportato l'ECAM Display:

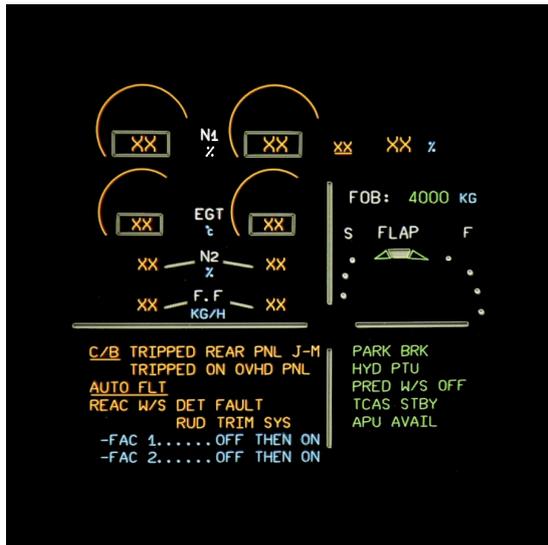
- All'inizio della procedura l'ECAM Display (upper) mostra una proiezione con delle croci ambrate per indicare che i parametri del motore non sono disponibili, poiché spento. (Figura 4.8 a)



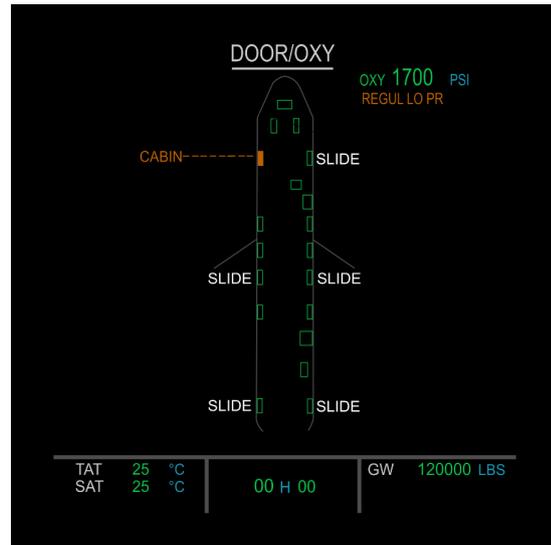
**Figura 4.7:** Step 2: Engine key

- L'ECAM Sistem Display (lower) evidenzia la schermata delle porte del velivolo, in maniera coerente con il termine della procedura di *Before Start* (Figura 4.8 b).
- L'ECAM Display, in seguito all'attivazione dell'*Engine mode selector*, inizia ad accendersi e a mostrare i parametri di  $N1$ ,  $N2$ ,  $EGT$  e  $F.F$ . Inizialmente i valori sono nulli (Figura 4.8 c).
- L'ECAM Sistem Display, in seguito alla selezione dell'*Engine Key*, mostra i parametri di pressione dell'olio e *start valve* (Figura 4.8 d).

La Figura 4.9 mostra il modo in cui i check devono essere effettuati dall'utente in VR; le schermate di Figura 4.8 sono correttamente visualizzate. In entrambi gli step compare una billboard, accompagnata da un messaggio vocale, in cui si chiede al pilota di ispezionare visivamente i display. Tali display sono stati creati manualmente inserendo delle immagini in WEAVR di tipologia *Sprite 2D*. La procedura a terra, infatti, non consente l'aggiornamento in tempo reale dei dati del display, per cui è necessario sostituire manualmente l'immagine ed utilizzarne una fissa per il check. La *Exit condition* prevede un'ispezione visiva (target *Visually inspected*) di 3 secondi per i display. Quando l'utente raggiunge la distanza prefissata per il check, rilevata grazie ai sensori del visore, l'obiettivo si rimpicciolisce ed avvisa l'utente dell'ispezione correttamente riuscita. La durata della proiezione delle billboard è impostata pari a 5 secondi, di conseguenza non è necessario inserire alcuna *Exit action* per la loro rimozione.



(a) ECAM Display old



(b) ECAM Sistem Display old



(c) ECAM Display new

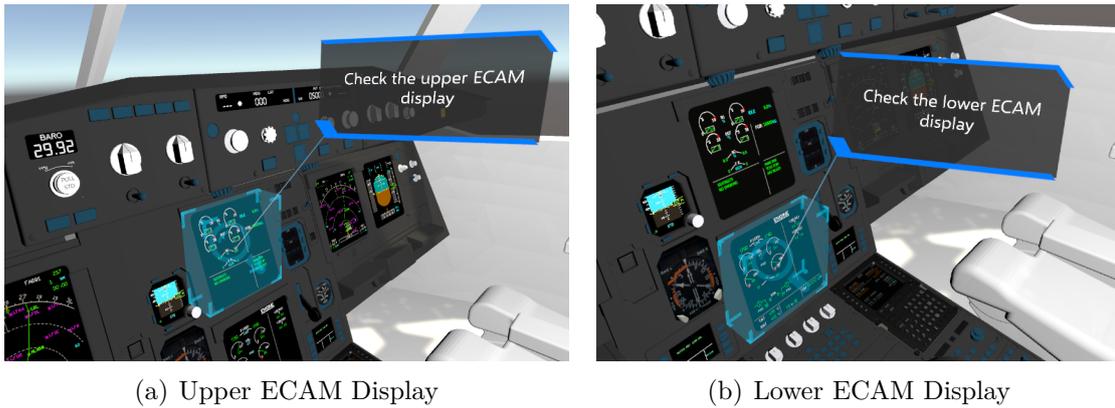


(d) ECAM Sistem Display new

Figura 4.8: Modifica dell'ECAM

**Step 5: Engine Manual Starter ON** Lo step successivo prevede l'accensione manuale del motore.

L'Engine Manual starter è un tasto di tipo *executable*; per attivarlo è necessario sollevare una gabbia di protezione chiamata **Engine guard**, a sua volta identificata come *Object* di tipo *Two way switch*.



**Figura 4.9:** Step 3-4: Check ECAM display

L'utente è invitato da una billboard, sincronizzata con un messaggio audio, ad accendere il manual starter; egli, dunque, effettua uno switch della gabbietta per consentirne l'apertura e poi procede all'accensione.



**Figura 4.10:** Step 5: Engine Manual Starter

Le *Exit conditions* riguardano:

- Apertura della gabbietta -> *Two way switch* settato su *up*
- Esecuzione del tasto -> *Engine manual started executed*

Quando queste condizioni si realizzano, l'interruttore viene visualizzato in **ON**, con l'inserimento di una apposita immagine che ne indica lo stato di accensione. La billboard precedentemente mostrata viene eliminata e sostituita da una che

indica lo stato di accensione del manual starter, con la dicitura "Current State: ON", della durata di 1 secondo.

Con l'accensione manuale del motore viene richiesto al pilota un nuovo controllo dei display dell'ECAM, ripetendo i passaggi di ispezione visiva applicati negli step 3-4 (Figura 4.9). In particolare il pilota deve verificare:

- L'incremento della pressione dell'olio e l'allineamento della *start valve* sul **lower ECAM Display**, ossia il display inferiore dell'ECAM.
- L'incremento, sull'**upper ECAM Display**, del valore *N2* (numero di giri del gruppo compressore-turbina di alta pressione) fino al 60%. Il caso ideale, inoltre, prevede che il valore *N1* (numero di giri del gruppo compressore-turbina di bassa pressione) arrivi fino al 20%.

**Step 6: Switch Engine Master ON** L'utente, seguendo l'indicazione riportata nella apposita billboard e riprodotta vocalmente mediante un messaggio audio, dovrà azionare l'**Engine Master Switch**. Si tratta di un *VR object* con comportamento *Two way switch*.



**Figura 4.11:** Step 6: Engine Master switch

In Figura 4.11 viene illustrato il modo in cui il pilota innesta lo switch. L'interazione tra il controller e il *VR object* permette l'azionamento del pulsante. Questa operazione deve essere effettuata sia nel caso di *Engine Start* manuale che automatico. La *Exit condition* prevede il posizionamento su *up* del *Two way switch*. Tra le *Exit actions* figurano sia la rimozione della billboard di guida, sia l'attivazione del suono del motore, realizzata mediante un *toggle object*. Il pilota avrà in questo modo un feedback uditivo sull'effettiva accensione del motore. Completa lo step

l'attivazione temporanea, della durata di un secondo, di una billboard che riporta il messaggio "Current state ON".

Il pilota ispeziona l'**upper ECAM display** per controllare nuovamente i parametri del Motore 2.

**Step 7: Engine Manual Starter OFF** In seguito all'ispezione dei parametri del motore e appurata la loro correttezza, il pilota può procedere allo spegnimento del manual starter. Il motore, infatti, è acceso e non ha più bisogno del manual starter. L'utente visualizza una billboard che lo invita a spegnere questo interruttore. La gabbia di protezione è ancora aperta, in seguito ai passaggi dello Step 5. Con lo spegnimento del manual starter, viene visualizzata un'immagine sovrapposta all'interruttore che ne indica il nuovo stato: OFF.



**Figura 4.12:** Step 7: Engine Manual Starter OFF

Il pilota è dunque invitato anche a chiudere la gabbia di protezione, come indicato in Figura 4.12.

**Step 8: Engine mode selector NORM** Si tratta dell'operazione inversa rispetto allo step 1: l'**Engine Mode Selector** viene riportato al suo stato iniziale: il *Three way switch* viene impostato su *middle* attraverso una rotazione opposta rispetto allo Step 1. In questo modo compare una billboard che avvisa lo stato corrente dell'interruttore.

**Checklist** Il pilota ha correttamente completato la procedura. Una billboard compare per avvisare l'utente che ha a disposizione 20 secondi per controllare tutti i passaggi svolti, controllando la checklist completa della procedura presente nella schermata visualizzata sul tablet (Figura 4.13).



Figura 4.13: Engine start completed



Figura 4.14: Engine Start: Checklist

In Figura 4.14 si nota che viene aggiunto un tick accanto ad ogni operazione correttamente completata; al termine della procedura tutte le task risulteranno spuntate.

**End Procedure** L'ultimo step della *Engine Start procedure* ha lo scopo di informare l'utente che le operazioni sono state correttamente portate a termine; la procedura può essere ripetuta o, in alternativa, l'utente può decidere di effettuarne un'altra. Non sono presenti condizioni di uscita. Vengono quindi implementate in WEAVR le seguenti *Exit actions*:

- Rimozione di tutte le billboard precedentemente mostrate
- Disattivazione della schermata dei parametri del motore nell'ECAM Sistem Display e attivazione della schermata di default
- Disattivazione della schermata del motore acceso nell'ECAM Display e attivazione della schermata di default
- Spegnimento del motore mediante il ripristino in posizione *down*, quindi OFF, dell'Engine Master Switch

L'ultimo nodo viene collegato al *Procedure Starter* in cui, tramite l'esecuzione del tasto **play**, l'utente sceglie di avviare di una nuova procedura, come visualizzato in 4.15.



**Figura 4.15:** Procedure starter

La procedura di *Engine Start* viene dunque implementata con una programmazione sequenziale in WEAVR mediante la creazione di nodi.

Il flow chart dell'implementazione della procedura è riportato in Tabella 4.1.

## 4.2 On Ground Emergency Evacuation

Questa procedura è descritta nel dettaglio nell'elaborato di tesi del mio collega Giovanni Sgarra.

OGGETTO	AZIONE
Nodo 1	Scene preparation
Nodo 2 Engine mode selector	IGN/Start
Nodo 3	"Current state: ON"
Nodo 4 Engine Key	Press
Nodo 5 ECAM Display	Check
Nodo 6 ECAM Sistem Display	Check
Nodo 7 Engine Guard	UP
Nodo 8 Engine Manual Starter	ON
Nodo 9 ECAM Sistem Display	Check
Nodo 10 ECAM Display	Check
Nodo 11 Engine Master Switch	UP
Nodo 12	"Current state: ON"
Nodo 13 ECAM Display	Check
Nodo 14 Engine Manual Starter	OFF
Nodo 15	"Current state: OFF"
Nodo 16 Engine Guard	DOWN
Nodo 17 Engine mode selector	Middle
Nodo 18	"Current state: NORM"
Nodo 19 Checklist	
Nodo 20 End procedure	

**Tabella 4.1:** Engine Start procedure: nodi in WEAVR

L'avvio della procedura di *On Ground Emergency Evacuation* presuppone il completamento delle checklist di Before Start e Before Take-off.

EMER EVAC	
AIRCRAFT / PARKING BRK.....	STOP / ON
ATC (VHF 1).....	NOTIFY
CABIN CREW (PA).....	ALERT
ΔP (only if MAN CAB PR has been used).....	CHECK ZERO
● If ΔP not at zero:	
CAB PR MODE SEL.....	MAN
V/S CTL.....	FULL UP
ALL ENG MASTERS.....	OFF
ALL FIRE pb (ENGs & APU).....	PUSH
ALL AGENTS (ENGs & APU).....	AS RQRD
■ If evacuation required:	
EVACUATION.....	INITIATE
■ If evacuation not required:	
CABIN CREW AND PASSENGERS (PA).....	NOTIFY

**Figura 4.16:** On Ground Emergency Evacuation checklist

La checklist dell'*On Ground Emergency Evacuation* per l'A320 è riportata in Figura 4.16.

In questa situazione si verifica un'emergenza al motore che costringe i piloti e i passeggeri ad evacuare l'aereo: si tratta di una *abnormal operation* da condurre solo in casi particolari. Entrambi i piloti hanno il controllo dei comandi; durante l'accelerazione del velivolo a terra, con i motori accesi, avviene un incendio al Motore 2.



**Figura 4.17:** Engine 2 Fire

Il pilota nota immediatamente l'ECAM Display che notifica l'emergenza al motore. La preparazione della scena della procedura, dunque, prevede la presenza della schermata dell'ECAM di Figura 4.17 con i motori già accesi e il *parking brake* settato su OFF, poiché il velivolo non è in situazione di parcheggio. Vengono settati tutti gli *Object* in maniera opportuna per l'esecuzione della procedura.

La procedura inizia con il velivolo già arrestato, poiché statica. Il pilota:

- Aziona il *parking brake*, ruotandolo su ON.
- Notifica l'ATC (*Air Traffic Control*) della situazione di emergenza, comunicando successivamente con il personale di bordo.
- Spegne il motore in avaria con l'apposito *Engine Master Switch*.

- Aziona il tasto *Engine Fire Button*, in seguito al sollevamento dell'apposita gabbia di protezione, che attiva due estintori con spruzzatori azionati elettricamente. Con l'attivazione dei due *Agents* viene permesso lo scarico di ogni estintore.

La procedura di emergenza con il motore in avaria può prevedere, a questo punto, due situazioni differenti:

1. Evacuazione non richiesta
2. Evacuazione richiesta

Se non fosse necessaria l'evacuazione, poiché l'estintore riesce a spegnere correttamente l'incendio nel motore, il pilota può proseguire nelle operazioni di volo *standard*.

