

POLITECNICO DI TORINO

---



**Politecnico  
di Torino**

Tesi di Laurea Magistrale in Ingegneria Aerospaziale

**Progettazione di procedure per la  
formazione di piloti in realtà virtuale  
mediante software Pacelab WEAVR e  
confronto con il tradizionale  
Computer Based Training**

Relatore  
Dr. Stefano Primatesta

Correlatore:  
Ing. Stefano Mapelli

Laureando  
Alpe Davide

---

Aprile 2024





# Sommario

In questo elaborato di tesi viene presentato un progetto nato dalla collaborazione tra il Politecnico di Torino e l'azienda TXT Group con l'obiettivo di creare un ambiente capace di riprodurre fedelmente la cabina di pilotaggio di un Airbus A320, tramite la realtà virtuale, in cui i piloti possano apprendere e testare procedure operative al fine di sostituire efficacemente i tradizionali mezzi di apprendimento.

Il lavoro è stato svolto interamente in stretta collaborazione con la collega Annalisa Cantiani e prevede lo sviluppo di due procedure operative "statiche", *Parking* e *After Landing*, che non richiedono la presenza di un simulatore di volo, e due "dinamiche", *Take off* e *Landing*, che interagiscono con il simulatore *XPlane* per ottenere i dati di volo in tempo reale. Le procedure sono state sviluppate nell'ambiente grafico *Unity* per mezzo del plug-in *WEAVR*, fornito dall'azienda; inoltre, per quelle dinamiche, si è usufruito di un altro strumento fornito da TXT, chiamato *SimulationHub*, che permette il collegamento tra *WEAVR* e altri software esterni a *Unity*, quindi anche un simulatore di volo.

In questo documento si introduce l'ambiente della realtà virtuale con una ricerca allo stato dell'arte e si focalizza l'attenzione sul *Training VR* in diversi settori evidenziando i suoi pregi e difetti; vengono illustrati i diversi dispositivi hardware e i software utilizzati per poi proseguire con la descrizione e l'implementazione delle procedure operative sviluppate. Infine viene introdotto un test pensato ed effettuato su un campione di persone per valutare l'efficacia delle procedure dinamiche create. Sfruttando il simulatore fisico dell'Airbus A320, presente nel laboratorio di simulazione del volo al Politecnico di Torino, è stato possibile far riprodurre la procedura imparata tramite realtà virtuale o tramite addestramento classico *Computer Based*, CB, ed effettuarne una valutazione con un metodo creato appositamente.

I risultati dimostrano che la realtà virtuale non solo si è rivelata un mezzo di apprendimento efficace, ma in quest'ambito ha superato le aspettative, andando a superare, con un buon margine, i punteggi del tradizionale addestramento *Computer Based*; in questo modo è stato possibile stabilire che questo nuovo mezzo di apprendimento è davvero una risorsa indispensabile per l'addestramento dei piloti del futuro.



# Indice

<b>Elenco delle figure</b>	VIII
<b>1 Introduzione</b>	1
1.1 Obiettivi . . . . .	2
1.2 TXT Group . . . . .	2
1.3 Outline . . . . .	3
<b>2 Stato dell'arte</b>	5
2.1 Realtà estesa . . . . .	5
2.1.1 Realtà virtuale . . . . .	5
2.1.2 Realtà aumentata . . . . .	7
2.1.3 Realtà mista . . . . .	7
2.2 Realtà virtuale in campo aeronautico . . . . .	9
2.2.1 Training piloti in VR e confronto con metodo tradizionale . . . . .	9
2.2.2 Manutenzione aeronautica . . . . .	16
2.3 Altri campi di interesse della realtà virtuale . . . . .	22
2.3.1 Medicina . . . . .	22
2.3.2 Automotive . . . . .	25
2.3.3 Turismo . . . . .	26
<b>3 Hardware e software</b>	28
3.1 Software . . . . .	28
3.1.1 SteamVR . . . . .	28
3.1.2 Unity . . . . .	29
3.1.3 Pacelab WEAVR . . . . .	31
3.1.4 WEAVR Simulation Hub . . . . .	35
3.1.5 Visual Studio . . . . .	36
3.1.6 X-Plane . . . . .	36
3.1.7 X-Plane Connect . . . . .	37
3.2 Hardware . . . . .	38
3.2.1 HTC Vive Pro . . . . .	38
3.2.2 Simulatore A320 . . . . .	40

<b>4</b>	<b>Procedure operative statiche</b>	<b>42</b>
4.1	After Landing . . . . .	43
4.2	Parking . . . . .	55
<b>5</b>	<b>Procedure operative dinamiche</b>	<b>57</b>
5.1	Architettura del progetto . . . . .	57
5.1.1	SimulationHub . . . . .	57
5.1.2	Client X-Plane . . . . .	60
5.1.3	SimulationHub script . . . . .	61
5.1.4	uDesktopDuplication plugin . . . . .	62
5.2	Approach & Landing . . . . .	65
5.2.1	Piano di volo e configurazione MCDU . . . . .	65
5.2.2	Avvio della procedura . . . . .	67
5.2.3	Descrizione della procedura . . . . .	68
5.3	Take Off & Climb . . . . .	87
<b>6</b>	<b>Confronto VR - Computer Based Training</b>	<b>89</b>
6.1	Introduzione del test . . . . .	89
6.2	Struttura del test . . . . .	90
6.3	Risultati e analisi dei dati del test sperimentale . . . . .	96
<b>7</b>	<b>Conclusioni</b>	<b>102</b>
<b>A</b>	<b>Informativa sulla privacy</b>	<b>103</b>
<b>B</b>	<b>Metodo di valutazione</b>	<b>108</b>
	<b>Riferimenti bibliografici</b>	<b>109</b>

# Elenco delle figure

1.1	TXT Group logo . . . . .	2
2.1	Un visore di realtà virtuale: HTC VIVE Pro . . . . .	5
2.2	Visore e relative periferiche . . . . .	6
2.3	Esempio di realtà aumentata attraverso uno smartphone . . . . .	7
2.4	Esempio di ologramma in mixed reality . . . . .	8
2.5	Spettro della Mixed Reality . . . . .	9
2.6	Addestramento Pilot Training Next PTN . . . . .	12
2.7	Architettura ImmersiFLY . . . . .	13
2.8	Flusso dati di ImmersiFLY . . . . .	13
2.9	Esempio di simulatore VR militare . . . . .	15
2.10	Schema di un sistema BCI . . . . .	16
2.11	GMF 4.0 . . . . .	18
2.12	MRO.AIR . . . . .	18
2.13	Italpresse Gauss . . . . .	19
2.14	HoloLens 2 . . . . .	19
2.15	Microsoft HoloLens . . . . .	21
2.16	Architettura PoC . . . . .	21
2.17	Applicazione del PoC . . . . .	22
2.18	Possibilità della Realtà Aumentata . . . . .	23
2.19	Esempio di formazione chirurgica . . . . .	24
2.20	AR in medicina . . . . .	25
2.21	Framework dell'applicazione . . . . .	27
2.22	Ambiente di sviluppo Unity . . . . .	27
3.1	Ambiente virtuale SteamVR . . . . .	29
3.2	Home page Unity - Hierarchy . . . . .	30
3.3	WEAVR Enviroment . . . . .	31
3.4	WEAVR Enviroment . . . . .	32
3.5	WEAVR Enviroment . . . . .	33
3.6	WEAVR Enviroment . . . . .	34
3.7	Procedure Editor e Procedure Inspector . . . . .	35
3.8	Procedure Editor e Procedure Inspector . . . . .	35
3.9	Procedure Editor e Procedure Inspector . . . . .	37
3.10	HTC Vive Pro . . . . .	38



3.11 Riproduzione cockpit Airbus A320 . . . . .	40
4.1 After Landing checklist . . . . .	43
4.2 Schermata di benvenuto . . . . .	44
4.3 Tablet schermate . . . . .	45
4.4 After Landing benvenuto . . . . .	46
4.5 Step 1: Ground Spoiler RETRACT . . . . .	47
4.6 Step 2-3-4-5: Configurazione <i>Exterior lights</i> . . . . .	48
4.7 Step 6: PM Procedures . . . . .	49
4.8 Step 7: Radar/PWS OFF . . . . .	50
4.9 Step 8: ENG mode selector NORM . . . . .	51
4.10 Step 9: Flap ZERO . . . . .	51
4.11 Step 10: TCAS/ATC STBY . . . . .	52
4.12 Maschere APU Master button . . . . .	53
4.13 Step 11: APU Master switch ON . . . . .	53
4.14 Maschere APU Start button . . . . .	54
4.15 Step 12: APU Start ON . . . . .	54
4.16 Step 13: End step . . . . .	55
4.17 Parking checklist . . . . .	56
5.1 Architettura SimHub . . . . .	58
5.2 Files ICD . . . . .	59
5.3 SimHubReader script . . . . .	62
5.4 . . . . .	63
5.5 SimulationHubIcd script . . . . .	64
5.6 Monitor Board . . . . .	64
5.7 MCDU . . . . .	65
5.8 Schermate MCDU . . . . .	66
5.9 Configurazione scena X-Plane . . . . .	67
5.10 Scelta procedura nel <i>Client XPlane</i> . . . . .	68
5.11 Visuale X-Plane Approach & Landing . . . . .	68
5.12 . . . . .	70
5.13 Maschere altitudine autopilota . . . . .	70
5.14 Step 1: Set altitudine . . . . .	71
5.15 Configurazione Element script . . . . .	72
5.16 Invio comandi a X-Plane 11 . . . . .	72
5.17 Step 2: Approach . . . . .	73
5.18 Step 3: AP2 . . . . .	74
5.19 Step 4: Guardare altitudine . . . . .	75
5.20 Step 5: Flap 1 . . . . .	75
5.21 Step 6: Guardare velocità . . . . .	76
5.22 Step 7: Flap 2 . . . . .	77
5.23 Step 8-9-10-11: Flap 3 e Flap FULL . . . . .	78
5.24 Maschere sistema di pressione . . . . .	78
5.25 Step 12: Set QNH . . . . .	79
5.26 Landing checklist . . . . .	80

5.27	Step 14: Checklist . . . . .	80
5.28	Maschere ECAM . . . . .	81
5.29	Step 15: Landing gear DOWN . . . . .	82
5.30	Checklist ECAM superiore aggiornata . . . . .	82
5.31	Step 16: Spoiler armed . . . . .	83
5.32	. . . . .	84
5.33	Step 17: Auto BRK LOW . . . . .	84
5.34	Step 19: Flare . . . . .	85
5.35	Maschera ECAM inferiore - REVERSE . . . . .	86
5.36	Step 20: Reverse . . . . .	86
5.37	Step 22: IDLE . . . . .	87
6.1	Home page della piattaforma Rally con esempio di calendario . . . . .	91
6.2	Esempio diapositiva CBT . . . . .	92
6.3	Questionario iniziale . . . . .	93
6.4	Training VR e valutazione al simulatore . . . . .	94
6.5	Home page NASA TLX app . . . . .	95
6.6	. . . . .	95
6.7	Analisi dei risultati ottenuti tramite script Matlab boxplot . . . . .	99
6.8	Analisi dei risultati ottenuti dal NASA TLX tramite script Matlab boxplott	100
6.9	Risultati dei test sperimentali con candidati aerospaziali . . . . .	101
6.10	Risultati dei test sperimentali con candidati non aerospaziali . . . . .	101
B.1	Metodo di valutazione . . . . .	108

# Capitolo 1

## Introduzione

La realtà virtuale è un campo tecnologico in continua evoluzione che ambisce a creare un'esperienza immersiva simulando la presenza fisica in un ambiente tridimensionale. Sviluppata a partire dagli ultimi decenni soprattutto in ambito videoludico, negli ultimi anni è cresciuta in modo esponenziale e la sua influenza si è estesa in diversi settori come quello industriale, medico, scientifico e formativo. La notevole diffusione della realtà virtuale è dovuta in gran parte al progresso tecnologico e scientifico; infatti lo sviluppo di visori *VR* più accessibili e la miniaturizzazione dell'hardware, insieme al miglioramento del software e della sensoristica, hanno reso questa tecnologia più diffusa e accessibile al pubblico.

In particolare, nel settore aerospaziale, la *virtual reality* ha rivoluzionato i processi di progettazione e prototipazione, consentendo agli ingegneri di immergersi completamente in un ambiente 3D e facilitando quindi la progettazione sperimentale e la revisione dei prototipi. Assume un ruolo determinante anche nella formazione di piloti, permettendogli di addestrarsi in scenari realistici e di affrontare situazioni di emergenza in totale sicurezza[1]. Ad esempio, Il Centro Italiano Ricerche Aerospaziali (CIRA) si dedica all'applicazione della VR nella manutenzione aeronautica dal 2004, sviluppando tecniche e software di simulazione immersiva sia per clienti industriali che istituzionali, migliorando notevolmente l'efficacia della formazione e l'interazione con l'ambiente virtuale.

I numerosi benefici della realtà virtuale hanno quindi convinto grandi aziende e comunità internazionali ad investire nella ricerca in quest'ambito, come ha fatto l'Unione Europea finanziando, nel 2008, il progetto *Vision*[2]. Questo progetto è finalizzato a migliorare la VR nelle applicazioni aerospaziali e a definire le caratteristiche principali delle interfacce nella tecnologia della realtà virtuale, ovvero l'interazione e la visualizzazione tridimensionale, per migliorare flessibilità e prestazioni in modo da ridurre i costi, soprattutto per i più importanti prodotti virtuali del settore aeronautico, come le cabine di pilotaggio e gli interni della fusoliera.

## 1.1 Obiettivi

La presente tesi si propone di definire procedure operative all'interno di un ambiente di realtà virtuale basato sulla piattaforma *Pacelab WEAVR*, con l'obiettivo di consentire agli utenti di immergersi nella cabina di pilotaggio del velivolo Airbus A320, familiarizzando con gli elementi essenziali e memorizzando le fasi principali del volo. *Pacelab WEAVR*, sviluppato dalla *TXT Group*, offre la possibilità di creare procedure guidate che migliorano il livello di apprendimento degli utenti, riducendo al contempo i costi e i tempi di formazione.

A questo scopo, in questo progetto di tesi sono state implementate quattro procedure operative: due statiche, *Parking* e *After Landing* e due dinamiche, *Take Off & Climb* e *Approach & Landing*. Le statiche si riferiscono a procedure svolte dai piloti a terra e sono state implementate come integrazione a quelle elaborate nelle tesi precedenti[3][4][5][6][7][8][9] e sono servite anche come strumento per familiarizzare con la piattaforma di sviluppo; le procedure operative dinamiche, invece, sono costituite da azioni da eseguire durante il volo e richiedono, dunque, di interfacciarsi con un simulatore di volo, *X-Plane*, in quanto è necessario interagire con la cabina di pilotaggio a seconda dei parametri di volo visualizzati.

Dopo aver analizzato le applicazioni VR e la differenza tra formazione VR e tradizionale con una ricerca allo stato dell'arte per vedere vantaggi e svantaggi delle diverse tecnologie, si è giunti all'obiettivo finale di questo progetto: il confronto tra addestramento in realtà virtuale e Computer Based Training CBT, attraverso un test sperimentale svolto su un campione di persone volontarie. Il test consiste nell'addestramento di due persone alla volta, una in VR e l'altra con CBT, per un determinato di tempo, al termine del quale entrambi provano a replicare quanto imparato, sfruttando il simulatore di volo fisico dell'Airbus A320 presente nel Laboratorio di Simulazione del Volo al Politecnico di Torino. La loro prestazione viene valutata tramite apposito metodo sviluppato a priori e viene, in seguito, effettuato un confronto tra tutti i candidati.

## 1.2 TXT Group

L'azienda TXT Group è un'organizzazione multinazionale con una vasta presenza nel settore dell'Information Technology e della consulenza[10]. Le sue attività spaziano dall'offerta di



**Figura 1.1:** TXT Group logo

servizi di consulenza strategica e implementazione di soluzioni tecnologiche all'ingegneria e

al supporto nella gestione del ciclo di vita dei prodotti. L'azienda fornisce servizi e soluzioni a una vasta gamma di settori industriali, tra cui *Aerospace*, *Automotive*, la *Defense* e *Logistic*.

### Storia e fondazione

TXT Group è stata fondata nel 1989 e ha gradualmente ampliato la propria presenza globale attraverso acquisizioni strategiche e partnership, come l'acquisizione del gruppo PACE nel 2016, attraverso cui ha potuto fornire nuove soluzioni di software digitali come il prodotto *Pacelab WEAVR*, utilizzato per questo progetto di tesi, che è stato utilizzato da *Leonardo Company* per le applicazioni di addestramento aeronautico. Nel corso degli anni, l'azienda ha sviluppato una solida reputazione nel campo dell'innovazione tecnologica e della consulenza di alto livello.

### Innovazione e Ricerca e Sviluppo

*TXT Group* dedica risorse significative alla ricerca e allo sviluppo, con l'obiettivo di rimanere all'avanguardia nel settore dell'Information Technology e dell'ingegneria. Come dimostra questo progetto di tesi, l'azienda collabora spesso con istituti di ricerca, università e partner industriali per sviluppare soluzioni innovative e anticipare le future esigenze del mercato.

### Presenza Globale

L'azienda è presente, attraverso sedi e uffici, in diversi paesi in tutto il mondo. Questa distribuzione geografica consente di servire clienti internazionali e di adattarsi alle specifiche normative e culturali dei vari mercati.

## 1.3 Outline

Questo elaborato di tesi si divide in una prima parte di ricerca bibliografica sulla realtà virtuale e sui vari sistemi hardware e software utilizzati e in una seconda parte in cui si descrivono le procedure operative implementate e il test sperimentale.

Inizialmente, nel secondo capitolo, viene fornita una panoramica sulla tecnologia della realtà estesa, concentrandosi sulla **realtà virtuale**. In particolare, si approfondisce l'utilizzo del VR nel campo della formazione dei piloti e sulle differenze con altre tipologie di addestramento, per poi proseguire con il suo impiego nell'ambito della manutenzione aeronautica, un altro campo fondamentale di applicazione. Successivamente, viene analizzato il contributo della realtà virtuale in altri ambiti come in quello medico, in cui vengono approfondite le tecniche di applicazione, quello automobilistico e quello turistico, dove viene visto un esempio pratico di applicazione.

Nel terzo capitolo, vengono descritte le principali componenti **hardware** e **software** utilizzate nel progetto di tesi. Tra i software analizzati si trova l'ambiente grafico *Unity*, in cui è visibile il modello 3D del cockpit, *Pacelab WEAVR*, applicazione sviluppata da TXT Group per produrre procedure di addestramento, il simulatore *X-Plane* e i suoi relativi

strumenti di collegamento; nella componentistica hardware, invece, viene dettagliato il simulatore di volo presente nel Laboratorio di Simulazione del Volo, sfruttato per i test sperimentali e la strumentazione VR.

Nel quarto e nel quinto capitolo si passa alla parte pratica del progetto, in cui sono state sviluppate quattro procedure operative di volo: due *statiche* e due *dinamiche*. Le procedure statiche sono così definite in quanto non richiedono l'interazione con un simulatore di volo in tempo reale e sono effettuate a terra. Queste sono:

- **After Landing:** descritta in dettaglio in questo elaborato;
- **Parking:** approfondita nell'elaborato di tesi della mia collega Annalisa Cantiani.

Quelle dinamiche, invece, sono procedure effettuate in volo e richiedono quindi che l'utente abbia a disposizione i dati di volo in tempo reale e, dunque, un ambiente di simulazione che operi in sincronia con l'utente. Sono state prodotte:

- **Take off & Climb:** trattata nell'elaborato della mia collega;
- **Approach & Landing:** descritta approfonditamente in questa tesi.

Nella parte relativa alle procedure dinamiche viene, inoltre, dettagliata l'architettura di comunicazione *WEAVR - SimulationHub - X-Plane* e il relativo codice implementato in C++ che permette il collegamento per la parte *X-Plane - SimHub*.

Nel sesto e ultimo capitolo, viene analizzato un test sperimentale condotto su un campione di 66 persone, sfruttando la procedura operativa di *Take off*, descritta nell'elaborato. L'obiettivo di questo test è quello di capire se l'addestramento in realtà virtuale può portare dei vantaggi rispetto al più tradizionale Computer Based Training.

# Capitolo 2

## Stato dell'arte

### 2.1 Realtà estesa

Per definire con precisione l'ambiente di simulazione, occorre innanzitutto comprendere il concetto di realtà estesa, ovvero l'insieme di tre tecnologie o, meglio, di tre strumenti di esperienza interattiva e immersiva: Realtà virtuale, Realtà aumentata e Realtà mista [11].

#### 2.1.1 Realtà virtuale

Il termine *realtà virtuale* **VR** descrive un'ambientazione creata artificialmente mediante l'uso di sistemi informatici, progettata per offrire all'utilizzatore un'esperienza di immersione che può essere *parziale* o *completa* [12]. L'individuo che interagisce con questa dimensione virtuale percepisce di essere situato in un contesto simulato e questa percezione varia a seconda delle capacità tecniche della tecnologia impiegata. La realtà virtuale *completa*



**Figura 2.1:** Un visore di realtà virtuale: HTC VIVE Pro

diventa accessibile attraverso l'impiego di dispositivi specifici, solitamente connessi a un elaboratore dotato della necessaria capacità computazionale per garantirne il funzionamento ottimale. Sebbene non tutte le seguenti periferiche siano essenziali, il loro utilizzo può significativamente arricchire la qualità dell'esperienza:

- **Visore.** Incorpora schermi, spesso ad alta definizione, posizionati vicino agli occhi per escludere qualsiasi percezione del mondo esterno, immergendo completamente l'utente in un ambiente digitale. I modelli base, talvolta realizzati con materiali riciclabili come il cartone, utilizzano uno smartphone inserito in uno spazio dedicato per generare l'esperienza VR. Al contrario, i dispositivi più sofisticati sono dotati di una varietà di sensori progettati per tracciare i movimenti della testa e trasmettere queste azioni all'interno dell'ambiente virtuale, arricchendo l'esperienza dell'utente con un livello di interattività e risposta al movimento più avanzato.
- **Cuffie on-ear.** Alcuni modelli di visori sono dotati di cuffie regolabili che, posizionate vicino alle orecchie, trasmettono direttamente all'utente i suoni, ricreando nei modelli più avanzati un effetto sonoro surround per rendere più immersiva l'esperienza.
- **Controller.** Costituiscono l'unico mezzo per interagire dinamicamente con l'ambiente virtuale. Riconoscibili grazie alla loro somiglianza con i tradizionali joystick, questi dispositivi sostituiscono tali strumenti nonché mouse, tastiera, trackball e altri sistemi di input tradizionali; permettono di compiere movimenti, impartire comandi, afferrare oggetti nell'ambiente virtuale, interagire con elementi dell'interfaccia sintetica come pannelli o tastiere digitali, arricchendo l'esperienza utente con un livello di interazione senza precedenti.
- **Stazioni.** Nel contesto della realtà virtuale è cruciale ricordare l'esistenza del mondo fisico circostante per evitare incidenti, come urtare contro un muro o inciampare in ostacoli. A questo scopo, sono stati sviluppati dispositivi specifici, noti come *base stations*, che, posizionati strategicamente, consentono di mappare e delimitare lo spazio fisico in cui l'utente può muoversi in modo sicuro. Questi dispositivi, interfacciandosi con il visore, creano una griglia virtuale che segnala i confini dell'area sicura non appena l'utente si avvicina al limite; nei modelli di visore più avanzati, sono integrate anche telecamere di profondità che rilevano e proiettano in tempo reale le sagome degli ostacoli nell'ambiente circostante, aumentando ulteriormente la sicurezza durante l'uso della realtà virtuale.



**Figura 2.2:** Visore e relative periferiche

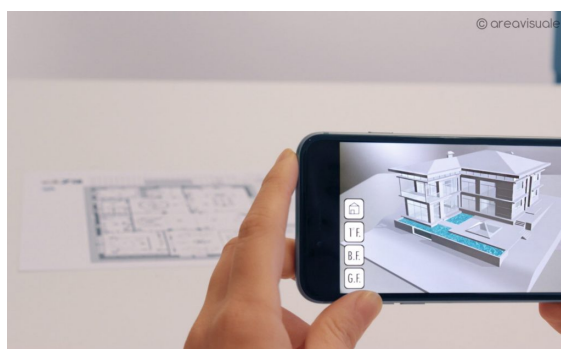
La versione *parziale* della realtà virtuale, utilizzando esclusivamente un visore, anche *stand-alone*, ovvero che non necessita di un hardware esterno per essere utilizzato perché usufruisce



di quello integrato nel casco stesso, limita l'interazione dell'utente all'interno dell'ambiente digitale. Questo tipo di realtà virtuale può consistere principalmente nella visualizzazione di contenuti video preregistrati a 360 gradi, permettendo agli utenti di guardarsi intorno per scoprire dettagli aggiuntivi; tuttavia, in questo contesto, è improbabile che l'utente possa muoversi liberamente o interagire in modo significativo con l'ambiente virtuale proposto.

### 2.1.2 Realtà aumentata

La realtà aumentata **AR** rappresenta una tecnologia che integra elementi e informazioni digitali nel mondo reale, arricchendolo visivamente. Attraverso l'utilizzo di dispositivi come smartphone, tablet, computer con webcam o occhiali specializzati che proiettano immagini direttamente sulla retina, gli utenti possono visualizzare dati, grafiche e media sovrapposti alla loro vista normale. Questa sovrapposizione mira a fornire un'esperienza complementare alla realtà, aggiungendo contenuti interattivi e multimediali in tempo reale all'ambiente fisico in cui si trova l'utente. L'impiego di questa tecnologia risulta



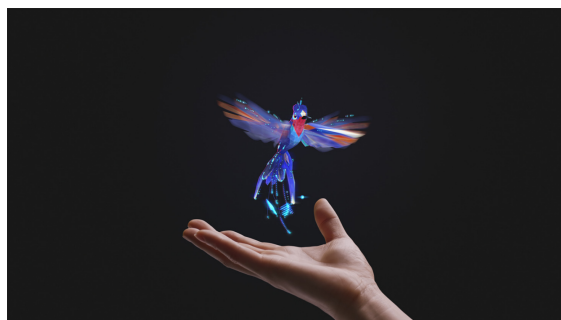
**Figura 2.3:** Esempio di realtà aumentata attraverso uno smartphone

intuitivo e diretto, beneficiando dell'evoluzione di dispositivi mobili, quali smartphone e tablet, inclusi quelli appartenenti a fasce di prezzo medio-basse. Essendo dotati di GPS per la localizzazione, bussola per l'orientamento, connessione internet per l'acquisizione di dati e capacità di riproduzione di video in streaming, permettono un'integrazione efficace con l'ambiente circostante. Attraverso l'uso della fotocamera, è possibile sovrapporre all'immagine reale vari strati di informazioni digitali, migliorando l'esperienza visiva con dati aggiuntivi, indicazioni direzionali sul terreno, oggetti tridimensionali interattivi, o video esplicativi relativi a prodotti o luoghi, il tutto visualizzato in tempo reale. Questa sinergia tra componenti hardware e software permette l'accesso a un'ampia gamma di applicazioni pratiche e informative, rendendo la realtà aumentata uno strumento versatile e accessibile per aumentare la percezione del mondo circostante.

### 2.1.3 Realtà mista

La realtà mista **MR** rappresenta la fusione tra realtà virtuale e realtà aumentata, offrendo un livello di immersione superiore mediante l'integrazione di elementi virtuali nel contesto del mondo reale. Questa tecnologia consente agli utenti di interagire con il proprio ambiente

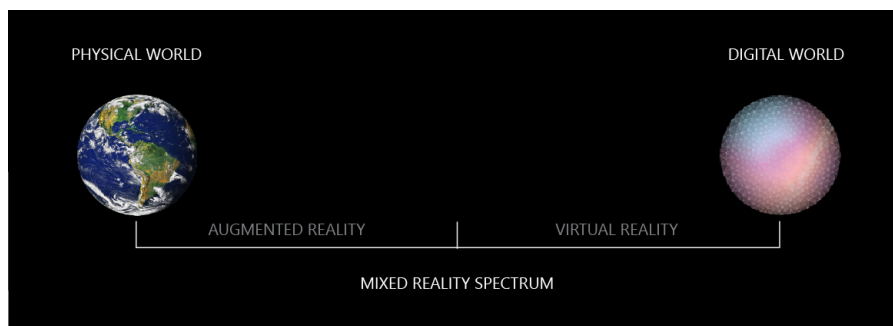
circostante senza il pericolo di collisioni fisiche, aggiungendo la possibilità di interfacciarsi con ologrammi e oggetti digitali che appaiono tangibili. Essendo caratterizzata da un dinamismo senza precedenti, la MR elimina efficacemente le barriere tra il reale e il virtuale, posizionandosi come una delle innovazioni più significative e impattanti nel panorama tecnologico attuale. L'accesso alla realtà mista avviene attraverso l'utilizzo di un dispositivo



**Figura 2.4:** Esempio di ologramma in mixed reality

indossabile, simile ai visori utilizzati per la realtà virtuale, comunemente denominato dispositivo olografico. Questo apparecchio si distingue per la sua capacità di integrare elementi digitali all'interno dell'ambiente fisico, sfruttando avanzate specifiche tecniche e un ampio campo visivo. Si distingue per tre principali componenti:

- **Display.** Permette agli utilizzatori di visualizzare sia l'ambiente reale che gli ologrammi proiettati, attraverso l'uso di lenti olografiche trasparenti con l'obiettivo di fondere la percezione del mondo fisico con elementi virtuali aggiunti, creando un'esperienza visiva multidimensionale.
- **Sensori.** Hanno il compito di analizzare l'ambiente circostante, oltre a monitorare i movimenti della testa e degli occhi dell'utente. Questi sensori operano grazie all'unità di misura inerziale di cui dispongono, comprendente accelerometro, giroscopio e magnetometro, che consente una precisa rilevazione del movimento e dell'orientamento. In aggiunta, il casco è dotato di una serie di telecamere, sia a luce visibile che a infrarossi, generalmente posizionate nella sua parte superiore, per catturare con dettaglio l'ambiente esterno e facilitare l'interazione tra l'utente e gli elementi virtuali presenti nell'ambiente reale. Si possono distinguere sensori *on-body* (per la persona) e *off-body* (per l'ambiente circostante), dispositivi aptici (come guanti e giubbotti) e sistemi microelettromeccanici *MEMS* per esperienze tattili senza l'ausilio di dispositivi indossabili [13].
- **Audio.** Questo componente sfrutta microfoni collocati in posizioni strategiche per catturare la voce dell'utente e altoparlanti integrati, situati vicino alle orecchie, per produrre un effetto sonoro spaziale. Questa configurazione audio avanzata non solo migliora la comunicazione all'interno dell'ambiente virtuale ma arricchisce anche l'esperienza immersiva, rendendo l'audio un elemento chiave per la percezione di un ambiente tridimensionale realistico.



**Figura 2.5:** Spettro della Mixed Reality

All'interno dello spettro in cui si colloca la realtà mista, visibile in Figura 2.5, si possono definire tre tipologie di esperienze possibili, precedentemente descritte:

- Nella parte sinistra dello spettro l'utente percepisce l'ambiente fisico senza avere l'impressione di averlo lasciato, quindi si parla di realtà aumentata;
- in centro si trova la fusione tra mondo virtuale e fisico, ovvero la realtà mista;
- nella parte destra dell'immagine, invece, l'individuo sperimenta un ambiente completamente digitale, la realtà virtuale, e riesce a percepire, tramite strumenti già citati, ciò che accade nell'ambiente fisico che lo circonda.

## 2.2 Realtà virtuale in campo aeronautico

### 2.2.1 Training piloti in VR e confronto con metodo tradizionale

Nel campo della formazione dei piloti, la VR è oggetto di interesse e sperimentazione: alcune istituzioni di formazione aeronautica hanno iniziato a utilizzare la realtà virtuale per l'addestramento al volo. Sebbene l'applicazione sia ancora nelle sue fasi iniziali, con lo sviluppo continuo della tecnologia e la riduzione dei costi, si prevede che giocherà un ruolo più importante nella formazione dei piloti in futuro [14]. Al momento, è sfruttata in alcune operazioni di base come:

- **Addestramento alle competenze di controllo di base:** i piloti possono addestrarsi in operazioni quali il decollo, l'aggiustamento dell'assetto di volo e l'atterraggio. In questi casi l'utente può simulare la cloche e il pedale dell'aereo attraverso il controller interattivo per rafforzare le competenze di volo.
- **Addestramento al volo strumentale:** si possono simulare varie condizioni meteorologiche e di volo, fornendo ai piloti l'opportunità di sperimentare scenari reali come condizioni di scarsa visibilità, volo notturno e volo strumentale. In realtà virtuale sono simulati il pannello degli strumenti e il sistema di navigazione per aiutare i piloti a migliorare le loro competenze e capacità decisionali.