Nella procedura implementata viene simulata la situazione di emergenza, che prevede anche l'evacuazione dal velivolo. Successivamente all'azionamento degli estintori, il pilota deve aspettare circa 30 secondi per valutare se l'incendio sia correttamente estinto, durante i quali effettua le seguenti ispezioni visive:

- Check della luce di emergenza, posizionata nel control pedestal panel, in posizione inferiore all'*Engine Master Switch*. Se la spia è ancora luminosa, l'emergenza sussiste.
- Check dell'ECAM Display. L'utente rileva ancora l'emergenza segnalata in Figura 4.17.
- Check dell'*Engine Fire Button*. Il tasto luminoso segnala che l'avaria del motore è ancora in corso e l'estinzione dell'incendio non si è verificata. Gli *Agents* illuminati indicano che la riserva di estinzione è esaurita e non c'è più possibilità di intervenire sull'incendio.

È necessaria l'evacuazione. Il pilota spegne anche il Motore 1, accertandosi che il *parking brake* sia innestato. Viene successivamente notificato all'ATC l'inizio della fase di evacuazione e il pilota attiva l'*Evacuation switch* a seguito del sollevamento dell'apposita gabbia di protezione. Gli assistenti di volo sono richiamati ad evacuare la cabina passeggeri, le luci di evacuazione lampeggiano sui pannelli e l'allarme sonoro di evacuazione è attivo.

Infine, il pilota si accerta nuovamente che gli *switch* del motore siano abbassati e conclude la procedura attivando i *Fire button* del Motore 1 e dell'APU. L'evacuazione in sicurezza è completata.

In Tabella 4.2 sono riportati i nodi della procedura di *On Ground Emergency Evacuation* implementata in WEAVR.

	OGGETTO	AZIONE
Nodo 1		Scene preparation
Nodo 2	Parking Brake	ON
Nodo 3		"Current state: ON"
Nodo 4		"Notify ATC"
Nodo 5	Cabin crew	Alert
Nodo 6		"Cabin crew at station"
Nodo 7	Engine 2 Master Switch	DOWN
Nodo 8		"Current state: OFF"
Nodo 9	Engine 2 Guard	UP
Nodo 10	Engine 2 Fire	PUSH
Nodo 11	Engine 2 Agents	ON
Nodo 12		"Wait 30 seconds"
Nodo 13	Engine 2 Fair Fail Light	Check
Nodo 14	ECAM	Check
Nodo 15	Engine 2 Fire	Check
Nodo 16		"Evacuation required"
Nodo 17	Engine 1 Master Switch	DOWN
Nodo 18		"Current state: OFF"
Nodo 19	Parking Brake	Check
Nodo 20		"Notify ATC"
Nodo 21	Evacuation Guard	UP
Nodo 22	Evacuation Switch	PUSH
Nodo 23	Cabin crew	Alert
Nodo 24		"EVACUATE"
Nodo 25	Horn Shut Off	PUSH
Nodo 26	Engine Masters	CHECK
Nodo 27	APU Guard	UP
Nodo 28	APU Fire	PUSH
Nodo 29	Engine 1 Guard	UP
Nodo 30	Engine 1 Fire	PUSH
Nodo 31	Checklist	
Nodo 32	End procedure	

**Tabella 4.2:** On Ground Emergency Evacuation procedure: nodi in WEAVR



# Capitolo 5

## Procedure in volo

Le procedure in volo sono operazioni dinamiche che il pilota esegue durante specifiche fasi del piano di volo. Esse necessitano dell'introduzione di un simulatore di volo, capace di rendere reale ed immersiva l'esperienza di training. Mentre le procedure a terra vengono eseguite senza la necessità di aggiornare i dati di volo, dunque senza il bisogno di introdurre un software di simulazione, le procedure dinamiche prevedono che l'utente interagisca coi dati di volo. Il flusso di dati viene generato da X-PLANE e il pilota esegue le procedure dinamiche in ambiente di realtà virtuale all'interno della piattaforma WEAVR.

La programmazione e la simulazione della missione reale dell'Airbus A320 vengono effettuate in X-PLANE. Nel Laboratorio di Simulazione del Volo del Politecnico di Torino è presente il simulatore fisico dell'A320 che consente di familiarizzare con l'ambiente procedurale e di valutare le potenzialità del Flight Control System di un velivolo commerciale [22]. L'ambiente di realtà virtuale è uno strumento innovativo che estende le potenzialità del simulatore procedurale, il quale non permette la simulazione di tutta la strumentazione di bordo e necessita della tecnologia VR per ricreare l'ambiente simulato e sfruttare gli equipaggiamenti non disponibili sul simulatore.

### 5.1 Piano di volo LIMC - LIMF

Il piano di volo è un componente cruciale affinché il computer di bordo del velivolo sia in grado di interagire con il pilota. Questo aspetto è ancora più rilevante per il sistema FBW<sup>1</sup> dell'Airbus A320. Tale tecnologia rappresenta una novità per l'aviazione civile, introdotta sin dalla presentazione dell'A320 ed in contrapposizione al suo gemello, il Boeing 737. Airbus ha implementato un sistema di comandi

---

<sup>1</sup>FBW: Fly By Wire

e attuatori controllati dall'impianto elettronico di bordo; i computer di bordo gestiscono numerosi aspetti del volo poiché i sistemi e i sottosistemi di bordo sono completamente digitalizzati e gestiti contemporaneamente dal pilota e dai cervelli elettronici di bordo [63]. Il sistema FBW dell'A320 utilizza cavi in fibra ottica per muovere le superfici alari collegate alla barra; i segnali elettrici, direttamente generati dal computer di bordo, percorrono la fibra ottica e raggiungono le superfici alari che vengono azionate con pistoni idraulici. Il pilota, muovendo la barra, genera un movimento registrato dal computer di bordo e tradotto in un segnale elettrico, che viene dunque inviato attraverso la fibra ottica. In questo modo il segnale è molto preciso e la stabilità del volo, controllata dall'autopilota, risulta migliorata.



Figura 5.1: MCDU A320

La pianificazione del volo avviene mediante la programmazione del **MCDU**, che costituisce l'interfaccia tra il pilota e il FMGS<sup>2</sup> e contiene un database con le

---

<sup>2</sup>FMGS: Flight Management and Guidance System

prestazioni del velivolo per consentire di stimare i consumi e i tempi necessari per il completamento della missione. IL FMS permette la guida verticale e laterale del velivolo, monitora l'inviluppo di volo e ottimizza la rotta; il computer di questo sistema si interfaccia con i sensori di bordo e trasmette i comandi che l'autopilota insegue. Queste informazioni vengono visualizzate dai piloti tramite gli appositi display. L'MCDU, rappresentato in Figura 5.1, contiene:

- **Display.** Riporta i dati relativi alle diverse fasi della missione.
- **Tastiera alfanumerica.** Permette al pilota di inserire i dati di interesse. Sono presenti anche i tasti per la selezione di determinate pagine dell'MCDU.

Si riportano le schermate caratteristiche affrontate nella programmazione completa dell'MCDU. Questa operazione viene effettuata manualmente per inserire in X-PLANE il piano di volo desiderato e per consentire all'utente di ripetere le procedure tramite il ricaricamento delle situazioni che vengono salvate inclusive del piano di volo per merito del plugin di *ToLiss*. Si tratta di un'operazione realizzata una sola volta senza la quale risulterebbe impossibile effettuare la missione.

**INIT** Prima dell'inizio del volo il pilota attiva la schermata *INIT*, riportata in Figura 5.2.

```

INIT
CO RTE          FROM/TO
NONE           LIMC/LIMF
ALTN/CO RTE
NONE
FLT NBR
AZ321
LAT            LONG
4537.0N       00844.2E
COST INDEX
30           WIND>
CRZ FL/TEMP   TROPO
FL100/-04.   36090
ENTRY OUT OF RANGE

```

**Figura 5.2:** INIT

Questa permette l'inizializzazione dell'intero piano di volo, che include la definizione degli aeroporti di partenza e di arrivo e la rotta da seguire. Vengono inseriti inoltre: numero del volo, quota di crociera desiderata e valore di *cost index*.

Quest'ultimo, per i velivoli della famiglia Airbus, rappresenta il rapporto tra il costo orario operativo del velivolo e il costo del combustibile e permette di dare un peso maggiore o minore al risparmio di combustibile durante la missione; un *cost index* basso corrisponde a velocità di salita e discesa più ridotte e a quote di volo maggiori, dando un peso maggiore al risparmio di combustibile.

Sono stati inseriti i seguenti dati:

- **FROM/TO:** LIMC/LIMF. Si sceglie di affrontare la missione che conduce all'aeroporto di Torino Caselle a partire dall'aeroporto di Milano Malpensa.
- **FLT NBR:** AZ321. Il numero di volo è facoltativo e può essere inserito casualmente in fase di simulazione.
- **COST INDEX:** 30. Il valore scelto è compreso nel range ammissibile e indica un peso maggiore verso il risparmio di combustibile.
- **CRZ FL:** FL100. Indica il livello di volo di crociera. Il *flight level* è un'indicazione di pressione espressa come altitudine sul livello del mare in unità di 100 piedi misurata secondo l'atmosfera ISA. Questa indicazione viene utilizzata dai controllori del traffico aereo per separare verticalmente gli aeromobili.
- **ALTN:** LIMA. Si tratta di un aeroporto alternativo in cui effettuare l'atterraggio. In questo caso è stato scelto l'aeroporto di Torino-Aeritalia.

Non vengono inseriti altri dati in questa sezione, ma si possono visualizzare quelli automaticamente riconosciuti dal computer muovendosi con le apposite frecce presenti nel tastierino.

**F-PLN** La modalità *F-PLN* consente al pilota di inserire i *waypoints* o le aerovie da inseguire durante il piano di volo. Definito l'aeroporto di partenza e di arrivo, in questa pagina (illustrata in Figura 5.3) il pilota indica le procedure **SID/STAR** per l'uscita dall'aeroporto di partenza e l'ingresso nell'aeroporto di arrivo.

- *SID - Standard Instrument Departure:* rotta di partenza strumentale che l'aeromobile deve seguire durante la fase di decollo.
- *STAR - Standard Arrival:* rotta di arrivo standard identificata in una procedura di avvicinamento tramite cui l'aereo procede dal punto di rotta al punto di avvicinamento [64].

SID e STAR hanno l'obiettivo di velocizzare il flusso del traffico aereo relativo a piste comuni attraverso l'utilizzo di percorsi specifici e limiti di velocità.

- **Departure from LIMC**



(a) Departure from LIMC

(b) Arrival at LIMF

Figura 5.3: F-PLN

- *Runway*: 35R
- *SID*: DOGU5L
- **Arrival at LIMF**
  - *Runway*: 36
  - *VIA*: TOP
  - *STAR*: TOP2A

**PERF** Le pagine delle prestazioni visualizzano i parametri delle prestazioni durante la fase di volo attualmente in corso. Si attiva mediante il tasto *PERF* ed è possibile visualizzare le schermate relative alle fasi precedenti e successive. In Figura 5.4 vengono riportate le schermate MCDU relative alle fasi di decollo (a) e di approccio (b).

- **TAKE OFF**: per questa fase si prevede l’inserimento delle velocità caratteristiche, la configurazione dei flap, l’altitudine alla quale avviene la transizione e l’inserimento delle temperature *FLEX*.

- $V_1 = 137$

Questo valore, espresso in nodi, rappresenta la *velocità di decisione*, ossia il limite oltre il quale il pilota deve necessariamente proseguire nel



(a) Takeoff

(b) Approach

**Figura 5.4:** PERF

decollo poiché un'eventuale arresto comporterebbe il fuoripista del velivolo, qualunque sia l'eventuale avaria riscontrata. Il valore inserito è coerente con i dati volo reali di un velivolo che parte dalla pista dell'aeroporto selezionato, ma può variare in base alle condizioni climatiche.

- $V_R = 141$

La *velocità di rotazione* è la velocità alla quale il pilota muove la barra per portare il velivolo all'incidenza di decollo. Tale valore deve necessariamente essere  $V_R \geq V_1$ , garantire sufficienti margini di sicurezza in caso di eventuali errori di pilotaggio ed essere inferiore alla velocità massima di rotolamento delle gomme.

- $V_2 = 144$

La *velocità di sicurezza al decollo* consente al velivolo di salire nella seconda parte di decollo con il gradiente stabilito dalla normativa ed è necessario che sia  $V_2 \geq 1.2V_{st}$ .

- $Flaps = 1$

La configurazione di decollo prevede i flaps estratti in prima posizione.

- *Transition altitude:* 6000

Rappresenta l'altitudine alla quale avviene la transizione all'atmosfera standard per la separazione verticale dei velivoli, seguendo i riferimenti delle altitudini. Il valore inserito, in piedi, è inferiore al livello di volo in crociera.

- *FLEX TO TEMP*: 43°C

Rappresenta la temperatura di decollo qualora questo avvenga con manetta in modalità *FLEX*. Questa temperatura definisce il livello di potenza erogata dal propulsore, che risulta inferiore alla potenza massima installata a bordo velivolo. In assenza di questo dato, è necessario procedere con il decollo in modalità *TOGA*. Solitamente si preferisce il decollo in modalità *FLEX* perché permette di ridurre le emissioni acustiche, evitare l'affaticamento delle gondole dei motori ed avere dunque vantaggio in termini di manutenzione.

- **APPR**: permette l'inserimento dei dati dell'aeroporto di destinazione (*QNH*, temperatura, intensità del vento), la configurazione dei flap e la *DH*<sup>3</sup>. Questi dati incidono nel calcolo delle velocità di approccio e atterraggio.

- *QNH* = 1013

Il valore di pressione atmosferica inserito è quello dell'atmosfera *ISA*.

- *TEMP* = 5°C

- *MAG WIND* = 0

L'intensità del vento è nulla poiché si suppone che sia assente.

- *DH* = 550

Tale valore è definito dall'ICAO come l'altezza più bassa durante la discesa in avvicinamento, viene inserito in piedi ed è misurato a partire dal livello del suolo.

L'inserimento di questa schermata nel MCDU viene effettuato durante la fase di crociera nella simulazione di missione condotta in questo lavoro. Questa fase non viene salvata, poiché si è ritenuto sufficiente inserire i parametri di *approach*, mettere in pausa la simulazione ad una distanza opportuna dall'aeroporto di destinazione e salvare la situazione utile per eseguire la procedura di avvicinamento evitando di simulare la crociera nei successivi voli.

## 5.2 Struttura del progetto

**Configurazione SimulationHub** SimulationHub viene utilizzato in questo progetto per permettere la comunicazione tra WEAVR e la simulazione del volo. La cartella principale di SimulationHub contiene le seguenti sotto-cartelle:

---

<sup>3</sup>DH: Decision Height

- *ClientUnity*: contiene l'eseguibile del *client Unity* e il relativo file *.bat* che definisce l'indirizzo del file ICD associato e il nome della *shared memory*.
- *ClientXPlane*: include l'eseguibile del codice del *client* implementato e il relativo file *.bat*, il quale specifica l'indirizzo del file ICD associato e l'identificativo della procedura da implementare. Poiché sono state implementate 3 procedure dinamiche nel codice, esse vengono identificate con i seguenti parametri interi:
  1. Takeoff
  2. Approach
  3. Landing

All'interno del codice è presente la lettura di questo dato, passato come argomento nella funzione *main*.

- *SimHub lib*. Contiene le librerie di SimulationHub divise nelle cartelle:
  - *include*: contiene i file *.h*, ossia l'intestazione delle funzioni contenute nella libreria. Tali file vengono inclusi nella dichiarazione del codice, mediante il comando di inclusione.
  - *lib*: include il corpo delle funzioni della libreria
  - *dll*: raggruppa i file *.dll*, ossia librerie dinamiche con codici utilizzabili da più programmi contemporaneamente.
- *ConfigFiles*. Include i file ICD (*.XML*) dei due client che comunicano con il server.
- *SimulationHubServer*: contiene l'eseguibile del server e il relativo file *.bat* che definisce l'indirizzo IP della macchina e della porta.
- *XPC lib*. Contiene l'*header* e il file sorgente di *X-plane Connect*.

**Struttura del Client *Unity*** WEAVR permette all'utente di effettuare una procedura, quando *Unity* è impostato in modalità *Game*. Ad ogni azione dell'utente corrisponde un invio di dati al server di SimulationHub. In maniera analoga, WEAVR legge un flusso di dati proveniente dal *client* opposto. Il *Client Unity* consente la comunicazione dei dati provenienti da WEAVR direttamente al server, il quale si occupa di trasmettere anche il flusso di dati in entrata. Questo *client* viene fornito come file eseguibile da TXT; il suo avviamento è previsto tramite un file *.bat*.

**Struttura del Client X-Plane** In questo lavoro di tesi il *Client X-Plane* è stato interamente sviluppato tramite codice C++ e permette di stabilire un collegamento diretto tra il simulatore di volo e il server di SimulationHub.

La sua struttura è la seguente:

- Inclusione delle librerie di sistema, di XPC e di SimulationHub.
- Dichiarazione delle funzioni presenti nel client:
  - *pause*: raggruppa il flusso di codice che permette la pausa della simulazione in X-PLANE. Infatti, qualora ad una determinata condizione di volo non corrisponda l'azione dell'utente sul relativo comando, la simulazione viene stoppata per consentire al pilota di eseguire il comando, dopo cui la simulazione riparte con l'inserimento del comando anche nel simulatore. In questo modo si evita che il pilota, tramite azioni sbagliate, condizioni l'andamento della simulazione in quanto le procedure implementate sono guidate.
  - *takeoff*: definisce la funzione per la procedura di decollo.
  - *approach*: dichiara la funzione relativa alla procedura di avvicinamento.
  - *landing*: inizializza la funzione associata alla procedura di atterraggio.

La struttura di queste funzioni prevede una programmazione sequenziale delle procedure operative in volo che l'utente si appresta ad affrontare in WEAVR. Vengono dunque implementati i vari step che, se correttamente eseguiti, sono passati alla simulazione e permettono di proseguire allo step successivo. Ogni procedura viene implementata in una funzione apposita e richiamata nel *main* in base alla scelta del parametro intero che l'utente inserisce all'avvio del file *.bat*.

- Creazione di una nuova interfaccia con SimulationHub tramite un puntatore a *simHub*.
- Connessione ad XPC (paragrafo 3.1.7) tramite la definizione dell'indirizzo IP del computer su cui viene eseguito il simulatore (*local host* 127.0.0.1) e creazione del socket tramite comunicazione UDP. Questo permette il totale controllo della simulazione del volo interamente da codice.
- Funzione **main**. È illustrata nello pseudo-codice riportato in Figura 5.5.

All'interno del corpo della funzione principale del codice sono presenti:

- Definizione delle variabili. Vengono utilizzate variabili di tipo *int* (numero associato alla procedura scelta, dimensioni dei *datarefs*, numero di *datarefs* in lettura), *float* (variabili di output WEAVR e valori numerici dei *datarefs*) e *char* (stringhe di *datarefs*).

- Connessione a SimulationHub tramite il richiamo delle funzioni dichiarate in *SimulationHubInterface.h*:
    - \* *Initialize*: lega le variabili, specificate dall'utente all'interno del file ICD, alle apposite etichette assegnate.
    - \* *GetLabel*: assegna ogni variabile del codice ad un'etichetta riconoscibile dal server di SimulationHub.
    - \* *Refresh*: aggiorna la lettura o la scrittura delle variabili.
- Questo permette di passare al server di SimulationHub tutte le variabili in lettura o in scrittura definite nel relativo file ICD.
- *switch* con multipli *case* che rimandano alla funzione associata alla procedura dinamica associata al parametro intero di scelta letto dal file *.bat*. Le specifiche funzioni verranno approfondite nei paragrafi dedicati alle procedure dinamiche.

**Struttura dei file ICD** In Figura 5.6 sono riportati i file ICD utilizzati. Si tratta di file che indicano l'elenco delle variabili a cui sono assegnate delle etichette e il loro tipo. Vengono inoltre indicati: nome del nodo, indirizzo IP della macchina, numero delle porte del client e del server di SimulationHub. L'elenco delle variabili si divide in:

- *Subscribed*: sono le variabili sottoscritte, che il client riceve in lettura.
- *Published*: sono le variabili pubblicate, che il client scrive.

**UNITY ICD** etichetta in lettura le variabili della simulazione e in scrittura le variabili relative all'interazione dell'utente con la cabina.

**XPLANE ICD**, viceversa, classifica come *Subscribed* le variabili da inviare a X-PLANE e come *Published* i dati della simulazione.

## 5.3 SimulationHub scripts

Per consentire la comunicazione da X-PLANE a WEAVR tramite SimulationHub si è reso necessario inserire degli scripts sviluppati in C# all'interno del progetto *Unity*.

**SimHubReader** Questo script si trova nella directory *Assets* ed è stato scritto per permettere la lettura e la memorizzazione in variabili dei dati di volo letti da X-PLANE.

La struttura del codice è la seguente:

```

int main(int argc, char** argv)
{
    //indice procedura passato dal file "Start Flight Procedure.bat"
    procedureChoice = atoi(argv[2]);

    //dichiarazione dei dataref da leggere in X-Plane
    datarefGet[datarefGetNum] =
    {
        //nomi dataref X-Plane
        "dataref_1";
        "dataref_2";
        ...
        "dataref_datarefGetNum";
    }

    //connessione a Simulation Hub e associazione delle etichette
    //del file XPLANE_ICD alle variabile interne al codice
    simHub->Initialize("path_ICD"); //path_ICD: indirizzo del file XPLANE_ICD

    simHub->GetLabel("labelName_1", variable_1, type_1);
    simHub->GetLabel("labelName_2", variable_2, type_2);
    ...
    simHub->GetLabel("labelName_n", variable_n, type_n);

    simHub->Refresh();

    //scelta della procedura da eseguire nel training VR
    switch (procedureChoice)
    {
        case 1:
            takeoff(datarefGet);
            break;

        case 2:
            approach(datarefGet);
            break;

        case 3:
            landing(datarefGet);
            break;
    }
}

```

Figura 5.5: Pseudo-codice: funzione *main*

- **Dichiarazione variabili.** Le variabili dichiarate si dividono in:
  - *public*: dati da visualizzare in *Unity* tramite stringhe inserite in un oggetto *Text*. Alcuni dati (velocità, radio altitude, stringhe del FMA<sup>4</sup>) vengono resi visibili in WEAVR in seguito all’attivazione della modalità *game* in quanto hanno lo scopo di informare il pilota delle condizioni di volo; altre variabili, invece, vengono scritte in *Unity* tramite stringhe ma non vengono visualizzate dal pilota (numero scelta procedura,  $V_{FE}$ , forze esercitate sul carrello) in quanto sono necessarie al soddisfacimento delle *Exit Condition*

<sup>4</sup>Flight Mode Annunciator

```
<?xml version="1.0" encoding="UTF-8"?>
<NodeConfiguration port="4900" address="127.0.0.1" name="UnityClient">
  <ServerConfig port="6500" address="127.0.0.1"/>
  - <SubscribedLabels>
    <label name="ProcedureChoice" description="Procedure choice" type="__int32"/>
    <label name="Altitude" description="Altitude [ft]" type="float"/>
    <label name="Airspeed" description="Airspeed [kts]" type="float"/>
    <label name="GearForce" description="Gear force [N]" type="float"/>
    <label name="V1" description="Decision Speed [kts]" type="float"/>
    <label name="VerticalSpeed" description="Vertical Speed [fpm]" type="float"/>
    <label name="APMode" description="Autopilot Lateral Mode" type="float"/>
    <label name="APPhase" description="Autopilot Phase" type="float"/>
    <label name="VFE" description="Flap Extended Velocity" type="float"/>
    <label name="QNH" description="QNH Atmosphere" type="float"/>
    <label name="WPT_CRS" description="Waypoint angle" type="float"/>
    <label name="RadioAltitude" description="Radio Altitude Height [ft]" type="float"/>
    <label name="VR" description="Rotate Speed" type="float"/>
  </SubscribedLabels>
  - <PublishedLabels>
    <label name="ParkingBrake" description="parking brake state" type="float"/>
    <label name="Flaps" description="flaps position" type="float"/>
    <label name="LandingGear" description="landing gear state" type="float"/>
    <label name="Spoiler" description="ground spoiler value" type="float"/>
    <label name="Thrust" description="thrust value" type="float"/>
    <label name="LatStick" description="lateral sidestick value" type="float"/>
    <label name="AutoPilot" description="autopilot state" type="float"/>
    <label name="AutoBrk" description="auto_brk_low state" type="float"/>
  </PublishedLabels>
</NodeConfiguration>
```

(a) UNITY ICD

```
<?xml version="1.0" encoding="UTF-8"?>
<NodeConfiguration port="4400" address="127.0.0.1" name="XPlaneClient">
  <ServerConfig port="6500" address="127.0.0.1"/>
  - <SubscribedLabels>
    <label name="ParkingBrake" description="parking brake state" type="float"/>
    <label name="Flaps" description="flaps position" type="float"/>
    <label name="LandingGear" description="landing gear state" type="float"/>
    <label name="Spoiler" description="ground spoiler value" type="float"/>
    <label name="Thrust" description="thrust value" type="float"/>
    <label name="LatStick" description="lateral sidestick value" type="float"/>
    <label name="AutoPilot" description="autopilot state" type="float"/>
    <label name="AutoBrk" description="auto_brk_low state" type="float"/>
  </SubscribedLabels>
  - <PublishedLabels>
    <label name="ProcedureChoice" description="Procedure choice" type="__int32"/>
    <label name="Altitude" description="Altitude [ft]" type="float"/>
    <label name="Airspeed" description="Airspeed [kts]" type="float"/>
    <label name="GearForce" description="Gear force [N]" type="float"/>
    <label name="V1" description="Decision Speed [kts]" type="float"/>
    <label name="VerticalSpeed" description="Vertical Speed [fpm]" type="float"/>
    <label name="APMode" description="Autopilot Lateral Mode" type="float"/>
    <label name="APPhase" description="Autopilot Phase" type="float"/>
    <label name="VFE" description="Flap Extended Velocity" type="float"/>
    <label name="QNH" description="QNH Atmosphere" type="float"/>
    <label name="WPT_CRS" description="Waypoint angle" type="float"/>
    <label name="RadioAltitude" description="Radio Altitude Height [ft]" type="float"/>
    <label name="VR" description="Rotate Speed" type="float"/>
  </PublishedLabels>
</NodeConfiguration>
```

(b) XPLANE ICD

Figura 5.6: File ICD utilizzati

ma non influenzano le azioni del pilota, che viene guidato dalla sequenza dei nodi della procedura.

– *float*: variabili necessarie allo script associate a dati numerici che non

vengono visualizzati in *Unity*. Sono valori di lettura necessari al soddisfacimento di determinate condizioni all'interno del codice.

- **Funzione Update.** È chiamata una volta per ogni frame e consente la lettura dei valori indicati nei file ICD. Per la visualizzazione nei *Text* è necessario convertire il valore letto in una stringa. All'interno di questa funzione sono presenti vari comandi *switch*.
  - **Modalità autopilota.** Questo *switch* prevede la scrittura della stringa corrispondente al valore numerico della modalità di autopilota letta tramite *dataref* da X-PLANE.
  - **Fase autopilota.** Analogamente, lo *switch* sulla fase dell'autopilota consente di stampare a video la stringa corrispondente al valore numerico letto tramite *dataref*.
  - **Scelta della procedura.** Si basa sulla lettura del dato numerico intero inserito nel file *.bat* relativo alla scelta della procedura da seguire. Nel caso di procedura di **decollo** vengono scritti i valori di  $V_1$ ,  $V_R$ ,  $V_{FE}$  e velocità verticale tramite stringhe. Al verificarsi di determinate condizioni su questi valori viene stampata una stringa su un'apposita casella di testo *Unity* che viene confrontata con una stringa inserita nella relativa *Exit Condition* del nodo della procedura; l'uguaglianza di queste due stringhe garantisce il soddisfacimento della condizione imposta e consente la prosecuzione della procedura. Nel caso di procedura di **atterraggio** il codice permette la lettura dei dati da X-PLANE relativi alla forza esercitata sulle ruote del carrello e alla radio altitudine. Tali dati vengono convertiti in stringhe, scritti in caselle di testo *Unity* e utilizzati per il soddisfacimento di apposite condizioni sia nel codice che nei nodi, in maniera analoga al decollo.

**SimulationHubIcd** Contiene la dichiarazione delle variabili pubbliche in accordo con la divisione *Subscribed - Published* del file **UNITY\_ICD**. Il nome delle variabili corrisponde all'etichetta assegnata ai dati nel file ICD.

**SimulationModel** Permette la dichiarazione delle strutture *Subscribed* e *Published* in lettura e scrittura, contenendo il modello di simulazione allocato in memoria condivisa.

L'abilitazione in WEAVR del pugin relativo alla simulazione consente la creazione del **Simulation Framework** tra gli oggetti della gerarchia. Nell'*Inspector* del Simulation Framework è necessario includere gli scripts *SimulationModel* e *SimHubReader* come componenti aggiuntivi. Il *SimHubReader* necessita inoltre

l'abbinamento delle caselle di testo associate alle corrispondenti variabili dichiarate nel codice.

## 5.4 Takeoff

La checklist della procedura di decollo implementata è riportata in Figura 5.7; si sceglie di eseguire la modalità di *takeoff standard* con intensità del vento inferiore ai 20 nodi in direzione  $X_w$ .