- **Simulazione di emergenza:** infine, è particolarmente rilevante la simulazione di avarie come guasto del motore, guasto del sistema e perdita dell'assetto direzionale; in questi casi il pilota impara a rispondere e risolvere correttamente i problemi, operando in un ambiente sicuro e ripetibile.

I mezzi di addestramento al volo tradizionali presentano principalmente la conoscenza sotto forma di testo, immagini e animazioni. Sebbene ciò possa fornire una certa base teorica agli studenti, è difficile per loro ottenere un'esperienza di volo immersiva. Tuttavia, con lo sviluppo della tecnologia VR, sempre più corsi di addestramento al volo iniziano a combinare la tecnologia VR con la formazione, in modo che gli studenti possano comprendere l'ambiente di volo reale, il terreno, la rotta, le procedure di volo e altre informazioni rilevanti coinvolte nel volo finale prima di entrare nel simulatore di volo per l'operazione pratica.

L'obiettivo sarebbe l'utilizzo della realtà virtuale nell'addestramento al volo finale che al momento si basa solo sul volo reale, nel quale l'istruttore di volo e lo studente condividono l'aereo per l'addestramento al volo pratico. Questa modalità, pur essendo il tipo ideale di training perché ci si trova effettivamente nella realtà, presenta diverse criticità, quali elevato tempo ed energia, possibili rischi per la sicurezza e costo operativo elevato.

Questo tipo di addestramento consiste nel preparare il pilota in modo che abbia un'approfondita comprensione e competenza delle procedure di decollo e atterraggio; infatti, deve assimilare gli standard di sicurezza, le pratiche di atterraggio e avere la capacità di rispondere a situazioni impreviste.

La progettazione dell'addestramento al volo finale su rotte di decollo e atterraggio è riassunta nei seguenti aspetti:

- Insegnamento in aula: i piloti apprendono conoscenze teoriche rilevanti, comprendono i principi di base e le procedure delle fasi di volo insieme ai manuali operativi standardizzati;
- Addestramento con simulazione: utilizzando simulatori di volo, il pilota sperimenta una simulazione realistica di tutti gli aspetti dell'operazione finale, in modo che possa esercitarsi nelle varie procedure e impari a gestire eventuali imprevisti;
- Addestramento in volo: in questa fase finale, il candidato si esercita nell'operazione di decollo e atterraggio nell'ambiente di volo reale. L'obiettivo è migliorare il livello di abilità del pilota e le sue capacità di gestione dello stress, attraverso l'esercizio pratico.

Questa tipologia di training ha il vantaggio di porre l'individuo nell'ambiente reale, ma presenta diverse limitazioni:

- **Spazio di formazione limitato:** l'addestramento al volo finale deve essere svolto nella rotta di decollo e atterraggio, ma, a causa dello spazio aereo limitato dell'area di addestramento, lo spazio di formazione sarà relativamente ridotto, aumentando la difficoltà e il costo;
- **Tempo di formazione limitato:** considerando il tempo totale dell'operazione, quello dedicato effettivamente alle singole procedure di decollo e atterraggio è esiguo, risultando in una minore efficienza dell'addestramento;

- **Influenza del meteo:** l'addestramento al volo finale è suscettibile a fattori meteorologici che possono portare a una riduzione dell'efficacia della formazione e del progresso della stessa;
- **Rischio di operazione reale:** per l'intera durata della prova è richiesto agli istruttori di volo di condurre l'insegnamento e la valutazione, il che può portare ad un eccessivo carico di lavoro, ignorando così possibili conflitti e aumentando i rischi operativi.

Per questi motivi è necessario esplorare altre modalità di addestramento come quella della realtà virtuale. Attualmente, la *European Aeronautic Defense and Space Company* utilizza la tecnologia VR per sviluppare il simulatore di addestramento in realtà virtuale dell'Airbus A380, che utilizza la combinazione di tecnologia VR e simulatore, attraverso occhiali VR e dispositivi di feedback aptico, in modo che i piloti possano addestrarsi nell'ambiente reale della cabina. Nell'addestramento finale, i piloti possono utilizzare simulatori di volo combinati con la tecnologia VR per condurre l'addestramento pratico finale, che risulta completamente immersivo, e percepire veramente il movimento dell'aereo, al fine di migliorare le abilità di volo e familiarizzare con le procedure. Addestrandosi nel simulatore, i piloti possono padroneggiare le abilità di volo in tutte le condizioni atmosferiche in un ambiente sicuro, riducendo notevolmente il costo, il rischio e la difficoltà dell'addestramento al volo reale.

Un altro esempio si può ritrovare nelle forze armate statunitensi che stanno integrando la realtà virtuale nella formazione iniziale al volo attraverso iniziative come il programma **Pilot Training Next** PTN. Questo programma combina l'addestramento basato sulla VR con quello del volo reale per ridurre il tempo totale necessario per completare la formazione per piloti universitari UPT [15]. L'USAF, United State Air Force Academy, ha riscontrato che i soggetti che utilizzano la VR trattenono le informazioni più a lungo e imparano a un ritmo più veloce rispetto ai metodi tradizionali; inoltre il processo di apprendimento più rapido potrebbe alla fine fornire un numero maggiore di piloti all'USAF che sono anche meglio addestrati. Sono stati condotti tre test, al momento, dove sono stati sottoposti metà dei candidati, tutti ufficiali militari, ad un addestramento di volo tradizionale e l'altra metà al nuovo PTN. I risultati sono stati commentati dal tenente colonnello Robert Knapp, capo delle operazioni del PTN, che non ha dubbi sul grande beneficio di questo addestramento e sostiene che, a differenza dell'addestramento tradizionale che richiede fino ad un anno di tempo, questo nuovo tipo di training è stato completato in un tempo compreso tra i sei gli otto mesi, con un notevole impatto positivo sui costi finali, dovuto in parte alle ridotte ore di volo reale e quindi alla minor spesa di carburante [16].

I simulatori di fascia alta contemporanei dispongono di input altamente realistici che includono cruscotti e strumentazione pari a quella a bordo degli aerei. Tuttavia, il livello di immersione non raggiunge quello del realismo; infatti, osservando lo schermo di un simulatore, è evidente che il mondo in cui si svolge un volo è virtuale e questo ambiente non può mai veramente rompere la barriera che permette ai piloti di sentirsi come se stessero effettivamente volando nel mondo reale. Stephen M. Casner sostiene, nell'articolo [17], che la mancanza di realismo in questi sistemi può spesso portare i piloti a essere impreparati per situazioni di emergenza reali. Basandosi su schemi di volo definiti a priori, quando un pilota incontra una situazione di emergenza durante l'addestramento simulato, può sembrare che stia semplicemente premendo un insieme di pulsanti per ottenere un risultato



**Figura 2.6:** Addestramento Pilot Training Next PTN

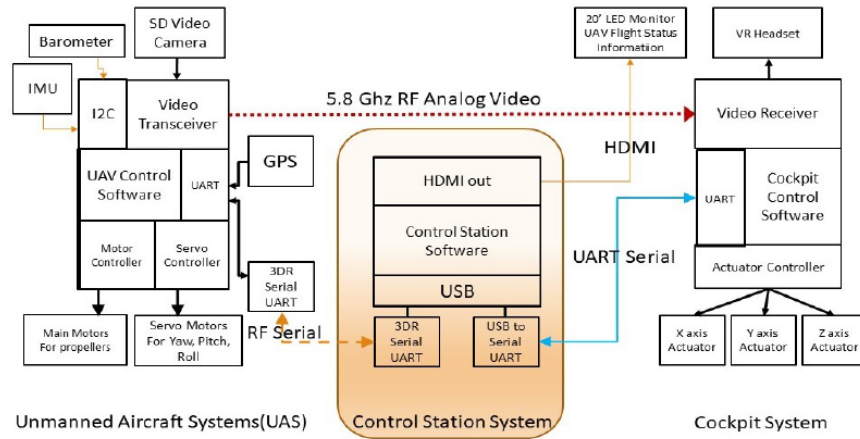
predeterminato e quindi è possibile che non percepisca l'urgenza della situazione perché l'emergenza virtuale è sistematica e le sessioni di addestramento possono essere molto ripetitive.

Si è notato che, quando le situazioni di guasto avvengono in momenti casuali, i tempi di reazione dei piloti sono spesso ordini di grandezza più lunghi e ciò si riflette nei dati sugli incidenti che mostrano che i piloti, che dovrebbero essere addestrati per questi eventi anormali, a volte non intraprendono le contromisure appropriate. Nel documento [18] viene descritto un sistema di simulazione di volo che è immersivo, economico e mira a permettere un addestramento al volo più efficiente: ImmersiFLY. Si tratta di un simulatore che consente ai piloti di sentirsi come se stessero pilotando un vero aereo, fornendo loro una visuale in prima persona FPV della cabina di pilotaggio e dell'ambiente circostante di un aeromobile senza pilota UAS, in modo da permettere all'utente di esplorare l'ambiente reale tramite movimenti della testa. Quindi, mentre l'aeromobile senza pilota vola nel mondo reale, i piloti sperimentano l'aerodinamica del pilotare un aereo di persona e, quando si verificano cambiamenti nelle condizioni dell'ambiente dell'aeromobile reale, come la direzione del vento, il pilota è in grado di notare i cambiamenti visivamente e fisicamente attraverso il visore montato sulla testa e attraverso i movimenti meccanici del simulatore a terra.

Si tratta di un sistema complesso, la cui architettura è mostrata in Figura 2.7, con tre principali sottosistemi:

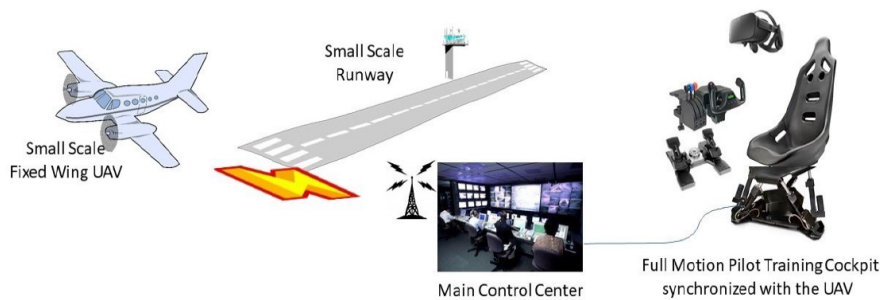
1. il sistema di aeromobili senza pilota UAS;
2. la stazione di controllo;

3. il sistema di simulazione della cabina di pilotaggio CSS;



**Figura 2.7:** Architettura ImmersiFLY

Affinché il sistema funzioni efficientemente, tutti e tre i sottosistemi devono comunicare tra loro. L'UAS manda un flusso video della sua cabina di pilotaggio, trasmesso tramite un segnale radio a frequenza di 5GHz, alla stazione di controllo; questo flusso viene successivamente visualizzato sul visore che fa parte del CSS. In parallelo, l'UAS invia anche parametri riguardanti la dinamica di volo (rollio, beccheggio, imbardata) alla stazione di controllo, per essere visualizzati sul monitor. La stazione di controllo invia quindi questi dati al CSS dove avviene l'attuazione della meccanica del sedile per replicare la dinamica dell'UAS. Perciò



**Figura 2.8:** Flusso dati di ImmersiFLY

il sistema include una postazione dotata di movimento, un visore e un UAS che il pilota può controllare; il visore fornisce agli utenti una vista dal vivo di ciò che vedrebbero se fossero effettivamente nella cabina di pilotaggio dell'aereo; inoltre, registra il movimento della testa del pilota attraverso l'uso di un sistema di misurazione inerziale, inviando i dati al modulo della telecamera, posta nel drone, che si muove secondo i movimenti della testa

del pilota. L'UAS trasmette a sua volta informazioni di volo e misurazioni dell'orientamento alla postazione del CSS per replicare il movimento tramite attuazione, come illustrato in Figura 2.8.

ImmersiFLY ha quindi l'obiettivo di fondere il volo reale con quello virtuale per dare in mano ai piloti un'esperienza quasi completamente realistica.

In sintesi, come evidenziato nell'articolo [14], si possono individuare numerosi vantaggi nell'impiego della VR nell'addestramento al volo finale:

- **Spazio di formazione illimitato:** La VR può creare una varietà di ambienti e scene virtuali in modo che i piloti possano svolgere l'addestramento al volo finale in un ambiente più ampio, non più limitato dallo spazio di addestramento reale; questo fornisce agli studenti più opportunità di pratica e una varietà di scenari di formazione. Allo stesso tempo, il contenuto di formazione può essere personalizzato in base ai dati di addestramento di ogni studente;
- **Ambiente personalizzato:** adottando il Modello Digitale di Elevazione DEM, si ricostruisce topografia del terreno attraverso dati reali di elevazione del terreno, ripristinando l'ambiente della scena reale. Questi scenari possono essere successivamente personalizzati in base alle diverse esigenze di addestramento;
- **Sicurezza e costo:** si riducono notevolmente i rischi e i pericoli per la sicurezza nel volo reale; gli studenti non devono affrontare le situazioni impreviste che possono sorgere durante il volo vero e proprio, ma le possono sperimentare durante la simulazione. Inoltre, la realtà virtuale permette di risparmiare sui costi del volo reale come quelli dovuti al carburante, alla manutenzione dell'aereo e quelli relativi alle spese dell'istruttore e, aggiungendo che gli studenti possono addestrarsi in contemporanea su più postazioni, il costo del sistema diminuisce gradualmente riducendo i tempi di addestramento;
- **Simulazione di emergenza:** si possono simulare molteplici guasti rimanendo in totale sicurezza e quindi verificare come il pilota reagirebbe senza dover tenere sotto controllo il movimento dell'aereo. Inoltre, si possono ideare simulazioni in base al livello di abilità dell'utente per migliorare la sua capacità di resistere alla pressione;
- **Controllo del progresso della formazione:** il progresso della formazione può essere controllato dall'istruttore di volo a cui è permesso di regolare la scena e il tempo di addestramento in base alla situazione e al progresso del pilota, tenendo anche conto degli obiettivi del corso, in modo da rendere l'apprendimento più efficace ed efficiente. Infine, il simulatore raccoglie i dati di formazione del pilota e li confronta con quelli di varie situazioni di volo per aiutare lo studente e l'insegnante a realizzare piani di lavoro personalizzati e a controllare il progresso nell'apprendimento.

Un altro punto di forza è che la realtà virtuale può essere combinata con molteplici tecnologie di sensori indossabili, permettendo di studiare processi fisiologici e cognitivi sollecitati da ambienti dinamici; per questo motivo si ipotizza grande potenzialità nel connubio tra VR e neurotecnologia.

Con il termine *neurotecnologia* ci si riferisce a sensori indossabili e sistemi intelligenti



comunemente utilizzati per raccogliere segnali dell'attività del sistema nervoso centrale provenienti dal cervello e dal corpo umano [19]. Recenti studi dimostrano come la formazione in VR possa essere combinata con numerose tecniche di imaging cerebrale, come la risonanza magnetica strutturale e funzionale MRI, l'elettroencefalografia EEG e la spettroscopia funzionale nell'infrarosso FNIRS [20]. Analizzando gli schemi di attivazione dei lobi frontali e della corteccia motoria, si notano interazioni simili tra l'allenamento in un ambiente VR e un ambiente reale, dimostrando la validità della VR. Oltre all'efficacia della formazione in VR, i suoi effetti psicologici e cognitivi possono anche essere misurati e monitorati utilizzando tecniche di neuroimaging per valutare, ad esempio, i livelli di stress; questo risulta utile per lo sviluppo di interfacce cervello-computer BCI, per osservare passivamente lo stato mentale dell'utente durante un compito e per determinare l'impatto di un prodotto sullo stesso.

Con il crescente interesse per la VR, aumenta la necessità di ricerca per andare a migliorare i visori, i simulatori e tutta la sensoristica con cui si interfaccia il visore. Soprattutto in ambito militare si osserva un particolare sviluppo di simulatori VR, come mostrato in figura 2.9, anche se rimangono diverse sfide riguardo alla combinazione di VR e neurotecnologia per l'addestramento piloti. Infatti, le misure dell'attività cerebrale come l'EEG sono molto sensibili e un sistema VR-EEG funzionante nella formazione al volo attualmente non esiste. Inoltre, nonostante esistano studi che dimostrano l'efficacia delle misure neuro-fisiologiche nell'apprendimento [21], i processi cognitivi e le risposte neurali corrispondenti ai compiti di volo e alla formazione, in aviazione, non sono ancora pienamente compresi. Come evidenziato



**Figura 2.9:** Esempio di simulatore VR militare

nell'articolo [22], ci sono quattro possibili direzioni di ricerca che possono essere intraprese per sviluppare l'integrazione tra VR e neurotecnologia:

1. Aumentare l'efficacia dei simulatori VR in confronto ai simulatori di volo tradizionali. Sono necessari confronti della performance di volo e del carico di lavoro mentale nel tempo tra diverse piattaforme di formazione; la performance di volo può essere misurata utilizzando indicatori di prestazione delle simulazioni.
2. Stabilire basi neurali e cognitive dell'apprendimento a lungo termine nella formazione al volo in VR. Si può creare un modello predittivo per quantificare l'apprendimento e la performance basandosi su dati fisiologici, inclusi EEG ed elettrocardiografia ECG.

Infatti, la frequenza cardiaca potrebbe essere un prezioso elemento per i progressi nella formazione e il carico di lavoro.

3. Ricercare una misura EEG oggettiva per valutare e tracciare l'apprendimento specifico dell'utente nell'addestramento al volo in VR. Il carico di lavoro cognitivo può essere misurato oggettivamente utilizzando caratteristiche temporali, spettrali e spaziali dell'EEG, ad esempio con l'uso dell'Indice di Impegno insieme all'Indice di Stabilità Cognitiva per controllare i progressi nella formazione.
4. Sviluppare un sistema BCI che può fornire feedback in tempo reale e migliorare i risultati della formazione nel volo in VR, schematizzato in figura 2.10. Lo scopo è determinare come i bio-marcatori dell'apprendimento possano essere tradotti in un feedback VR efficace e come l'interfaccia possa migliorare la curva di apprendimento nei piloti in tempo reale con l'obiettivo finale di validare il modello predittivo discusso in precedenza. Attualmente, l'Aeronautica Militare degli Stati Uniti ha già incorporato simulazioni di volo in VR con misure in tempo reale sullo stato cognitivo dell'utente basate su risposte agli eventi, intesi come tempi di reazione, compiti di monitoraggio, posizione della testa, movimenti e dati di tracciamento degli occhi [23], permettendo agli istruttori di fornire istruzioni per migliorare la curva di apprendimento dei piloti. L'intento è rendere la simulazione VR stessa adattiva, in termini di difficoltà del compito di volo, in base ai dati prelevati in tempo reale del livello di formazione, carico di lavoro e coinvolgimento.

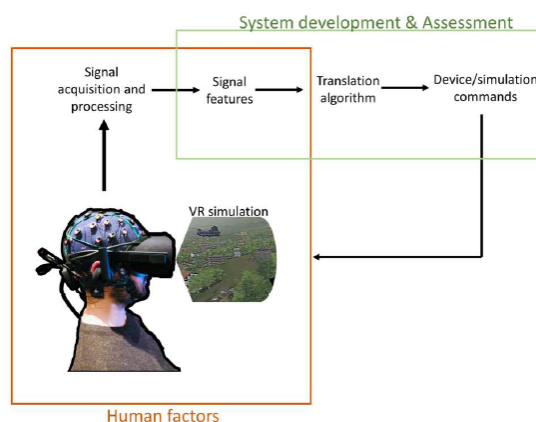


Figura 2.10: Schema di un sistema BCI

### 2.2.2 Manutenzione aeronautica

La manutenzione degli aerei è una caratteristica importante che influisce su sicurezza, affidabilità operativa, utilizzo ed economia degli aerei civili. Una manutenzione, ispezione e gestione efficienti degli aerei possono non solo migliorare l'efficienza del lavoro e risparmiare sui costi del volo, ma anche accelerare lo sviluppo dell'industria degli aerei civili. Il manuale

di manutenzione degli aerei è la base e il riferimento per gli ingegneri che si occupano della manutenzione per riparare e mantenere i velivoli.

In questo momento storico, al fine di ridurre i cicli di manutenzione, abbassarne i costi, ridurre i difetti di assemblaggio e i tassi di fallimento dei prodotti, evitando allo stesso tempo lo spreco di risorse, l'industria della manutenzione aeronautica sta introducendo gradualmente la tecnologia della realtà estesa.

La tecnologia *Mixed Reality* può mescolare i mondi fisico e digitale utilizzando immagini 3D, interazione naturale e percezione ambientale, e può essere applicata al campo della manutenzione degli aerei come dispositivo di assistenza. Il riconoscimento veloce e accurato degli oggetti è alla base per realizzare un'ampia gamma di applicazioni con la MR [24].

Il metodo tradizionale di formazione richiedeva che l'esperto o l'ingegnere trascorresse un certo periodo con il tirocinante per trasmetterne gli aspetti teorici e pratici. Uno studio ha menzionato che l'uso della formazione e-learning ha superato gli svantaggi dei modelli tradizionali di formazione faccia a faccia, in quanto migliora il livello di conoscenza teorica del tirocinante, arricchendone anche l'esperienza pratica [25]. La formazione basata sul computer CBT è attualmente una parte importante del processo di formazione per i tecnici di manutenzione aeronautica ATM, ma è limitato dalla complessità e dai costi associati allo sviluppo. Il più grande svantaggio del CBT, tuttavia, sono le sue restrizioni ambientali perché il CBT richiede l'accesso al computer e non comunica effettivamente con un ambiente di lavoro in tempo reale, essendo vincolato a quello della classe [26]. Gli studenti ATM necessitano di uno strumento di formazione che combini l'interattività della realtà virtuale con la potenza informativa del CBT. Questo strumento può essere un sistema di Realtà Aumentata integrato con la progettazione assistita da computer CAD. Pertanto, la tecnologia AR può essere il prossimo strumento in grado di sostituire la formazione basata sul computer.

Uno studio indica due modi in cui la formazione AR potrebbe ridurre gli errori, aumentando così la sicurezza dei passeggeri e minimizzando gli incidenti [27]. In primo luogo, la realtà aumentata risulta un metodo efficiente di recupero delle informazioni; secondo, l'AR potrebbe elaborare le informazioni facilitando la comprensione di concetti complessi in un arco di tempo più breve. Data la capacità del sistema AR di sovrapporre grafica e testo al mondo reale, si è arrivati alla conclusione che i sistemi AR completano l'elaborazione delle informazioni durante la manutenzione, migliorando il richiamo nella memoria a breve e a lungo termine [28]. La sovrapposizione grafica, infatti, è il principale vantaggio dell'AR, consentendo all'utente di accedere immediatamente a qualsiasi informazione pertinente, come procedure e criteri di ispezione, eliminando il problema della ricerca manuale delle istruzioni.

Si esaminano adesso alcuni applicativi, basati su queste tecnologie, che funzionano come strumenti di assistenza nella manutenzione degli aerei:

- **GMF 4.0** sfrutta la realtà aumentata come metodo di apprendimento per la manutenzione degli aerei e fornisce una procedura *step-by-step* per la guida ai componenti del velivolo nell'operazione *MRO* (Maintenance, Repair, and Overhaul). A partire da una singola scansione, può mostrare il modello 3D con animazione sulla sua funzionalità e sull'esecuzione della revisione generale. Inoltre, questa tecnologia aiuta a migliorare il tasso di errore umano e a costruire una cultura della sicurezza tra gli ingegneri novizi

[29];



**Figura 2.11:** GMF 4.0

- **MRO.AIR** è un'applicazione di realtà aumentata portatile per la riparazione di aerei, utilizzabile tramite smartphone e tablet, che fornisce assistenza vocale, fornendo all'utente la capacità di condurre un'esperienza collaborativa con un altro ingegnere. In questo modo, se l'addetto alla manutenzione riscontra qualunque tipo di difficoltà, può entrare in contatto con una figura più esperta per ricevere adeguata assistenza attraverso indicazioni in tempo reale disponibili nell'applicazione. Il limite di questo applicativo consiste nel fatto che, necessitando di un dispositivo portatile per essere utilizzato, il tecnico deve tenere in mano l'apparecchio mentre conduce l'operazione di riparazione e questo aspetto potrebbe ridurre leggermente l'efficienza dell'intera procedura [30];



**Figura 2.12:** MRO.AIR

- **Italpresse Gauss** combina l'uso di AR e VR insieme all'*Industrial Internet of things* per rendere la sua manutenzione efficace. Italpresse Gauss ha sviluppato un'applicazione utilizzata nelle operazioni di manutenzione e riparazione con l'obiettivo di renderle più veloci, ridurre i tempi di esecuzione e aumentarne la produttività. Questa tecnologia fornisce supporto agli ingegneri con informazioni procedurali in modelli 3D, mentre operano su una specifica parte dell'aereo per la revisione, al fine di ridurre gli errori umani. Questa applicazione, dotata di riconoscimento degli oggetti, aiuta ad aumentare

la capacità di scansione per rilevare quale procedura dovrebbe essere eseguita per risolvere i problemi di manutenzione [31];



**Figura 2.13:** Italtresse Gauss

- **HoloLens 2** nasce dalla collaborazione tra Airbus e Microsoft e fornisce una soluzione innovativa per migliorare di almeno il 30% l'esecuzione di compiti industriali [32]. Con l'uso di HoloLens 2, si può dimostrare che il controllo gestuale sta diventando sempre più avanzato, disponendo di un migliore riconoscimento delle mani durante la navigazione nello spazio fisico reale. In aggiunta, la combinazione del virtuale e del mondo reale incoraggia anche la collaborazione e l'interazione tra diversi individui.



**Figura 2.14:** HoloLens 2

Anche l'**ITAF** (Aeronautica Militare Italiana) [33] affronta il problema della manutenzione e i processi di guida e formazione tramite realtà aumentata e realtà mista. Infatti, sta valutando diverse tecnologie per capire come applicarle al sistema **C4I** (sistemi di Comando Controllo Comunicazioni Computer e Intelligenza), ovvero un moderno sistema militare le cui fasi operative richiedono un approccio innovativo per assicurare i servizi richiesti. La manutenzione eseguita sui sistemi C4I può essere divisa in due tipi principali:

- la manutenzione preventiva, basata su attività periodiche per mantenere le prestazioni del sistema all'interno dei parametri di progettazione;

- la manutenzione correttiva, per superare i guasti del sistema che ne riducono le prestazioni.

Per quanto riguarda la parte pratica della formazione del personale tecnico, adibito alla manutenzione sui sistemi C4I, l'ITAF utilizza normalmente sistemi reali, ovvero repliche di quelli operativi o parte di essi, dedicati alla formazione. Questo approccio richiede l'acquisizione di sistemi molto costosi utilizzati esclusivamente per la formazione che non possono essere utilizzati operativamente e devono essere mantenuti aggiornati come quelli operativi. I modelli virtuali di questi sistemi possono sostituire quelli reali, riducendo i costi e aumentando la flessibilità della formazione pratica. Dunque, considerando questi benefici, l'ITAF ha deciso di avviare una Prova di Concetto **PoC** sulla Realtà Mista, utilizzando la tecnologia *HoloLens* applicata alla manutenzione e alla formazione dei sistemi C4I.

L'equipaggiamento *HoloLens*, visibile in Figura 2.15, è un computer olografico completamente autonomo, che funziona con il sistema operativo Windows 10. È totalmente senza fili e permette di posizionare ologrammi nell'ambiente fisico, fornendo un modo per vedere il mondo attraverso un display. Gli ologrammi visibili mediante *HoloLens* appaiono realistici, possono muoversi, essere modellati e cambiare in base all'interazione con l'utente o l'ambiente fisico in cui sono visibili.

Questo dispositivo permette diverse modalità di interazione:

- **Suono spaziale:** permette all'utente di ascoltare l'audio binaurale che può simulare effetti spaziali, così che l'utente possa percepire e localizzare virtualmente un suono e identificarlo con precisione nell'ambiente;
- **Tracciamento dello sguardo:** consente all'utente di porre l'attenzione del dispositivo, in base al movimento della pupilla, verso ciò che egli percepisce, per migliorare la navigazione. L'*HoloLens* è in grado di determinare con precisione cosa mostrare e in quale posizione, creando illusioni 3D stereoscopiche;
- **Input gestuale:** permette all'utente di usare dei gesti predefiniti per eseguire varie funzioni. Ad esempio, per aprire su uno schermo il menù di navigazione dell'interfaccia utente, simile al tasto Windows su una tastiera, si usa il gesto *Bloom*, mentre il gesto *Air tap* viene usato per selezionare i comandi del menù (simile al click del tasto sinistro del mouse);
- **Supporto vocale:** permette di usare comandi vocali (simile a fare una domanda a Cortana, Siri, Google) e rende disponibile la funzione *Speech-To-Text*, ovvero il riconoscimento vocale.

Nell'articolo [33] si cerca di identificare l'architettura complessiva necessaria per l'adozione e la gestione di questa tecnologia nell'Aeronautica Militare Italiana. Si è pensato ad un'architettura abbastanza semplice in modo da ridurre i costi e il tempo necessario per sviluppare il sistema e, allo stesso tempo, per avere qualcosa che possa rappresentare le funzionalità principali di un futuro sistema operativo. Come rete di comunicazione si utilizza l'accesso a internet tramite rete mobile 4G; in particolare, il sito remoto è composto da un telefono cellulare 4G con funzione di tethering per connettere gli *HoloLens* tramite wifi. Il sito centrale è dotato di un PC con Windows connesso a Internet pubblico tramite la rete mobile 4G, come mostrato in Figura 2.16. L'applicazione utilizzata per connettere



Figura 2.15: Microsoft HoloLens

il PC agli *HoloLens* è il servizio pubblico di *Skype*, fornito da *Microsoft*, attraverso cui i due siti possono comunicare con voce, video e alcune azioni specifiche. Quest'architettura

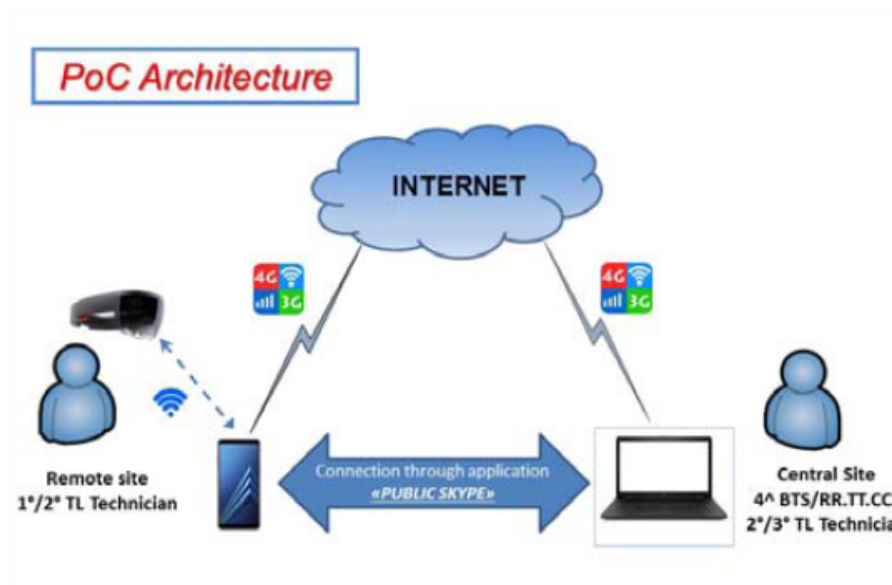


Figura 2.16: Architettura PoC

rappresenta una prima iterazione di questo percorso e risulta subito evidente una grande limitazione: l'uso di internet tramite rete mobile pone limiti sulla disponibilità della rete 4G in alcuni siti militari remoti, sulla larghezza di banda relativa alla forza del segnale radio e sulla sicurezza delle informazioni che, per alcuni sistemi militari, possono essere classificate. Questi problemi sono superabili attraverso la WAN privata dell'ITAF chiamata *AERONET*, disponibile in tutti i siti ITAF e come software di comunicazione si passerebbe ad un servizio *Skype* privato, in esecuzione su questa rete privata.