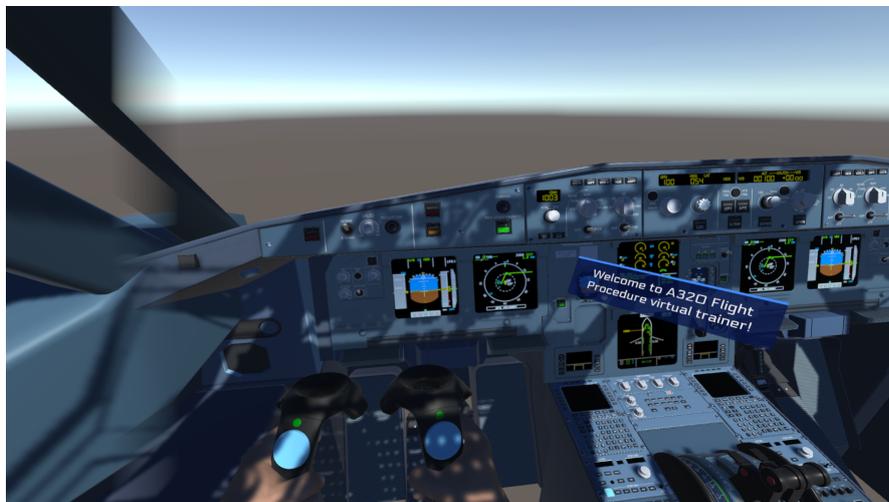
	PF	PM
<b>Takeoff</b> <sup>1</sup>	Announce – "TAKEOFF"	
<b>Standard Takeoff</b> <sup>2</sup> X-wind < 20 & No Tailwind	<ul style="list-style-type: none"> <li>• Thrust – N1 50% (1.05 EPR)</li> <li>• Brakes – Release</li> <li>• Thrust – FLX / TOGA <sup>3</sup></li> <li>• Sidestick – ½ FWD up to 80, Neutral by 100 <sup>4</sup></li> </ul>	Chrono – Start
<b>Standard Takeoff</b> <sup>2</sup> X-wind > 20 or with Tailwind	<ul style="list-style-type: none"> <li>• Thrust – N1 50% (1.05 EPR)</li> <li>• Brakes – Release</li> <li>• Thrust – N1 70% – FLX / TOGA <sup>5</sup></li> <li>• Sidestick – Full FWD up to 80, Neutral by 100</li> </ul>	
<b>PFD / ND</b>	Announce – FMA	Monitor <sup>6</sup> <ul style="list-style-type: none"> <li>• FMA – On PFD</li> <li>• FMS Position – On ND</li> </ul>
<b>80 Knots</b>		Check – Takeoff N1(EPR) Announce – "THRUST SET" Monitor – PFD & ENG Indications <sup>7</sup>
<b>100 Knots</b>	Check	Announce – "1 HUNDRED KNOTS"
<b>At V1 &amp; VR</b>		Announce – "V1" & "ROTATE"
<b>Positive Climb</b>	Order – "L/G UP"	Announce – "POSITIVE CLIMB"
<b>Thrust Reduction Altitude</b> <sup>8</sup>	Thrust Levers – CL	Packs – On (if applicable) <sup>9</sup>
<b>Acceleration Altitude</b>	Target Speed – Check FMA – Monitor <sup>10</sup>	
<b>F Speed</b> <sup>11</sup>	Flaps 1 – Order	Flaps 1 – Select
<b>S Speed</b>	Flaps 0 – Order	Flaps 0 – Select Spoilers – Disarm Ext Lights – Set <sup>12</sup>

Figura 5.7: Procedura di decollo

Per procedere all'esecuzione della procedura di decollo è necessario effettuare manualmente i seguenti passaggi:

- Avvio della modalità *game* in *Unity* quando l'utente indossa correttamente il visore ed accende i joystick.
- Avvio del server di SimulationHub tramite file *.bat*.
- Avvio del *Client Unity* tramite file *.bat*.
- Avvio del *Client X-Plane* tramite il file *.bat*.

**Preparazione della scena** Il nodo iniziale del *Procedure Editor* in WEAVR prevede una schermata di benvenuto per l'utente.



**Figura 5.8:** Welcome billboard

In Figura 5.8 è rappresentata la billboard che, sincronizzata con un messaggio vocale che ne riproduce il contenuto del testo, accoglie l'utente all'avvio della modalità di *game*. Vengono disabilitati momentaneamente i *Text* sovrapposti sul PFD e la procedura prosegue verso un nodo di divisione che porta al *takeoff* o al *landing*, in base alla condizione di uscita del nodo successivo. Infatti, per confermare la scelta della procedura di decollo, è necessario inserire due condizioni di uscita:

- $ProcedureChoice = 1$

Riguarda il valore intero di scelta della procedura inserito come ultimo argomento del file *.bat* di avvio del *Client X-Plane*.

- $V_1 = 137$

È una condizione necessaria a rafforzare il codice per assicurarsi che in memoria non sia letto casualmente il valore unitario di *ProcedureChoice*, che avvierebbe la procedura senza l'avvio effettivo del *Client X-Plane*. La condizione di lettura della  $V_1$  garantisce che il *Client X-Plane* sia effettivamente avviato poiché il valore letto corrisponde al dato inserito nel MCDU.

Il flusso della procedura, in caso di soddisfacimento delle condizioni di uscita, prosegue verso il nodo di **preparazione della scena** che prevede il posizionamento dei comandi di manetta, barra e leve di flap, parcheggio, spoiler e carrello in maniera coerente con X-PLANE. Vengono inoltre attivati i *Text* che permettono all'utente di visualizzare i dati di volo: airspeed,  $V_1$ , vertical speed, modalità e fase



(a) WEAVR



(b) X-PLANE

**Figura 5.9:** Welcome takeoff procedure

dell'autopilota e dell'autothrust. L'utente è accolto da una billboard, sincronizzata con un messaggio vocale, di benvenuto nella procedura di decollo (Figura 5.9). Non vengono inserite condizioni di uscita.

Nel *Client X-Plane*, in seguito alla lettura del *ProcedureChoice*, il codice entra nella funzione **takeoff**, che prevede l'inizializzazione di *datarefs* e comandi utilizzati. Si procede dunque alla lettura dei valori di  $V_1$  e  $V_R$  che vengono passati tramite SimulationHub a WEAVR.

**Step 0: Throttle 1/4** La prima azione dell'utente è l'azionamento della manetta, che viene portata leggermente in avanti per alzare il numero di giri del motore fino al 50% (1.05 EPR) in accordo con la *checklist* della procedura.



(a) WEAVR



(b) X-PLANE

**Figura 5.10:** Step 0: Throttle 1/4

Il pilota viene guidato da una billboard ad azionare la manetta, come mostrato in Figura 5.10. Un messaggio vocale è sincronizzato al testo della billboard. In questo nodo è già presente l'integrazione in WEAVR dei dati di volo di X-PLANE: infatti l'utente visualizza sul PFD i dati necessari alla procedura. La condizione da soddisfare in questo nodo riguarda il comportamento della manetta. Quest'ultima

viene configurata con un comportamento a stati, in maniera coerente con la manetta dell'Airbus A320; quando l'utente posiziona la manetta sullo stato corretto, la procedura prosegue verso il nodo successivo. In uscita dal nodo è prevista la rimozione delle billboard presenti.

Nel *Client X-Plane* è invece presente un ciclo **while** impostato sul numero di step complessivi. La programmazione sequenziale della procedura inizia con il *case 0* in cui viene inviato ad X-PLANE il *datarefs* contenente il valore della manetta. Se questa operazione risulta correttamente completata, il codice stampa in *console* l'avvenuta esecuzione del comando e viene incrementato il *procedureStep* che consente di proseguire al *case* successivo.

**Step 1: Sidestick 1/2 forward** In accordo con la *checklist* della procedura di decollo, l'utente viene ora invitato a muovere la barra in avanti. Si tratta di un'operazione utile ad aumentare il carico sulla ruota anteriore del carrello per migliorarne la capacità di sterzata e per conferire al velivolo un controllo positivo in imbardata.



**Figura 5.11:** Step 1: Sidestick 1/2 forward

L'utente riceve il messaggio di testo riprodotto su una billboard (come in Figura 5.11) e per mezzo di un audio. La barra, a cui viene conferito un comportamento a stati, deve essere mossa in avanti dall'utente; ciò costituisce una *Exit Condition* per il nodo che, se correttamente verificata, permette alla procedura di passare allo step successivo, rimuovendo tutte le billboard.

Nel *Client X-Plane* questa operazione viene programmata in un *case* analogo al precedente: viene inviato al simulatore il dato numerico di posizionamento

della barra tramite un *dataref*, incrementato il *procedureStep* e stampato a video il comando eseguito.

**Step 2: Parking Brake OFF** Quando il numero di giri del motore raggiunge il valore stabilito dalla procedura, l'utente deve rimuovere il freno di parcheggio.



**Figura 5.12:** Step 2: Parking Brake OFF

Il *parking brake* è dotato di un comportamento a due stati e l'utente lo disinnescia attraverso un click sul grilletto del controller. In Figura 5.12 viene illustrata la billboard che indica all'utente di rimuovere il freno di parcheggio; anche in questo caso il testo viene riprodotto da un messaggio vocale. Quando l'utente cambia lo stato del comando la condizione di uscita viene verificata. Sul PFD è possibile visualizzare i dati di volo: quando il comando di rimozione del *parking brake* viene correttamente ricevuto da X-PLANE, il dato di airspeed si aggiorna e si modifica in tempo reale.

Nel *Client X-Plane* questo step viene eseguito tramite l'invio del *datarefs* di impostazione del dato numerico del freno di parcheggio (0: OFF, 1: ON) nell'apposito *case*. Successivamente il *procedureStep* viene incrementato ed è possibile visualizzare la stampa a video di avvenuta disattivazione del freno di parcheggio.

**Step 3: Throttle FLX** È un passaggio analogo allo step 0. In seguito alla rimozione del freno di parcheggio, il velivolo aumenta la airspeed e si prepara a raggiungere i valori caratteristici di velocità per il decollo. Diventa dunque necessario che l'utente porti la manetta al massimo. In accordo con le procedure

commerciali, si sconsiglia di portare la manetta in TOGA<sup>5</sup>; per preservare lo stato del motore è preferibile decollare in modalità FLEX<sup>6</sup>.



(a) WEAVR



(b) X-PLANE

Figura 5.13: Step 3: Throttle FLX

L'utente viene guidato da una billboard (Figura 5.13) e da un messaggio audio di impostare la manetta in modalità FLX. Questo corrisponde ad impartire due

<sup>5</sup>Takeoff and Go Around:  $throttle = 1$

<sup>6</sup>Flex Takeoff con spinta ridotta:  $throttle = 0.87$

scatti alla manetta:

- $1/4 \rightarrow \text{CLB}$
- $\text{CLB} \rightarrow \text{FLX}$

La sincronizzazione con il simulatore consente all'utente di avvertire il suono dei motori che incrementano il numero di giri e di visualizzare l'aumento progressivo della airspeed sul PFD. La condizione di uscita dal nodo prevede che l'utente abbia portato la manetta allo stato richiesto, mentre in uscita è prevista la rimozione della billboard.

Nel *Client X-Plane* questa operazione viene gestita in maniera analoga al *case 0*. Il *dataref* relativo all'impostazione della manetta consente di portare la manetta allo stato successivo qualora venisse inviato il valore numerico 1. Poiché il pilota in questo step esegue due scatti in avanti della manetta, il codice prevede:

- Inizializzazione del valore numerico pari a 0.25 per lo stato della manetta, corrispondente al valore precedentemente settato.
- Invio del *dataref* con valore 1 per consentire alla manetta di portarsi in posizione CLB.
- Salvataggio del valore numerico corrispondente alla modalità CLB.
- Invio del *dataref* con valore 1 per consentire alla manetta di passare dallo stato CLB allo stato FLX.

Questa operazione si è rivelata necessaria in quanto, in seguito ad alcuni test, è emerso che X-PLANE non memorizza correttamente il valore numerico corrispondente alla modalità della manetta. Forzando il codice al salvataggio in memoria del dato numerico della manetta, il comando funziona correttamente. Viene successivamente incrementato il *procedureStep* e stampato a video il comando eseguito.

**Step 4: Sidestick Neutral** È un'operazione che si effettua al raggiungimento della condizione di velocità per preparare l'aereo alla rotazione. Questo step include dunque due nodi nel *Procedure Editor* di WEAVR:

- Nodo di raggiungimento della condizione di velocità. L'utente osserva l'incremento della velocità sul PFD fino al valore  $V = 100\text{kts}$ . Quando questa *Exit condition* risulta soddisfatta, un messaggio vocale annuncia "100 knots". Questa azione è tipica della procedura di Airbus, mentre Boeing avvisa i piloti al raggiungimento di 80 nodi.

- Nodo di azione sulla barra. L'utente è guidato da una billboard (Figura 5.14) a muovere la barra in posizione neutra, poiché si è raggiunta la velocità di riferimento. La *Exit condition* è impostata sul raggiungimento dello stato desiderato della barra. All'uscita dal nodo vengono rimosse tutte le billboard.



Figura 5.14: Step 4: Sidestick Neutral

Il *case* associato a questo step nel *Client X-Plane* è strutturato come segue:

- Inserimento in X-PLANE di una pausa al raggiungimento di  $V = 100kts$  tramite il richiamo della funzione *pause*, che invia il *dataref* di inserimento della pausa al simulatore. Si tratta di un comando utilizzato per prevedere la sincronizzazione con il comando eseguito dall'utente in WEAVR per un migliore controllo della simulazione di volo.
- Invio del *dataref* di impostazione del valore numerico della barra in seguito alla rimozione della pausa nella simulazione. Si procede dunque con l'incremento del *procedureStep* e la stampa a video del comando impartito.

**Step 5: Rotate** Si giunge dunque allo step in cui il pilota, agendo sulla barra, porta l'aereo in rotazione. Anche in questo caso l'utente esegue questa operazione al soddisfacimento delle opportune condizioni di velocità.

Questo step viene articolato in WEAVR mediante la creazione di 3 nodi.

- Nodo di raggiungimento della  $V_1$ . Al verificarsi della condizione  $V \geq V_1$  l'utente ascolta l'annuncio "V1" e il superamento della velocità di decisione (mostrata sul PFD mediante la lettura dai dati di volo del MCDU di X-PLANE) implica che l'utente deve necessariamente proseguire nel decollo.

- Nodo di raggiungimento della  $V_R$ . Si tratta della condizione che, se verificata, conduce il flusso della procedura nel nodo successivo. La velocità di rotazione viene letta dal MCDU.
- Nodo di azione sulla barra. Un messaggio vocale annuncia "ROTATE", mentre la billboard riportata in Figura 5.15 avvisa l'utente di ruotare la barra per consentire il distacco del velivolo dal suolo. La condizione di uscita riguarda il corretto raggiungimento dello stato della barra, a cui segue l'azione di rimozione delle billboard.

La struttura del *case* nel *Client X-Plane* è analoga allo step precedente. Al raggiungimento della condizione  $V \geq V_R$  viene inviato il comando di pausa alla simulazione per consentire all'utente di agire sulla barra; successivamente il codice rimuove la pausa e invia al simulatore il *dataref* di rotazione della barra. Il *procedureStep* viene incrementato e compare a video la stampa di avvenuta rotazione.

**Step 6: Gear UP** Al raggiungimento di un rateo di salita positivo il pilota ritrae il carrello. Questo step viene articolato in due nodi poiché è necessario verificare la condizione di avvenuto raggiungimento del *positive climb*, ossia un rateo di salita positivo.

- Nodo di raggiungimento di *positive climb*. Contiene la condizione di superamento del valore di velocità verticale con l'inserimento di un margine per garantire conservatività. Il valore del rateo di salita viene mostrato tramite un *Text* sul PFD, in posizione coerente con il reale PFD dell'A320, che ottiene il dato in lettura dalla simulazione di volo.
- Nodo di azione dell'utente sulla leva del carrello. In Figura 5.16 è illustrata la billboard con cui l'utente è informato di eseguire la retrazione del carrello. Quest'ultima è caratterizzata da un comportamento a due stadi, quindi la condizione di uscita prevede che sia impostata su UP.

Nel *Client X-Plane* viene scritto un codice che prevede la pausa della simulazione al raggiungimento del valore stabilito di rateo di salita positivo. Successivamente la pausa viene rimossa e il codice invia al simulatore il comando di retrazione del carrello. Il *case* si conclude con l'incremento del *procedureStep* e con la stampa a video di avvenuta retrazione del carrello.

**Step 7: AP1 ON** In questo step il pilota inserisce l'autopilota in seguito alla condizione di navigazione. Vengono inclusi i seguenti nodi.

- AP Mode. Si tratta del nodo di verifica del raggiungimento della modalità NAV indicata sul PFD. Questa modalità viene comunicata dal simulatore. Il



(a) WEAVR



(b) X-PLANE

**Figura 5.15:** Step 5: Rotate

pilota deve assicurarsi che la modalità di navigazione sia raggiunta prima di innescare l'autopilota.

- AP 1 ON. Viene mostrata all'utente una billboard (Figura 5.17 a) in cui si richiede l'attivazione dell'autopilota. Il tasto dell'autopilota è dotato di comportamento *executable*, quindi la condizione di uscita riguarda l'avvenuto innesto del comando. In Figura 5.17 (b) viene mostrato lo stato attuale dell'autopilota in seguito alla sua attivazione; questo è visibile anche sul Flight Mode Annunciator del PFD.



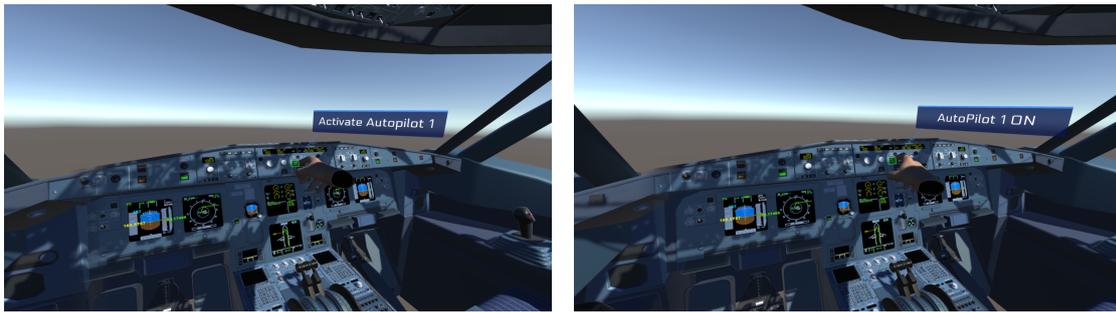
**Figura 5.16:** Step 6: Gear UP

Nel *Client X-Plane* il codice permette la pausa della simulazione quando si raggiunge la condizione di navigazione. Successivamente la pausa viene rimossa e il codice invia al simulatore il *dataref* relativo all'innesto dell'autopilota. Nel simulatore (Figura 5.17 c) è possibile visualizzare il corretto inserimento dell'autopilota sia sul PFD che sull'apposito tasto del FCU, che si illumina in seguito all'attivazione; quest'ultimo dettaglio non è visibile in WEAVR a causa dell'impossibilità di modificare la texture del modello grafico del cockpit dell'Airbus A320 di TXT, che prevede il tasto AP1 spento.

**Step 8: Throttle CLB** In fase di salita è necessario adeguare il nuovo livello di spinta dei motori. La modalità della manetta viene impostata su CLB e mantenuta in questa condizione per tutta la durata della missione fino alla fase di atterraggio. Per impostare la manetta in questa modalità, il pilota deve ricevere l'input dal PFD. Tale step include i seguenti nodi.

- Nodo di condizione sulla fase dell'autopilota. Nel Flight Mode Annunciator viene stampata a video la richiesta "CLB" per invitare l'autopilota ad impostare la conseguente modalità di spinta.
- Nodo di azione del pilota sulla manetta. Viene illustrata una billboard (Figura 5.18) che avvisa all'utente della nuova modalità di spinta richiesta. La condizione in uscita prevede che lo stato della manetta sia impostato correttamente.

Il codice del *Client X-Plane* è strutturato in maniera analoga allo Step 3: inizializzazione del valore della posizione delle leve della manetta in FLX e invio



(a) WEAVR - Activate AP1

(b) WEAVR - AP1 ON



(c) X-PLANE

**Figura 5.17:** Step 7: AP1 ON

del *dataref* di riduzione della manetta. Viene successivamente incrementato il *procedureStep* e stampato a video il comando eseguito.

**Step 9: Remove Flaps** Durante la fase di salita il pilota deve rimuovere i flap. Questa azione si verifica al raggiungimento della velocità di estensione del flap  $V_{FE}$ , ossia la massima velocità che il velivolo può raggiungere con i flap estesi.

- Nodo di raggiungimento della velocità  $V_{FE}$ . Viene utilizzato un *dataref* che stabilisce il prossimo indicatore sulla scala di velocità del PFD del capitano. L'utente osserva l'andamento della velocità; al verificarsi della condizione per cui il valore del *dataref* risulta minore di un margine la procedura prosegue nel nodo successivo.



**Figura 5.18:** Step 8: Throttle CLB

- Nodo di impostazione della leva del flap. L'utente viene guidato tramite la billboard rappresentata in Figura 5.19 alla rimozione del flap. Il velivolo decolla con i flaps settati a 0.25 e durante la salita vengono rimossi. La condizione di uscita riguarda il corretto raggiungimento dello stato della leva, che ha un comportamento di tipo *N-way switch*.



**Figura 5.19:** Step 9: Remove Flaps

Il codice del *Client X-Plane* che controlla la simulazione impone inizialmente una pausa al verificarsi della condizione su  $V_{FE}$ . Successivamente viene rimossa

la pausa e il simulatore riceve il comando di rimozione del flap tramite l'apposita funzione di *X-Plane Connect*.

**Step 10: Ground Spoiler Disarm** Al termine della fase di decollo il pilota disarma i ground spoiler. Si tratta dell'ultima operazione della procedura, che avviene senza alcuna condizione. Il velivolo sta effettuando la fase di salita.



(a) WEAVR



(b) X-PLANE

**Figura 5.20:** Step 10: Ground Spoiler Disarm

La billboard illustrata in Figura 5.20 invita l'utente a disattivare i ground spoiler, la cui leva è dotata di comportamento a stati. La condizione finale prevede

il corretto posizionamento della leva, a cui segue la rimozione della billboard precedentemente illustrata.

All'interno del *Client X-Plane* viene comandato l'invio del *dataref* di impostazione del valore 0 agli *speed brakes*. L'incremento del *procedureStep* permette la conclusione della procedura nel simulatore e l'uscita dal ciclo **while** della programmazione sequenziale. La stampa di avvenuta disattivazione dei ground spoiler consente la chiusura della schermata *console*. In Figura 5.20 (b) viene illustrata la schermata finale della procedura di decollo con le ultime operazioni correttamente completate nel simulatore.

**End Procedure** L'ultimo nodo della procedura serve per informare l'utente che ha correttamente completato le operazioni della *checklist* di decollo, come mostrato in Figura 5.21.



**Figura 5.21:** End procedure

La procedura viene correttamente eseguita.

**Riepilogo della procedura** In Tabella 5.1 si riporta un riepilogo degli step della procedura implementati in WEAVR e nel *Client X-Plane*.

In Figura 5.22 viene illustrato lo pseudo-codice utilizzato per la programmazione della procedura di decollo nel *Client X-Plane*.

## 5.5 Approach

La procedura di avvicinamento viene realizzata solo all'interno del *Client X-Plane* ed è simulata in X-PLANE in seguito al caricamento della situazione *approach*, che

	OGGETTO	AZIONE
Step 0	Throttle	N1 50% (1.05 EPR)
Step 1	Sidestick	Move 1/2 FWD
Step 2	Parking Brake	OFF
Step 3	Throttle	FLEX
Step 4	Sidestick	Move Neutral
Step 5		"ROTATE"
Step 6	Landing Gear	UP
Step 7	Autopilot	ON
Step 8	Throttle	CLIMB
Step 9	Flaps	REMOVE
Step 10	Ground Spoilers	DISARM

**Tabella 5.1:** Step della procedura di decollo

contiene il piano di volo correttamente impostato per le fasi di avvicinamento e di atterraggio. La programmazione sequenziale della procedura di *approach* contiene i seguenti step, eseguiti all'interno di un ciclo **while** impostato in maniera analoga alla funzione relativa al decollo. Ciascuno step prevede l'invio di *dataref* o comandi al simulatore tramite *X-Plane Connect*.

- **Step 0: Flap 1/4.** Al verificarsi della condizione di raggiungimento per l'estensione dei flap, il codice invia al simulatore il comando di estendere i flap alla prima posizione. Il pilota ha dunque terminato la fase di crociera condotta senza l'ausilio di ipersostentatori.
- **Step 1: Flap 1/2.** Analogamente, dopo un determinato intervallo di tempo, il pilota soddisfa la condizione di velocità tale da garantire l'estensione dei flap alla seconda posizione.
- **Step 2: QNH 1013.** In questo step viene azionato il selettore barometrico per impostare il riferimento di pressione al valore standard. Tale valore è inserito nel MCDU.
- **Step 3: Activate approach phase** È l'azione con cui il pilota attiva la fase di approccio azionando l'apposito pulsante sul FCU. Al simulatore viene inviato l'apposito comando da codice.
- **Step 4: Gear down.** L'estensione del carrello avviene attraverso l'invio al simulatore dell'apposito comando.

```

void takeoff(datarefGet[])
{
    //indice step in corso
    procedureStep = 0;

    //dichiarazione dei dataref da scrivere in X-Plane
    datarefSent[] =
    {
        //nomi dataref X-Plane
        "dataref_1";
        ...
        "dataref_n";
    }

    //dichiarazione dei command da inviare a X-Plane
    command[] =
    {
        //nomi command X-Plane
        "command_1";
        ...
        "command_n";
    }

    //procedura
    while (procedureStep < numTotStep)
    {
        //lettura dataref X-Plane
        getDREF(datarefGet[i]);

        simHub->Refresh();

        //step procedurali
        switch (procedureStep)
        {
            case 0:
                if (flightCondition)
                {
                    pause(1);           //attivazione pausa
                    sleep(seconds);     //pausa esecuzione codice
                    pause(0);           //disattivazione pausa

                    sendDREF(datarefSent[k]); //modifica dataref oppure
                    sendCOMM(command[k]);    //invio comando

                    procedureStep++;
                    std::cout << "consoleMessage"; //stampa a video
                }

                break;

            case 1:
                ...
            case numTotStep-1:
        }
    }
}

```

Figura 5.22: Pseudo-codice: funzione *takeoff*

- **Step 5: Ground Spoiler armed.** Vengono successivamente armati i ground spoiler, attraverso l'invio al simulatore dello stesso *dataref* utilizzato per la procedura di decollo, al valore -0.5.
- **Step 6: Autobrake Low ON.** In questo step il codice procede all'attivazione degli autobrakes attraverso l'invio al simulatore dell'apposito comando.
- **Step 7: RWY Turnoff Light.** Anche l'attivazione delle luci di spegnimento

pista del carrello principale avviene attraverso l'invio ad X-PLANE dell'apposito comando.

- **Step 8: Landing lights.** Le luci di atterraggio vengono azionate tramite comando; quest'ultimo viene ripetuto per le luci posizionate su entrambi i lati del velivolo.
- **Step 9: Nose landing lights.** Le luci anteriori di atterraggio vengono accese nuovamente tramite l'invio del comando al simulatore.
- **Step 10: Flap 3/4.** Terminate le precedenti operazioni, il velivolo è in grado di rispondere positivamente all'estensione completa dei flap. Si procede gradualmente all'attivazione della terza posizione della leva degli ipersostentatori.
- **Step 11: Flap 1.** La condizione sulla  $V_{FE}$  viene soddisfatta anche per l'estensione completa dei flap in quarta posizione.
- **Step 12: Autopilot 2 ON.** L'approccio termina con la preparazione all'atterraggio strumentale. Il pilota aziona entrambi gli autopiloti per garantire ridondanza al velivolo.

## 5.6 Landing

La procedura di *landing* è accuratamente spiegata nell'elaborato di tesi del mio collega Giovanni Sgarra.

In Tabella 5.2 si riportano gli step implementati in WEAVR e nel *Client X-Plane* per la corretta esecuzione della procedura di atterraggio riportata in Figura 5.23.

	PF	PM
<b>Approx. 30 Feet</b>	Flare – Perform <sup>2</sup> Thrust – Idle <sup>3</sup>	Attitude – Monitor <sup>4</sup>
<b>Touchdown <sup>5</sup></b>	Reverse – Idle or Max	Spoilers – Check / Announce: "SPOILERS / NO SPOILERS" <sup>6</sup>  Reversers – Check Announce: "REVERSE GREEN / NO REVERSE"
<b>Brakes</b>	As Required <sup>7</sup>	Deceleration – Check / Announce: "DECCEL / NO DECCEL"
<b>70 Knots</b>	Thrust Levers – Reverse Idle	"70 KNOTS" – Announce <sup>8</sup>
<b>Taxi Speed</b>	Reversers – Stow <sup>9</sup>	
<b>Before 20 Knots</b>	Autobrake – Disengage	

Figura 5.23: Procedura di atterraggio

	OGGETTO	AZIONE
Step 0	Throttle	REVERSE
Step 1	Throttle	REVERSE IDLE
Step 2	Auto BRK LOW	DISENGAGE
Step 3	Throttle	IDLE

**Tabella 5.2:** Step della procedura di atterraggio

Il settaggio della scena è concorde con il termine della procedura di avvicinamento. Il pilota si prepara ad effettuare l'atterraggio strumentale con entrambi gli autopiloti inseriti. I dati di volo letti da WEAVR tramite *Text* e visualizzabili sul PFD sono:

- **Radio Altitude:** rappresenta l'altitudine di riferimento in fase di atterraggio.
- **Airspeed:** è un'indicazione necessaria per le velocità caratteristiche successive al *touch down* del velivolo.
- **Modalità e fase dell'autopilota:** permette di azionare la fase di *landing* e di riconoscere il *roll out* del velivolo a terra.
- **Inserimento dell'auto-thrust.** Consente di regolare la spinta in base al livello di manetta inserito.

Al verificarsi della condizione  $F_{lg} \neq 0$  (forze sulle ruote del carrello non nulle) il pilota aziona la manetta in modalità REV<sup>7</sup> per effettuare una importante decelerazione su pista. La comunicazione con il simulatore conferisce al pilota una completa immersività nella simulazione: l'audio di X-PLANE consente al pilota di avere una totale percezione della riduzione di altitudine e di velocità. Il *flare* viene automaticamente effettuato per merito dei due autopiloti innestati in fase di avvicinamento e del piano di volo correttamente inserito. In seguito al *touch-down* sulla pista, il velivolo inizia la fase di *roll out*, che viene visualizzata sul PFD del pilota.

Durante la decelerazione viene annunciato il raggiungimento dei 70 nodi e il pilota muove nuovamente la manetta, portandola in modalità REV IDLE<sup>8</sup>. Quando il velivolo raggiunge  $V = 30kts$  l'utente disattiva l'Autobrake Low, mentre al raggiungimento del valore  $V = 10kts$  è necessario impostare la manetta in posizione IDLE, a cui corrisponde una spinta nulla.