Questa tecnologia è stata sperimentata in diversi scenari manutentivi come nei sistemi radar di controllo del traffico aereo ATCR, nei sistemi di comunicazione radio G/A/G, visibile in Figura 2.17, e nei sistemi di navigazione radio con risultati molto promettenti;

i tecnici guidati da remoto sono stati in grado di eseguire le attività di manutenzione pianificate. Tuttavia, l'introduzione della MR nella manutenzione con un equipaggiamento come *HoloLens*, richiede una revisione delle attuali procedure di manutenzione e una formazione specifica del personale tecnico.



Figura 2.17: Applicazione del PoC

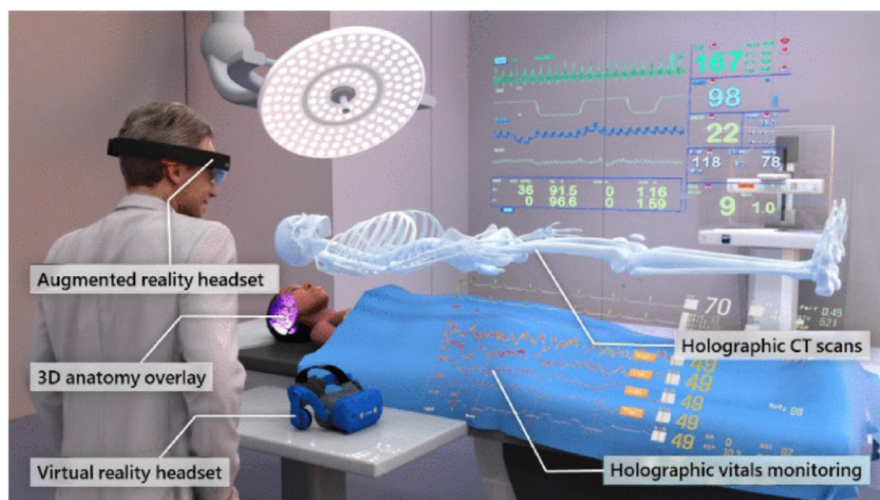
## 2.3 Altri campi di interesse della realtà virtuale

La tecnologia della Realtà Virtuale sta gradualmente avendo un impatto significativo su vari settori industriali.

### 2.3.1 Medicina

Le tecnologie hardware e software di realtà virtuale stanno evolvendo rapidamente per soddisfare le specifiche esigenze della medicina e della pratica sanitaria, e sono destinate a diventare tecnologie pilastro sia all'interno sia al di fuori dell'ospedale, integrando l'assistenza oltre la struttura ospedaliera, deframmentando il percorso di cura del paziente. AR e VR offrono un mezzo senza pari per interagire con le informazioni digitali, abbattendo le barriere tra il mondo reale e il mondo digitale; la capacità di integrare l'assistenza sanitaria digitale con la pratica clinica può portare a un miglioramento della prestazione e a migliori risultati medici [34]. Un'illustrazione dell'applicazione dell'AR è mostrata nella Figura 2.18, dove i dati del paziente in tempo reale e le rappresentazioni 3D dei dati anatomici possono essere sovrapposti davanti alla vista del chirurgo; il medico infatti indossa un headset AR molto leggero che sovrappone dati alla vista del mondo reale.





**Figura 2.18:** Possibilità della Realtà Aumentata

AR e VR svolgono un ruolo significativo nella formazione della prossima generazione di professionisti chirurgici. Queste tecnologie permettono lo sviluppo di nuovi modelli di apprendimento digitale consentendo un'educazione alle scienze di base, apprendimento basato su scenari, gestione delle crisi e esame di casi clinici in modo coinvolgente e interattivo. VR e AR possono potenzialmente migliorare le opportunità di formazione per consentire un'acquisizione delle competenze più rapida ed evitare che i pazienti siano esposti a chirurghi nella fase iniziale della loro curva di apprendimento, anche se sono state espresse preoccupazioni riguardo al fatto che queste tecnologie siano, al momento, al di sotto del livello richiesto per la padronanza in alcune aree di pratica [35].

Attraverso azioni mentali o fisiche, gli umani acquisiscono conoscenza e informazioni contestuali su un compito che viene eseguito; queste informazioni diventano cruciali quando qualcuno ha bisogno di eseguire l'intervento in un momento successivo, poiché, spesso, le informazioni su come eseguire una determinata procedura diventano chiare solo quando lo scenario contestuale viene presentato all'umano. Ciò ha ispirato approcci di apprendimento applicati in contesti educativi che si basano sulla teoria dell'apprendimento situato, sostenendo che l'educazione e la formazione funzionano meglio quando la conoscenza è appresa nel contesto in cui verrà sfruttata [36]. La formazione chirurgica basata su VR si è dimostrata uno strumento didattico efficiente e conveniente, poiché dagli studi si evince come gli utenti sviluppino competenze tecniche paragonabili a quelle sviluppate tramite tecniche standard di formazione chirurgica, con il vantaggio di prestazioni nettamente migliorate in termini di competenze non tecniche e autostima, come mostrato in Figura 2.19. AR e VR possono consentire una prepianificazione dettagliata delle procedure mediche, una migliore identificazione di eventuali incidenti, revisioni postoperatorie e interazioni remote. Molti progetti di ricerca e clinici indagano l'uso di AR e VR nel contesto dell'esplorazione multidimensionale delle informazioni di imaging medico, come le scansioni di tomografia computerizzata CT o l'angiografia a risonanza magnetica MRI.

L'obiettivo è la pianificazione diagnostica e terapeutica con la capacità di ricostruire e



**Figura 2.19:** Esempio di formazione chirurgica

visualizzare immagini complesse in 3D e simulare procedure. Questo è particolarmente rilevante nel contesto della pianificazione chirurgica perché una procedura complessa specifica per il paziente può essere provata e possono essere immaginati diversi scenari, come i margini di resezione del tumore e la valutazione della distanza, come valutato da Hansen [37], utilizzando un approccio AR nella chirurgia del fegato.

Nell'esempio presentato nella Figura 2.20, Pratt et al. hanno condotto una serie di studi dove hanno utilizzato l'AR di *Microsoft, HoloLens*, per la chirurgia di ricostruzione degli arti, sovrapponendo le informazioni derivanti dalla scansione di tomografia computerizzata sul corpo di un paziente per guidare le incisioni operative. Inoltre, l'uso di questo dispositivo non compromette la sterilità ambientale grazie alla sua natura contenuta e alla capacità di essere utilizzato con gesti della mano e controllo vocale. Con minimi cambiamenti alla procedura, hanno concluso che il visore *Microsoft HoloLens* si è dimostrato uno strumento potente che ha il potenziale per ridurre il tempo anestetico e per fornire supporto remoto al chirurgo operante [38].

Oltre ai benefici, vanno prese in considerazione anche le possibili limitazioni di queste tecnologie. Ad esempio, la quantità di informazioni presentate ad un chirurgo tramite AR durante un intervento potrebbe diventare distraente o imprecisa a causa di problemi di latenza o di disallineamento dei dati di imaging o delle annotazioni. Pertanto, è necessario visualizzare solo i dati importanti o elaborare un metodo per filtrare le diverse categorie di informazione. Hansen et al. hanno proposto una metodologia per la riduzione delle richieste cognitive, dove hanno sviluppato una serie di algoritmi e gradienti di texture per la valutazione della distanza nella chirurgia del fegato, con l'obiettivo di aumentare l'accettazione dell'AR in sala operatoria [37].

Un altro fattore chiave è assicurare al personale clinico la fedeltà anatomica di qualsiasi immagine, in particolare quando potrebbe esserci un intervento guidato da remoto. Le principali sfide rimanenti per la ricerca futura includono il miglioramento dell'esperienza nei simulatori VR/AR immersivi di movimento, questioni relative al fattore umano e lo svolgimento di ampi studi clinici.



Figura 2.20: AR in medicina

### 2.3.2 Automotive

Nel settore automobilistico, la tecnologia VR è stata utilizzata dai produttori di attrezzature originali OEM nell'analisi della produzione, nel processo di assemblaggio, negli studi ergonomici e nei test drive. Ad esempio, Ford dispone di un laboratorio VR, presso il Michigan Development Center denominato Ford Immersive Vehicle Environment Lab FIVE, utilizzato per validare lo stile e l'artigianalità del veicolo oltre all'ergonomia. Un altro esempio è il Virtual Innovation Center VIC di Jaguar e Land Rover nel Regno Unito, un centro che guida lo sviluppo della VR per applicazioni automobilistiche, con prestazioni avanzate di hardware e software. Studi indicano che la dimensione del mercato della realtà virtuale nell'automotive in Nord America era di 82 milioni di dollari nel 2017, passando a 130 milioni nel 2018 e si prevede una crescita esponenziale nell'applicazione della VR nel campo automobilistico negli anni successivi [39].

In generale, in questo settore, la VR ha portato benefici in diversi aspetti:

- **Design:** Il design automobilistico è un processo che richiede continue modifiche e revisioni, con la necessità di tornare più volte alle decisioni precedenti prima che l'auto raggiunga finalmente la produzione. Questa caratteristica è stata vista come uno degli aspetti più costosi e che richiedono più tempo. Infatti, come affermato da Gomes e Zachmann [40], le fasi iniziali del design possono incidere fino al 70% del costo totale di un prodotto; quindi, la VR può ridurre costi e tempi sostituendo i modelli fisici con quelli virtuali. Un'altra utilità della VR nel design è la possibilità di avere team multidisciplinari e distribuiti in tutto il mondo, per lavorare insieme sul prototipo allo stesso tempo; a questo proposito, Lehner e De Fanti [41] hanno testato la VR per lo sviluppo di grandi veicoli, dimostrando una riduzione dei costi e dei tempi e un aumento della qualità;
- **Prototipazione Virtuale:** I Prototipi Virtuali sono, in alcuni casi, utilizzati per sostituire i modelli fisici. Nel processo decisionale, la prototipazione virtuale può semplificare le procedure ed evitare l'effetto "collo di bottiglia" che si manifesta da

errori nelle fasi iniziali dello sviluppo di un componente e revisioni costanti, portando alla necessità di ricostruire i modelli fisici [42];

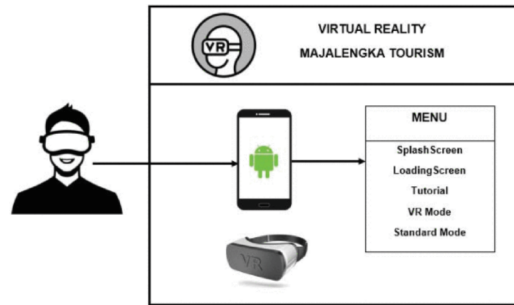
- **Produzione:** L'applicazione della VR alla produzione è chiamata *Virtual Manufacturing* ed è stata definita come l'uso della VR o dei computer per lo sviluppo di un prodotto. I suoi vantaggi vanno dal miglioramento del processo decisionale alla riduzione dei costi, al potenziamento delle misure di rischio e al controllo dei processi di produzione [43];
- **Assemblaggio Virtuale:** Correlato alla produzione, è possibile effettuare l'assemblaggio e il disassemblaggio di oggetti virtuali. Questo permette la valutazione del benessere dei lavoratori e delle misure di salute e sicurezza grazie alla possibilità di creare rappresentazioni virtuali dei luoghi di lavoro. È stato osservato che offre un enorme vantaggio nel processo di design e nelle nuove implementazioni di strumenti migliorando la qualità del prodotto e riducendo il tempo di immissione sul mercato [44];
- **Formazione:** È stato dimostrato che la formazione in realtà mista delle operazioni di servizio automobilistico è preferita dai tirocinanti rispetto agli approcci tradizionali basati sull'osservazione [45]. Inoltre, gli studi hanno mostrato che, con l'addestramento simulato, il completamento dei compiti è migliorato rispetto alla formazione su attrezzature reali, passando dal 50% di successo senza alcun errore al 66%. In aggiunta, come evidenziato per i compiti di assemblaggio e disassemblaggio, le informazioni ricordate dopo due e quattro settimane sono maggiori per i partecipanti formati con la VR rispetto ad altri sistemi.

### 2.3.3 Turismo

Molti hotel nel mondo hanno utilizzato la tecnologia della realtà estesa nella manutenzione, nei servizi agli ospiti e nel servizio in camera, facendo uso di ologrammi con avatar e display interattivi. Nel rendere queste innovazioni, come la realtà aumentata e la realtà virtuale, fruibili per le persone, gli smartphone e i relativi gadget sono strumenti essenziali. L'uso della VR nel settore dei viaggi e del turismo può portare cambiamenti nelle decisioni relative alla destinazione e può essere un mezzo produttivo per eliminare il divario tra ciò che i turisti si aspettano e ciò che in realtà sarà l'esperienza [46]. La ricerca sull'uso della realtà virtuale nel turismo riscontra un atteggiamento positivo da parte dei clienti, principalmente per via della sensazione di immersione che risulta essere già essa stessa un forma di attrazione turistica, e incrementa l'interesse nel visitare la località reale [47].

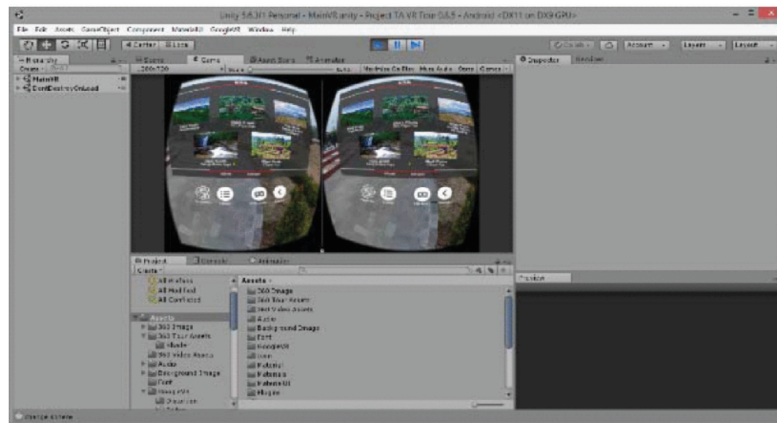
Un'applicazione di questo concetto si può ritrovare nella Reggenza di Majalengka in Indonesia, località con un notevole potenziale turistico da sviluppare. Si tratta di una serie di video a 360 gradi prodotti direttamente nella località turistica, messi insieme in un unico menù principale e presentati sotto forma di realtà virtuale con la funzione di tracking, tramite sistema GPS, per simulare l'esperienza di camminata nelle attrazioni. Il dispositivo utilizzato è uno smartphone con l'applicazione turistica di realtà virtuale installata [48].

Il framework dell'applicazione è mostrato in Figura 2.21 e si può notare come il menù progettato abbia funzionalità come schermate iniziali, schermate di caricamento, tutorial,



**Figura 2.21:** Framework dell'applicazione

modalità di realtà virtuale e modalità standard. Gli utenti possono scegliere la visualizzazione della realtà virtuale utilizzando gli occhiali *Google Cardboard*, un dispositivo di realtà virtuale molto economico che necessita di uno smartphone per essere utilizzato. Lo sviluppatore



**Figura 2.22:** Ambiente di sviluppo Unity

utilizza il software *Unity* versione 5.6 per creare un'applicazione di realtà virtuale che usa un dispositivo *Android*, Figura 2.22, e, per scattare foto a 360 gradi, invece, sfrutta il dispositivo *Samsung Gear 360*, perché il processo di scatto può essere fatto rapidamente e senza editing nella fase di post-processing.

Per la validazione di questo sistema è stato proposto un questionario composto da 10 domande con punteggi da 1 a 4; dai risultati emerge che l'86,6% delle persone ha espresso interesse nell'uso dell'applicazione di realtà virtuale, l'86,6% ha convenuto che l'applicazione sia facile da usare e l'81,6% ha dichiarato di sentirsi come se fossero sul luogo di un'attrazione turistica durante l'utilizzo.

# Capitolo 3

## Hardware e software

In questo capitolo viene descritta tutta la componentistica hardware e software utilizzata per simulare la cabina dell'Airbus A320 nell'ambiente virtuale e per realizzare le procedure operative statiche e dinamiche.

### 3.1 Software

Il progetto di tesi è stato realizzato presso il laboratorio di Simulazione del Volo del Dipartimento di Ingegneria Meccanica e Aerospaziale del Politecnico di Torino, quindi tutti i software che verranno analizzati sono installati nel computer di tale laboratorio. Questi software sono essenziali per l'integrazione tra l'ambiente di realtà virtuale e la simulazione di volo, due ambiti chiave del progetto, con l'obiettivo di creare un'interfaccia fluida e funzionale per l'utente, in grado di rendere l'esperienza il più immersiva possibile.

I software utilizzati si possono raggruppare in tre categorie principali:

- **Software per la realtà virtuale:** fondamentali per la fruizione dell'ambiente virtuale e dell'interazione con esso (*SteamVR*, *Unity*, *WEAVR*).
- **Software per la simulazione di volo:** necessario per rendere realistica l'esperienza di pilotaggio, riproducendo con accuratezza la dinamica e le condizioni di volo (*X-Plane 11*).
- **Software per il collegamento dei client:** sono utilizzati per integrare la realtà virtuale con la simulazione di volo, assicurando la sincronizzazione tra i vari componenti del progetto (*Simulation Hub*, *X-Plane connect*, *Visual Studio*).

Per quanto riguarda la provenienza dei software, il progetto si avvale di soluzioni *open source*, scaricate e installate direttamente dai siti ufficiali, di software con licenza forniti da Txt Group e di software acquistati appositamente per il progetto attraverso store ufficiali.

#### 3.1.1 SteamVR

SteamVR è una piattaforma di realtà virtuale sviluppata da Valve Corporation, progettata per immergere l'utente in ambienti virtuali tridimensionali interattivi [49]. È compatibile

con vari dispositivi di realtà virtuale, come HTC Vive, Oculus Rift, Valve Index e altri visori compatibili con OpenVR e ne consente l'utilizzo in diverse applicazioni come l'ambiente Unity.

Prima di entrare nella *Home* di SteamVR, visibile in Figura 3.1, occorre configurare



**Figura 3.1:** Ambiente virtuale SteamVR

l'ambiente fisico dove si vuole operare. Si devono, innanzitutto, posizionare le *Base Station* che, tramite infrarossi, rilevano lo spazio disponibile nella realtà e delimitano un confine virtuale entro cui è permesso il movimento in VR; successivamente viene calibrato il visore e i relativi controller e il software diventa disponibile per lavorare in background quando viene richiamato da applicativi compatibili che, in questo specifico caso, saranno Unity e un suo plug-in, WEAVR, permettendo il corretto funzionamento dell'hardware.

In questo progetto, in particolare nelle procedure statiche, vengono utilizzati alcuni *prefabs*, ovvero dei pacchetti pronti all'uso, che forniscono determinate funzioni in VR:

- **Teleport point:** consente il teletrasporto in un punto configurato a priori nello spazio virtuale, utilizzando l'apposito trackpad per il puntamento disponibile sul controller. In questo progetto è stato utilizzato per permettere lo spostamento tra il sedile del pilota e quello del copilota nella cabina di pilotaggio dell'Airbus A320.
- **Teleport area:** permette il teletrasporto all'interno di un'area predefinita; quindi, in questo caso, vi è la possibilità di muoversi in infiniti punti all'interno di quest'area che, in questo progetto, è l'intero pavimento della cabina di pilotaggio.
- **Hand Poses:** consente di visualizzare la posa della mano quando, in realtà virtuale, la si avvicina ad un oggetto interagibile della scena, con posizionamenti diversi a seconda del comportamento di quest'ultimo.

### 3.1.2 Unity

*Unity* è un motore grafico ideato per lo sviluppo di contenuti multimediali come videogiochi, che offre agli sviluppatori un potente ambiente integrato per la creazione di giochi e applicazioni interattive 2D, 3D, realtà virtuale VR e realtà aumentata AR [50]. Nasce nel 2005 dalla *Unity Technologies* e nel corso degli anni è diventato uno dei *game engine*

più popolari grazie alla sua facilità d'uso e alla vasta gamma di funzionalità. *Unity* è un ambiente multi-piattaforma in quanto consente di sviluppare giochi e app che possono essere distribuiti su oltre 25 piattaforme diverse, come sistemi desktop (Windows, macOS, Linux), mobile (iOS, Android), console (PlayStation, Xbox, Nintendo Switch), Web e VR/AR; dispone di un visual editor intuitivo, mostrato in figura 3.2, che facilita la progettazione di scene senza dover creare un codice complesso; vanta una vasta libreria di risorse pronte all'uso, come modelli 3D, texture, script e anche strumenti aggiuntivi per estendere le funzionalità dell'editor di *Unity*; inoltre, supporta diversi linguaggi di programmazione come *C++* e *Java*, ma al suo interno viene utilizzato *C#*.

In questo progetto vengono utilizzate due versioni differenti di *Unity*, in quanto le procedure statiche sviluppate vanno ad aggiungersi a quelle prodotte in un progetto di tesi precedente [9] in cui, per evitare di perdere il lavoro fatto, non è stato possibile aggiornare né la versione di *Unity*, la *2019.4.28f1*, né quella di *WEAVR*; mentre, per quanto riguarda le procedure dinamiche è stato aggiornato *Unity* alla versione *2021.3.16f1* e, su indicazione di *TXT*, anche *WEAVR* è stato installato in un nuovo progetto nella sua versione più recente. Nella



**Figura 3.2:** Home page Unity - Hierarchy

Figura 3.2, come accennato in precedenza, viene mostrata la schermata principale di *Unity* che dispone di sei pannelli principali:

- **Hierarchy:** in questa sezione vengono raccolti gli oggetti di scena, i *game object*, tra loro interconnessi tramite gerarchia del tipo *parent-child*: possono essere presenti tanti oggetti figlio connessi ad un solo padre.
- **Scene:** in questa finestra, selezionata in figura, viene mostrato l'ambiente dove si costruisce il modello 3D attraverso i *game object* e i relativi comportamenti.
- **Game:** contiene la scena definitiva, non direttamente modificabile, che apparirà una volta lanciata la simulazione. Nel caso di questo progetto, si avvia la simulazione in



VR della cabina dell'Airbus A320 e l'utente può interagire con i vari oggetti di scena abilitati.

- **Inspector:** finestra relativa al *game object* selezionato che permette di modificarne le proprietà, andando ad agire sui suoi *component*.
- **Project:** in questa sezione si può navigare nella *directory* del progetto dove sono presenti tutti gli elementi che permettono il corretto funzionamento dell'applicazione come Asset, script, audio, video, texture, modelli 3D e materiali.
- **Console:** questa finestra mostra warning ed errori per avvisare l'utente una volta lanciata la scena.

### 3.1.3 Pacelab WEAVR

*Pacelab WEAVR* è un software che permette progettazione e sviluppo di sistemi di formazione virtuale. Fornisce una piattaforma editor integrata, che supporta l'intero processo di produzione dalla progettazione alla consegna finale. Il suo approccio visivo, basato su template, richiede poche o nessuna competenza di programmazione o scripting, permettendo agli esperti in materia di creare compiti in modo efficiente e senza coinvolgere disegnatori 3D o ingegneri del software. Viene rilasciato, sotto licenza, insieme ad un manuale di utilizzo [51] e si installa direttamente importando in *Unity* il pacchetto *.unitypackage*, essendo sviluppato come suo *plug-in*.

*Pacelab WEAVR* comprende tre moduli principali:

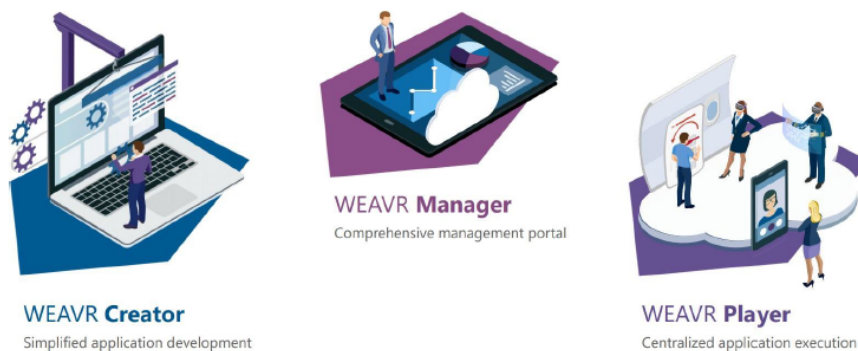


Figura 3.3: WEAVR Enviroment

- **WEAVR Creator:** è uno strumento che fornisce librerie per la creazione di contenuti basati su procedure in realtà virtuale e mista. Il *Creator* consente la definizione di comportamenti e animazioni di elementi 3D, implementando strumenti base con comportamento e animazioni standardizzati come pulsanti e leve, camere e script di movimento e un editor di flussi di lavoro che consente agli utenti di modellare procedure definendo passaggi, gruppi di passaggi, flussi di navigazione e animazioni.

- **WEAVR Manager:** Il modulo *Manager* raccoglie e distribuisce contenuti sviluppati all'interno del *Creator*, consente la gestione di utenti, gruppi e assegnazioni di lavoro, oltre al controllo del monitoraggio in tempo reale.
- **WEAVR Player:** Il *Player* consente di accedere, selezionare ed eseguire le procedure disponibili in diverse modalità, una volta esportate dal *Creator*. Quest'applicazione è disponibile su vari sistemi inclusi PC desktop, dispositivi mobili e dispositivi VR, come HTC Vive, Oculus e Microsoft HoloLens. Il *Player* deve connettersi a *WEAVR Manager* tramite un login per aggiornare lo stato delle sue procedure esistenti e scaricarne di nuove; inoltre, consente anche la condivisione di una procedura tra più studenti.

## Installazione

Dopo aver creato un account su *WEAVR Manager* e aver quindi fatto il login, si scarica dal sito ufficiale il pacchetto contenente *WEAVR Creator*. Per installarlo bisogna aprire il software *Unity* e andare su *Window > Package Manager > Add Package from tarball...* e selezionare il pacchetto appena scaricato, come mostrato in Figura 3.4, ed infine attivare la licenza andando su *WEAVR > Intro* per poi fare l'accesso con le proprie credenziali.

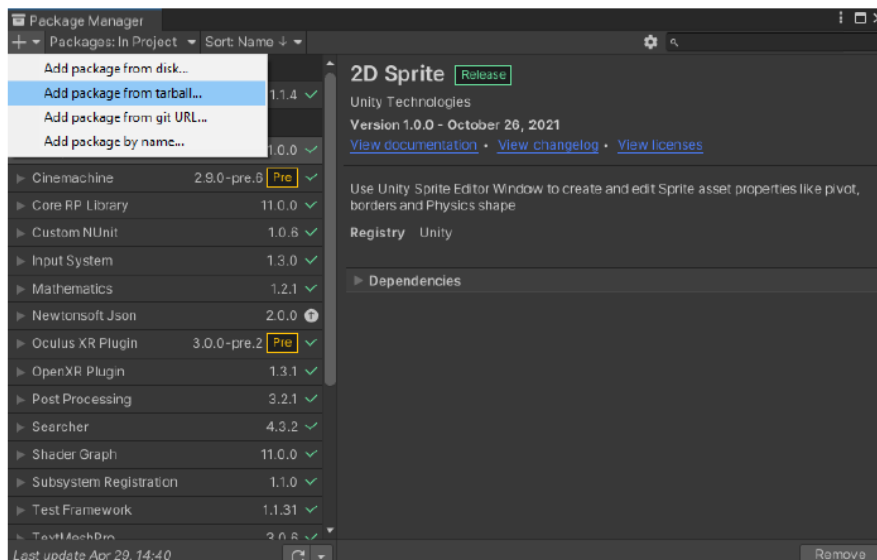


Figura 3.4: WEAVR Environment

## Setup scena

Prima di poter utilizzare il tool, è necessario configurare la scena andando su *WEAVR > Setup Scene*; successivamente bisogna configurare *WEAVR* in base al visore di realtà virtuale utilizzato cliccando su *Project Settings > XR Plug-In Management*, abilitando il plug-in *OpenXR* tra la lista di quelli disponibili e, infine, scegliendo su *Interaction Profiles*

quello relativo al visore che si vuole configurare che in questo specifico caso sarà *HTC Vive Controller Profile*.

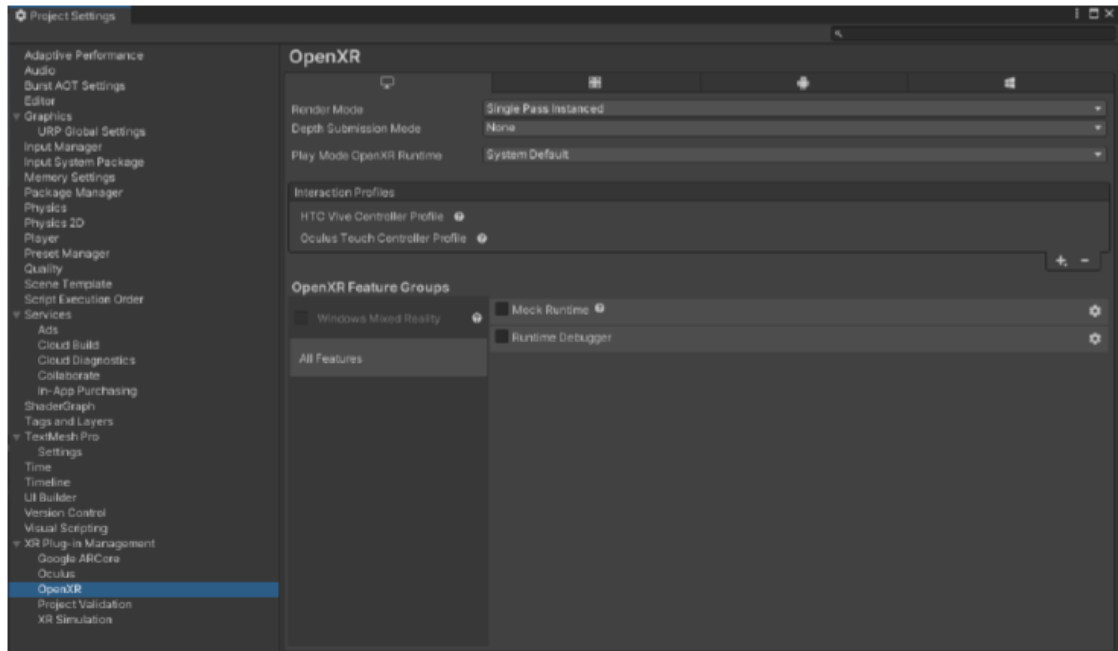


Figura 3.5: WEAVR Environment

## Component

*WEAVR* contiene al suo interno degli script predefiniti che su *Unity* vengono definiti componenti e si possono aggiungere ad un *Game Object* per fornire un determinato comportamento, selezionandoli dal pannello *Inspector*. Questi componenti sfruttano la fisica implementata in *Unity* per rendere le interazioni più naturali, specialmente in ambiente VR. Si elencano e descrivono brevemente i principali *Components* utilizzati in questo progetto:

- **WEAVR Object:** questo componente permette di interagire con un oggetto in ambiente virtuale; per poterlo afferrare è necessario impostare il layer dell'oggetto su *Grabbable*.
- **Executable:** assegnando questo comportamento, l'oggetto esegue una determinata azione quando si verifica una condizione definita a priori.
- **Push Button:** questa funzione permette di premere un tasto osservando il comportamento di un vero bottone.
- **Switch:** questo comportamento trasforma un oggetto, ad esempio un interruttore con più di due vie, in uno switch, assegnandogli diverse posizioni nello spazio.

## Procedure Editor

In questo pannello avviene la creazione di procedure guidate e per abilitarlo occorre andare su *WEAVR > Procedures > Procedure Editor*. La procedura consiste in una programmazione sequenziale costituita da diversi step in successione chiamati nodi e il passaggio dal precedente al successivo avviene al soddisfacimento di una determinata condizione.

Nel *Procedure Editor* vengono mostrati i vari nodi creati tramite una modalità di vista a grafi, come si vede in Figura 3.6, e durante l'esecuzione di una procedura diventeranno di colore verde man mano che vengono svolti.

Esistono tre modalità di esecuzione di una procedura:

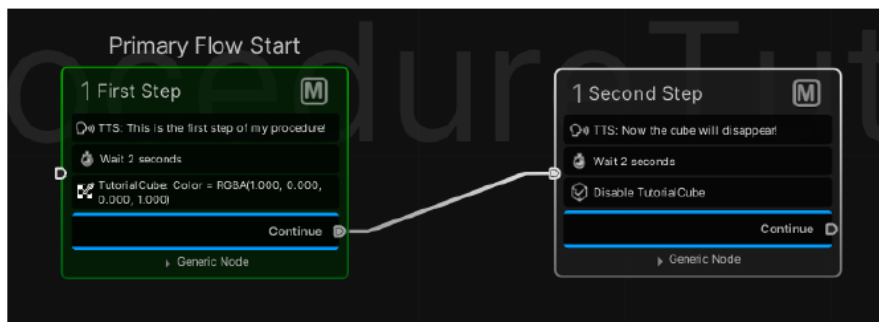


Figura 3.6: WEAVR Environment

- **Automatic Execution Mode:** viene interamente gestita da *WEAVR* ed è utilizzata per mostrare all'utente come dovrebbe essere eseguita l'intera procedura. In questo caso l'utente è quasi totalmente uno spettatore in quanto è richiesta un'interazione minima con l'ambiente come il movimento della camera per osservare gli step.
- **Guided Execution Mode:** in questo caso l'utente è guidato nell'esecuzione della procedura e deve interagire con gli oggetti, in modo coerente con quanto definito dalla procedura, per avanzare nei vari step, senza tuttavia poter navigare tra i nodi a piacimento, in quanto obbligato a procedere cronologicamente.
- **Feedback Execution Mode:** in questa modalità l'utente non è guidato in alcun modo e deve quindi procedere nell'esecuzione autonomamente ricordandosi ordine e tipo di azione da eseguire; si tratta, dunque, di una modalità utile per testare l'apprendimento di una procedura.

## Procedure Inspector

In questo menù si definiscono le caratteristiche dei nodi che sono suddivisi in tre sezioni:

- **Enter Action:** sono azioni che vengono eseguite all'ingresso del nodo.
- **Exit Conditions:** si tratta della condizione o condizioni di uscita che, una volta soddisfatte, permettono il passaggio al nodo successivo.

- **Exit Action:** sono azioni che si eseguono automaticamente quando la condizione di uscita è soddisfatta, prima di accedere allo step successivo.

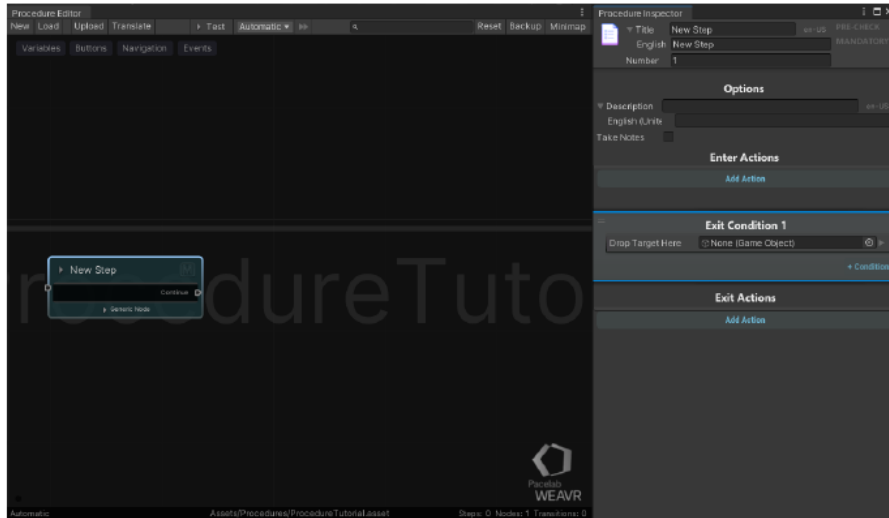


Figura 3.7: Procedure Editor e Procedure Inspector

### 3.1.4 WEAVR Simulation Hub

*WEAVR Simulation Hub* è un modulo concepito per creare connessioni tra diversi punti di simulazione, facilitando lo scambio di informazioni e permettendo, dove necessario, la condivisione di dati tramite memoria condivisa[52]. Diversamente da un sistema *Plug&Play*, il *SimHub* necessita di essere specificamente configurato per ogni specifica simulazione. Le sue componenti chiave sono descritte di seguito e mostrate in figura 3.8:

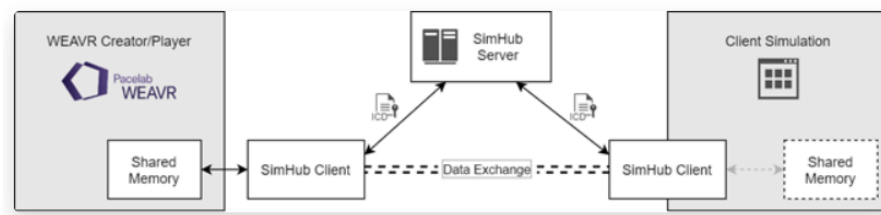


Figura 3.8: Procedure Editor e Procedure Inspector

- **SimHub Server:** monitora i *client* connessi e gestisce ogni nuova connessione o disconnessione; quando si connette un nuovo *client*, il server riceve il file *ICD*, *Interface Control Document*, contenente dati, da quest'ultimo e lo informa su quali altri *client* deve connettersi per avere tutti i dati richiesti. In pratica, i *client* scambiano dati tra loro utilizzando il protocollo di messaggi fornito. Si tratta un'applicazione autonoma

che utilizza una connessione TCP per la gestione dei *client* e una connessione UDP per lo scambio di dati. In questo progetto è stato fornito da *TXT* come eseguibile, lanciabile tramite file *.bat*.

- **SimHub Client:** gestisce lo scambio di dati con altri *client* ed ha anche la capacità di accedere alla memoria condivisa per leggere e scrivere dati utilizzati dalla simulazione; ogni *client* fornisce al server un file *ICD* per scambiare informazioni con un altro *client*. In questo progetto il *client* che si occupa delle informazioni lato *Unity* è stato fornito da *TXT*, mentre l'altro che comunica con *X-Plane* è stato progettato in questo lavoro in *C++*.
- **Shared Memory:** *SimHub* comprende una libreria dedicata per l'accesso alla memoria condivisa permettendo allo sviluppatore di scrivere un codice standard per accedere alla memoria condivisa anche senza usare il linguaggio di programmazione *C++*; infatti, *WEAVR*, nonostante sia scritto in *C#*, può accedere alla memoria condivisa attraverso questa libreria.

### 3.1.5 Visual Studio

Visual Studio è un ambiente di sviluppo integrato *IDE* prodotto da Microsoft [53]; è utilizzato per sviluppare applicazioni per computer, web, servizi cloud, e mobile, supportando vari linguaggi di programmazione come *C#*, *VB.NET*, *C++*, e *F#*. Visual Studio facilita il ciclo di vita completo dello sviluppo del software, dall'ideazione al codice, fino al test e al deployment grazie alla sua interfaccia utente che offre numerose funzionalità, tra cui l'editing intelligente del codice, il debugging avanzato e strumenti per analizzare le prestazioni dell'applicazione. In questo progetto si utilizza la versione 2019 presente nel PC del laboratorio.

In questo ambiente sono stati usati due linguaggi di programmazione:

- *C++* per scrivere il codice del *client X-Plane* in modo che fosse compatibile con *SimulationHub*.
- *C#* per alcuni script inseriti in *WEAVR* per delle specifiche funzioni che verranno descritte nei capitoli successivi.

### 3.1.6 X-Plane

*X-Plane* è un avanzato software di simulazione di volo, disponibile per una varietà di piattaforme e sistemi operativi, che si posiziona al pari di *Microsoft Flight Simulator* per il suo realismo e la sua completezza nel panorama dei simulatori aeronautici a livello mondiale [54]. Il suo approccio alla simulazione del volo si basa sulla Teoria dell'Elemento di Pala, una metodologia che analizza ogni parte dell'aeromobile come un elemento distinto, modellandone il comportamento aerodinamico; questo permette di ricostruire il volo dell'intero aeromobile attraverso l'integrazione dei vari effetti di forze e accelerazioni, offrendo così una rappresentazione fedele e dettagliata di diversi tipi di velivoli. *X-Plane* è progettato sia per addestrare i nuovi piloti, fornendo un ambiente di apprendimento sicuro e controllato, sia per permettere ai piloti più esperti di esplorare scenari di volo avanzati. Include un database ampio di aeroporti, accuratamente modellati in 3D e supporta la realtà

virtuale, permettendo agli utenti di immergersi completamente nelle simulazioni di volo, per un'esperienza ancora più vicina alla realtà.

La versione acquistata nel computer del Laboratorio di Simulazione del Volo del Politecnico di Torino, per Windows, è X-Plane 11 che include una variegata flotta di velivoli, che spazia dagli aerei leggeri agli aerei di linea, fino agli elicotteri; i modelli in questione sono: 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. Tuttavia è stato acquistato separatamente un velivolo della famiglia *A320* per svolgere questo lavoro di tesi; il velivolo in questione è denominato *Airbus A321 ToLiss*, mostrato in Figura 3.9, ed è fornito sotto licenza ufficiale Airbus. Presenta un modello del cockpit con texture di alta qualità, due *MCDU* e autopiloti indipendenti, sistemi Fly-By-Wire di altamente realistici e sistemi di bordo riprodotti con precisione. Questo modello è dotato di numerosi *datarefs*, ovvero informazioni, accessibili



**Figura 3.9:** Procedure Editor e Procedure Inspector

tramite stringhe, rilasciate da *X-Plane*, in grado di poter gestire ogni aspetto della simulazione. Inoltre, dispone di un plug-in, rilasciato insieme al modello dell'aereo, chiamato *ToLiss Interactive Simulation Control System*, che permette di salvare la configurazione del velivolo e della scena, generata da *X-Plane*, in ogni momento della simulazione, per poterla successivamente ricaricare per ripetere lo stesso scenario, mantenendo invariati tutti i parametri. Infine, questo aereo garantisce la totale compatibilità con il simulatore fisico presente in laboratorio, in quanto non si riscontrano differenze di comandi e pilotaggio negli aerei appartenenti alla famiglia *Airbus A320*.

### 3.1.7 X-Plane Connect

*X-Plane Connect XPC* è un toolbox di ricerca open source sviluppato dalla *NASA* per facilitare l'interazione con *X-Plane*. *XPC* permette agli utenti di controllare e interagire con il velivolo attraverso un flusso di dati e impiega funzioni sviluppate in vari linguaggi informatici come *C*, *C++*, *Java*, *Matlab*, *Python* in tempo reale. Lo scambio di dati con il

simulatore avviene tramite *UDP* (User Datagram Protocol) [55].

Per utilizzarlo è necessario installare un plug-in in *X-Plane* e in questo caso sono state integrate le sue librerie per stabilire il collegamento con *SimHub* nella scrittura del *Client X-Plane*. Attraverso questo tool, è possibile modificare gli stati del velivolo nella simulazione, utilizzando stringhe di codice che vengono inviate al simulatore (*X-Plane*), ovvero *datarefs* e comandi.

XPC è scaricabile gratuitamente dalla piattaforma *Github* che fornisce un file *.zip* da estrarre all'interno della directory *plugins* di *X-Plane*, mentre il file *xplaneConnect.c* deve essere inserito all'interno della *Visua Studio Solution* del *Client X-Plane*. Le funzioni utilizzate nel *Client* creato sono:

- **openUDP:** configura un *socket UDP* per facilitare la connessione attraverso una specifica porta all'indirizzo IP del computer in cui è in esecuzione X-Plane.
- **getDREF:** riceve uno o più valori del *dataref*, inviati come stringa attraverso il *socket*.
- **sendDREF:** trasmette uno o più valori del *dataref*, inviati come stringa attraverso il *socket*; è necessario che il *dataref* in questione sia di tipo *writable*.
- **sendCOMM:** trasmette un comando come stringa attraverso il *socket*.

## 3.2 Hardware

Le componenti hardware utilizzate sono il visore VR, il PC del laboratorio e il simulatore fisico dell'airbus A320.

### 3.2.1 HTC Vive Pro

L'*HTC Vive Pro*, visibile in Figura 3.10, è un visore per la realtà virtuale e rappresenta un'evoluzione del precedente modello *HTC Vive*, offrendo miglioramenti significativi in termini di qualità visiva, comfort e immersione nell'esperienza VR [56].



Figura 3.10: HTC Vive Pro



## Qualità dell'immagine

L'*HTC Vive Pro* vanta una risoluzione combinata di 2880 x 1600 pixel, 1440 x 1600 pixel per occhio, che rappresenta un significativo aumento rispetto alla risoluzione del suo predecessore; questo miglioramento riduce lo *screen door effect*, l'effetto griglia, aumentando la nitidezza e la chiarezza delle immagini. Utilizza display AMOLED, caratterizzato da colori vivaci e neri profondi, migliorando l'immersione e la percezione della profondità.

## Audio

L'*HTC Vive Pro* dispone di cuffie integrate ad alta risoluzione, completamente regolabili in altezza e direzione per il massimo comfort, con supporto per l'audio spaziale 3D, rendendo l'audio direzionale e più realistico.

## Comfort e design

Il design è stato studiato per svolgere sessioni prolungate di utilizzo; il visore include una cinghia per la testa disegnata con un sistema di regolazione della dimensione e del bilanciamento del peso, rendendolo più comodo da indossare. Inoltre è permessa una regolazione che permette agli utenti di regolare la distanza tra le lenti per adattarsi meglio alla distanza interpupillare, migliorando il comfort visivo e riducendo l'affaticamento oculare.

## Tracciamento e sensori

Supporta il sistema di tracciamento *SteamVR Tracking 2.0*, che offre un'area di gioco più ampia e un tracciamento più preciso del movimento di testa e mani. Include inoltre diversi sensori come accelerometri, giroscopi e sensori di prossimità, che contribuiscono a migliorare la precisione del tracciamento.

## Controller

I controller dell'*HTC Vive Pro* sono progettati per offrire un'esperienza d'uso confortevole e intuitiva, avendo una forma che si adatta comodamente alla mano, con un'impugnatura curva che facilita la presa durante sessioni di gioco prolungate e sono dotati di vari pulsanti, tra cui un grilletto, un touchpad circolare, pulsanti di sistema e menù; il grilletto offre un feedback tattile, mentre il touchpad supporta input multi-direzionali e gesti. I controller includono anche un sistema di feedback aptico che simula la sensazione di toccare o interagire con oggetti nel mondo virtuale.

## Base station

Le base station dell'*HTC Vive Pro* svolgono un ruolo fondamentale nel tracciamento della posizione e dei movimenti del visore e dei controller in uno spazio tridimensionale. Per un tracciamento ottimale, le base station devono essere posizionate agli angoli opposti della stanza, preferibilmente montate su treppiedi o fissate al muro a un'altezza di circa 2 metri, inclinate verso il basso di 30-45 gradi; sono in grado di coprire efficacemente un'area di 10 metri quadrati ciascuna, con un angolo complessivo di 120 gradi.

### 3.2.2 Simulatore A320

Il simulatore di volo Airbus A320 presso il Dipartimento di Ingegneria Meccanica ed Aerospaziale DIMEAS del Politecnico di Torino, basato su X-Plane, rappresenta un'eccellente risorsa didattica e di ricerca. L'utilizzo di X-Plane come piattaforma di simulazione consente di ottenere un'alta fedeltà nella riproduzione delle dinamiche di volo, grazie alla suo avanzato modello di volo e alla capacità di simulare con precisione una vasta gamma di condizioni atmosferiche e di emergenza.

La replica in scala 1:1 del cockpit dell'Airbus A320, mostrata in Figura 3.11, include tutti



**Figura 3.11:** Riproduzione cockpit Airbus A320

gli elementi essenziali per il volo. I componenti principali che si possono trovare in questo simulatore sono:

- **PFD: Primary Flight Display** Schermo che mostra le informazioni di volo primarie come assetto, quota, velocità, rateo di salita e heading; è presente un display per il pilota e uno per il copilota.
- **ND: Navigation Display** Schermo che indica la rotta impostata tramite MCDU, oltre a waypoints ed eventuali aeroporti; è presente un display per il pilota e uno per il copilota.
- **FCU: Flight Control Unit** svolge la funzione di elemento di interfaccia tra il pilota e il Flight Management System: questo dispositivo consente al pilota di selezionare e impostare i vari modi di navigazione e di attivare l'autopilota, facilitando così la gestione del percorso di volo e delle performance dell'aeromobile. Attraverso la FCU, i piloti possono inserire velocità, altitudine, direzione e altri parametri di volo essenziali, ottenendo un controllo preciso e personalizzato sul comportamento del simulatore.
- **MCDU: Multi-function Control and Display Unit** Si tratta di un mini computer, con interfaccia tramite tastiera, che permette di impostare e modificare il piano di volo.

- **ECAM: Electronic Centralized Aircraft Monitor** Si compone di due schermi, uno superiore e uno inferiore, posti in centro al cockpit, in cui è possibile visualizzare diverse schermate per il monitoraggio dei sistemi di bordo del velivolo.
- **EFIS: Electronic Flight Instrument System** Si tratta di un pannello di strumentazione di volo elettronica, uno dalla parte del pilota e un altro da quella del copilota, che incorpora i selettori presenti sul pannello di controllo, nonché gli indicatori luminosi di *Master Warning* e *Caution Warning*.
- **Throttle** Manetta per il controllo della potenza dei motori.
- **Speed brake** Leva per il controllo degli aerofreni.
- **Flap lever** Leva per il controllo di *flaps* e *slats*.
- **Parking brake** Leva per l'inserimento del freno di stazionamento.
- **Rudder pedals** Sono le pedaliera e sono presenti sia per il capitano che per il primo ufficiale e permettono il controllo di timone e freni dei carrelli di atterraggio.
- **Sidesticks** Permettono il controllo di alettoni ed equilibratore e sono presenti sia per il capitano che per il primo ufficiale.

Tuttavia, in questo simulatore, mancando l'*Overhead Panel* e gli switch per l'accensione dei motori, la simulazione deve essere avviata a motori accesi e con l'aereo pronto per la procedura di taxi.

### Architettura

Il simulatore utilizza un computer principale che gestisce il comparto video attraverso due proiettori per l'*out of window* della cabina dell'A320 e tre monitor nascosti all'interno della struttura della cabina, di cui due renderizzano il *Primary Flight Display*, PFD, e il *Navigation Display*, ND, uno per il pilota e l'altro per il copilota, mentre il terzo mostra l'ECAM superiore ed inferiore. Questo PC è dotato di hardware avanzato per gestire tutta l'esigenza grafica: infatti monta un processore *Intel i7* e una scheda video *NVIDIA GTX 1080 Ti*. Ad esso sono collegate diverse periferiche attraverso un hub USB di collegamento:

- il pannello di controllo dell'FCU, per gestire i comandi dell'autopilota;
- due pedaliera, per controllare il timone e i freni a terra;
- un sidestick, per la gestione dell'assetto del velivolo;
- la manetta, per il controllo della spinta dei propulsori;
- quattro microcontrollori, per il controllo degli switch fisici come leve e pulsanti;
- un minicomputer, *Intel NUC*, per la gestione dell'MCDU lato pilota.

## Capitolo 4

# Procedure operative statiche

Con il termine **SOP** si definiscono le *Standard Operating Procedure*, ovvero un insieme di procedure, stabilite dalla compagnia aerea, che aiutano i piloti nell'operare un aeromobile commerciale in modo sicuro e coerente [57]. Le SOP forniscono istruzioni su come eseguire un determinato compito e si possono suddividere in *standard*, quindi per tutte le operazioni di routine, e *abnormal*, seguite solo in situazioni di emergenza come in presenza di guasti. Inoltre, definiscono uno standard da seguire e ciò assicura che la procedura venga eseguita sempre allo stesso modo da persone diverse e sempre in un ordine specifico.

Gli obiettivi delle SOP sono, dunque, molteplici:


- Assicurare che l'aeromobile sia pilotato correttamente, in conformità con le linee guida del produttore;
- Utilizzare in modo appropriato le risorse e le funzionalità dell'aeromobile;
- Promuovere la sicurezza operativa e l'efficienza operativa;
- Promuovere l'adesione alla filosofia operativa del produttore e della compagnia aerea;
- Prevenire errori da parte dell'equipaggio.


Queste procedure ricoprono tutti gli aspetti del volo, partendo dalle fasi iniziali di accensione e collaudo dei sistemi, fino alla fase finale di parcheggio e spegnimento. Si trovano descritte in un manuale che rimane a disposizione dei piloti nella cabina di pilotaggio e prevedono compiti specifici sia per il pilota, *Pilot Flying PF*, che per il copilota, *Pilot Monitoring PM*. In questo lavoro di tesi sono state sviluppate due procedure operative di routine a terra attraverso il software *Pacelab WEAVR: Parking e After Landing*. Per loro implementazione è stato seguito il manuale di addestramento, creato a scopo simulativo, per la famiglia di aerei *Airbus A32NX* [58] e l'elaborato si aggiunge al lavoro svolto negli anni precedenti nelle tesi [3], [4], [5], [6], [7], [8], [9].

In questo elaborato viene descritta dettagliatamente solo la procedura di *After Landing*, in quanto quella di *Parking* è trattata in modo approfondito nella tesi della mia collega Annalisa Cantiani.


## 4.1 After Landing


La procedura di *After Landing* appartiene alla categoria delle *standard procedures*, in quanto viene ripetuta per ogni volo. Si tratta di una serie di azioni da eseguire in seguito all'atterraggio, ma prima della fase di *Parking*; in questa fase si predispone l'aereo per arrivare in sicurezza al parcheggio designato. In Figura 4.1 è rappresentata la checklist da seguire per completare questa procedura. Per poter integrare le procedure con quelle già

 <b>AFTER LANDING</b>	<b>A320</b>
(PF) GRND SPLRS.....DISARM	
(PF) EXTERIOR LIGHTS.....SET	
(PM) RADAR/PWS.....OFF	
(PM) ENG MODE SELECTOR.....NORM	
(PM) FLAPS.....RETRACT	
(PM) TCAS/ATC.....STBY	
(PM) APU.....START	



**Normal**





**Procedures**

**Figura 4.1:** After Landing checklist

esistenti nel lavoro [9], le indicazioni presenti nelle procedure, così come il comparto audio, sono in lingua inglese.

All'avvio della procedura, l'utente indossa il visore di realtà virtuale e si trova all'interno della cabina di pilotaggio dell'Airbus A320, seduto sul sedile del pilota; davanti a se può notare un tablet e, collegato ad esso tramite linea tratteggiata, uno schermo in cui, attraverso un testo, viene dato il benvenuto e le istruzioni su come proseguire, come mostrato in Figura 4.2. Lo schermo con il testo scritto è l'oggetto di *WEAVR* denominato **Billboard** e, attraverso la funzione **Text To Speech TTS**, può essere accompagnato da un messaggio vocale, sincronizzato con questo elemento, che legge il contenuto del testo.



Figura 4.2: Schermata di benvenuto

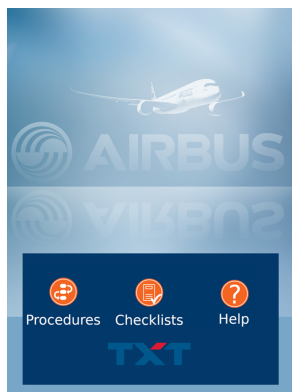
L'utente viene, quindi, invitato a prendere il tablet per portarselo più vicino e ad interagire con esso: è un oggetto di tipo **grabbable** e quindi può essere preso e spostato, avvicinando la mano, comandata dal movimento del controller nello spazio reale, fino al punto in cui l'elemento viene evidenziato in verde e premendo e mantenendo il grilletto del joystick per poi rilasciarlo nel punto desiderato.

### Tablet

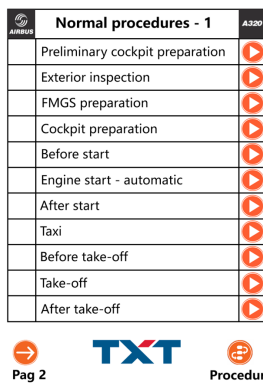
Nella schermata principale, visibile in Figura 4.3(a), si notano tre possibili opzioni, configurate come tasti **Executable** in *WEAVR*:

- **Procedures:** contiene l'elenco delle procedure eseguibili in VR, visibili in Figura 4.3(b) e (c). Ad ogni procedura è associato un tasto *Play*, anch'esso configurato come **Executable**, che, se premuto, permette di iniziare la relativa procedura, seguendo l'*Exit Condition* che porta al ramo specifico nella *Procedure Editor*. Sono presenti due pagine di procedure standard, anche se non tutte sono state ancora implementate e per navigare tra le pagine è necessario premere i pulsanti **Pag 1** e **Pag 2**, mentre, tramite il tasto **Procedures** si torna alla schermata principale;
- **Checklist:** permette di visualizzare le pagine di checklist generiche dell'A320, la prima è mostrata in Figura 4.3(d). In ognuna di esse, nella parte inferiore, sono presenti i tasti per cambiare pagina **Prev** o **Next** e per tornare al menu principale **Home**;
- **Help:** mostra un tutorial su come navigare nel tablet, su come iniziare la procedura voluta e permette la visualizzazione delle relative checklist.

Vengono adesso descritti singolarmente gli step della procedura che corrispondono ai vari nodi implementati in *WEAVR*, andando a evidenziare *Enter Actions*, *Exit Conditions* e *Exit Action* per ogni nodo.



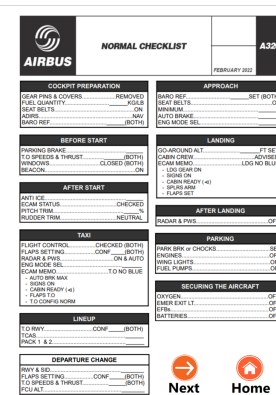
(a) Home page



(b) Normal procedures pag.1



(c) Normal procedures pag.2



(d) Checklist

Figura 4.3: Tablet schermate

### Step 0: Setup scena

Avviata la procedura *After Landing*, l'utente vede la schermata di benvenuto di inizio procedura, mostrata in Figura 4.4. Da questo momento non si può più interagire, ma solo spostare il tablet, che mostra la schermata con la checklist dettagliata sugli step da compiere che, una volta eseguiti, vengono contrassegnati da un *tick* verde. Viene mostrata una **Billboard** di benvenuto, analoga a quella mostrata all'avvio, sempre sincronizzata ad un messaggio vocale che ne riproduce il testo. In *WEAVR* vengono impostati tutti gli interruttori e i pulsanti rilevanti ai fini della procedura nella posizione coerente con la scena, ovvero come si troverebbero al termine dell'atterraggio:

- *Speed brakes* in posizione FULL;
- *Runway lights*, *Land lights*, *Strobe light*, *Nose light* su ON;
- *Radar PWS* su AUTO;
- *Flaps* in posizione FULL;



Figura 4.4: After Landing benvenuto

- *Engine switch mode* in posizione NORM;
- *TCAS/ATM* su TA/RA;
- *APU Master button* e *APU Start button* su OFF.

### Step 1: Ground Spoiler RETRACT

Preparata la scena, si procede con l'avvio vero e proprio della procedura. Il primo passo consiste nel retrarre gli aerofreni che sono stati utilizzati per rallentare l'aereo durante l'atterraggio.

- **Enter Action:** Si predispone una *billboard* in cui viene chiesto di retrarre gli spoiler portando la relativa leva, collegata allo schermo tramite linea tratteggiata ed evidenziata con un contorno verde, in posizione RETRACT. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;
- **Exit condition:** La leva degli aerofreni è configurata in *WEAVR* come un *Linear N-Way Switch*, ovvero uno switch con multiple posizioni possibili, tre in questo caso, *FULL*, *1/2* e *RETRACT*. La *exit condition* si verifica quando l'utente porta la leva in posizione RETRACT;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste due azioni: la rimozione della *billboard* in precedenza descritta e l'aggiunta del *tick* sul tablet in corrispondenza del relativo step eseguito.

### Step 2: Runway lights OFF

Successivamente si procede con il settaggio delle luci esterne del velivolo.

- **Enter Action:** Si predispone una *billboard* in cui viene chiesto di portare su OFF l'interruttore delle *Runway lights* che è collegato allo schermo tramite linea tratteggiata ed evidenziato con un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;





**Figura 4.5:** Step 1: Ground Spoiler RETRACT

- **Exit condition:** L'interruttore è configurato in *WEAVR* come un *Rotary Two-Way Switch*, ovvero uno switch con due posizioni possibili, *ON* e *OFF*. La *exit condition* si verifica quando l'utente posiziona lo switch su *OFF*;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, la *billboard* in precedenza descritta viene rimossa e si prosegue al nodo successivo.

### Step 3: Land lights OFF

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di portare in posizione *RETRACT* i due interruttori delle *Land lights*, destra e sinistra, che sono collegati allo schermo tramite linea tratteggiata ed evidenziati con un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione *TTS*;
- **Exit condition:** Entrambi gli interruttori sono configurati in *WEAVR* come *Rotary N-Way Switch*, ovvero uno switch con multiple posizioni possibili, tre in questo caso, *ON*, *OFF* e *RETRACT*. La *exit condition* si verifica quando l'utente posiziona entrambi gli switch in posizione *RETRACT*;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, la *billboard* in precedenza descritta viene rimossa e si prosegue al nodo successivo.

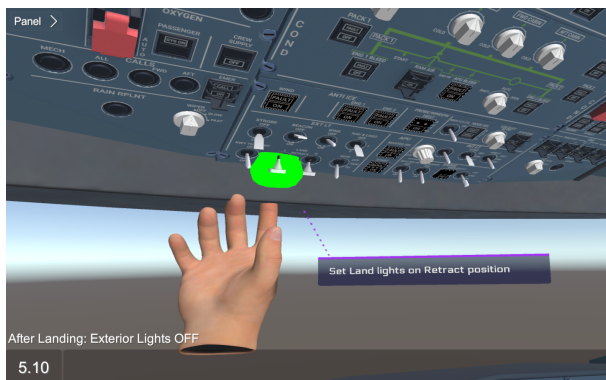
### Step 4: Strobe lights AUTO

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di portare su *AUTO* l'interruttore delle *Strobe lights* che è collegato allo schermo tramite linea tratteggiata ed evidenziato con un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione *TTS*;

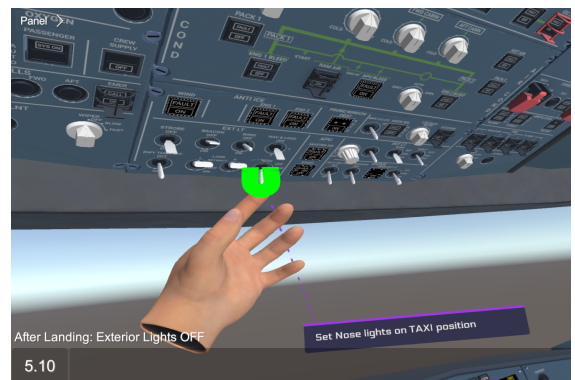
- **Exit condition:** L'interruttore è configurato in *WEAVR* come un *Rotary N-Way Switch*, ovvero uno switch con multiple posizioni possibili, tre in questo caso, *ON*, *AUTO* e *OFF*. La *exit condition* si verifica quando l'utente posiziona lo switch su *AUTO*;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, la *billboard* in precedenza descritta viene rimossa e si prosegue al nodo successivo.

### Step 5: Nose lights TAXI

- **Enter Action:** Si predispone una *billboard* in cui viene chiesto di portare su *TAXI* l'interruttore delle *Nose lights* che è collegato allo schermo tramite linea tratteggiata ed evidenziato con un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;
- **Exit condition:** L'interruttore è configurato in *WEAVR* come un *Rotary N-Way Switch*, ovvero uno switch con multiple posizioni possibili, tre in questo caso, *TAKE OFF*, *TAXI* e *OFF*. La *exit condition* si verifica quando l'utente posiziona lo switch su *TAXI*;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste due azioni: la rimozione della *billboard* in precedenza descritta e l'aggiunta del *tick* sul tablet in corrispondenza dello step relativo alle *Exterior lights*.



(a) Land lights



(b) Nose lights

Figura 4.6: Step 2-3-4-5: Configurazione *Exterior lights*

### Step 6: PM Procedures

Conclusa la configurazione delle *Exterior lights*, la procedura richiede che sia il copilota a continuare ad eseguire gli step, mentre il pilota rimane concentrato nel condurre l'aereo alla piazzola di parcheggio designata.

- **Enter Action:** Si predispone una *billboard* in cui viene chiesto di usare la funzione *teleport* per trasferirsi dal sedile del pilota a quello del copilota. Per attivarla è necessario

premere e mantenere premuto il trackpad del controller, per poi rilasciarlo una volta puntata correttamente l'area in cui si vuole teletrasportare. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;

- **Exit condition:** Come *exit condition* si predispone un controllo sulla distanza tra la camera VR, ovvero il punto in cui si trovano gli occhi virtuali dell'utente e il punto centrale del sedile interessato. Per cui, se questa distanza risulta inferiore a quella impostata, ovvero 0.7 m, la condizione risulta verificata;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, la *billboard* in precedenza descritta viene rimossa e si prosegue al nodo successivo.