<sup>7</sup>Reverse: livello di spinta massimo negativo  $throttle = -1$

<sup>8</sup> $throttle = -0.1$



## Capitolo 6

# Conclusioni

Il progetto di tesi realizzato nasce dalla volontà di esplorare il settore del *Training&Simulation* dei piloti aeronautici e si inserisce nella tematica di addestramento del personale per mezzo della realtà virtuale. Il lavoro intende dunque essere incentrato sulla branca della simulazione del volo, che sfrutta la tecnologia VR per conferire ai piloti un grado di formazione efficiente e superiore alla tecnica tradizionale CBT.

TXT Group, mediante la piattaforma PACELAB WEAVR, propone tecniche di addestramento basate sulla ripetizione di procedure operative da parte dell'utente. Il lavoro di tesi condotto sfrutta queste tecniche di *training* abbinate alla simulazione del volo: per rendere efficace l'apprendimento di procedure operative dinamiche da parte del pilota si rende necessaria la fruizione di un software di simulazione capace di interagire con l'ambiente di realtà virtuale in cui l'utente esegue le procedure guidate.

L'implementazione delle **procedure a terra** di *Engine Start* e *On Ground Emergency Evacuation* all'interno di questo lavoro di tesi ha un duplice scopo: ampliamento della rete di procedure a terra implementate nei lavori di tesi precedentemente condotti con TXT ed acquisizione di familiarità con WEAVR al fine di apprendere tutti gli ingredienti necessari alla realizzazione di una procedura.

I limiti del simulatore procedurale sono stati superati attraverso la digitalizzazione dell'intero cockpit dell'Airbus A320 con tecnica virtual reality. Ciascun componente del cockpit è stato ricreato in un modello tridimensionale rappresentato in *Unity* a cui viene associato un determinato oggetto in modo univoco. Le procedure statiche implementate permettono l'interazione con il cockpit stesso, tramite sequenze di nodi WEAVR che includono sequenze di azioni eseguite in *Unity* quando l'utente interagisce con gli elementi del modello 3D.

Il risultato finale permette di guidare l'utente nell'esecuzione delle operazioni necessarie al completamento delle procedure. All'interno dell'immagine 3D fornita

dal visore vengono mostrati dei box e delle frecce che riportano suggerimenti sulle operazioni da svolgere, oltre a operazioni sui punti in cui trovare i selettori o i display di interesse.

Le procedure a terra circoscrivono la programmazione al solo utilizzo di *Unity* e WEAVR. L'interazione con la cabina di pilotaggio è tuttavia completa: l'utente ha piena sensibilità, nei limiti della tecnologia VR a disposizione, di tutte le operazioni che effettua e dei comandi con cui si interfaccia. Il risultato si può considerare soddisfacente poiché il pilota riesce ad avere un ottimo controllo degli eventi, è maggiormente coinvolto nella scena ed è in grado di concentrarsi più approfonditamente sulla procedura al fine di migliorare l'efficacia dell'apprendimento, come riportato nell'analisi del questionario della tesi [1]. Vi è dunque un superamento, supportato dall'avanzamento tecnologico che prevede una sempre maggiore diffusione delle tecniche di realtà virtuale nel quinquennio 2022-2027, della tradizionale tecnica di apprendimento *computer based*. La bontà del risultato raggiunto è accompagnata da un'ottima usabilità dell'ambiente WEAVR. Nell'implementazione delle procedure a terra non è stato infatti necessario realizzare degli script in linguaggio C# poiché il plugin si prestava ad essere intuitivo nella sua interfaccia grafica.

Questo lavoro di tesi ha permesso il superamento di alcune criticità emerse nell'analisi di valutazione finale dei lavori precedenti, riportate in [1] e riguardanti le procedure statiche:

- **Possibilità di richiamare le billboard.** Tale limite condiziona l'utente che non riesce ad individuare prontamente l'istruzione. Per consentire all'utente di visualizzare le billboard senza limiti di tempo si è ritenuto sufficiente eliminare il *timeout* di comparsa delle billboard, che vengono rimosse solo in seguito all'azione dell'utente sul comando indicato.
- **Mancanza di indicazioni sul cockpit A320.** Si tratta di un limite grafico, prontamente risolto alla richiesta di un nuovo modello di cockpit A320 a TXT. Tale modello si è rivelato necessario anche per l'aggiunta del sidestick che non è incluso nel modello utilizzato per le procedure a terra. Il nuovo modello include anche una texture completa dell'interno della cabina.
- **Incremento del numero di nodi della procedura.** La procedura *On Ground Emergency Evacuation* include un numero abbondante di nodi e si presenta come la procedura statica più lunga implementata. Per aumentare la difficoltà del *training* si ritiene opportuno procedere al test delle procedure in volo.

Il questionario di autovalutazione delle procedure implementate in realtà virtuale non è stato condotto per via del persistere del quadro epidemiologico legato all'emergenza sanitaria. Non sarebbe stato infatti apprezzabile un riscontro di

eventuali miglioramenti rispetto al test condotto nel precedente lavoro di tesi [1], a cui si rimanda per la visione del risultato generale limitato ad un campione ristretto di utenti.

Lo step successivo del lavoro si incentra sulla realizzazione delle **procedure in volo**. Si tratta di un passaggio fondamentale per la formazione del pilota. Mentre le procedure a terra vengono implementate con l'obiettivo di fornire al pilota un grado di familiarità elevata con il cockpit, le procedure di volo abbinano questo proposito alla formazione del pilota durante la missione: l'utente, al termine delle procedure dinamiche, deve aver acquisito un grado di confidenza tale da riconoscere i principali dati di volo che condizionano le azioni sui comandi della cabina.

Per la digitalizzazione delle procedure in volo è stato necessario ricorrere a molteplici software, capaci di fornire una comunicazione diretta tra la simulazione di volo e l'ambiente di realtà virtuale. *X-Plane Connect* viene utilizzato per condurre la missione all'interno di X-PLANE. Il *Client X-Plane* viene elaborato per garantire una connessione con il *server* di SimulationHub, il quale permette il passaggio dei dati al *Client Unity*. È stato necessario realizzare diversi codici per consentire ai vari ambienti di operare correttamente; parte del percorso formativo di questo lavoro di tesi ha dunque previsto l'apprendimento di nozioni basilari di programmazione C++ e C#.

Il progetto di realizzazione delle procedure di volo può essere ulteriormente migliorato ed ampliato in futuro. È infatti possibile proporre alcune future modifiche da apportare alla realizzazione delle procedure e all'analisi dei risultati, al fine di conferire valore all'intero progetto sviluppato.

1. **Comunicazione.** La realizzazione delle procedure dinamiche si basa su una necessaria comunicazione tra la simulazione del volo e l'ambiente di realtà virtuale in cui il pilota si forma. In questo progetto di tesi la comunicazione è unilaterale: i dati di interesse della simulazione del volo arrivano all'interno di WEAVR. Ciò è possibile attraverso lo script di lettura dei dati di interesse, che fa fede ai file ICD comunicati al server. I dati ritenuti fondamentali per la formazione del pilota nell'esecuzione delle procedure vengono stampati a video in realtà virtuale tramite caselle di testo che leggono dallo script. Si tratta dunque di una comunicazione unilaterale X-PLANE  $\rightarrow$  WEAVR. Qualsiasi tipo di comunicazione in WEAVR viene effettuata tramite l'aggiunta del *Cockpit Element*; nel *binding* di questo componente è possibile scegliere la modalità di comunicazione (lettura o scrittura). La versione di WEAVR fornitaci da TXT non garantisce il corretto funzionamento della modalità di scrittura, che impedisce la comunicazione bi-laterale X-PLANE  $\Leftrightarrow$  WEAVR. Questo limite viene attualmente superato inserendo una pausa nel *Client X-Plane* al verificarsi della condizione per cui l'utente effettua una determinata azione; tale

condizione viene comunicata in WEAVR, l'utente esegue il comando in VR ed il *Client X-Plane* invia il *dataref* o comando al simulatore. La sincronizzazione viene tuttavia garantita, ma è auspicabile un miglioramento relativo alla scrittura in modo che ogni comando dell'utente sia corrisposto ad una variabile scritta in WEAVR, pubblicata dal relativo ICD e letta nel *Client X-Plane* che invia il comando alla simulazione solo al verificarsi della condizione di lettura.

2. **Dimensioni della cabina.** Il nuovo modello di cockpit dell'Airbus A320, fornito da TXT, ha riscontrato un lieve problema di dimensioni e si presenta più piccolo sia della cabina reale che del modello utilizzato nelle procedure statiche. Questo potrebbe non conferire all'utente un corretto realismo della situazione.
3. **Questionario di valutazione.** Per stabilire numericamente l'influenza della digitalizzazione delle procedure dinamiche è auspicabile condurre un'attività di indagine basata su un questionario in cui l'utente esprime il proprio grado di coinvolgimento, la controllabilità degli oggetti virtuali e la stimolazione sensoriale in base agli input conferiti dalla procedura e dai dati di volo. È auspicabile che l'indagine preveda un campione sufficiente di utenti che testino la simulazione VR e il tradizionale metodo CBT al fine di provare l'efficienza del metodo sperimentato in questo progetto di tesi.
4. **Flussi video.** Non è incluso nell'obiettivo di questo progetto di tesi, ma uno step successivo del lavoro può incentrarsi sulla realizzazione di un simulatore VR che mostri a video la schermata del mondo esterno alla cabina e dei display completamente interattivi.
5. **Creazione di un'app di simulazione per il training.** Al termine dell'implementazione di tutte le procedure dinamiche, si può pensare di realizzare un'app a ciclo chiuso, mediante l'inserimento di un tablet in maniera simile alle procedure a terra. Tale tablet eliminerebbe l'inserimento manuale del numero intero *ProcedureChoice* nel file *.bat* di avvio del *Client X-Plane* e consentirebbe al pilota di scegliere la procedura da effettuare; in questo modo WEAVR dovrebbe scrivere il numero di *ProcedureChoice* e passarlo al client. Questa operazione non è tuttavia eseguibile attualmente per due limiti principali:
  - Impossibilità di usare il *write* nel *binding*.
  - Impossibilità di aprire X-PLANE da codice e di caricare l'apposita situazione automaticamente in background, poiché è necessario utilizzare il plugin di *ToLiss* manualmente.

Si tratta di un suggerimento futuro che renderebbe concluso il prodotto e fruibile come applicazione per la formazione globale del pilota.

Il risultato raggiunto dal lavoro di tesi consente al pilota di effettuare correttamente le procedure di decollo e atterraggio seguendo i dati della simulazione di volo e rispettando le condizioni imposte poiché, essendo una procedura guidata, si è scelto di implementare gli step in modo che all'utente non sia concessa la possibilità di errore, tale da condizionare il controllo della simulazione. Da questo progetto emerge l'importanza della simulazione per la formazione dei piloti. In seguito all'apprendimento delle procedure dinamiche in realtà virtuale, il pilota dovrebbe essere in grado di immergersi nel simulatore fisico dell'Airbus A320 e di effettuare le procedure con un maggior grado di consapevolezza dei comandi da eseguire. Con il perfezionamento della tecnologia VR nei prossimi anni sarà inoltre possibile effettuare un vero e proprio confronto realistico tra un training in realtà virtuale ed uno con il simulatore fisico del velivolo.

Si ritiene infatti che la realtà virtuale costituisca un potenziale strumento per la formazione globale del personale aeronautico, visto il grado di diffusione di questa tecnologia nel corso degli anni e dei vantaggi che ne conseguono.



# Appendice A

## Cockpit A320

Vengono illustrati i pannelli principali dell'Airbus A320 [65] e descritti tutti i comandi con cui il pilota interagisce nell'esecuzione delle procedure implementate in questo progetto di tesi. La strumentazione viene suddivisa in blocchi per facilitare l'individuazione degli equipaggiamenti necessari.

### A.1 Control pedestal



Figura A.1: Control pedestal panel 3D

Il **Control pedestal Panel** si trova al centro tra i due piloti.

Contiene i seguenti comandi, utilizzati nelle procedure ed evidenziati in Figura A.2.

- **Engine Key.** È contenuto nell'ECAM Control Panel, che include i selettori del sistema ECAM per la visualizzazione riportata sul System display. Viene utilizzato per la visualizzazione del motore durante l'accensione automatica o manuale (Figura A.2 b).
- **Engine Mode Selector.** Si tratta di un interruttore a 3 porte (Figura A.2 c):
  - *Crank*: la start valve si apre se l'interruttore di accensione manuale viene utilizzato, con  $N_2$  inferiore al 10%. Non si verifica l'accensione.
  - *Norm*: consente il funzionamento quando il motore è acceso.
  - *IGN/Start*: fornisce una accensione continua se abbinata ai master.
- **Engine Master Switch.** Si tratta di un interruttore a 2 porte:
  - *OFF*: le fuel valves si chiudono, il motore si spegne e i canali del FADEC vengono resettati.
  - *ON*: apre le fuel valves e avvia l'accensione automatica o manuale del motore.

È presente un interruttore per ogni motore (Figura A.2 c).

- **Parking brake.** È una manopola che necessita la rotazione in senso orario per l'attivazione (Figura A.2 d). Quando il parking brake viene attivato, gli altri modi di frenata sono automaticamente disabilitati.
- **Speed brake.** È una leva che consente di ridurre la velocità in volo. Ad altitudini elevate i freni di velocità non dovrebbero essere estesi più della metà. Nella procedura dinamica del *Decollo* questa leva si utilizza per disattivare i *Ground Spoilers*. Alzando la leva del freno di velocità si armano gli spoiler a terra che massimizzano la frenata e scaricano il peso del velivolo sul carrello. Possono essere utilizzati anche per massimizzare la resistenza aerodinamica, come nel caso dell'atterraggio, in cui essi sono già attivi. La leva è raffigurata in Figura A.2 e.
- **Flap lever.** La leva dei flaps (Figura A.2 f) impartisce a flaps e slats il funzionamento simultaneo verso la posizione comandata. La leva deve essere estratta dal fermo prima dell'utilizzo. Quando la leva è diversa dalla posizione neutral, vengono azionati flap e slat tramite i sistemi idraulici ed elettrici. La leva consente di muovere la posizione di flaps e slats in 4 posizioni differenti, nei quali è presente un incastro per il posizionamento.

- **Throttle.** La manetta dell'A320, situata nel cuore del control pedestal panel e rappresentata in Figura A.3, è costituita da due leve che regolano il livello di spinta di ciascun propulsore. Le leve possono essere mosse solo manualmente; il loro spostamento induce un segnale al FADEC che ne calcola la valutazione associata alla spinta. Se le leve sono posizionate in un fermo, il FADEC calcola la spinta associata a quel fermo, mentre se la manetta è posizionata tra due fermi il FADEC seleziona il limite corrispondente al fermo superiore. Ci sono 5 posizioni definite:
  - Takeoff/Go-Around **TOGA**
  - Flex/Max Continuous Thrust **FLX/MCT**
  - Climb **CL**
  - Forward **IDLE**
  - Max Reverse **REV FULL**

Per correttezza, si sottolinea che Airbus utilizza il sistema Auto-Thrust anziché il sistema Auto-Throttle sfruttato da Boeing. L'Auto-Thrust si attiva prima del decollo, quando vengono definite le velocità caratteristiche del decollo (dipendenti dalla pista, dalle condizioni atmosferiche e dal velivolo) e il piano di volo.