Figura 4.7: Step 6: PM Procedures

### Step 7: Radar/PWS OFF

- **Enter Action:** Si predispone una *billboard* in cui viene chiesto di portare su OFF l'interruttore del *Radar/PWS*, ovvero spegnere la funzione *Predictive wind shear*, che è collegato allo schermo tramite linea tratteggiata ed evidenziato con un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;
- **Exit condition:** L'interruttore è configurato in *WEAVR* come un *Rotary Two-Way Switch*, ovvero uno switch con due posizioni possibili, *OFF* e *AUTO*. La *exit condition* si verifica quando l'utente posiziona lo switch su *OFF*;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste due azioni: la rimozione della *billboard* in precedenza descritta e l'aggiunta del *tick* sul tablet in corrispondenza del relativo step eseguito.



Figura 4.8: Step 7: Radar/PWS OFF

#### Step 8: ENG mode selector NORM

- **Enter Action:** Si predispono una *billboard* in cui viene chiesto di verificare tramite ispezione visiva che lo switch di *Engine mode selector*, che è collegato allo schermo tramite linea tratteggiata ed evidenziato con un contorno verde, sia sulla posizione NORM. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;
- **Exit condition:** L'ispezione visiva consiste nel guardare, per un determinato tempo e ad una certa distanza, l'*Object* configurato su *Unity*. La *exit condition* si verifica quando l'utente si avvicina allo switch lo osserva fino al completamento del timer impostato;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste due azioni: la rimozione della *billboard* in precedenza descritta e l'aggiunta del *tick* sul tablet in corrispondenza del relativo step eseguito.

#### Step 9: Flap ZERO

- **Enter Action:** Si predispono una *billboard* in cui viene chiesto di retrarre i flap portando la relativa leva, collegata allo schermo tramite linea tratteggiata ed evidenziata con un contorno verde, in posizione ZERO. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;
- **Exit condition:** La leva degli ipersostentatori è configurata in *WEAVR* come un *Linear N-Way Switch*, ovvero uno switch con multiple posizioni possibili, cinque in questo caso, *FULL*, *3*, *2*, *1* e *ZERO*. La *exit condition* si verifica quando l'utente porta la leva in posizione ZERO;



Figura 4.9: Step 8: ENG mode selector NORM

- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste due azioni: la rimozione della *billboard* in precedenza descritta e l'aggiunta del *tick* sul tablet in corrispondenza del relativo step eseguito.



Figura 4.10: Step 9: Flap ZERO

### Step 10: TCAS/ATC STBY

- **Enter Action:** Si predispone una *billboard* in cui viene chiesto di ruotare lo switch in posizione STBY che è collegato allo schermo tramite linea tratteggiata ed evidenziato da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;
- **Exit condition:** L'interruttore è configurato in *WEAVR* come *Rotary N-Way Switch*, ovvero uno switch con multiple posizioni possibili, cinque in questo caso, *TA/RA*, *TA ONLY*, *XPNDR*, *ALT RPTG OFF* e *STBY*. La *exit condition* si verifica quando l'utente posiziona lo switch in posizione *STBY*;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste due azioni: la rimozione della *billboard* in precedenza descritta e l'aggiunta del *tick* sul tablet in corrispondenza del relativo step eseguito.



Figura 4.11: Step 10: TCAS/ATC STBY

### Step 11: APU Master switch ON

- **Enter Action:** Si predispone una *billboard* in cui viene chiesto di premere il tasto, per attivare l'alimentazione dell'APU, che è collegato allo schermo tramite linea tratteggiata ed evidenziato da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;
- **Exit condition:** Il pulsante è configurato in *WEAVR* come *Push button*, ovvero un tasto che ha due configurazioni che si alternano ogni volta in cui viene premuto: ON e OFF; per rendere riconoscibile lo stato in cui si trova il pulsante sono state create

appositamente due maschere, mostrate in Figura 4.12, che si attivano e disattivano in base alla condizione del tasto. La *exit condition* si verifica quando l'utente preme il pulsante e lo porta in posizione ON;

- **Exit Action:** Appena la *exit condition* viene soddisfatta, la *billboard* in precedenza descritta viene rimossa e si prosegue al nodo successivo.

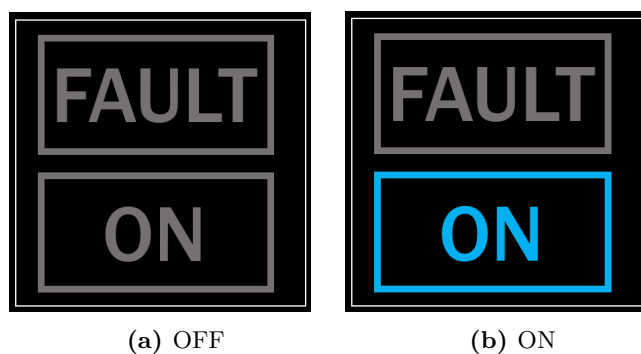


Figura 4.12: Maschere APU Master button



Figura 4.13: Step 11: APU Master switch ON

#### Step 12: APU Start ON

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di premere il tasto, che è collegato allo schermo tramite linea tratteggiata ed evidenziato da un contorno

verde, per avviare l'APU. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;

- **Exit condition:** Il pulsante è configurato in *WEAVR* come *Push button*, ovvero un tasto che ha due configurazioni che si alternano ogni volta in cui viene premuto: *AVAIL* e *ON*; per rendere riconoscibile lo stato in cui si trova il pulsante sono state create appositamente due maschere, mostrate in Figura 4.14, che si attivano e disattivano in base alla condizione del tasto. La *exit condition* si verifica quando l'utente preme il pulsante e lo porta in posizione *ON*;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, avviene la rimozione della *billboard* in precedenza descritta e l'aggiunta del *tick* sul tablet in corrispondenza del relativo step eseguito. Inoltre, essendo lo step finale della procedura di *After landing*, viene riabilitata l'interfaccia del tablet.

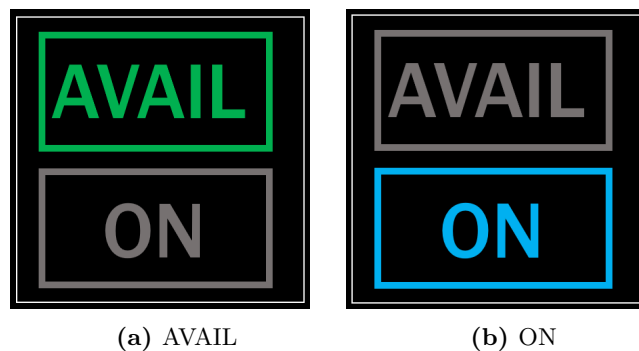


Figura 4.14: Maschere APU Start button



Figura 4.15: Step 12: APU Start ON



### Step 13: End step


Nell'ultimo nodo della procedura viene dichiarato, tramite *billboard* e TTS, che quest'ultima è stata completata con successo che adesso è possibile osservare sul tablet che tutti gli step sono stati completati; occorre premere il tasto *Procedures* e, in seguito, *Checklist*, in modo da visionare la checklist generica dove si vedono tutte le procedure dell'Airbus A320: quelle completate sono contrassegnate da una spunta verde come visto in precedenza per i singoli step. Terminato un timer di 20 secondi, impostato in *WEAVR* tramite il comando *Wait* come *exit action*, è possibile tornare alla schermata principale e iniziare una nuova procedura.



Figura 4.16: Step 13: End step

## 4.2 Parking

La procedura di *Parking* appartiene alla categoria delle *standard procedures* in quanto viene ripetuta per ogni volo. Si tratta di una serie di azioni da eseguire in seguito alla procedura di *After landing*, ma prima dell'ultima SOP, *Securing the Aircraft*, in cui si procede nel mettere in sicurezza l'aereo. In questa fase si predispongono l'aereo per il parcheggio attraverso gli step visibili in Figura 4.17 in cui è mostrata la checklist da seguire per completare questa procedura.

 AIRBUS	<b>PARKING</b>	<b>A320</b>
(PF) ACC PRESSURE.....	CHECK	
(PF) PARKING BRK.....	ON	
(PF) ENG MASTER SWITCH.....	OFF	
(PF) EXTERIOR LIGHT.....	OFF	
(PF) SLIDES.....	CHECK DISARMED	
(PF) BELT SIGN.....	OFF	
(PM) ANTI ICE.....	OFF	
(PM) APU BLEED.....	ON	
(PM) FUEL PUMPS.....	OFF	



**Normal**



**Procedures**

Figura 4.17: Parking checklist

## Capitolo 5

# Procedure operative dinamiche

In questo capitolo vengono trattate le procedure dinamiche; con questo termine si fa riferimento alle SOP che il pilota esegue durante il volo, interfacciandosi, quindi, con i dati di volo. Per questo motivo si rende necessaria la presenza di un simulatore di volo, *X-Plane*, che fornisca tali informazioni in tempo reale e con un ambiente grafico che riproduca fedelmente l'*out of window*, ovvero la vista esterna, della cabina di pilotaggio.

Il flusso dati proviene, dunque, da *X-Plane* e l'utente sperimenta le procedure nella cabina di pilotaggio virtuale attraverso *WEAVR* e *Unity*, sfruttando queste informazioni. La connessione e lo scambio di dati tra le due piattaforme sono resi possibili attraverso il modulo *SimulationHub server* e i suoi due *client*: il *client Unity* e il *client X-Plane*.

Dopo aver dettagliato l'architettura di questo progetto, si procede con la descrizione delle due procedure dinamiche create: *Take Off & Climb* e *Approach & Landing*. In questo lavoro viene spiegata passo per passo la procedura di *Approach & Landing*, mentre quella di *Take Off & Climb* viene approfondita nell'elaborato della mia collega Annalisa Cantiani e quindi sarà descritta brevemente.

### 5.1 Architettura del progetto

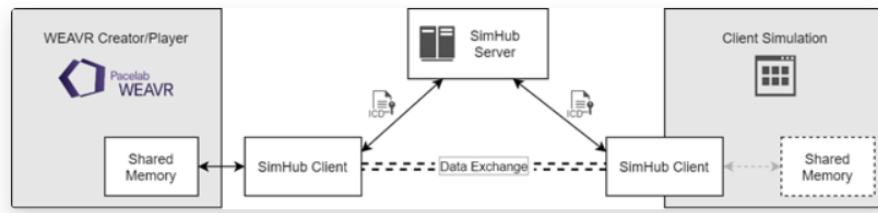
All'interno della cartella contenente il progetto si trovano due cartelle:

- *Procedure dinamiche A320*: si tratta del progetto in *Unity* in cui ci sono i vari *Assets* e la directory principale di *WEAVR*, oltre alle procedure sviluppate;
- *SimulationHub*: in questa cartella ci sono tutti gli elementi relativi al modulo *SimulationHub*, nonché il codice del tool *XPlane Connect*.

#### 5.1.1 SimulationHub

Il modulo *SimulationHub*, in questo progetto, viene utilizzato per permettere la comunicazione tra *X-Plane* e *WEAVR* e la sua architettura è mostrata in Figura 5.1. All'interno della sua directory principale si possono trovare le seguenti cartelle:

- **Client Unity**: include due componenti principali: l'eseguibile del client e un file batch *.bat* associato. Quest'ultimo è progettato per specificare due parametri prima dell'avvio



**Figura 5.1:** Architettura SimHub

dell'applicazione: l'indirizzo del file *ICD*, *Interchangeable Component Documentation*, correlato, chiamato *UNITY\_ICD*, e il nome dello spazio di memoria condivisa, *SimHubShm*, utilizzato. Questo *client* permette la comunicazione tra *WEAVR* e il *Server*, tramite l'utilizzo della *Shared Memory* e viene direttamente fornito pronto all'uso da TXT;

- **Client XPlane:** contiene la *Visual Studio Solution* del *Client*, ovvero il suo codice sorgente;
- **ConfigFiles:** questa cartella contiene i due file *ICD* dei due *client*: *UNITY\_ICD* e *XPLANE\_ICD*, entrambi in formato *.XML*. Si tratta di file contenenti l'elenco delle variabili a cui sono assegnate delle etichette e il loro tipo. Al loro interno si riporta l'indirizzo IP del PC, *127.0.0.1*, il numero della porta del *client*: 4400 per il *Client XPlane* e 4900 per il *Client Unity* e il numero della porta del server di *SimHub*: 6500. Le variabili possono essere:
  - *Published*: variabili che il *client* scrive, quindi pubblicate;
  - *Subscribed*: variabili che il *client* legge, quindi sottoscritte.

Per quanto riguarda il file *XPLANE\_ICD*, le variabili pubblicate sono quelle relative ai dati di simulazione come velocità, quota o rateo di salita, provenienti da *X-Plane*, mentre quelle sottoscritte sono quelle da inviare al simulatore, come la percentuale di manetta, posizione dei flap o del carrello di atterraggio. Invece, per *UNITY\_ICD* accade l'esatto opposto: scrive le variabili da inviare al simulatore e legge quelle provenienti da esso, quindi i dati della simulazione. In Figura 5.2 sono riportati in dettaglio i file *ICD*;

- **SimHub lib:** contiene le librerie di *SimHub*, raggruppate nelle seguenti cartelle:
  - *dll*: contiene i file *.dll*, ossia le librerie dinamiche;
  - *include*: raggruppa le intestazioni delle funzioni delle librerie: i file *.h*. Questi vanno inclusi nel codice mediante il comando di inclusione;
  - *lib*: include i file *.lib*, ossia le librerie al cui interno si trova il codice delle funzioni di *SimHub*;
- **SimulationHub Server:** contiene l'eseguibile del *Server*, file *.exe*, e il relativo file *.bat* in cui sono dichiarati l'indirizzo IP della macchina, *127.0.0.1*, e la porta di comunicazione, *6500*;

```

<?xml version="1.0" encoding="UTF-8"?>
<NodeConfiguration name="UnityClient" address="127.0.0.1" port="4900">
  <ServerConfig address="127.0.0.1" port="6500"/>
  <SubscribedLabels>
    <label name="ProcedureChoice" type="_int32" description="Procedure choice"/>
    <label name="Altitude" type="float" description="Altitude [ft]"/>
    <label name="Airspeed" type="float" description="Airspeed [kts]"/>
    <label name="GearForce" type="float" description="Gear force [N]"/>
    <label name="V1" type="float" description="Decision Speed [kts]"/>
    <label name="VerticalSpeed" type="float" description="Vertical Speed [fpm]"/>
    <label name="APMode" type="float" description="Autopilot Lateral Mode"/>
    <label name="APPhase" type="float" description="Autopilot Phase"/>
    <label name="N1" type="float" description="Engine rpm"/>
    <label name="Altitude_foot" type="float" description="Altitude[ft]"/>
    <label name="WPT_CRS" type="float" description="Waypoint angle"/>
    <label name="VR" type="float" description="Rotate Speed"/>
  </SubscribedLabels>
  <PublishedLabels>
    <label name="ParkingBrake" type="float" description="parking brake state"/>
    <label name="Flaps" type="float" description="flaps position"/>
    <label name="LandingGear" type="float" description="landing gear state"/>
    <label name="Spoiler" type="float" description="ground spoiler value"/>
    <label name="Thrust" type="float" description="thrust value"/>
    <label name="LatStick" type="float" description="lateral sidestick value"/>
    <label name="AutoPilot" type="float" description="autopilot state"/>
    <label name="AutoBrk" type="float" description="auto brk_low state"/>
    <label name="Nuovo1" type="float" description="auto_brk_low state"/>
    <label name="Nuovo2" type="float" description="auto_brk_low state"/>
    <label name="Nuovo3" type="float" description="auto_brk_low state"/>
  </PublishedLabels>
</NodeConfiguration>

```

## (a) UNITY ICD

```

<?xml version="1.0" encoding="UTF-8"?>
<NodeConfiguration name="XPlaneClient" address="127.0.0.1" port="4400">
  <ServerConfig address="127.0.0.1" port="6500"/>
  <SubscribedLabels>
    <label name="ParkingBrake" type="float" description="parking brake state"/>
    <label name="Flaps" type="float" description="flaps position"/>
    <label name="LandingGear" type="float" description="landing gear state"/>
    <label name="Spoiler" type="float" description="ground spoiler value"/>
    <label name="Thrust" type="float" description="thrust value"/>
    <label name="LatStick" type="float" description="lateral sidestick value"/>
    <label name="AutoPilot" type="float" description="autopilot state"/>
    <label name="AutoBrk" type="float" description="auto brk_low state"/>
    <label name="Nuovo1" type="float" description="auto_brk_low state"/>
    <label name="Nuovo2" type="float" description="auto_brk_low state"/>
    <label name="Nuovo3" type="float" description="auto_brk_low state"/>
  </SubscribedLabels>
  <PublishedLabels>
    <label name="ProcedureChoice" type="_int32" description="Procedure choice"/>
    <label name="Altitude" type="float" description="Altitude [ft]"/>
    <label name="Airspeed" type="float" description="Airspeed [kts]"/>
    <label name="GearForce" type="float" description="Gear force [N]"/>
    <label name="V1" type="float" description="Decision Speed [kts]"/>
    <label name="VerticalSpeed" type="float" description="Vertical Speed [fpm]"/>
    <label name="APMode" type="float" description="Autopilot Lateral Mode"/>
    <label name="APPhase" type="float" description="Autopilot Phase"/>
    <label name="N1" type="float" description="Engine rpm"/>
    <label name="Altitude_foot" type="float" description="Altitude[ft]"/>
    <label name="WPT_CRS" type="float" description="Waypoint angle"/>
    <label name="VR" type="float" description="Rotate Speed"/>
  </PublishedLabels>
</NodeConfiguration>

```

## (b) XPLANE ICD

Figura 5.2: Files ICD

- **XPC lib:** al suo interno si trovano le librerie di funzioni di *XPlane Connect* e il relativo codice sorgente.

### 5.1.2 Client X-Plane

Questo *client* gestisce la comunicazione tra *X-Plane* e il *Server*, tramite l'utilizzo della *Shared Memory* ed è stato sviluppato in *C++* appositamente per questo progetto, basandosi sul lavoro già svolto in parte nelle tesi [7] e [8].

#### Architettura del Client

Il codice è strutturato nel seguente modo:

- Attraverso apposito comando, si includono le librerie di sistema, di *SimHub* e di *XPlane Connect*;
- Vengono dichiarate le funzioni usate nel client:
  - **pause**: permette di mettere in pausa la simulazione su *X-Plane*; in questo modo, raggiunta una determinata condizione nel volo e diventando, dunque, obbligatorio eseguire un'azione, il pilota ha il tempo di capire ed eseguire questo compito, prendendosi tutto il tempo necessario. Una volta soddisfatto il compito richiesto, la simulazione riparte e il comando viene inviato al simulatore per rendere effettivo il cambiamento. Attraverso questa funzione è, quindi, possibile eseguire una procedura guidata senza il tempo sia un fattore rilevante per la simulazione;
  - **takeoff**: si tratta della funzione per la procedura di *Take Off & Climb*;
  - **approach**: si tratta della funzione per la procedura di *Approach & Landing*.

Le due funzioni riguardanti le procedure dinamiche sono organizzate in modo coerente con l'ordine degli step presente su *WEAVR*: viene indicizzato il numero degli step all'interno di un ciclo *while*, contenente uno *switch* con *case* pari al numero degli step. In ogni *case* si controlla lo stato di una o più variabili in uscita da *WEAVR* e, quando l'utente ha compiuto l'azione corretta in VR, si entra in un *if* e si invia il comando eseguito ad *X-Plane*.

- Attraverso un puntatore si crea un'interfaccia con *SimulationHub*;
- Si avvia la connessione ad *XPC*, definendo l'indirizzo IP della macchina, *127.0.0.1*, mediante la funzione, definita in *xplaneConnect.h*, *openUDP*;
- Si definisce la funzione *main*, ovvero la funzione principale del codice:
  - Si definiscono le variabili: *int*, per il numero della procedura scelta, numero e dimensioni dei *datarefs*, *float*, per le variabili in uscita da *WEAVR* e i valori dei *datarefs*, e *char*, per le stringhe dei *datarefs*.
  - Si avvia la connessione con *SimHub Server*, utilizzando le funzioni dichiarate in *SimulationHubInterface.h*:
    - \* **Initialize**: apre il collegamento assegnando le apposite etichette alle variabili presenti nei file ICD;
    - \* **GetLabel**: assegna alle variabili dei file ICD una variabile presente nel codice;
    - \* **Refresh**: avvia lettura e scrittura delle variabili precedentemente citate.

- Infine è presente uno *switch* con due *case* per scegliere la funzione relativa alla procedura che si vuole eseguire. La scelta si effettua passando come argomento al *main* il numero voluto: 1 o 2.

## X-Plane Connect

Questo tool fornisce un'interfaccia per facilitare la connessione tra il *client XPlane* e *X-Plane* stesso; è costituito dal codice principale, *xplaneConnect.c*, e da una libreria in cui sono dichiarate le funzioni utilizzabili nel client, *xplaneConnect.h*. Nel codice sono state utilizzate le seguenti funzioni:

- **openUDP**: permette di aprire una connessione con XPC, inserendo l'indirizzo IP del PC in cui si esegue *X-Plane*;
- **getDREF**: prende il valore di uno specifico *dataref* di *X-Plane*, come ad esempio velocità o quota;
- **sendDREF**: scrive un determinato valore su uno specifico *dataref* di *X-Plane*, come ad esempio la percentuale di manetta. Il *dataref* in questione deve essere di tipo *writable*;
- **sendCOMM**: manda un comando come stringa ad *X-Plane*. La lista dei comandi, così come quella dei *datarefs*, disponibili è visibile all'interno del simulatore.

### 5.1.3 SimulationHub script

Per consentire una comunicazione efficace tra *WEAVR*, *SimHub* e *X-Plane*, si è reso necessario inserire alcuni script nella directory *Assets* del progetto in *Unity*, elaborati in linguaggio *C#*.

#### SimHubReader

Permette la lettura e il salvataggio dei dati di volo importati da *X-Plane*, nelle relative variabili ed è organizzato nel seguente modo:

- *Dichiarazione variabili*: sono tutte di classe *public*, costituite da stringhe inserite nell'oggetto *Text* e vengono visualizzate in *Unity* con lo scopo di informare il pilota sui parametri di volo in tempo reale. Si tratta di altitudine, velocità, rateo di salita e numero di giri dei motori;
- *Funzione Update*: permette la lettura dei valori dal file ICD. Questi valori si presentano come *float* e, per essere visualizzati come testo in VR, devono prima essere convertiti in formato stringa tramite apposito comando.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using TXT.WEAVR.SimulationData;
4  using UnityEngine;
5  using UnityEngine.UI;
6
7  [Script Unity (1 riferimento ad asset) | 0 riferimenti]
8  public class SimHubReader : MonoBehaviour
9  {
10     public SimulationModel simFramework;
11     public Text Altitude_foot;
12     public Text Airspeed;
13     public Text M1;
14     public Text VerticalSpeed;
15
16     // Start is called before the first frame update
17     [Messaggio Unity | 0 riferimenti]
18     void Start()
19     {
20     }
21
22     // Update is called once per frame
23     [Messaggio Unity | 0 riferimenti]
24     void Update()
25     {
26         Airspeed.text = simFramework.SubscribedLabels.Airspeed.ToString();
27         Altitude_foot.text = simFramework.SubscribedLabels.Altitude_foot.ToString();
28         M1.text = simFramework.SubscribedLabels.M1.ToString();
29         VerticalSpeed.text = simFramework.SubscribedLabels.VerticalSpeed.ToString();
30     }
31 }

```

Figura 5.3: SimHubReader script

## SimulationModel

All'interno di questo script sono dichiarate le strutture *Published* e *Subscribed*, rispettivamente per scrittura e lettura, e viene indicata la memoria condivisa a cui accedere, *SimHubShm*. Inserendo questo script, tra gli oggetti della gerarchia in *Unity* si crea l'*Object*, *Simulation Framework*, nel cui *inspector* vanno inseriti gli script *SimulationModel* e *SimHubReader*; in quest'ultimo è necessario andare ad abbinare le caselle di testo alle variabili dichiarate, come mostrato in Figura 5.4.

## SimulationHubICD

Al suo interno sono dichiarate le variabili *public* divise in *Subscribed* e *published*, secondo quanto riportato nel file *UNITY ICD*; lo script è riportato in figura 5.5.

### 5.1.4 uDesktopDuplication plugin

Questo tool è un *plugin* di *Unity* ed è scaricabile gratuitamente da *GitHub*. Una volta correttamente importato su *Unity*, compare nella gerarchia la voce *Monitor Board*; si tratta di un pannello, attraverso cui viene proiettato il desktop del computer su cui è in esecuzione *Unity* e si attiva quando si entra in modalità *Game*. In questo progetto, questo strumento viene sfruttato per fornire un'immagine realistica e coerente con la simulazione all'utente, mentre esegue una procedura: è, dunque, possibile proiettare l'*out of window* della cabina di pilotaggio, in modo da ricreare un'esperienza il più immersiva possibile. Operativamente, una volta impostata la giusta visuale su *X-Plane*, è sufficiente attivare la modalità *Game* su *Unity* per poi ridurre a icona e lasciare la finestra di *X-Plane* a tutto schermo.

Per poter usufruire di un'esperienza migliore, è necessario adattare lo schermo, mostrato in figura 5.6, all'interno di *Unity*, modificando i parametri nella finestra *Inspector*. Per rendere

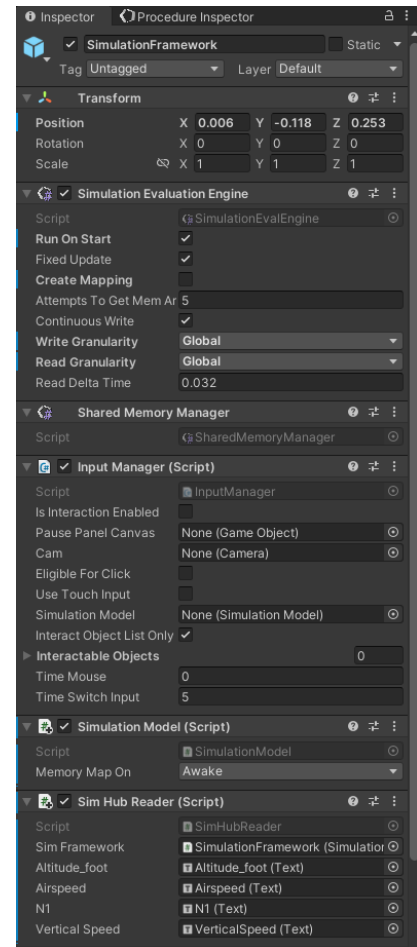


```

7
8 using UnityEngine;
9 using TXT.WEAVR.Core;
10 using TXT.WEAVR.Simulation;
11 using System.Collections.Generic;
12
13 namespace TXT.WEAVR.SimulationData
14 {
15
16     public class SimulationModel : MonoBehaviour
17     {
18         public enum MapOn { None, Awake, Start }
19         [DoNotExpose]
20         public MapOn memoryMapOn = MapOn.Awake;
21         [ShowPrimitivesOnly]
22         [SimulationVariable(SharedMemoryAccess.Read, @"SimHubShm")]
23         public SubscribedLabels SubscribedLabels { get; set; }
24         [ShowPrimitivesOnly]
25         [SimulationVariable(SharedMemoryAccess.Write, @"SimHubShm")]
26         public PublishedLabels PublishedLabels { get; set; }
27         void Awake()
28         {
29             if (memoryMapOn == MapOn.Awake)
30             {
31                 SimulationEvalEngine.Instance.MapToSharedMemory(this);
32             }
33         }
34
35         void Start()
36         {
37             if (memoryMapOn == MapOn.Start)
38             {
39                 SimulationEvalEngine.Instance.MapToSharedMemory(this);
40             }
41         }
42
43     }
44 }

```

(a) SimulationModel script



(b) Simulation Framework

Figura 5.4

l'esperienza più immersiva, è necessario rendere lo schermo curvo, in modo che avvolga lateralmente la cabina di pilotaggio, senza lasciare quindi parti esterne dell'ambiente *Unity* visibili agli occhi dell'utente; per ottenere questo risultato si modificano i seguenti parametri:

- Si scala lo schermo per adattare la risoluzione a quella del desktop:  $1920 \times 1080$ ; per farlo si imposta il fattore *Scale* nel seguente modo:  $x = 1.92$ ,  $y = 1.08$ ,  $z = 1$ ;
- *Bend width* = 1.92;
- *Bend Radius* = 5;
- *Thickness* = 1;
- infine, occorre traslare lo schermo manualmente nelle tre dimensioni in modo che sia centrato con il punto centrale della cabina.