- **Sidestick.** La barra laterale viene installata da Airbus (Boeing monta degli Yokes) per permettere al pilota di eseguire le manovre con una sola mano, consentendo un'esperienza più confortevole con maggiori spazi per interagire con i sistemi di bordo. In Figura A.4 si riporta il modello del sidestick dell'A320. Si tratta di un componente fondamentale per impartire i comandi alle superfici primarie e favorire il movimento nei piani longitudinale e latero-direzionale del velivolo.

## A.2 Overhead panel

L'**Overhead Panel** include i selettori dell'interfaccia con i sistemi di bordo. Dell'overhead panel sono stati utilizzati i seguenti tasti, evidenziati in Figura A.5:

- **Engine Fire Button.** Viene rivestito da una gabbia di protezione (*guard*). Quando rilasciato, l'*engine fire* illumina la spia al suo interno (Figura A.5 b), silenzia l'allarme, disattiva l>IDG <sup>1</sup>, interrompe l'alimentazione del FADEC e

---

<sup>1</sup>IDG: Integrated Drive Generator

chiude le valvole di sicurezza. Gli Engine Fire Button sono affiancati dagli *Engine Agents*. Ogni motore è dotato di due estintori azionabili in caso di incendio ed alimentati elettricamente. Gli Agents permettono lo scarico dell'estintore nel tentativo di estinguere l'eventuale incendio.

- **Engine Manual Starter.** È un interruttore rivestito da una gabbia di protezione (Figura A.5 c) che permette l'avviamento manuale del motore a terra.
  - *ON*: la start valve si apre se l'Engine Mode Selector è in uno stato differente dal NORM ed  $N_2$  è inferiore al 15%. Si attiva una spia blu luminosa per confermarne lo stato di accensione.
  - *OFF*. La start valve del motore è chiusa.
- **Exterior Lights.** Utilizzate nella procedura di approccio, le luci esterne consentono all'aereo di essere visibile dalle torri di controllo per il decollo e l'atterraggio. Il pilota, nella procedura di *approach*, aziona i seguenti interruttori, raffigurati in Figura A.5 (d):
  - *Runway Turnoff Light Switch*: la posizione *ON* illumina le luci di spegnimento pista montate sul carrello anteriore. Esse si disattivano automaticamente quando il carrello viene retratto.
  - *Landing Lights Switch*: sono luci di atterraggio che vengono azionate in coppia poiché sono poste su due lati. Quando sono impostate su *ON*, le luci di atterraggio garantiscono un'illuminazione completamente estesa in prossimità del carrello posteriore.
  - *Nose Selector Light Switch*: permette l'accensione delle luci di decollo e taxi in maniera coerente con la modalità impostata sullo switch.
- **Emergency evacuation panel.** Si tratta del pannello utilizzato nella procedura di *On Ground Emergency Evacuation*, rappresentato in Figura A.5 (e). In particolare, in questa procedura, il pilota utilizza i seguenti tasti:
  - *Evacuation switch*: protetto dalla apposita *Evacuation Guard*, questo interruttore avvisa gli assistenti di volo di evacuare la cabina passeggeri, azionando i segnali di evacuazione mediante apposite luci che lampeggiano sui pannelli di prua e poppa degli assistenti. La spia di questo tasto lampeggia in rosso quando azionata su *ON*, accompagnata da un allarme acustico. L'avviso di evacuazione può essere annullato premendo una seconda volta sull'interruttore.
  - *Horn Shut Off*: viene azionato dal pilota per silenziare il suono di evacuazione nell'abitacolo.

## A.3 Instrument panel

L'**Instrument Panel** è composto da un *Center Panel* per la visualizzazione dei display costituenti il *glass cockpit* e da un *Glareshield* per l'esecuzione dei comandi di FCU ed EFIS.

- Nelle procedure a terra e in volo è fondamentale che il pilota impari a leggere correttamente gli **ECAM Display**, evidenziati in Figura A.6 a. I dati visualizzabili sui Display sono relativi a:
  - Indicazioni sui parametri di motore e fuel
  - Avvisi di avvertenza o segnalazioni di guasti, divisi in:
    - \* *Livello 3*: richiesta azione immediata, segnalazione in rosso
    - \* *Livello 2*: avvertimento all'equipaggio, segnalazione in arancio
    - \* *Livello 1*: monitoraggio, colorazione in arancio
  - Dati di volo permanenti
  - Posizione di flap e slat

Il pilota può visualizzare fino a 12 pagine di sistema sull'ECAM Display.

- L'Instrument Panel contiene, tra i comandi utilizzati nelle procedure implementate, anche la leva del **Gear Lever** (Figura A.6 b). Questa fornisce segnali elettrici alle unità di interfaccia di controllo del carrello di atterraggio che controllano l'alimentazione idraulica.
  - la posizione **UP** ritrae il carrello di atterraggio e ne permette lo stivaggio in seguito al decollo.
  - La posizione **DOWN** abbassa il carrello di atterraggio nella fase di *approach* che precede l'atterraggio.
- Il **Flight Control Unit - FCU** permette l'interfaccia tra pilota e sistema di navigazione del velivolo (Figura A.6 c). Di questo pannello vengono utilizzati i seguenti pulsanti:
  - **Autopilot**. I due autopiloti dell'A320 attivano o disattivano l'autopilota selezionato in base alle modalità di guida riportate in FCU e MCDU. Gli autopiloti utilizzano timone, alettoni ed equilibratore per ottenere il profilo di volo desiderato. Vengono utilizzati nelle procedure dinamiche di decollo, approccio e atterraggio; in particolare, qualora fosse in procinto di effettuare un avvicinamento strumentale, il pilota dovrebbe azionare i due autopiloti simultaneamente per garantire maggiore ridondanza al velivolo.

- **APPR Button.** Questo pulsante viene azionato durante la procedura di *avvicinamento*. Consente di attivare le modalità LOC<sup>2</sup> e G/S<sup>3</sup> in caso di atterraggio ILS tramite piano di volo o le modalità APP NAV, FINAL e FINAL APP in caso di *non precision approach*.

La gestione del FCU può avvenire in modalità *MANAGED* (in cui il pilota insegue il piano di volo inserito nel sistema FMGS) e *SELECTED* (in cui l'autopilota insegue i target di velocità e altitudine inseriti dal pilota).

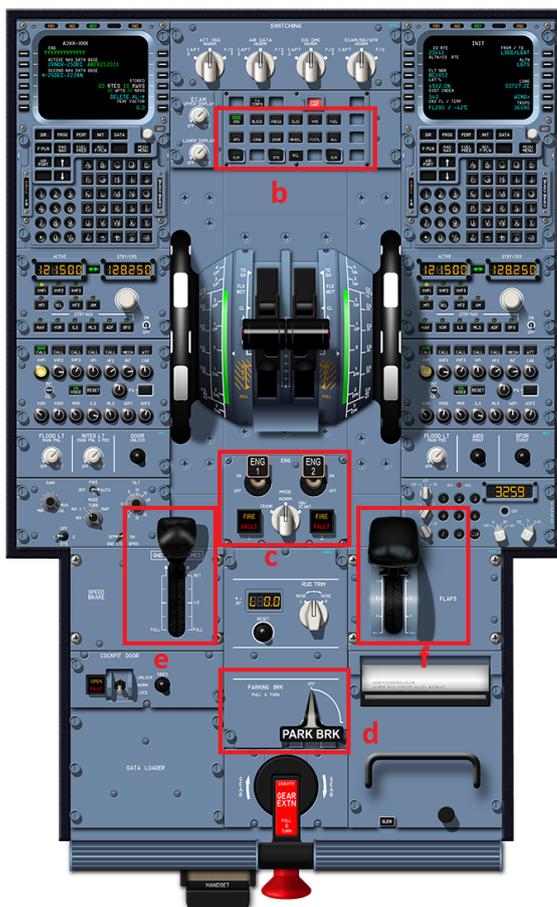
- In Figura A.6 (d) vengono evidenziati gli interruttori del freno automatico: **AUTO/BRK**. Per l'atterraggio si seleziona un Auto-Brake LOW, che fornisce la velocità di decelerazione selezionata attraverso una pressione progressiva ai freni per 4 secondi dopo l'apertura del Ground Spoiler. Il livello massimo viene solitamente utilizzato al decollo. Quando sono attivati, una spia blu ne indica lo stato di accensione; in seguito all'atterraggio essi verranno spenti dal pilota.
- L' **EFIS Control Panel** (Figura A.6 e) viene utilizzato per il controllo di PFD e ND. Il pilota, in fase di avvicinamento, utilizza il **selettore barometrico** per selezionare il riferimento barometrico. Tirando il selettore in fase di *approach*, il pilota imposta due modalità sul PFD:
  - *QNH 1013*: per impostare il riferimento barometrico a 1013 hPa.
  - *STD*: per utilizzare il riferimento barometrico dell'atmosfera standard.
- **PFD: Primary Flight Display** importa tutte le indicazioni di base del velivolo (Figura A.6 f).
  - Sull'orizzonte artificiale il pilota legge tutti i dati di **assetto**. Il **Flight Director** mostra le barre relative agli angoli di rollio e assetto desiderati per mantenere il velivolo nella traiettoria di interesse.
  - Nella parte inferiore è presente il dato di **imbardata**.
  - Ai lati dell'orizzonte artificiale sono presenti le indicazioni della **velocità indicata** (a sinistra), dell'**altitudine** e del **rateo di salita o discesa** del velivolo (a destra).
  - In alto a destra sono riportati gli stati di funzionamento di **automanetta** e **autopilota**.

---

<sup>2</sup>LOC: Localizer per fornire una guida azimutale

<sup>3</sup>G/S: Glideslope per definire il corretto profilo di discesa verticale

- In fase di atterraggio si visualizzano indicazioni sullo stato di funzionamento del sistema ILS in basso a sinistra.
- La parte superiore del PFD è chiamata **Flight Mode Annunciator** e si divide in 5 colonne che riportano: stato di funzionamento dell'automanetta, modi di controllo dell'autopilota nel piano verticale e laterale, indicazioni per la fase di avvicinamento e stato dell'autopilota.
- Accanto al PFD è presente il **ND: Navigation Display**. Riporta le informazioni di navigazione e rotta (Figura A.6 g). Le informazioni visualizzate da questo pannello dipendono dalla modalità EFIS selezionata.
  - *NAV*: velivolo centrato sulla rotta impostata. È una modalità utilizzata per decollo e atterraggio e permette di visualizzare una vista d'insieme.
  - *PLAN*: scena centrata sui waypoints. È una modalità utilizzata durante la programmazione del MCDU per verificare che le informazioni inserite nel computer di bordo siano correttamente riportate.
  - *ARC*: mostra il velivolo nella rotta con un campo di vista limitato. È una modalità utilizzata dal pilota durante l'esecuzione della missione.



(a) Control pedestal 2D



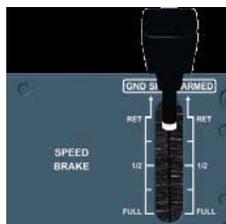
(b) Engine key



(c) Mode selector e switch



(d) Parking brake



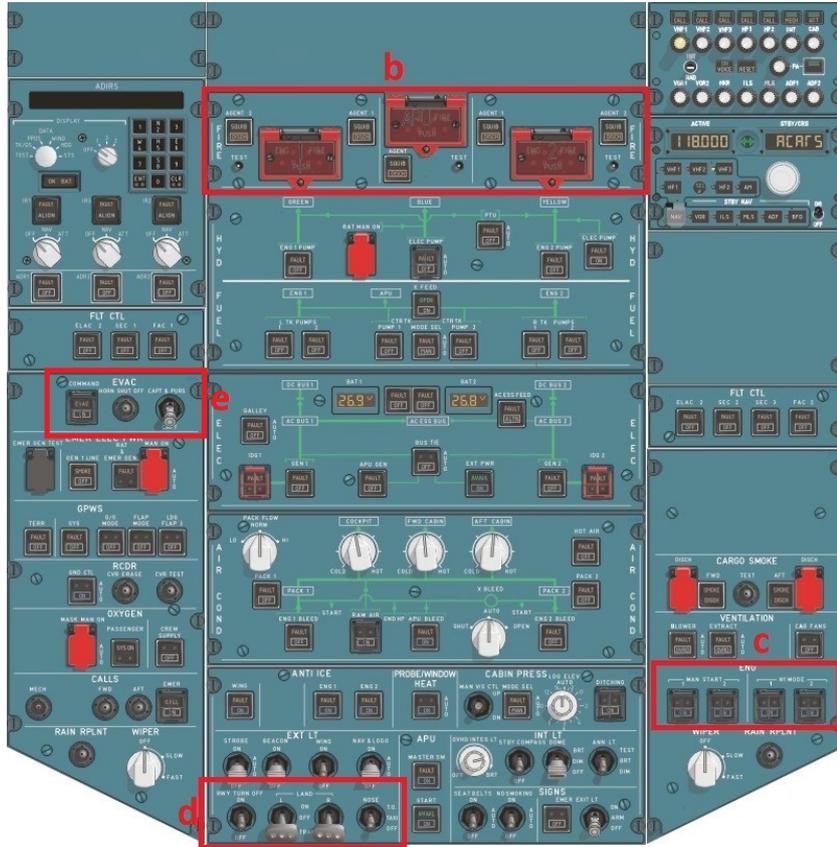
(e) Speed Brakes



(f) Flap lever

Figura A.2: Control pedestal panel

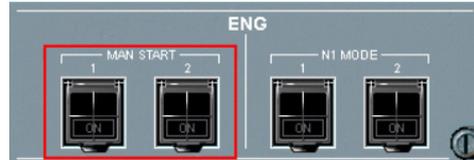




(a) Overhead panel 2D



(b) Engine e APU fire



(c) Engine manual starter



(d) Exterior Lights



(e) Emergency evacuation panel

Figura A.5: Overhead panel



(a) Instrument panel 2D



(b) Gear lever



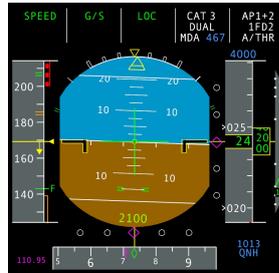
(c) FCU



(d) AutoBrake



(e) EFIS



(f) PFIS



(g) ND

Figura A.6: Instrument panel



# Bibliografia

- [1] Giuseppe Brizzi. «Digitalizzazione e analisi dell'addestramento del personale aeronautico attraverso gli strumenti della realtà virtuale= Digitalisation and analysis of the training of aircraft personnel through the use of virtual reality instruments». Tesi di laurea mag. Politecnico di Torino, 2020 (cit. alle pp. iii, 3, 43, 96, 97).
- [2] Andrea Pavone. «Digitalizzazione e analisi di procedure operative aeronautiche per l'addestramento di piloti in ambiente di realtà virtuale.= Digitization and analysis of aeronautical operating procedures for the training of pilots in a virtual reality environment.» Tesi di laurea mag. Politecnico di Torino, 2020 (cit. alle pp. iii, 3, 43).
- [3] Frederic Vianale. «Analisi di procedure operative per l'addestramento del personale aeronautico in ambiente di realtà virtuale= Analysis of operative procedures for aeronautical personnel in virtual reality environment». Tesi di laurea mag. Politecnico di Torino, 2020 (cit. alle pp. iii, 3, 43).
- [4] Lorenzo Fava. «Digitalizzazione di procedure operative per l'addestramento del personale aeronautico in ambiente di realtà virtuale.= Digitalization of operating procedures for the training of aviation personnel in a virtual reality environment.» Tesi di laurea mag. Politecnico di Torino, 2020 (cit. alle pp. iii, 3, 43).
- [5] Peter A Hancock. «The future of simulation». In: *Human factors in simulation and training* (2009), pp. 169–186 (cit. a p. 1).
- [6] Barry H Kantowitz. «Selecting measures for human factors research». In: *Human factors* 34.4 (1992), pp. 387–398 (cit. a p. 1).
- [7] Christina S Morris, Peter A Hancock e Ed C Shirkey. «Motivational effects of adding context relevant stress in PC-based game training». In: *Military Psychology* 16.2 (2004), pp. 135–147 (cit. a p. 1).
- [8] Clark Aldrich. *Simulations and the future of learning: An innovative (and perhaps revolutionary) approach to e-learning*. John Wiley & Sons, 2003 (cit. a p. 1).

- 
- [9] Matthias Oberhauser e Daniel Dreyer. «A virtual reality flight simulator for human factors engineering». In: *Cognition, Technology & Work* 19.2 (2017), pp. 263–277 (cit. alle pp. 1, 2, 19).
- [10] Brian D Kelly. «Flight Deck Design and Integration for Commercial Air Transport». In: *Human Factors for Civil Flight Deck Design* (2004) (cit. a p. 1).
- [11] Alexandra D Kaplan, Jessica Cruit, Mica Endsley, Suzanne M Beers, Ben D Sawyer e PA Hancock. «The effects of virtual reality, augmented reality, and mixed reality as training enhancement methods: A meta-analysis». In: *Human factors* 63.4 (2021), pp. 706–726 (cit. alle pp. 2, 3, 12, 13).
- [12] Eduardo Salas, Clint A Bowers e Lori Rhodenizer. «It is not how much you have but how you use it: Toward a rational use of simulation to support aviation training». In: *The international journal of aviation psychology* 8.3 (1998), pp. 197–208 (cit. a p. 2).
- [13] Robert L Helmreich, Ashleigh C Merritt e John A Wilhelm. «The evolution of crew resource management training in commercial aviation». In: *The international journal of aviation psychology* 9.1 (1999), pp. 19–32 (cit. a p. 2).
- [14] Shakil Ahmed. «A review on using opportunities of augmented reality and virtual reality in construction project management». In: *Organization, technology & management in construction: an international journal* 10.1 (2018), pp. 1839–1852 (cit. alle pp. 2, 21, 22).
- [15] JJ Kozak, PA Hancock, EJ Arthur e ST Chrysler. «Transfer of training from virtual reality». In: *Ergonomics* 36.7 (1993), pp. 777–784 (cit. a p. 2).
- [16] Jennifer Whyte. «Industrial applications of virtual reality in architecture and construction». In: *Journal of Information Technology in Construction (ITcon)* 8.4 (2003), pp. 43–50 (cit. a p. 3).
- [17] Peter A Hancock e Robert R Hoffman. «Keeping up with intelligent technology». In: *IEEE Intelligent Systems* 30.1 (2015), pp. 62–65 (cit. a p. 3).
- [18] Endel Tulving e Donald M Thomson. «Encoding specificity and retrieval processes in episodic memory.» In: *Psychological review* 80.5 (1973), p. 352 (cit. a p. 3).
- [19] Duncan R Godden e Alan D Baddeley. «Context-dependent memory in two natural environments: On land and underwater». In: *British Journal of psychology* 66.3 (1975), pp. 325–331 (cit. a p. 3).
- [20] *TXT e-solutions*. URL: <https://www.txtgroup.com/it/> (cit. a p. 4).
- [21] PACE INNOVATION HUB. *XR - Extended Reality*. URL: <https://pace.txtgroup.com/products/extended-reality/> (cit. alle pp. 5, 13).

- [22] G. Guglieri. «Flight Simulation: Architecture of Flight Simulation. Dispense del Corso di Simulazione del Volo». In: (mar. 2021) (cit. alle pp. 9, 10, 39, 61).
- [23] Pawel Dymora, Bartosz Kowal, Miroslaw Mazurek e Sliwa Romana. «The effects of Virtual Reality technology application in the aircraft pilot training process». In: *IOP Conference Series: Materials Science and Engineering*. Vol. 1024. 1. IOP Publishing. 2021, p. 012099 (cit. alle pp. 10, 18).
- [24] William R Sherman e Alan B Craig. *Understanding virtual reality—Interface, application, and design*. 2003 (cit. a p. 10).
- [25] *Virtual Reality Market Stumbles in 2018*. URL: <https://www.ccsinsight.com/blog/virtual-reality-market-stumbles-in-2018/> (cit. a p. 11).
- [26] *Virtual Reality & Flight Training*. URL: <https://vrpilot.aero/virtual-reality-flight-training/> (cit. alle pp. 12, 18).
- [27] Yeling Jiang, Mesut Akdere e Flavio Destri Lobo. «Evaluation and assessment of virtual reality-based simulated training: exploring the human–technology frontier». In: *European Journal of Training and Development* (2021) (cit. alle pp. 13, 19).
- [28] Donald L Kirkpatrick. «Techniques for evaluation training programs». In: *Journal of the American Society of Training Directors* 13 (1959), pp. 21–26 (cit. a p. 13).
- [29] Stefan Auer, Jens Gerken, Harald Reiterer e Hans-Christian Jetter. «Comparison Between Virtual Reality and Physical Flight Simulators for Cockpit Familiarization». In: *Mensch und Computer 2021*. 2021, pp. 378–392 (cit. alle pp. 13–15).
- [30] Jesse Orlansky, Joseph String e Paul R Chatelier. *The cost-effectiveness of military training*. Rapp. tecn. INSTITUTE FOR DEFENSE ANALYSES ALEXANDRIA VA, 1982 (cit. a p. 14).
- [31] Matthias Oberhauser, Daniel Dreyer, Reinhard Braunstingl e Ioana Koglbauer. «What’s Real About Virtual Reality Flight Simulation?» In: *Aviation Psychology and Applied Human Factors* (2018) (cit. a p. 15).
- [32] Vsevolod Peysakhovich, Louis Monnier, Mélanie Gornet e Stéphane Juaneda. «Virtual reality vs. real-life training to learn checklists for light aircraft». In: *Eye-Tracking in Aviation. Proceedings of the 1st International Workshop (ETAVI 2020)*. ISAE-SUPAERO, Université de Toulouse; Institute of Cartography and ... 2020, pp. 47–53 (cit. a p. 15).
- [33] *KLM City Hopper introduce il Virtual Reality training per i piloti*. URL: <https://italiavola.com/2020/10/29/klm-city-hopper-introduce-il-virtual-reality-training-per-i-piloti/> (cit. a p. 15).

- [34] *La vision di Leonardo per il training dei futuri piloti militari*. URL: <https://www.leonardocompany.com/it/news-and-stories-detail/-/detail/simulation-based-trainer-how-an-idea-can-become-a-successful-product> (cit. a p. 16).
- [35] *Are the VR flight simulators the future of pilot training?* URL: <https://www.aircharterservice.com/about-us/news-features/blog/are-vr-flight-simulators-the-future-of-pilot-training> (cit. alle pp. 17, 19).
- [36] *La piattaforma Pacelab WEAVR per creare soluzioni di training in realtà virtuale (VR) per le nuove generazioni di piloti e manutentori*. URL: <https://www.txtgroup.com/it/news-events/news-press-releases/la-piattaforma-pacelab-weavr-per-creare-soluzioni-di-training-in-realt%C3%A0-virtuale-vr-per-le-nuove-generazioni-di-piloti-e-manutentori> (cit. a p. 17).
- [37] Vitor Reis e Cláudia Neves. «Application of virtual reality simulation in firefighter training for the development of decision-making competences». In: *2019 International Symposium on Computers in Education (SIIE)*. IEEE, 2019, pp. 1–6 (cit. a p. 19).
- [38] Fabrizio Corelli. «Sistemi di realtà virtuale per l'addestramento dei vigili del fuoco». Tesi di laurea mag. Politecnico di Torino, 2019 (cit. a p. 20).
- [39] Tazama U St. Julien e Chris D Shaw. «Firefighter command training virtual environment». In: *Proceedings of the 2003 conference on Diversity in computing*. 2003, pp. 30–33 (cit. a p. 20).
- [40] LabVr - Università degli Studi di Siena. *Il laboratorio di Realtà Virtuale dell'Università di Siena lancia il Progetto Pilota per il Training in Realtà Virtuale nel settore edile*. URL: [https://api.santachiaralab.unisi.it/api/v1/website/images/1574071846554\\_labvr\\_app\\_edile\\_presentazione.pdf](https://api.santachiaralab.unisi.it/api/v1/website/images/1574071846554_labvr_app_edile_presentazione.pdf) (cit. a p. 21).
- [41] Ajang Behzadi. «Using augmented and virtual reality technology in the construction industry». In: *American journal of engineering research* 5.12 (2016), pp. 350–353 (cit. a p. 21).
- [42] Sevilay Demirkesen e David Arditi. «Construction safety personnel's perceptions of safety training practices». In: *International Journal of Project Management* 33.5 (2015), pp. 1160–1169 (cit. a p. 21).
- [43] Yihai Fang, Jochen Teizer e Eric Marks. «A framework for developing an as-built virtual environment to advance training of crane operators». In: *Construction Research Congress 2014: Construction in a Global Network*. 2014, pp. 31–40 (cit. a p. 21).

- [44] *Addestramento militare in realtà virtuale*. URL: <https://www.augmenta.it/addestramento-militare-realta-virtuale/> (cit. a p. 22).
- [45] *Come si addestra l'esercito con la realtà virtuale e aumentata in Italia*. URL: <https://www.vice.com/it/article/53yxaa/addestrare-esercito-con-realta-virtuale-e-aumentata-in-italia-> (cit. a p. 22).
- [46] La Stampa. «Il Poli “addestra” i militari della NATO con un simulatore di realtà virtuale». In: (ott. 2019). URL: <https://www.lastampa.it/torino/2019/10/02/news/il-poli-addestra-i-militari-della-nato-con-un-simulatore-di-realta-virtuale-1.37624940> (cit. a p. 22).
- [47] Xinxiong Liu, Jing Zhang, Guoxiang Hou e Zenan Wang. «Virtual reality and its application in military». In: *IOP Conference Series: Earth and Environmental Science*. Vol. 170. 3. IOP Publishing. 2018, p. 032155 (cit. a p. 23).
- [48] Ajey Lele. «Virtual reality and its military utility». In: *Journal of Ambient Intelligence and Humanized Computing* 4.1 (2013), pp. 17–26 (cit. a p. 23).
- [49] *Addestramento degli astronauti con la realtà virtuale*. URL: <https://www.experenti.eu/realta-virtuale/training/training-vr-astronauti/> (cit. a p. 23).
- [50] R Bowen Loftin. «Virtual environments for aerospace training». In: *Proceedings of WESCON'94*. IEEE. 1994, pp. 384–387 (cit. a p. 24).
- [51] *Realtà Virtuale: rivoluzione nel campo della medicina*. URL: <https://www.digitalforbusiness.com/realta-virtuale-medicina/#> (cit. a p. 25).
- [52] Greg S Ruthenbeck e Karen J Reynolds. «Virtual reality for medical training: the state-of-the-art». In: *Journal of Simulation* 9.1 (2015), pp. 16–26 (cit. a p. 25).
- [53] Yu Wang, Shuxiang Guo, Yaxin Li, Takashi Tamiya e Yu Song. «Design and evaluation of safety operation VR training system for robotic catheter surgery». In: *Medical & biological engineering & computing* 56.1 (2018), pp. 25–35 (cit. alle pp. 25, 26).
- [54] *SteamVR*. URL: <https://store.steampowered.com/steamvr?l=italian> (cit. a p. 27).
- [55] Kelvin Valentino, Kevin Christian e Endra Joelianto. «Virtual reality flight simulator». In: *Internetworking Indonesia Journal* 9.1 (2017), pp. 21–25 (cit. a p. 28).
- [56] TXT e-solutions. *PACELAB WEAVR - USER MANUAL* (cit. a p. 30).
- [57] *WEAVR SimulationHub*. URL: <https://help.pace.de/xr/weavr/2.0.0/advanced-features/weavr-simulation-hub/index.html> (cit. a p. 33).

- [58] *X-Plane 11*. URL: <https://www.x-plane.com/> (cit. a p. 35).
- [59] *NASA - XPlane Connect*. URL: <https://github.com/nasa/XPlaneConnect> (cit. a p. 37).
- [60] *VIVE Pro*. URL: <https://www.vive.com/eu/product/vive-pro/> (cit. a p. 37).
- [61] Airbus. *A320 FCTM* (cit. a p. 43).
- [62] Fly Uk. *Airbus A320/A321 - Normal Checklist* (cit. a p. 44).
- [63] *I due gemelli: B-737 e A320*. URL: <https://www.ilpost.it/antoniadini/2012/01/01/i-due-gemelli-b-737-e-a320/> (cit. a p. 62).
- [64] *SIDs and STARs*. URL: [https://www.skybrary.aero/index.php/SIDs\\_and\\_STARs](https://www.skybrary.aero/index.php/SIDs_and_STARs) (cit. a p. 64).
- [65] *A320 Displays and Panels*. URL: [http://www.a320dp.com/A320\\_DP/panels.html](http://www.a320dp.com/A320_DP/panels.html) (cit. a p. 101).

# Ringraziamenti

*Dedicato a mia nonna*

Desidero riservare un breve spazio a coloro i quali hanno contribuito alla conclusione del mio percorso.

Un ringraziamento speciale va al Dott. Stefano Primatesta, che ha il merito di averci supportato nelle varie fasi del lavoro, nella stesura dell'elaborato e nell'aver sempre mostrato disponibilità. Ringrazio calorosamente il Prof. Guglieri per l'opportunità di tesi concessami e per essere stato un punto di riferimento didattico nel percorso di Laurea Magistrale. Non posso infine esimermi dal ringraziare l'ing. Angelo Gino e tutto il team di TXT per il supporto fornito nelle varie tematiche del progetto.

I ringraziamenti più sentiti vanno a mia madre, mio padre, mio fratello, mia nonna e tutti i componenti della mia famiglia che mi hanno permesso di intraprendere questo percorso formativo a Torino. Siete il faro più luminoso del mio cammino.

Ringrazio una persona speciale per aver quotidianamente creduto in me e per avermi affiancato sin dall'inizio del percorso accademico: Marisa. Senza di te, probabilmente, mi sarei già arreso.

È doveroso rivolgere una dedica a chi ha contribuito a rendere unico ogni rientro a casa, il *Bello Clebbo*, che mi ha sempre regalato momenti di serenità. E grazie a te, Pier, per essere un mio punto di riferimento da più di 10 anni.

Infine ringrazio tutti gli amici e i colleghi conosciuti nel mio percorso accademico, tra cui Giovanni, che ha condiviso con me tutte le difficoltà del lavoro di tesi.

Con l'augurio di poter condividere e festeggiare con voi tutti i traguardi che il futuro mi riserverà.

Francesco