```
1 using System.Runtime.InteropServices;
2
3 namespace TXT.WEAVR.SimulationData
4 {
5     [StructLayout(LayoutKind.Sequential, Pack = 1)]
6     public struct PublishedLabels
7     {
8         public float ParkingBrake;
9         public float Flaps;
10        public float LandingGear;
11        public float Spoiler;
12        public float Thrust;
13        public float LatStick;
14        public float AutoPilot;
15        public float AutoBrk;
16        public float Nuovo1;
17        public float Nuovo2;
18        public float Nuovo3;
19    }
20
21    [StructLayout(LayoutKind.Sequential, Pack = 1)]
22    public struct SubscribedLabels
23    {
24        public int ProcedureChoice;
25        public float Altitude;
26        public float Airspeed;
27        public float GearForce;
28        public float V1;
29        public float VerticalSpeed;
30        public float APMode;
31        public float APPhase;
32        public float N1;
33        public float Altitude_foot;
34        public float WPT_CRS;
35        public float VR;
36    }
37 }
```

Figura 5.5: SimulationHubled script

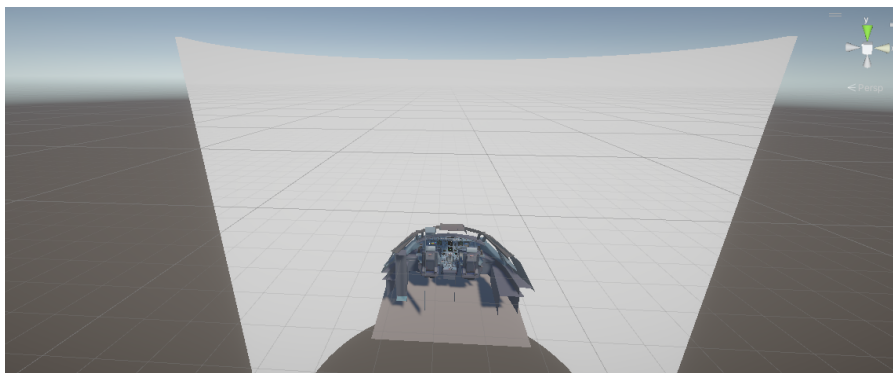


Figura 5.6: Monitor Board

## 5.2 Approach & Landing

In questo elaborato viene descritta in maniera approfondita la procedura di *Approach & Landing*; prima di analizzare la procedura vera e propria, occorre capire come è stato configurato lo scenario di volo.

### 5.2.1 Piano di volo e configurazione MCDU

Entrambe le procedure dinamiche sono svolte nello stesso piano di volo: partenza dall'aeroporto di Milano Malpensa LIMC e arrivo a Torino Caselle LIMF. Per pianificare il volo si ricorre ad uno strumento essenziale dell'Airbus A320: l'MCDU; questa costituisce l'interfaccia tra il pilota e il *Flight Management System* FMS e contiene un database con le performance del velivolo in modo da stimare i consumi, i tempi e le velocità ottimali della missione.

L'MCDU è mostrata in Figura 5.7 ed è formata da:



Figura 5.7: MCDU

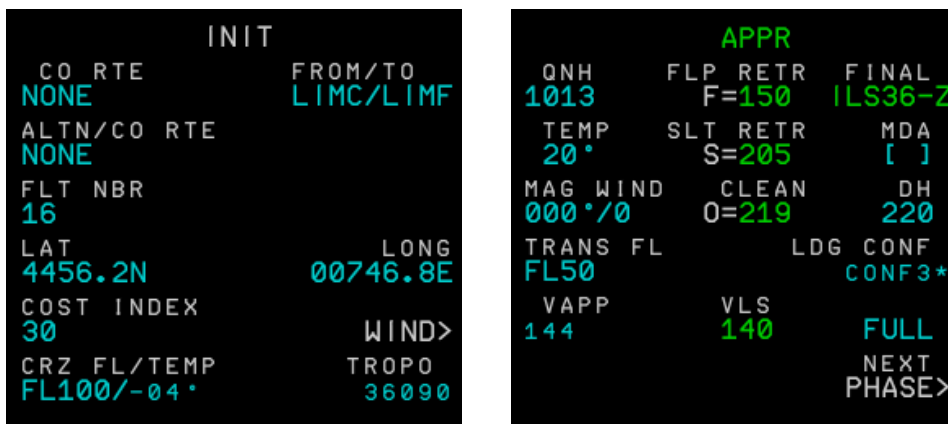
- **Display:** per mostrare i dati da inserire e visualizzare le varie schermate delle fasi di volo;

- **Tastiera alfanumerica:** per inserire i parametri richiesti dal computer di bordo e selezionare le pagine da visualizzare.

Per iniziare la configurazione del volo si parte dalla pagina **INIT**, in cui devono essere compilati diversi campi:

- **FROM/TO:** si inseriscono le sigle dell'aeroporto di partenza e di arrivo rispettivamente;
- **FLT NBR:** è il codice identificativo del volo;
- **COST INDEX:** è un parametro che indica quanto si vuole volare prestando attenzione al risparmio combustibile: si sceglie *30*, andando a selezionare una condizione in cui si vuole risparmiare sul costo del carburante;
- **CRZ FL:** indica la quota di crociera. Il valore *FL100* significa *Flight Level 100*, dove ogni unità corrisponde a *100 ft*, misurati dal livello del mare. Quindi, impostando questo parametro, si sceglie una quota di crociera di *10000 ft*.

Successivamente, selezionando la pagina **F-PLN**, si imposta la rotta da seguire, scegliendo i *waypoint* del percorso. Infine, si visualizza la pagina **PERF**, dove vengono visualizzati i parametri del volo e viene richiesto di inserire dei dati relativi alla fase di *TAKE OFF*, prima di partire, e a quella di *APPROACH*, durante il volo, mostrata in Figura 5.8. Nella



(a) INIT

(b) PERF-APPR

Figura 5.8: Schermate MCDU

pagina **PERF**, relativa all'*APPROACH*, vengono impostati i seguenti parametri:

- **QNH:** indica la pressione di riferimento per l'aeroporto di arrivo;
- **TEMP:** indica la temperatura dell'aria all'aeroporto di arrivo;
- **MAG WIND:** è il valore relativo all'intensità del vento e si presuppone che sia nulla per questa simulazione;

- **VAPP:** è la velocità con cui si effettua l’approccio finale alla pista di atterraggio, nonché la velocità di atterraggio;
- **DH:** la sigla sta per *Decision High* e indica l’altezza da terra minima, misurata in piedi dal livello del suolo, per l’avvicinamento alla pista.

## 5.2.2 Avvio della procedura

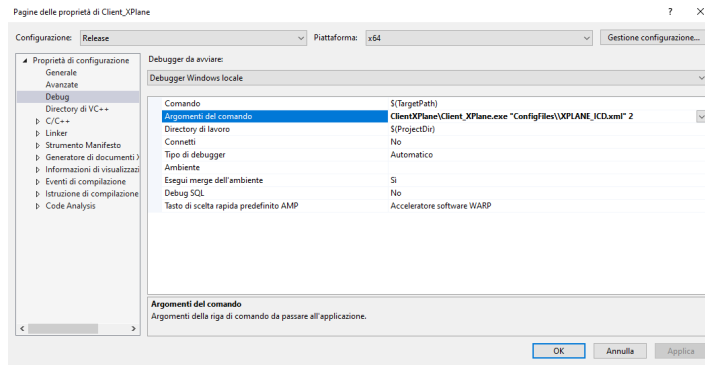
Per avviare correttamente la procedura, occorre seguire i seguenti step:

1. Aprire il progetto "*Procedure dinamiche A320*" in *Unity*;
2. Aprire *SimulationHub Server*, tramite apposito file *.bat*;
3. Aprire *Client Unity*, tramite apposito file *.bat*;
4. Aprire *Client XPlane*, attraverso la *Visual Studio Solution*;
5. Avviare *X-Plane* e impostare il volo scegliendo il velivolo *ToLiss A321 V1.3 Std Def.* Attendere il caricamento del volo, impostare la visuale come mostrato in Figura 5.9 (a) e , successivamente, caricare la scena per la procedura di *Approach & Landing*, vedi Figura 5.9 (b);

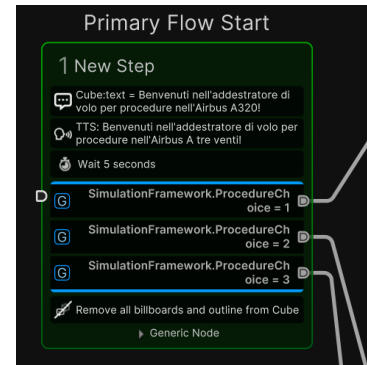


**Figura 5.9:** Configurazione scena X-Plane

6. Assicurarsi che l’utente abbia indossato correttamente il visore e che SteamVR sia aperto;
7. Avviare la modalità *game* in *Unity*;
8. Fare il *Run* del codice in visual studio dopo aver impostato dalla finestra di debug il numero della procedura, in questo caso 2, che viene passato come argomento al *main*, come in Figura 5.10 (a). In questo modo, attraverso la variabile *ProcedureChoice*, viene passato a *WEAVR* il segnale per discriminare la procedura da avviare nel primo nodo, Figura 5.10 (b);



(a) Finestra di debug



(b) Nodo con ProcedureChoice

Figura 5.10: Scelta procedura nel *Client XPlane*

9. Come ultimo passo, bisogna impostare la finestra di *X-Plane* a tutto schermo, Figura 5.11, in modo che il tool *uDesktopDuplication* restituisca all'utente in VR la corretta visuale.

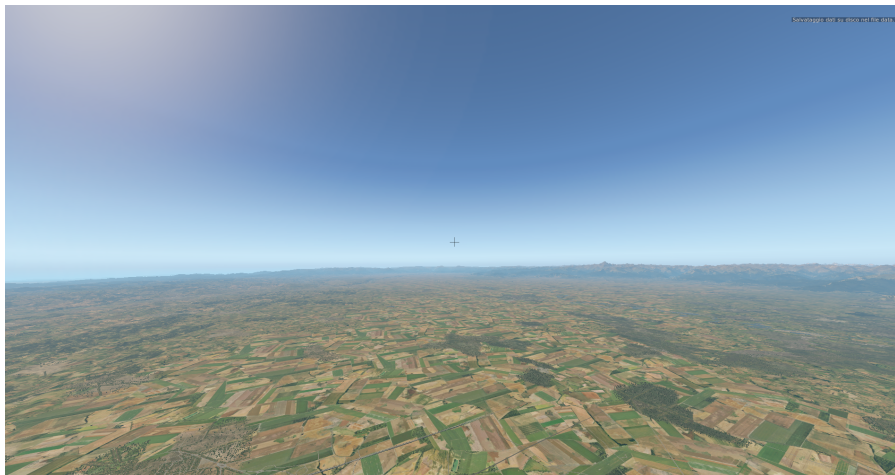


Figura 5.11: Visuale X-Plane Approach &amp; Landing

### 5.2.3 Descrizione della procedura

In questo progetto si è scelto di unire la procedura di *Approach* a quella di *Landing* in modo da renderla equivalente sia come difficoltà che come numero di step da eseguire a quella di *Take off*, in previsione dei test su un campione di persone.

L'intera fase di *Landing* viene condotta con l'aiuto strumentale *ILS*, *Instrumental Landing System*, in modo da ottenere sempre lo stesso percorso e velocità di discesa in ogni ripetizione della procedura ed eliminare, quindi, eventuali errori dell'utente; questo sistema fornisce un ausilio di precisione per l'avvicinamento alla pista, impiegando due fasci radio che

costituiscono una guida verticale e orizzontale alla strumentazione e al pilota, durante l'atterraggio. I due componenti sono:

- **Glidescope:** fornisce il corretto profilo di discesa verticale;
- **Localizer:** fornisce la corretta guida azimutale.

Viene adesso descritta la procedura passo per passo, come avvenuto per la procedura statica.

### Step 0: Setup scena

Dopo aver avviato la modalità *Game* di *Unity*, se passato correttamente alla funzione *main* del *client XPlane* il numero 2 come argomento, relativo alla procedura di *Approach & Landing*, *WEAVR* seleziona automaticamente questa procedura e passa al nodo per la preparazione della scena, in cui vengono abilitate alcune maschere per pulsanti e schermi e posizionate le leve nella posizione corretta per la simulazione. In particolare, le azioni principali sono le seguenti:

- Vengono abilitate le maschere relative al PFD e al ND, mostrate in Figura 5.12 (a) e (b);
- Viene abilitata la maschera per la pressione standard di riferimento;
- Viene abilitata la maschera dell'altitudine dell'autopilota con valore a 10000 ft, quota da cui inizia la procedura;
- Viene disabilitata la stringa di testo relativa alla velocità verticale, in quanto non viene utilizzata per alcun controllo in questa procedura, mentre, invece, sono abilitate le stringhe per la velocità, la quota e il numero di giri dei motori;
- Vengono impostati gli stati di: manetta in posizione *CLIMB*, carrello di atterraggio in posizione *RETRACT* e freno di stazionamento *OFF*.

In questo primo nodo di benvenuto, viene mostrata all'utente una *billboard* dove viene evidenziata la procedura scelta e dopo un timer di 10 secondi, impostato tramite comando *Wait*, si prosegue al nodo successivo, rimuovendo la *billboard*. All'avvio del *client XPlane*, si entra nella funzione *Approach* e la simulazione viene messa in pausa, mediante l'apposita funzione creata che manda un *command* ad *X-Plane*. In seguito, il codice passa all'interno del ciclo *while*, in cui avviene la lettura dei *datarefs*, riguardanti velocità, quota e numero di giri, per poi inviarli al *SimHub server* che, a sua volta, li manda a *WEAVR*, dove vengono convertiti per essere poi visualizzati come stringa di testo; questa lettura avviene in continuazione fino al termine della procedura.

### Step 1: Set altitude

Questo è il primo step vero e proprio della procedura e consiste nell'impostare il target di altitudine per l'autopilota.

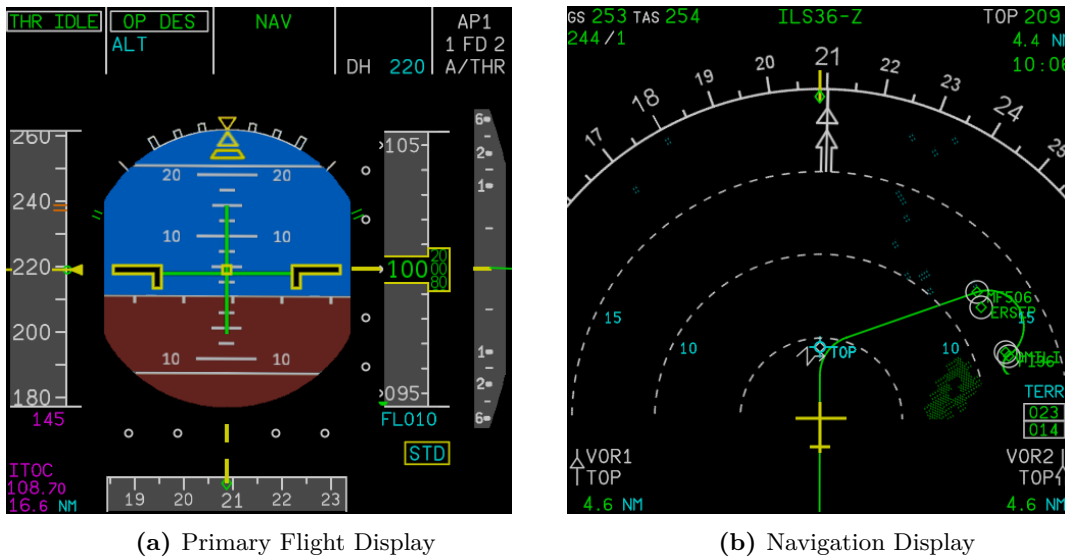


Figura 5.12

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di tirare il tasto relativo all'altitudine dell'autopilota che è collegato allo schermo tramite linea tratteggiata ed evidenziato da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS. Inoltre, per facilitare l'interazione con questo tasto, vengono disabilitati i componenti *Box Collider* di due pulsanti vicini: AP1 e AP2;
- **Exit condition:** Il tasto è configurato in *WEAVR* come *Executable*, ovvero un elemento con cui si interagisce tramite click del grilletto posteriore del controller. La *exit condition* si verifica quando l'utente aziona il suddetto elemento;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste diverse azioni: viene riabilitato il *Box Collider* di AP2, la maschera dello schermo relativo all'autopilota viene sostituita, come mostrato in Figura 5.13 da (a) a (b), si esegue l'animazione del tasto con cui si è interagito, si cambia lo stato del tasto da OFF a ON, mediante la funzione di *WEAVR* chiamata *Call Method - GoToState (string)* e avviene la rimozione della *billboard* in precedenza descritta.

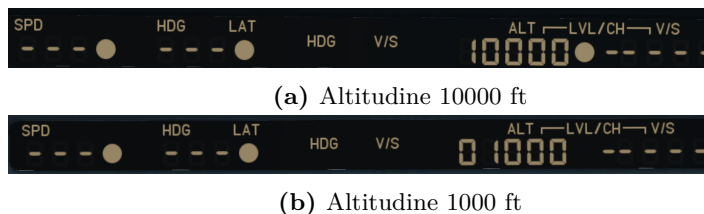


Figura 5.13: Maschere altitudine autopilota





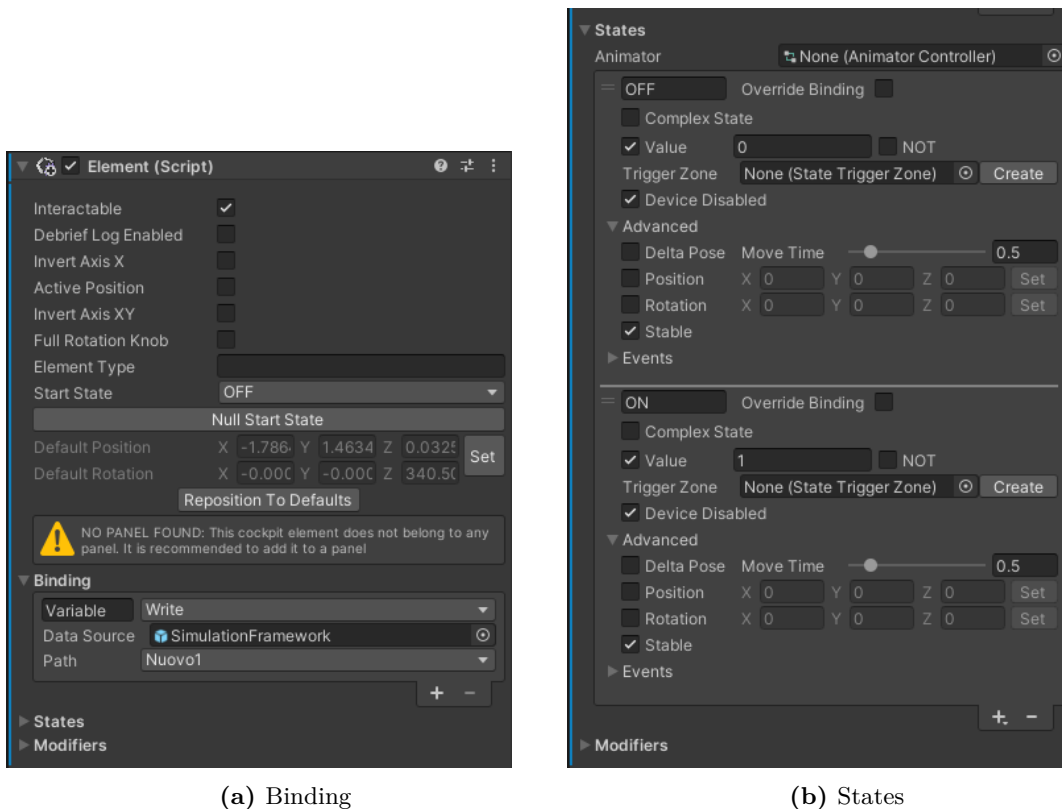
**Figura 5.14:** Step 1: Set altitude

La funzione *Animation* consiste nel far compiere un movimento, scelto a priori, all'elemento per cui viene richiamata; in questo caso il pulsante si muove come se fosse tirato verso l'esterno di pochi millimetri.

Per quanto riguarda la funzione *GoToState*, essa permette a *WEAVR* di comunicare con *SimHub*; ogni elemento a cui viene associata questa funzione deve essere configurato, impostando degli stati, almeno due, in cui esso può trovarsi. La configurazione deve essere effettuata nell'*Inspector*, andando ad aggiungere il componente *Element (script)*, mostrato in Figura 5.15, in cui è possibile indicare lo stato di partenza, impostare la variabile del file ICD a cui viene legato questo elemento e impostarne gli stati. Nella sezione *Binding*, si lega l'elemento alla variabile in uscita da *WEAVR*, in questo caso *Nuovo1*, e si dichiara se questa è utilizzata in scrittura, *write*, o in lettura, *read*.

Nella sezione *States*, invece, si creano gli stati dell'elemento, assegnando un nome che viene riconosciuto all'interno di *WEAVR* dalla funzione *GoToState*, in questo caso OFF e ON, e indicando per ogni stato un valore che esso deve fornire alla variabile in uscita che viene poi letta dal *client XPlane*.

Per quanto riguarda la parte di codice nel *client XPlane*, relativa a questo step, è mostrata in Figura 5.16. Si entra nel *case 0* dello *switch* e si effettua un controllo sulla variabile in uscita da *WEAVR*: se l'utente ha effettuato l'azione in VR, lo stato viene cambiato in ON, andando a scrivere il valore 1.0 nella variabile e quindi si entra nell'*if*. Viene tolta la pausa della simulazione e vengono inviata l'azione ad *X-Plane*, mediante due funzioni: *sendDREF*, per impostare il valore 1000 dell'altitudine e *sendCOMM*, per il *pull* del pulsante in cabina. Infine, viene incrementato l'indice *procedurestep* per passare al *case* successivo. Questa tipologia di codice si ripete per tutti gli altri step della procedura.



(a) Binding

(b) States

Figura 5.15: Configurazione Element script

```

switch (procedureStep)
{
case 0: //set altitude
{
if(weavrOut[8] == 1.0)
{
flagPause = pause(0);

datarefSentValues[0][0] = 1000.0F;
sendDREF(clientXPlane, datarefSent[0], datarefSentValues[0], datarefSentSizes[0]);
sleep(0.1);
sendCOMM(clientXPlane, command[5]);
sleep(0.1);
procedureStep++;
cout << "set altitude\n";
}
};
break;
}

```

Figura 5.16: Invio comandi a X-Plane 11

## Step 2: Approach

- **Enter Action:** Si predispone una *billboard* in cui viene chiesto di premere il pulsante per attivare la modalità *APPROACH* dell'autopilota, che è collegato allo schermo tramite linea tratteggiata ed evidenziato da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS. Inoltre, per

facilitare l'interazione con questo tasto, viene disabilitato il componente *Box Collider* del tasto analizzato in precedenza per impostare l'altitudine dell'autopilota;

- **Exit condition:** Il pulsante è configurato in *WEAVR* come *Executable*, ovvero un elemento con cui si interagisce tramite click del grilletto posteriore del controller. La *exit condition* si verifica quando l'utente aziona il suddetto elemento;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste diverse azioni: la maschera del tasto viene sostituita con quella corrispondente alla sua attivazione, si esegue l'animazione del pulsante con cui si è interagito, si cambia lo stato del tasto da OFF a ON, mediante la funzione di *WEAVR* chiamata *Call Method - GoToState (string)* e avviene la rimozione della *billboard* in precedenza descritta.



Figura 5.17: Step 2: Approach

### Step 3: AP2

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di premere il pulsante AP2 per attivare la modalità dell'autopilota per l'atterraggio strumentale in autonomia, che è collegato allo schermo tramite linea tratteggiata ed evidenziato da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS. Inoltre, per facilitare l'interazione con questo tasto, viene disabilitato il componente *Box Collider* del tasto APPR;
- **Exit condition:** Il pulsante è configurato in *WEAVR* come *Executable*, ovvero un elemento con cui si interagisce tramite click del grilletto posteriore del controller. La *exit condition* si verifica quando l'utente aziona il suddetto elemento;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste diverse azioni: la maschera del tasto viene sostituita con quella corrispondente alla sua attivazione, si

esegue l'animazione del pulsante con cui si è interagito, si cambia lo stato del tasto da OFF a ON, mediante la funzione di *WEAVR* chiamata *Call Method - GoToState (string)* e avviene la rimozione della *billboard* in precedenza descritta.



Figura 5.18: Step 3: AP2

#### Step 4: Guardare altitudine

- **Enter Action:** Si predispongono una *billboard* in cui viene chiesto di osservare l'evoluzione della stringa di testo di colore viola, relativa alla quota di volo, in quanto, al di sotto dei 7000 piedi sarà necessario estrarre i flap. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS. Inoltre, per facilitare l'interazione con il simulatore fisico, in previsione dei test, è stata aggiunta, sotto la *billboard*, l'immagine relativa al PFD evidenziando l'ubicazione reale della misura dell'altitudine;
- **Exit condition:** La variabile relativa all'altitudine viene mandata dal *SimHub Server* a *WEAVR*, in cui avviene lettura e aggiornamento della stessa in modo continuo. Si effettua, dunque, un controllo su questa variabile e la *exit condition* si verifica quando il valore che essa assume è minore di 7000;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste due azioni: la rimozione della *billboard* in precedenza descritta e quella dell'immagine del PFD.

#### Step 5: Flap 1

- **Enter Action:** Si predispongono una *billboard* in cui viene chiesto di muovere la leva dei flap in posizione 1, che è collegata allo schermo tramite linea tratteggiata ed evidenziata da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;



Figura 5.19: Step 4: Guardare altitudine

- **Exit condition:** La leva è configurata in *WEAVR* come *Executable*, ovvero un elemento con cui si interagisce tramite click del grilletto posteriore del controller. La *exit condition* si verifica quando l'utente aziona il suddetto elemento;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste diverse azioni: si esegue l'animazione della leva con cui si è interagito, si cambia lo stato dell'elemento da 0 a 1, mediante la funzione di *WEAVR* chiamata *Call Method - GoToState (string)* e avviene la rimozione della *billboard* in precedenza descritta.



Figura 5.20: Step 5: Flap 1

### Step 6: Guardare velocità

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di osservare l'evoluzione della stringa di testo di colore giallo, relativa alla velocità di volo, in quanto, al di sotto dei 205 nodi sarà necessario estrarre ulteriormente i flap. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS. Inoltre, per facilitare l'interazione con il simulatore fisico, in previsione dei test, è stata aggiunta, sotto la *billboard*, l'immagine relativa al PFD evidenziando l'ubicazione reale della misura della velocità;
- **Exit condition:** La variabile relativa all'altitudine viene mandata dal *SimHub Server* a *WEAVR*, in cui avviene lettura e aggiornamento della stessa in modo continuo. Si effettua, dunque, un controllo su questa variabile e la *exit condition* si verifica quando il valore che essa assume è minore di 205;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste due azioni: la rimozione della *billboard* in precedenza descritta e quella dell'immagine del PFD.



Figura 5.21: Step 6: Guardare velocità

### Step 7: Flap 2

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di muovere la leva dei flap in posizione 2, che è collegata allo schermo tramite linea tratteggiata ed evidenziata da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;
- **Exit condition:** La leva è configurata in *WEAVR* come *Executable*, ovvero un elemento con cui si interagisce tramite click del grilletto posteriore del controller. La *exit condition* si verifica quando l'utente aziona il suddetto elemento;

- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste diverse azioni: si esegue l'animazione della leva con cui si è interagito, si cambia lo stato dell'elemento da 1 a 2, mediante la funzione di *WEAVR* chiamata *Call Method - GoToState (string)* e avviene la rimozione della *billboard* in precedenza descritta.



Figura 5.22: Step 7: Flap 2

### Step 8-9-10-11: Flap 3 e Flap FULL

Come per gli step visti in precedenza, si estendono i flap man mano che l'aereo rallenta. In particolare:

- **Flap 3:** si estendono i flap passando dalla posizione 2 alla 3, attraverso *GoToState* e animazione, quando la velocità scende sotto i 180 nodi;
- **Flap FULL:** si estendono i flap passando dalla posizione 3 alla FULL, attraverso *GoToState* e animazione, quando la velocità scende sotto i 150 nodi.

Anche questi step, come i precedenti riguardanti le estensioni dei flap, sono preceduti da un nodo che chiede di tenere sotto controllo l'evoluzione della velocità, fino ad arrivare a quella target per l'estensione.

### Step 12: Set QNH

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di premere il pulsante per cambiare sistema di riferimento della pressione, da STD a QNH, che è collegato allo schermo tramite linea tratteggiata ed evidenziato da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;



(a) Guardare velocità



(b) Flap 3



(c) Guardare velocità



(d) Flap FULL

Figura 5.23: Step 8-9-10-11: Flap 3 e Flap FULL

- **Exit condition:** Il pulsante è configurato in *WEAVR* come *Executable*, ovvero un elemento con cui si interagisce tramite click del grilletto posteriore del controller. La *exit condition* si verifica quando l'utente aziona il suddetto elemento;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste diverse azioni: la maschera dello schermo relativo al riferimento della pressione viene sostituita, come mostrato in Figura 5.24 da (a) a (b), per entrambi i display di pilota e copilota, si esegue l'animazione del tasto con cui si è interagito e avviene la rimozione della *billboard* in precedenza descritta.



(a) Display STD



(b) Display QNH

Figura 5.24: Maschere sistema di pressione





Figura 5.25: Step 12: Set QNH

### Step 13-14: Checklist

Lo step 13 è un nodo di attesa in quanto viene chiesto di controllare l'altitudine fino al raggiungimento di 2700 piedi di quota per poi andare a visionare e completare la checklist per l'atterraggio. La sua descrizione è analoga a quella dello step 4, trattato in precedenza. Per quanto riguarda il nodo successivo, viene richiesta l'ispezione visiva della checklist:

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di avvicinarsi al pannello ECAM superiore per ispezionare la checklist, mostrata in Figura 5.26, che è collegato allo schermo tramite linea tratteggiata ed evidenziato da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;
- **Exit condition:** L'ispezione visiva prevede che l'utente si avvicini ad una certa distanza preimpostata dall'oggetto, per un determinato periodo di tempo. La *exit condition* si verifica quando l'utente si avvicina a distanza minore di 0.6 metri, rimanendoci per 3 secondi di tempo;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, è prevista solo la rimozione della *billboard* in precedenza descritta.

### Step 15: Landing gear DOWN

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di muovere la leva del carrello di atterraggio per la sua estrazione, che è collegata allo schermo tramite linea tratteggiata ed evidenziata da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;



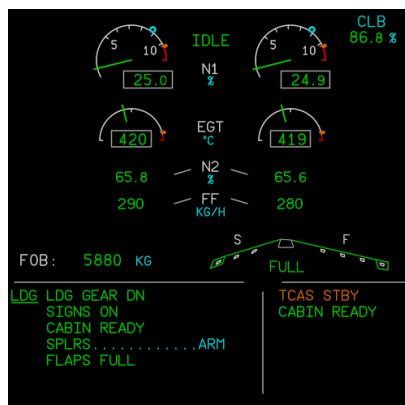
Figura 5.26: Landing checklist



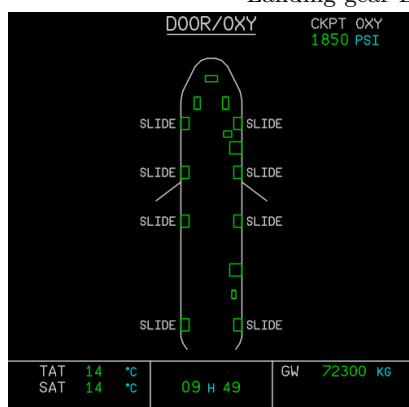
Figura 5.27: Step 14: Checklist

- **Exit condition:** La leva è configurata in *WEAVR* come *Executable*, ovvero un elemento con cui si interagisce tramite click del grilletto posteriore del controller. La *exit condition* si verifica quando l'utente aziona il suddetto elemento;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste diverse azioni: si esegue l'animazione della leva con cui si è interagito, si cambia lo stato dell'elemento da UP a DOWN, mediante la funzione di *WEAVR* chiamata *Call Method - GoToState (string)*, si sostituisce la maschera della checklist che corrisponde al pannello ECAM

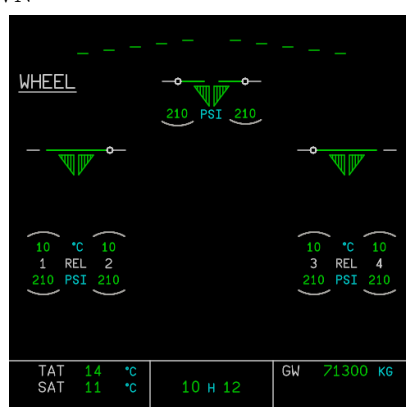
superiore, mostrata in precedenza, con quella aggiornata in base allo step effettuato, come si vede in Figura 5.28 (a) e anche quella relativa al pannello inferiore, che passa da (b) a (c) e avviene la rimozione della *billboard* in precedenza descritta.



(a) Checklist ECAM superiore aggiornata  
- Landing gear DOWN



(b) Pagina ECAM inferiore - DOOR/OXY



(c) Pagina ECAM inferiore - Landing gear DOWN

Figura 5.28: Maschere ECAM

### Step 16: Spoiler armed

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di alzare la leva degli aerofreni per armarli, che è collegata allo schermo tramite linea tratteggiata ed evidenziata da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;
- **Exit condition:** La leva è configurata in *WEAVR* come *Executable*, ovvero un elemento con cui si interagisce tramite click del grilletto posteriore del controller. La *exit condition* si verifica quando l'utente aziona il suddetto elemento;



Figura 5.29: Step 15: Landing gear DOWN

- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste diverse azioni: si esegue l'animazione della leva con cui si è interagito, si cambia lo stato dell'elemento da DISARMED a ARMED, mediante la funzione di *WEAVR* chiamata *Call Method - GoToState (string)*, si sostituisce la maschera della checklist che corrisponde al pannello ECAM superiore, mostrata in precedenza, con quella aggiornata in base allo step effettuato, come si vede in Figura 5.30 e avviene la rimozione della *billboard* in precedenza descritta.

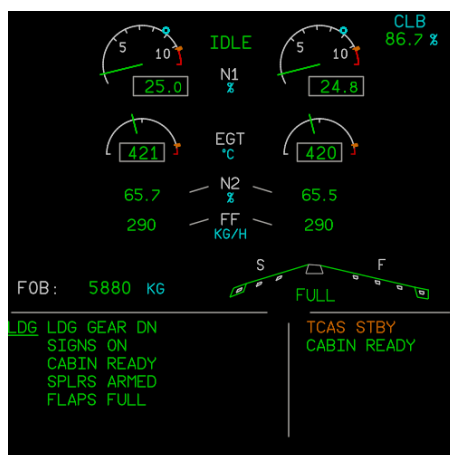


Figura 5.30: Checklist ECAM superiore aggiornata



Figura 5.31: Step 16: Spoiler armed

#### Step 17: Auto BRK LOW

- **Enter Action:** Si predispone una *billboard* in cui viene chiesto di premere il pulsante per attivare la frenata automatica a bassa intensità, che è collegato allo schermo tramite linea tratteggiata ed evidenziato da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;
- **Exit condition:** Il pulsante è configurato in *WEAVR* come *Executable*, ovvero un elemento con cui si interagisce tramite click del grilletto posteriore del controller. La *exit condition* si verifica quando l'utente aziona il suddetto elemento;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste diverse azioni: la maschera del pulsante viene sostituita per mostrare la sua attivazione, come mostrato in Figura 5.32 (a), viene nuovamente cambiata la maschera della checklist, visibile in Figura 5.32 (b), si esegue l'animazione del tasto con cui si è interagito e avviene la rimozione della *billboard* in precedenza descritta.

#### Step 18-19: Flare

Lo step 18 è un nodo di attesa in quanto viene chiesto di controllare l'altitudine fino al raggiungimento di 1000 piedi di quota, per poi effettuare la manovra di flare. La sua descrizione è analoga a quella dello step 4, trattato in precedenza.

Per quanto riguarda il nodo successivo, viene richiesto di portare indietro lo stick per effettuare la manovra di flare:

- **Enter Action:** Si predispone una *billboard* in cui viene chiesto di portare lo stick indietro per effettuare la manovra di flare, che è collegato allo schermo tramite linea tratteggiata ed evidenziato da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;



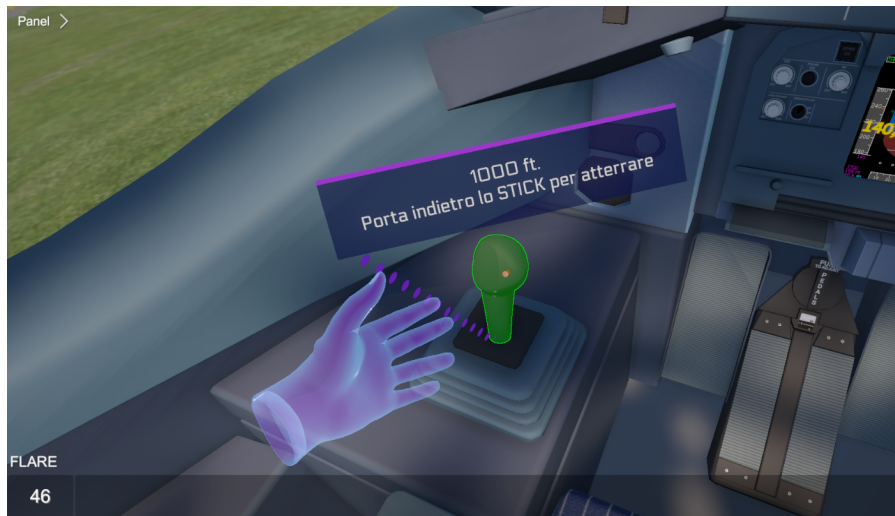
(a) Maschera Auto BRK LOW ON (b) Checklist aggiornata Auto BRK LOW

Figura 5.32



Figura 5.33: Step 17: Auto BRK LOW

- **Exit condition:** Lo stick è configurato in *WEAVR* come *Executable*, ovvero un elemento con cui si interagisce tramite click del grilletto posteriore del controller. La *exit condition* si verifica quando l'utente aziona il suddetto elemento;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste diverse azioni: si esegue l'animazione dello stick con cui si è interagito, si cambia lo stato dell'elemento da *NEUTRAL* a *BACKWARD*, mediante la funzione di *WEAVR* chiamata *Call Method - GoToState (string)*, avviene la rimozione della *billboard* in precedenza descritta e inizia un timer di 4 secondi, mediante la funzione *Wait*; durante questi secondi avviene il *touchdown* e si prosegue allo step successivo.



**Figura 5.34:** Step 19: Flare

#### Step 20: Reverse

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di portare la manetta in posizione REVERSE per attivare l'inversione di spinta, che è collegata allo schermo tramite linea tratteggiata ed evidenziata da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;
- **Exit condition:** La manetta è configurata in *WEAVR* come *Executable*, ovvero un elemento con cui si interagisce tramite click del grilletto posteriore del controller. La *exit condition* si verifica quando l'utente aziona il suddetto elemento;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste diverse azioni: si esegue l'animazione della manetta con cui si è interagito, si cambia lo stato dell'elemento da CLIMB a REVERSE, mediante la funzione di *WEAVR* chiamata *Call Method - GoToState (string)*, viene sostituita la maschera del pannello ECAM inferiore con quella aggiornata, come si nota in Figura 5.35, e avviene la rimozione della *billboard* in precedenza descritta.

#### Step 21-22: IDLE

Lo step 21 è un nodo di attesa in quanto viene chiesto di controllare la velocità fino al raggiungimento di 50 nodi, per poi portare la manetta in posizione IDLE. La sua descrizione è analoga a quella dello step 6, trattato in precedenza.

Per quanto riguarda l'ultimo nodo, viene richiesto di portare la manetta in posizione IDLE:

- **Enter Action:** Si predispose una *billboard* in cui viene chiesto di portare la manetta in posizione IDLE, che è collegata allo schermo tramite linea tratteggiata ed evidenziata da un contorno verde. Questo messaggio è accompagnato dalla lettura vocale del testo attraverso la funzione TTS;

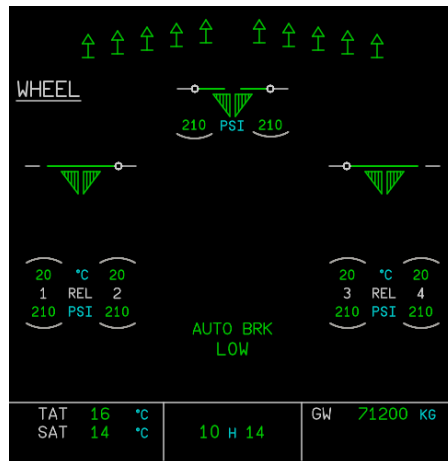


Figura 5.35: Maschera ECAM inferiore - REVERSE



Figura 5.36: Step 20: Reverse

- **Exit condition:** La manetta è configurata in *WEAVR* come *Executable*, ovvero un elemento con cui si interagisce tramite click del grilletto posteriore del controller. La *exit condition* si verifica quando l'utente aziona il suddetto elemento;
- **Exit Action:** Appena la *exit condition* viene soddisfatta, sono previste diverse azioni: si esegue l'animazione della manetta con cui si è interagito, si cambia lo stato dell'elemento da REVERSE a IDLE, mediante la funzione di *WEAVR* chiamata *Call Method - GoToState (string)* e avviene la rimozione della *billboard* in precedenza descritta.





Figura 5.37: Step 22: IDLE

### 5.3 Take Off & Climb

Questa procedura è analizzata in dettaglio nell'elaborato della collega Annalisa Cantiani e, dunque, in questo lavoro sono solo riportati i vari step implementati in *WEAVR*.

- **Step 0:** Setup della scena. Vengono impostati i vari comandi nelle posizioni corrette e vengono abilitate le maschere degli schermi;
- **Step 1:** Take off clearance. L'utente deve effettuare un'ispezione visiva per verificare che la pista sia libera e dichiararlo ad alta voce;
- **Step 2:** ECAM inspection. Viene richiesta un'ispezione visiva per leggere la checklist sull'ECAM superiore, in cui sono elencati passi successivi della procedura;
- **Step 3:** Auto BRK MAX. L'utente deve premere il pulsante per impostare la frenata automatica, in caso di decollo abortito, sulla massima intensità disponibile;
- **Step 4:** Spoiler ARMED. Viene richiesto di sollevare la leva degli aerofreni per armarli;
- **Step 5:** Flap 1. L'utente deve portare la leva per estrarre gli ipersostentatori in posizione 1;
- **Step 6:** T.O. CONFIG. Si richiede di premere il pulsante del *Take off configuration* per completare la checklist;
- **Step 7:** Guardare checklist. L'utente deve ispezionare nuovamente il pannello in cui è descritta la checklist per verificare di averla completata;
- **Step 8:** Set altitude. L'utente deve premere il pulsante relativo all'altitudine dell'autopilota per impostare il target di 10000 ft;

- **Step 9:** Parking Brake off. Viene richiesto di rimuovere il freno di stazionamento;
- **Step 10:** Stick forward. L'utente deve muovere in avanti lo Stick per incrementare la trazione durante il rullaggio;
- **Step 11:** Throttle 1/4. L'utente deve dare un po' di potenza ai motori portando leggermente in avanti la manetta;
- **Step 12:** N1 check. Viene richiesta un'ispezione visiva sul pannello ECAM superiore per verificare che i giri motore aumentino correttamente;
- **Step 13:** Throttle FLEX. Viene richiesto di portare la manetta in posizione FLEX per il decollo;
- **Step 14:** 100 Knots. Viene annunciato quando la velocità arriva a 100 nodi;
- **Step 15:** Stick neutral. L'utente deve riportare lo stick in posizione neutra;
- **Step 16:** V1. Viene annunciato quando la velocità arriva a V1, ovvero 127 nodi;
- **Step 17:** Rotate. L'utente deve riportare lo stick in posizione neutra;
- **Step 18:** Positive climb. Viene enunciata questa frase quando il rateo di salita supera i 2000 ft/min;
- **Step 19:** Landing gear retract. Viene richiesto di retrarre il carrello di atterraggio;
- **Step 20:** Guardare altitudine. L'utente deve prestare attenzione alla quota e aspettare che si raggiungano i 2300 ft di altitudine;
- **Step 21:** Throttle CLB. Viene richiesto di portare la manetta in posizione CLIMB;
- **Step 22:** AP1. L'utente deve premere il pulsante per attivare l'autopilota;
- **Step 20:** Guardare altitudine. L'utente deve prestare attenzione alla quota e aspettare che si raggiungano i 3100 ft di altitudine;
- **Step 21:** Flap ZERO. Viene richiesto di muovere i flap in posizione 0;
- **Step 22:** Spoiler DISARMED. L'utente deve abbassare la leva degli aerofreni per disarmarli;

## Capitolo 6

# Confronto VR - Computer Based Training

### 6.1 Introduzione del test

Il metodo di formazione aeronautico più diffuso è il *Computer Based Training* **CBT** che consiste nello sfruttare tecnologie informatiche per fornire contenuti didattici ed esperienze di apprendimento interattivo. Questa forma di apprendimento è diventata una componente integrante dei programmi di formazione aeronautica grazie alla sua efficacia nel migliorare la ritenzione delle conoscenze e lo sviluppo delle competenze. Il CBT consente ai professionisti dell'aviazione, come piloti, assistenti di volo e personale di manutenzione, di completare i moduli di addestramento al proprio ritmo e convenienza e può coprire una vasta gamma di argomenti, dai principi di base del volo, alle procedure operative standard, alla gestione delle emergenze, utilizzando software per computer, presentazioni multimediali e simulazioni interattive [59].

Il CBT presenta numerosi vantaggi rispetto all'insegnamento tradizionale come:

- **Efficienza dell'apprendimento:** Questa tipologia di formazione si è dimostrata in grado di migliorare l'efficienza dell'apprendimento. Infatti, degli studi dimostrano che il CBT permette agli studenti di trattenere le informazioni meglio e per periodi più lunghi rispetto ai metodi di apprendimento tradizionali, grazie alla sua natura interattiva, che coinvolge gli alunni e incoraggia una partecipazione attiva. Inoltre, il CBT consente agli apprendisti di procedere al proprio ritmo, assicurando che comprendano pienamente i materiali di formazione, prima di passare al modulo successivo. Questo approccio permette agli individui di concentrarsi su aree in cui potrebbero necessitare di più tempo e pratica, migliorando i risultati complessivi dell'apprendimento. In aggiunta, i moduli di formazione basati su computer possono essere facilmente aggiornati per incorporare le ultime regolamentazioni, procedure e avanzamenti tecnologici del settore;
- **Costi-efficienza:** I metodi di formazione tradizionali spesso richiedono risorse, come simulatori e materiali didattici fisici, che possono essere dispendiosi in termini di tempo e costi di manutenzione. Al contrario, il CBT riduce la necessità di materiali didattici ed elimina i costi associati alla loro produzione e distribuzione. Inoltre, l'addestramento

basato su computer elimina la necessità di spese di viaggio e alloggio per i partecipanti, poiché può essere accessibile da remoto da qualsiasi luogo con una connessione internet, rendendo la formazione più accessibile e riducendone il costo complessivo;

- **Flessibilità:** Con il CBT, è possibile accedere ai materiali didattici e completare i moduli secondo la propria convenienza, eliminando la necessità di orari fissi e permettendo agli individui di bilanciare i propri impegni lavorativi con quelli di apprendimento. Questa flessibilità è particolarmente vantaggiosa per piloti e assistenti di volo che hanno orari irregolari e potrebbero non avere il tempo o la disponibilità per le sessioni di formazione tradizionali in aula.

Tuttavia il CBT ha una grande limitazione: l'esperienza pratica. Infatti, il CBT può eccellere nella teoria e nella conoscenza procedurale, ma non è in grado di fornire esperienza pratica che è un elemento essenziale nell'addestramento aeronautico, necessitando, quindi, di metodi di addestramento complementari.

Un'alternativa a questa tipologia di apprendimento è la realtà virtuale: con la VR si può coniugare la potenza formativa del CBT con l'esperienza pratica, addestrandosi in ambiente virtuale e usufruendo, quindi, di tutti i vantaggi ad esso collegati. Nell'articolo [60] viene condotto uno studio sull'efficacia della realtà virtuale nella formazione degli assistenti di volo: viene insegnato come aprire diversi tipi di uscita sopra l'ala di un aereo in situazioni di emergenza. Nel test sono coinvolti 72 partecipanti a cui viene sottoposto l'addestramento: metà in ambiente virtuale e l'altra metà in formato cartaceo. Dai risultati emerge la superiorità della realtà virtuale come metodo di apprendimento con un punteggio medio sugli step corretti di 25.88 contro 17.43 per quanto riguarda l'addestramento tradizionale. Tuttavia, essendo la VR ancora in fase di sviluppo, non sono presenti in letteratura molti casi di studio sulla sua effettiva efficacia rispetto al CBT. Uno studio comparativo è stato tentato nei lavori di Brizzi [5] e Pavone [6] al Politecnico di Torino, ma causa pandemia, lo studio è stato condotto su un campione di 5 persone e quindi non è risultato scientificamente attendibile.

Questo progetto di tesi si conclude quindi con un test sperimentale per verificare l'efficacia della realtà virtuale nell'addestramento piloti, sfruttando le procedure operative dinamiche descritte nel capitolo precedente.

## 6.2 Struttura del test

### Partecipanti

Per effettuare questo test sperimentale è stato scelto un campione di studenti volontari all'interno del Politecnico di Torino, tutti ingegneri o aspiranti tali, di cui circa la metà Aerospaziali. In totale si sono selezionati 66 partecipanti di età compresa tra i 19 e i 24 anni, a cui è stato fatto firmare il modulo per il consenso del trattamento dei dati personali e della privacy, visibile in Appendice A. Inizialmente, erano stati selezionati solo studenti iscritti al corso di ingegneria aerospaziale, per garantire una conoscenza di base del velivolo e delle sue meccaniche; successivamente, il campione è stato ampliato con studenti di altre discipline ingegneristiche, dal momento che, nonostante ottenessero punteggi inferiori, questi ultimi presentavano un approccio più neutro nei confronti del

test. Infatti, alcuni ingegneri aerospaziali possedevano una conoscenza pregressa troppo dettagliata delle Procedure Operative Standard SOP dell'A320, fatto che rischiava di falsare i risultati dell'esperimento. Per organizzare l'afflusso delle persone è stata utilizzata una piattaforma gratuita, denominata *Rally* [61], in cui è stato possibile creare un calendario con date e orari disponibili per la prenotazione, in modo che ogni partecipante potesse, attraverso un link, dare la sua disponibilità in base ai propri impegni, senza sovrapporsi con gli altri candidati.

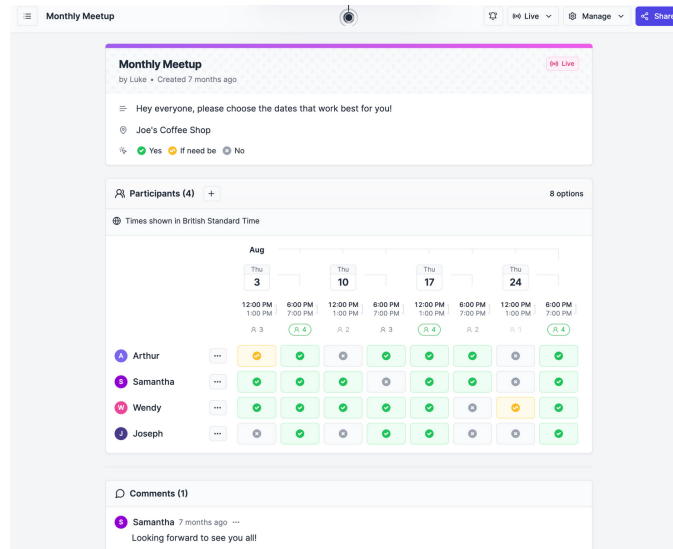


Figura 6.1: Home page della piattaforma Rally con esempio di calendario

## Hardware e software

Si è usufruito del visore *HTC VIVE PRO* collegato al computer del laboratorio per la simulazione in realtà virtuale della procedura operativa da testare, un notebook per mostrare le diapositive preparate per l'addestramento *Computer Based* e, infine, il simulatore fisico, descritto nel Capitolo 3, messo a disposizione dal Politecnico di Torino nel Laboratorio di Simulazione del Volo per verificare l'apprendimento della suddetta procedura. L'architettura software è quella analizzata nel capitolo precedente, che include, quindi, *Unity*, *Pacelab WEAVR*, *SimHub* e *X-Plane*.

## Fase 1: Introduzione

Per effettuare la prova si è scelto di procedere con due candidati alla volta, che vengono impegnati per circa un'ora di tempo. Innanzitutto vengono introdotte le modalità della prova: un partecipante proverà l'addestramento in realtà virtuale mentre l'altro tramite metodo tradizionale CBT; per scegliere il tipo di addestramento si è scelto di avvalersi del classico lancio della moneta, in modo da mantenere il più possibile l'aleatorietà dei risultati. Si sfrutta la sola procedura operativa del *Take-off* in quanto ha una durata, in VR, di circa

3/4 minuti e questo consente di non allungare troppo i tempi di addestramento poiché la procedura di *Landing*, sebbene di difficoltà simile, presenta al suo interno molti tempi di attesa dove l'utente non deve compiere azioni.

Per quanto riguarda il CBT sono state preparate delle diapositive, di cui un esempio in Figura 6.2, dove vengono illustrati gli step da eseguire, esattamente come in VR, e sono stati concessi 15 minuti di tempo per studiare la procedura, con la possibilità di muoversi liberamente tra le slides. Invece, chi segue l'addestramento in realtà virtuale ripete l'intera procedura per 4 volte, in modo da allineare le tempistiche con l'altro metodo di apprendimento.

In particolare, nella diapositiva mostrata, si possono individuare alcuni aspetti in comune

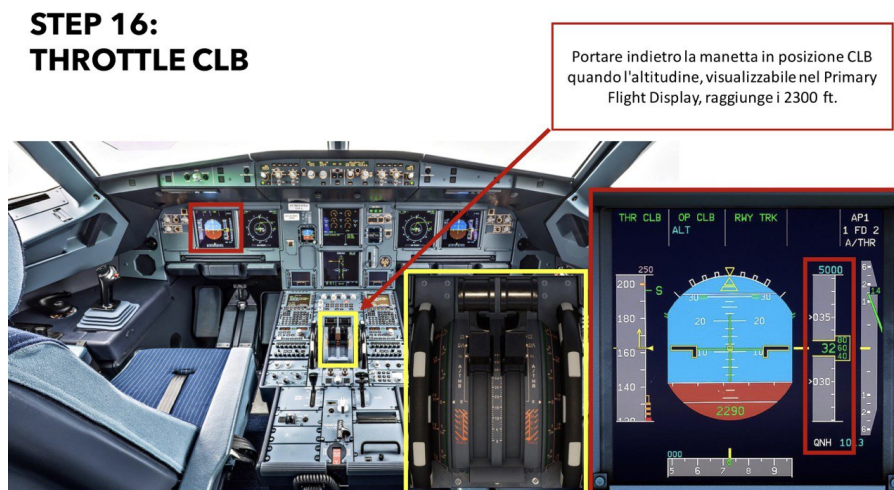


Figura 6.2: Esempio diapositiva CBT

con tutte le altre:

- In centro viene mostrato il cockpit dell'Airbus A320, il cui modello corrisponde a quello del simulatore fisico del laboratorio, anche se semplificato;
- L'elemento, pulsante o leva che è protagonista di questo particolare step della procedura è riquadrato in giallo, per poi, sempre tramite riquadro dello stesso colore, essere ingrandito per mostrarlo con maggior dettaglio;
- In alto a destra nella diapositiva, nel riquadro rosso, è presente la descrizione dello step e una freccia che indica la posizione dell'elemento in questione nel cockpit;
- In alcune slide, come in quella mostrata, viene ingrandito anche il PFD, in cui si mette in risalto il parametro di volo da tenere sotto controllo per quel determinato step;
- Infine, in alto a sinistra viene riportato il nome e il numero dello step.

Prima di iniziare viene fatto un rapido sondaggio, mostrato in Figura 6.3, per capire se ci sono eventuali esperienze pregresse con la realtà virtuale, nell'utilizzo di simulatori di volo o di pilotaggio, per tenerne conto nei risultati.

Partecipante
Nome:
Cognome:
Addestramento (VR /CB):
Età:
Professione:
Anno accademico:
Dimestichezza con VR:
Esperienze con simulatori di volo fisici e non (videogiochi):
Esperienze di pilotaggio:
Conoscenza della procedura di decollo di un aereo:

**Figura 6.3:** Questionario iniziale

## Fase 2: Training

Il candidato che segue il CBT inizia subito, mentre per chi deve usare il VR è prevista una breve procedura di prova, che non ha elementi in comune con quella di *Take-off*, ma serve per adattarsi all'ambiente virtuale e rappresenta un tutorial su come eseguire le azioni richieste. Terminato il tutorial un supervisore avvia la procedura da imparare e rimane a seguire il candidato nel caso di necessità e per il riavvio della procedura; l'altro supervisore, invece, rimane a disposizione della persona che segue l'addestramento tramite diapositive, per eventuali domande.

## Fase 3: Valutazione al simulatore

Terminata la fase di addestramento per entrambi i candidati, si passa alla valutazione di quanto appreso tramite il simulatore fisico dell'Airbus A320. Al simulatore, il partecipante si siede sul sedile del pilota, un supervisore su quello del copilota, per valutare l'esecuzione della procedura, e l'altro supervisore prepara la scena e verifica che il decollo proceda come previsto.

La valutazione avviene tramite un metodo appositamente preparato che consiste in un punteggio che va da 0 a 19, un punto per ogni step corretto, con eventuale sottrazione di punti in base al tipo di errore commesso: step dimenticato, pulsante premuto errato, valore impostato sbagliato e altri. La scheda di valutazione è mostrata in Appendice B.

## Fase 4: Seconda valutazione al simulatore

Ai partecipanti viene chiesta la disponibilità per ritornare una seconda volta in laboratorio a distanza di una settimana circa: più precisamente tra i 6 e gli 8 giorni dalla prova precedente. Il secondo tentativo si distingue dal primo, in quanto non consiste nell'addestramento, ma nella sola prova pratica, con l'obiettivo di capire se a distanza di tempo e senza ripensare a casa alla procedura imparata, nei limiti del possibile, o guardare video tutorial, la formazione in realtà virtuale, eseguendo effettivamente la procedura, sia più efficace, per memorizzare, rispetto al CBT.



**Figura 6.4:** Training VR e valutazione al simulatore

### Fase 5: Questionario NASA TLX

Infine, viene proposto un questionario sviluppato dalla NASA in grado di valutare, attraverso opportuni parametri, il carico di lavoro sottoposto ad una persona in un determinato task: *NASA TLX*. Il *NASA Task Load Index* è uno strumento che consente agli utenti di eseguire valutazioni del carico di lavoro soggettivo su operatori che lavorano con vari sistemi di interfaccia uomo-macchina. Originariamente sviluppato come un questionario cartaceo da Sandra Hart [62] del *NASA Ames Research Center ARC* negli anni '80, il *NASA TLX* è diventato lo standard d'eccellenza in una vasta gamma di applicazioni come cockpit di aerei, postazioni di comando, simulazioni e test di laboratorio. Incorporando una procedura di valutazione multidimensionale, il *NASA TLX* deriva un punteggio di carico di lavoro complessivo basato su una media ponderata delle valutazioni su sei sottogruppi:

- **Mental Demand:** si tratta dello sforzo mentale richiesto da un determinato compito;
- **Physical Demand:** ovvero lo sforzo fisico richiesto da un certo task;
- **Temporal Demand:** indica quanto incide il tempo a disposizione nello svolgimento di un determinato lavoro;
- **Performance:** si riferisce a quanto il candidato ha avuto successo nel compiere il task e a quanto è soddisfatto di ciò che ha fatto in termini di performance;
- **Effort:** è una combinazione di quanto sforzo fisico e mentale è stato richiesto per avere una determinata performance;
- **Frustration:** indica quanto il candidato si sia sentito insicuro, irritato e stressato durante un task.

In questo progetto si è usufruito della versione mobile di *NASA TLX*, scaricabile gratuitamente per sistemi *iOS* direttamente dall'*App Store*. Il vantaggio di avere l'applicazione risiede nel fatto che non è più necessario effettuare i calcoli dei punteggi ed è, inoltre,



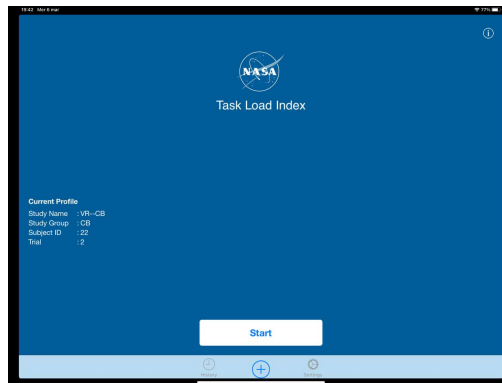
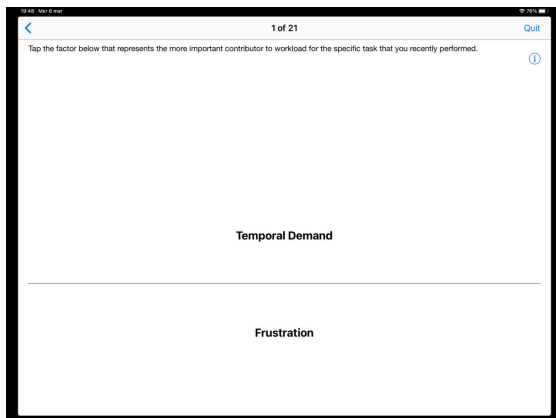
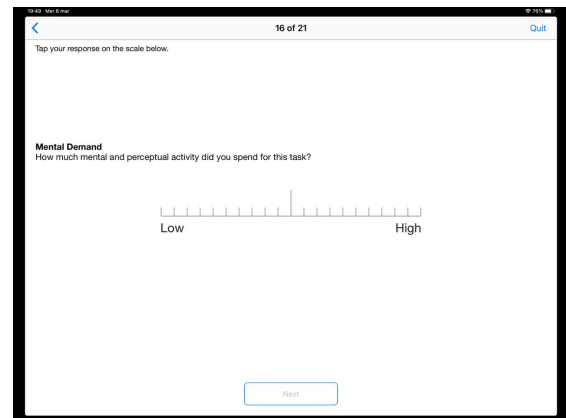


Figura 6.5: Home page NASA TLX app

possibile mantenere in memoria i punteggi di ogni candidato. Ai candidati viene fornito un *IPAD* con cui è possibile effettuare il test. Inizialmente, vengono spiegate le categorie sopra elencate e, successivamente, si procede ad un confronto tra queste categorie, prese a coppie, in cui viene richiesto di scegliere quale delle due è stata più rilevante; infine, per ognuna di esse, è necessario indicare quanto il sottogruppo in questione sia stato rilevante, attraverso una scala graduata, come mostrato in Figura 6.6. Al termine del questionario viene elaborato il punteggio del carico di lavoro in una scala da 0, minimo carico di lavoro, a 100, massimo carico di lavoro.



(a) Confronto tra sottogruppi



(b) Assegnazione punteggio su scala graduata

Figura 6.6

### 6.3 Risultati e analisi dei dati del test sperimentale

	1° TRIAL	2° TRIAL	TLX
VR1	16.9	18.4	46
VR2	15	17.4	72.67
VR3	15.9	16.2	42.33
VR4	17.4	16.2	40
VR5	14.9	16	64.33
VR6	19	19	31.33
VR7	16.7	18.2	37.67
VR8	17.2	18.5	41.88
VR9	17.4	18	48.33
VR10	15.9	16.9	25.33
VR11	15.4	16.8	56
VR12	15.5	16.6	60
VR13	13.9	13.9	62.33
VR14	14.4	16.2	53.33
VR15	16.3	16.7	8.33
VR16	16.2	16.9	40
VR17	15.1	16.2	57.33
VR18	15.2	15.6	53.33
VR19	17	17.8	53
VR20	16.4	16.9	70.67
VR21	13.2	15.9	78.67
VR22	15.4	15.4	34.33
VR23	16.5	16.9	43
VR24	15.9	16.4	51
VR25	17.6	17.9	36.33
VR26	16.2	16.8	57.6
VR27	16	15.8	46.89
VR28	14.9	15.9	29.6
VR29	18.7	18.1	32.67
VR30	13.5	14.3	34
VR31	16	16.4	59.33
VR32	15.7	15.5	68
VR33	16	15.7	66

Tabella 6.1: Risultati VR dei test sperimentali

I dati dei test sono stati raccolti in due tabelle, in cui sono visibili i punteggi ottenuti da ogni candidato, sia per il primo che per il secondo trial. Le persone, per questioni di privacy, sono identificate con la sigla **VR**, seguita da un numero, per chi ha seguito l'addestramento in realtà virtuale e con l'identificativo **CB**, sempre seguito da un numero, se formati con computer based training. Inoltre viene anche riportato il relativo punteggio del *NASA TLX*.

	1° TRIAL	2° TRIAL	TLX
CB1	17	18.4	68.33
CB2	19	18.2	53.67
CB3	13.6	12.5	24.33
CB4	16.4	14.4	75.33
CB5	17.7	17	72
CB6	19	18.2	69.67
CB7	17.5	17.5	64
CB8	14.2	13.3	60.33
CB9	16.6	15.2	62.54
CB10	14.4	13.5	55
CB11	12.7	12.2	62.33
CB12	14.6	14.1	70.67
CB13	9	10.4	87.67
CB14	16.7	15.7	54.67
CB15	16.1	15.6	54
CB16	14.8	14.5	10.33
CB17	14.3	14.7	50.67
CB18	14.6	14.2	58.3
CB19	14.8	16	61.67
CB20	15.2	15.2	65.3
CB21	15.4	15.1	81
CB22	15.5	13.9	55.67
CB23	14.5	15.5	57.33
CB24	13.9	13	48.25
CB25	15.6	14.6	80.29
CB26	14.5	13	71
CB27	14.9	14	63
CB28	16	14.9	53.54
CB29	16.5	15.3	58
CB30	15	12	24
CB31	11.4	12.6	53
CB32	12.6	11.7	66
CB33	13.4	13.1	72.33

Tabella 6.2: Risultati CB dei test sperimentali

Nelle tabelle 6.1 e 6.2 sono compresi tutti i candidati, quindi sia ingegneri aerospaziali che ingegneri di altri orientamenti; la suddivisione dei punteggi nelle varie categorie viene analizzata in seguito.

Per l'analisi dei dati, un primo aspetto rilevante è il calcolo del **punteggio medio** tra tutti i candidati di categoria VR e CB, suddiviso anche tra primo e secondo tentativo, visibile in tabella 6.3.

<b>MEDIA</b>	<b>VR</b>	<b>CB</b>
<b>1° Trial</b>	15.98	15.07
<b>2° Trial</b>	16.65	14.53

**Tabella 6.3:** Media dei punteggi VR e CB

Successivamente viene calcolata la **varianza** dei risultati, riportata in tabella 6.4. Questo aspetto consente di verificare e quantificare in quale misura i dati ottenuti si discostano dal valore medio; un valore basso indica uniformità dei risultati, mentre una varianza alta suggerisce una maggiore dispersione dei dati.

Inoltre, questi valori forniscono informazioni sull'affidabilità e sulla coerenza dei dati: ai fini statistici, si prediligono valori di varianza bassi, ovvero risultati molto simili tra loro, con la conseguenza di un maggiore validità del metodo adottato.

<b>VARIANZA</b>	<b>VR</b>	<b>CB</b>
<b>1° Trial</b>	1.30	2.02
<b>2° Trial</b>	1.16	1.93

**Tabella 6.4:** Varianza dei punteggi VR e CB

Per visualizzare graficamente i risultati, è stato utilizzato uno script *MATLAB* basato su *boxplot*. Questa funzione in *MATLAB* è un utile strumento per rappresentare la distribuzione di un insieme di dati numerici. Questi grafici forniscono una chiara panoramica delle proprietà di dispersione e centralità dei dati. Inserendo una matrice di dati come input, il *boxplot* mostra una scatola che indica i quartili Q1, Q2 e Q3, equivalenti rispettivamente al 25°, 50° e 75° percentile dei dati, con una linea che rappresenta la mediana.

I percentile sono misure che dividono un set di dati in 100 parti uguali: il 25° percentile indica il valore al di sotto del quale si trova il 25% dei dati (il bordo inferiore della scatola), il 50° percentile rappresenta la mediana e il 75° percentile indica il valore al di sotto del quale si trova il 75% dei dati (il bordo superiore della scatola).

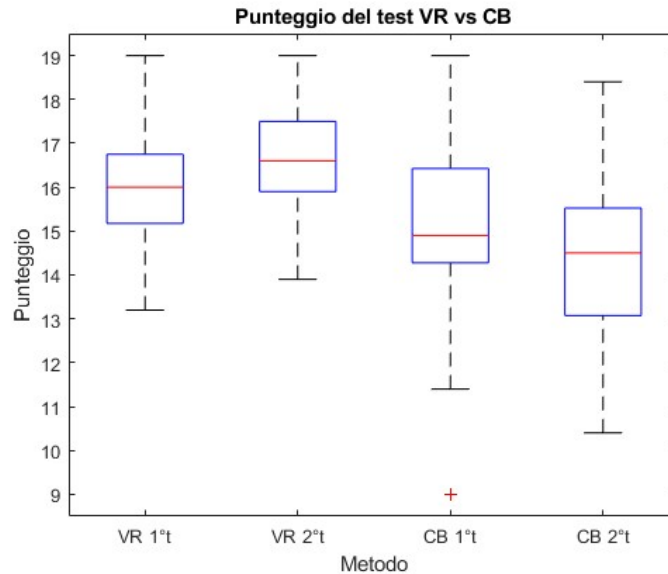
Le *whisker* estese dalla scatola possono rappresentare due cose diverse:

- Valori minimi e massimi inclusi nella distribuzione principale dei dati;
- Valori non inclusi negli outlier, ovvero i dati che si discostano significativamente dalla distribuzione principale.

L'analisi dei risultati dei candidati sottoposti alla formazione VR e CB rivela differenze significative, come chiaramente evidenziato dal grafico presentato in Figura 6.7. In particolare, il grafico mostrato nella Figura 6.8 dimostra che i punteggi medi ottenuti dai candidati che hanno ricevuto la formazione VR risultano superiori rispetto a quelli che hanno ricevuto la formazione CB.

Tuttavia, uno degli aspetti più rilevanti, emersi dall'analisi, riguarda il confronto tra i punteggi del primo e secondo tentativo: i candidati che hanno ricevuto la formazione VR tendono a migliorare nel tempo, mentre quelli che hanno eseguito l'addestramento tramite CB, mostrano l'andamento opposto, con un peggioramento dei risultati.

Questo divario indica che la realtà virtuale applicata in ambito formativo potrebbe favorire la memoria a lungo termine, probabilmente anche per il fatto di eseguire realmente le azioni e coinvolgendo, quindi, la memoria muscolare, rispetto alla formazione CB, che sembra invece sollecitare principalmente la memorizzazione a breve termine. Inoltre, dal grafico,



**Figura 6.7:** Analisi dei risultati ottenuti tramite script Matlab boxplot

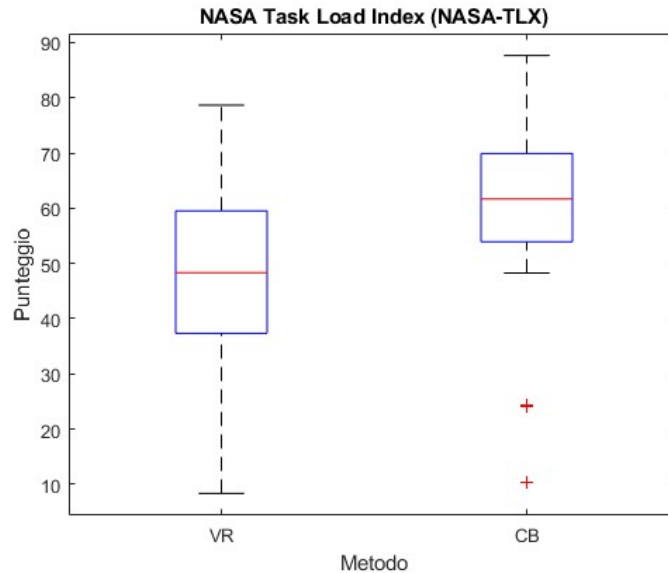
emerge una maggiore varianza nei risultati CB rispetto a quelli VR, visibile nell'ampio range tra i punteggi massimi e minimi. In aggiunta, la mediana della *scatola* non è per nulla centrata per quanto riguarda i risultati CB, e le whiskers risultano più estese. Questi risultati anomali potrebbero essere dovuti alle esperienze pregresse dei candidati o alle differenze individuali nella capacità di memorizzazione e apprendimento.

Infine, dall'analisi delle tabelle di valutazione, il cui *sample* è visibile in Appendice B, emerge una maggiore difficoltà nelle fasi di *rullaggio* e *climb*, da parte dei candidati che hanno seguito la formazione CB. Questa differenza è probabilmente attribuibile alla mancanza di esperienza pratica nell'eseguire gli step in un determinato momento sul velivolo in simulazione, avendo solo studiato questi aspetti da un punto di vista puramente teorico.

Nel grafico mostrato in Figura 6.8, invece, sono analizzati i risultati ottenuti dalla compilazione del questionario *NASA TLX* che permette di valutare il carico di lavoro percepito dal candidato durante la fase di apprendimento.

Come si evince dal grafico, i punteggi ottenuti dai candidati CB risultano abbondantemente più alti di quelli VR, indicando che quest'ultimi percepiscono un minor carico di lavoro. Procedendo con l'analisi delle singole categorie del questionario, si nota che il punteggio più alto è attribuito alla categoria *frustration*, suggerendo che alla formazione CB venga associata un'esperienza più monotona e meno coinvolgente. Infatti, gli utenti che hanno provato la realtà virtuale sostengono di percepirla come una nuova tecnologia da esplorare

e, quindi, meno stressante anche da utilizzare.



**Figura 6.8:** Analisi dei risultati ottenuti dal NASA TLX tramite script Matlab boxplott

Infine, vengono riportati due ulteriori grafici, elaborati tramite *Excel*, che mostrano un confronto dei risultati distinguendo la categoria degli ingegneri aerospaziali, Figura 6.9, da quella delle altre discipline ingegneristiche, Figura 6.10.

Come prevedibile, i punteggi degli ingegneri aerospaziali sono, in media, superiori sia per VR che per CB confrontandoli con i rispettivi addestramenti nell'altra categoria. L'aspetto rilevante di questo studio è che il trend generale rimane invariato rispetto a quanto analizzato in precedenza: si osserva un miglioramento dei risultati VR nel secondo tentativo, dopo una settimana, e un peggioramento, per quanto riguarda i candidati CB.

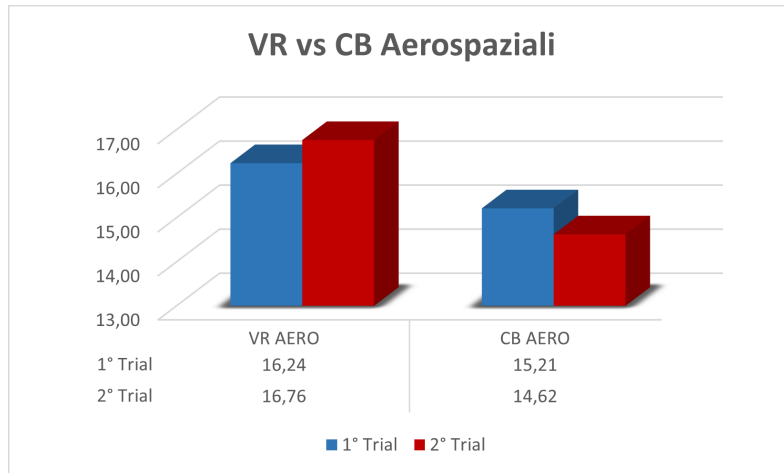


Figura 6.9: Risultati dei test sperimentali con candidati aerospaziali

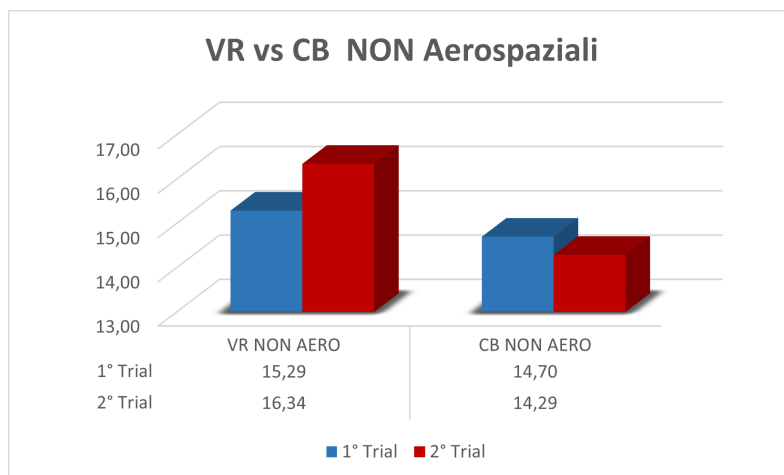


Figura 6.10: Risultati dei test sperimentali con candidati non aerospaziali

## Capitolo 7

# Conclusioni

La tesi si focalizza sull'impiego della realtà virtuale per l'addestramento dei piloti, con un occhio di riguardo alla simulazione del volo, con un approccio che promette di superare le metodologie tradizionali.

Il progetto sfrutta la piattaforma Pacelab WeaVR di *TXT Group* per creare simulazioni VR interattive delle procedure operative, con l'obiettivo di potenziare l'apprendimento dinamico delle manovre di volo. Questo include la digitalizzazione del cockpit dell'Airbus A320 nell'ambiente grafico *Unity* e l'implementazione di procedure a terra, quali *Parking* e *After Landing*, e di procedure in volo, come *Take off* e *Landing*.

In questo modo si è ottenuto un sistema che guida l'utente nell'esecuzione delle procedure operative, assistendolo con aiuti visivi e vocali nell'ambiente VR, consentendo un'esperienza totalmente immersiva.

Il progetto ha superato diverse sfide poste dai lavori precedenti, come la comunicazione bidirezionale tra simulazione di volo e ambiente VR, l'ottimizzazione delle dimensioni del cockpit virtuale e l'introduzione della visualizzazione esterna alla cabina di pilotaggio per aumentare l'immersione.

Il lavoro si conclude con un test sperimentale volto a fornire dati concreti sull'efficacia della VR nell'apprendimento e a concretizzare lo sviluppo delle procedure operative. Non solo si è dimostrato che l'apprendimento tramite VR è più efficace rispetto al tradizionale *Computer Based Training*, ma risulta anche meno pesante e frustrante dal punto di vista del carico di lavoro mentale.

Sicuramente si possono prevedere miglioramenti futuri sia dal punto di vista delle procedure, fornendo la possibilità all'utente di commettere errori e, quindi, di aumentare la sua consapevolezza attraverso un'esperienza personalizzabile, sia dal punto di vista del codice in *C++*, in quanto andrebbe prevista anche in questo caso la possibilità di commettere errori e quindi sarebbe necessario comunicare in modo molto più massivo con il simulatore di volo *X-Plane*.



**Appendice A**

**Informativa sulla privacy**



## INFORMATIVA AI SENSI DELL'ART. 13 DEL REGOLAMENTO GENERALE SULLA PROTEZIONE DEI DATI UE 679/2016 PER LA PARTECIPAZIONE ALLO STUDIO

Gentile partecipante,

chiediamo la sua disponibilità a collaborare ad un progetto di tesi, in collaborazione con il Dipartimento di Ingegneria Meccanica e Aerospaziale (DIMEAS) del Politecnico di Torino

Al fine di informarla circa lo scopo e le caratteristiche dello studio, la invitiamo a leggere attentamente quanto riportato di seguito, affinché lei possa decidere in modo consapevole e libero se partecipare.

I collaboratori coinvolti in questo progetto restano a disposizione per rispondere alle sue eventuali domande tramite la seguente mail:

[s301588@studenti.polito.it](mailto:s301588@studenti.polito.it)

*Responsabile scientifico:* Dr. Stefano Primatesta.

*Collaboratori:* Davide Alpe, Annalisa Cantiani.

### *OBIETTIVI E BENEFICI DELLA RICERCA*

Lo scopo del progetto è valutare l'efficacia dell'apprendimento di una procedura di volo tramite realtà virtuale e quindi il confronto con le tecniche di apprendimento tradizionali.

La valutazione è effettuata attraverso un test da svolgere al simulatore di volo presente in laboratorio, in seguito ad un addestramento in VR o tramite approccio Computer-based.

La sua partecipazione ci permetterà di raccogliere i dati essenziali per procedere con il progetto di tesi.

### *MODALITÀ E DURATA*

La invitiamo a dare la sua disponibilità alla partecipazione tramite apposita piattaforma, scegliendo una data.

Il test in laboratorio si compone di due fasi:

(i) **fase di training** in cui vengono mostrate delle slides riguardanti la procedura di decollo del velivolo Airbus A320, nel caso di addestramento Computer-based, o in cui viene fatta svolgere la suddetta procedura in realtà virtuale;

(ii) **fase di test** al simulatore fisico in cui, sia i candidati che hanno svolto il training VR che quelli che lo hanno svolto il Computer-based training, dovranno replicare la procedura di decollo appena memorizzata.

La valutazione del test sarà data da un punteggio variabile da 0 a 19 a seconda degli errori commessi. Ai candidati verrà chiesto di ritornare la settimana successiva al training per sottoporsi nuovamente al solo test al simulatore.

**La durata complessiva dell'esperimento sarà di circa 45 minuti, variabile a seconda del tipo di addestramento svolto.**

#### *PARTECIPAZIONE VOLONTARIA*

La partecipazione è **assolutamente volontaria** e il rifiuto di partecipare non comporterà alcuna conseguenza negativa. Inoltre, se dovesse cambiare idea e volesse ritirarsi, in qualsiasi momento è libero/a di farlo senza dover fornire alcuna spiegazione.

#### *RISCHI E POTENZIALI BENEFICI DERIVANTI DALLO STUDIO*

Il test è stato sottoposto ad una scrupolosa valutazione dei rischi e dei benefici a breve termine per il partecipante, ma anche a lungo termine per le future applicazioni del nostro studio. Il nostro obiettivo principale, nel breve e nel lungo termine, è quello di massimizzare i vantaggi e ridurre al minimo il danno del partecipante, nell'interesse del benessere del singolo e della società nel suo complesso.

Sebbene nel breve termine il partecipante non riuscirà a misurare il beneficio di questo studio, nel lungo termine si prevede un beneficio per la comunità scientifica che studia l'interazione tra uomini e ambiente virtuale per applicazioni di training del personale aeronautico.

Il test che effettueremo con i volontari non prevede alcun rischio maggiore o aggiuntivo rispetto a quelli che gli esseri umani incontrano durante la vita quotidiana.

Inoltre, se il volontario dovesse sentirsi a disagio sarebbe libero di abbandonare il test in qualsiasi momento.

#### *DATI DI CONTATTO*

**Titolare del trattamento dei dati** è il Politecnico di Torino, con sede in Corso Duca degli Abruzzi, n. 24, 10129 – Torino, nella persona del Rettore.

Il dato di contatto del Titolare è PEC: [politecnicoditorino@pec.polito.it](mailto:politecnicoditorino@pec.polito.it)

Per ulteriori informazioni e chiarimenti: [privacy@polito.it](mailto:privacy@polito.it)

Il **Responsabile della protezione dati** (“DPO”) del Politecnico di Torino, al quale gli interessati possono rivolgersi per questioni relative al trattamento dei loro dati personali e all'esercizio dei loro diritti, è contattabile ai seguenti indirizzi: [dpo@polito.it](mailto:dpo@polito.it); PEC: [dpo@pec.polito.it](mailto:dpo@pec.polito.it)

#### *PRINCIPI, BASE GIURIDICA E FINALITÀ DEL TRATTAMENTO*

Nel rispetto dei principi di liceità, correttezza, trasparenza, adeguatezza, pertinenza e necessità di cui all'art. 5, paragrafo 1, del GDPR, il Politecnico di Torino, in qualità di Titolare, provvederà al trattamento dei suoi dati personali ai sensi dell'art. 6, paragrafo 1, lettere:

a) (“l'interessato ha espresso il consenso al trattamento dei propri dati personali per una o più specifiche finalità ”);

e) (“il trattamento è necessario per l'esecuzione di un compito di interesse pubblico o connesso all'esercizio di pubblici poteri di cui è investito il titolare”);

nel perseguimento delle finalità istituzionali connesse al progresso nella ricerca scientifica come previsto dallo Statuto di Ateneo. In particolare, lo sperimentatore le chiederà di fornire alcuni dati personali, quali: nome, cognome, e-mail, sesso, età.

Queste informazioni, così come i dati che emergeranno nel corso della ricerca, sono importanti per il corretto svolgimento dello studio.

I dati raccolti saranno utilizzati soltanto per il progetto di tesi.

#### *TRASFERIMENTO DEI DATI*

I dati trattati per le finalità di cui sopra verranno comunicati, o saranno comunque accessibili, ai dipendenti e collaboratori assegnati ai competenti uffici del Politecnico di Torino, che saranno adeguatamente istruiti dal Titolare.

I dati relativi alla ricerca potranno essere comunicati in forma anonima e/o aggregata ad altri soggetti, quali, ad esempio, altre Università, istituzioni e organismi pubblici e privati aventi finalità di ricerca, limitatamente ad informazioni prive di dati identificativi e per scopi storici o scientifici.

La divulgazione dei risultati (ad esempio mediante pubblicazione di articoli scientifici, partecipazione a convegni, redazione di tesi ecc.) avverrà soltanto con modalità che rendano difficile la sua identificazione.

La gestione e la conservazione dei dati personali raccolti dal Politecnico di Torino avviene su sistemi ubicati all'interno dell'Ateneo e/o esterni di fornitori di alcuni servizi necessari alla gestione tecnico – amministrativa che, ai soli fini della prestazione richiesta, potrebbero venire a conoscenza dei dati personali degli interessati e che saranno debitamente nominati come Responsabili del trattamento a norma dell'art. 28 del GDPR. I dati salvati saranno conservati con un sistema a doppia codifica.

I dati raccolti non saranno oggetto di trasferimento verso un Paese non appartenente all'Unione Europea (c.d. Paese terzo).

#### *CONSERVAZIONE DEI DATI RACCOLTI ALLA TERMINE DELLA RICERCA*

I dati raccolti verranno conservati tramite un sistema a doppia codifica su server interno con accesso tramite password. I dati personali, dopo 5 anni dalla conclusione della tesi, verranno distrutti.

#### *DIRITTI DEL PARTECIPANTE*

Il partecipante ha il diritto di chiedere al Titolare del trattamento, conformemente agli artt. 15 e ss. del GDPR,

- l'accesso ai propri dati personali ed a tutte le informazioni di cui all'art. 15 del GDPR;
- la rettifica dei propri dati personali inesatti e l'integrazione di quelli incompleti;
- la cancellazione dei propri dati, fatta eccezione per quelli contenuti in atti che devono essere obbligatoriamente conservati dall'Ateneo, e salvo che sussista un motivo legittimo prevalente per procedere al trattamento;
- la limitazione del trattamento nelle ipotesi di cui all'art. 18 del GDPR.
- di opporsi al trattamento dei dati personali, fermo quanto previsto con riguardo alla necessità ed obbligatorietà del trattamento dati per poter fruire dei servizi offerti;
- di revocare il consenso eventualmente prestato per i trattamenti non obbligatori dei dati, senza con ciò pregiudicare la liceità del trattamento basata sul consenso prestato prima della revoca;
- alla portabilità dei dati.

Se desidera esercitare qualsiasi dei suoi diritti, puoi rivolgerti al Titolare del trattamento.

Il candidato, prendendo visione di questo documento

#### **AUTORIZZA**

il Politecnico di Torino, ai sensi degli artt. 96 e 97 della Legge in materia di protezione del diritto d'autore e di altri diritti connessi al suo esercizio n. 633 del 22 aprile 1941 nonché dell'art. 10 codice civile, a:

- esercitare i diritti previsti dagli artt. 12 e ss. della Legge n. 633/1941;

#### **DICHIARA**

- di aver letto e accettato i termini e le condizioni di trattamento dei dati personali con le modalità e per le finalità indicate nell'Informativa resa ai sensi del Regolamento Generale sulla protezione dei dati (Regolamento EU 2016/679 – noto anche come “GDPR”);
- di aver autorizzato la riproduzione e la trasmissione dei contenuti video, nei termini ed alle condizioni di seguito indicati per le finalità istituzionali;
- di assumere la piena responsabilità delle dichiarazioni rese, sollevando l'Università da qualsiasi pretesa e/o azione di terzi;
- di essere informata/o e consapevole del fatto che per alcune trasmissioni è ammesso il download;
- di rinunciare a qualunque corrispettivo per l'utilizzo, la riproduzione e la diffusione delle immagini.

Il/la sottoscritto/a vieta altresì l'uso delle immagini in contesti che ne pregiudicano la dignità personale ed il decoro.

*RECLAMO*

Ha il diritto di rivolgersi al Garante per la protezione dei dati personali secondo le modalità indicate al seguente link: <https://www.garanteprivacy.it/web/guest/home/docweb/-/docwebdisplay/docweb/4535524>

# Appendice B

## Metodo di valutazione

	RPOCEDURA TAKE OFF	T.O. CLEARANCE	ECAM INSP.	AUTO BRK MAX	SPOILER ARMED	FLAP 1	T.O. CONFIG	SET ALT	PARK BRK OFF	STICK FRWD	THROTTLE 1_4
0	BENE AL MOMENTO GIUSTO										
-1	STEP DIMENTICATO										
-0.5	GUARDARE POSTO SBAGLIATO										
-0.5	NON ENUNCIARE										
-0.5	PULSANTE SBAGLIATO ADIACENTE										
-0.8	PULSANTE SBAGLIATO LONTANO										
-0.5	POSIZIONE SBAGLIATA										
-0.8	LEVA SBAGLIATA										
-0.5	VALORE SETTATO ERRATO										
-0.5	MOMENTO/VALORI ERRATI										
-0.3	ERR ORDINE ADIACENTE A TERRA										
-0.5	ERR ORDINE ADIACENTE IN VOLO										
-1	ERR ORDINE CONCETTUALE										

	RPOCEDURA TAKE OFF	N1 CHECK	THROTTLE CLB	STICK NEUTRAL	STICK BACKWARD	LANDING GEAR RET	THROTTLE CLB	AP1	FLAP 0	SPOILER DISARMED
0	BENE AL MOMENTO GIUSTO									
-1	STEP DIMENTICATO									
-0.5	GUARDARE NEL POSTO SBAGLIATO									
-0.5	NON ENUNCIARE									
-0.5	PULSANTE SBAGLIATO ADIACENTE									
-0.8	PULSANTE SBAGLIATO LONTANO									
-0.5	POSIZIONE SBAGLIATA									
-0.8	LEVA SBAGLIATA									
-0.5	VALORE SETTATO ERRATO									
-0.5	MOMENTO/VALORI ERRATI									
-0.3	ERRORE DI ORDINE ADIACENTE A TERRA									
-0.5	ERRORE DI ORDINE ADIACENTE IN VOLO									
-1	ERRORE DI ORDINE CONCETTUALE									

Figura B.1: Metodo di valutazione

# Bibliografia

- [1] «Simulazioni VR nel settore aerospaziale». In: URL: <https://www.vrmp.it/tech/simulazioni-vr-nel-settore-aerospaziale/> (cit. a p. 1).
- [2] «Voli più sicuri grazie alla realtà virtuale». In: URL: <https://cordis.europa.eu/article/id/86614-safer-flights-thanks-to-virtual-reality/it#:~:text=> (cit. a p. 1).
- [3] Frederic Vianale. *Analisi di procedure operative per l'addestramento del personale aeronautico in ambiente di realtà virtuale*. 2020 (cit. alle pp. 2, 42).
- [4] Lorenzo Fava. *Digitalizzazione di procedure operative per l'addestramento del personale aeronautico in ambiente di realtà virtuale*. 2020 (cit. alle pp. 2, 42).
- [5] Giuseppe Brizzi. *Digitalizzazione e analisi dell'addestramento del personale aeronautico attraverso gli strumenti della realtà virtuale*. 2020 (cit. alle pp. 2, 42, 90).
- [6] Andrea Pavone. *Digitalizzazione e analisi di procedure operative aeronautiche per l'addestramento di piloti in ambiente di realtà virtuale*. 2020 (cit. alle pp. 2, 42, 90).
- [7] Giovanni Sgarra. *Integrazione del software Pacelab WEAVR con il simulatore di volo X-Plane per il training dei piloti in ambiente di realtà virtuale*. 2021 (cit. alle pp. 2, 42, 60).
- [8] Francesco Salvatore Tartaglia. *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*. 2021 (cit. alle pp. 2, 42, 60).
- [9] Beniamino Barberis. *Development of Virtual Reality Pilot Training for A320 Standard Operating Procedures in Unity/Pacelab WEAVR*. 2023 (cit. alle pp. 2, 30, 42, 43).
- [10] «TXT Group». In: URL: <https://www.txtgroup.com/it/> (cit. a p. 2).
- [11] «Il metaverso tra realtà digitale e aumentata: innovazione tecnologica e catena del valore». In: URL: [https://www.researchgate.net/publication/358667569\\_Il\\_metaverso\\_tra\\_realta\\_digitale\\_e\\_aumentata\\_innovazione\\_tecnologica\\_e\\_catena\\_del\\_valore](https://www.researchgate.net/publication/358667569_Il_metaverso_tra_realta_digitale_e_aumentata_innovazione_tecnologica_e_catena_del_valore) (cit. a p. 5).
- [12] «Realtà virtuale, aumentata e mista: differenze e applicazioni». In: URL: <https://www.internet4things.it/industry-4-0/realta-virtuale-aumentata-e-mista-differenze-e-applicazioni/> (cit. a p. 5).

- [13] «Realtà mista: come espandere digitalmente la dimensione fisica per aumentare la produttività». In: URL: <https://www.digital4.biz/executive/realta-mista-come-espandere-digitalmente-la-dimensione-fisica-per-aumentare-la-produttivita/> (cit. a p. 8).
- [14] Chengyi Zeng, Xu Chen, Zhibin Ge, Yuming Wang, Weiwei Cao e Chengqian Zhang. «An exploration of final flight training based on virtual reality technology». In: *2023 International Conference on Computer Applications Technology (CCAT)*. 2023, pp. 287–290. DOI: [10.1109/CCAT59108.2023.00060](https://doi.org/10.1109/CCAT59108.2023.00060) (cit. alle pp. 9, 14).
- [15] Elizabeth Pennington, Riley Hafer, Erin Nistler, Todd Seech e Chad Tossell. «Integration of Advanced Technology in Initial Flight Training». In: *2019 Systems and Information Engineering Design Symposium (SIEDS)*. 2019, pp. 1–5. DOI: [10.1109/SIEDS.2019.8735628](https://doi.org/10.1109/SIEDS.2019.8735628) (cit. a p. 11).
- [16] «Pilot Training Next begins third iteration January 2020». In: URL: <https://www.jbsa.mil/News/News/Article/2047294/pilot-training-next-begins-third-iteration-january-2020/> (cit. a p. 11).
- [17] S.M. Casner, Stephen M Casner, Richard W Geven e Kent T Williams. «The Effectiveness of Airline Pilot Training for Abnormal Events». In: *Human factors : the journal of the Human Factors and Ergonomics Society*. 55.3 (2013-06). ISSN: 0018-7208 (cit. a p. 11).
- [18] Irvin Steve Cardenas, Chaisay Nicholas Letdara, Brian Selle e Jong-Hoon Kim. «ImmersiFLY: Next Generation of Immersive Pilot Training». In: *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2017, pp. 1203–1206. DOI: [10.1109/CSCI.2017.212](https://doi.org/10.1109/CSCI.2017.212) (cit. a p. 12).
- [19] R. Andorno M. Ienca. «Towards new human rights in the age of neuroscience and neurotechnology». In: *Life Sci Soc Policy* 13.5 (2017). URL: <https://www.doi.org/10.1186/s40504-017-0050-1> (cit. a p. 15).
- [20] Ying-hui Chou, Carol P. Weingarten, David J. Madden, Allen W. Song e Nan-kuei Chen. «Applications of Virtual Reality Technology in Brain Imaging Studies». In: *Virtual Reality in Psychological, Medical and Pedagogical Applications*. A cura di Christiane Eichenberg. Rijeka: IntechOpen, 2012. Cap. 10. DOI: [10.5772/48445](https://doi.org/10.5772/48445). URL: <https://doi.org/10.5772/48445> (cit. a p. 15).
- [21] Angelica M. Tinga, Tycho T. de Back e Max M. Louwerse. «Non-invasive neurophysiological measures of learning: A meta-analysis». In: *Neuroscience Biobehavioral Reviews* 99 (2019), pp. 59–89. ISSN: 0149-7634. DOI: <https://doi.org/10.1016/j.neubiorev.2019.02.001>. URL: <https://www.sciencedirect.com/science/article/pii/S0149763418304202> (cit. a p. 15).
- [22] Evy van Weelden, Maryam Alimardani, Travis J. Wiltshire e Max M. Louwerse. «Advancing the Adoption of Virtual Reality and Neurotechnology to Improve Flight Training». In: *2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS)*. 2021, pp. 1–4. DOI: [10.1109/ICHMS53169.2021.9582658](https://doi.org/10.1109/ICHMS53169.2021.9582658) (cit. a p. 15).



- [23] Benjamin Bell, Elaine Kelsey, Benjamin Nye e Winston (“Wink”) Bennett. «Adapting Instruction by Measuring Engagement with Machine Learning in Virtual Reality Training». In: *Adaptive Instructional Systems*. A cura di Robert A. Sottolare e Jessica Schwarz. Cham: Springer International Publishing, 2020, pp. 271–282. ISBN: 978-3-030-50788-6 (cit. a p. 16).
- [24] Hongchao Wang, Dan Huang e Hongli Sun. «Application of Mixed Reality Object Recognition Technology in Aircraft Maintenance Manual». In: *2022 12th International Conference on Information Technology in Medicine and Education (ITME)*. 2022, pp. 618–622. DOI: [10.1109/ITME56794.2022.00132](https://doi.org/10.1109/ITME56794.2022.00132) (cit. a p. 17).
- [25] Paul Milgram e Fumio Kishino. «A Taxonomy of Mixed Reality Visual Displays». In: *IEICE Trans. Information Systems* vol. E77-D, no. 12 (dic. 1994), pp. 1321–1329 (cit. a p. 17).
- [26] Safinaz Mustapha, Cyprian Aidan Chong e M. N. Mohammed. «Review on the Usage of Mixed Reality and Augmented Reality Assisted Learning Tool in Aircraft Maintenance». In: *2021 IEEE 9th Conference on Systems, Process and Control (ICSPC 2021)*. 2021, pp. 168–173. DOI: [10.1109/ICSPC53359.2021.9689118](https://doi.org/10.1109/ICSPC53359.2021.9689118) (cit. a p. 17).
- [27] T. Haritos e N.D. Macchiarella. «A mobile application of augmented reality for aerospace maintenance training». In: *24th Digital Avionics Systems Conference*. Vol. 1. 2005, 5.B.3–5.1. DOI: [10.1109/DASC.2005.1563376](https://doi.org/10.1109/DASC.2005.1563376) (cit. a p. 17).
- [28] U. Neumann e A. Majoros. «Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance». In: *Proceedings. IEEE 1998 Virtual Reality Annual International Symposium (Cat. No.98CB36180)*. 1998, pp. 4–11. DOI: [10.1109/VRAIS.1998.658416](https://doi.org/10.1109/VRAIS.1998.658416) (cit. a p. 17).
- [29] «Augmented Reality on Aircraft Maintenance». In: URL: <https://www.youtube.com/watch?v=f4EQAm02jas> (cit. a p. 18).
- [30] «MRO.AIR - Artificial Intelligent Reality». In: URL: <https://www.youtube.com/watch?v=MYddQcSEXbQ> (cit. a p. 18).
- [31] «AR/VR Augmented Operations and Maintenance». In: URL: [https://www.youtube.com/watch?v=1A1gQyebX\\_0](https://www.youtube.com/watch?v=1A1gQyebX_0) (cit. a p. 19).
- [32] «Airbus drives innovation and accelerates production with Azure mixed reality and HoloLens 2». In: URL: <https://www.youtube.com/watch?v=lxjC4Z05qh8> (cit. a p. 19).
- [33] Pompeo Piedimonte e Silvia Liberata Ullo. «Applicability of the Mixed Reality to Maintenance and Training Processes of C4I Systems in Italian Air Force». In: *2018 5th IEEE International Workshop on Metrology for AeroSpace (MetroAeroSpace)*. 2018, pp. 559–564. DOI: [10.1109/MetroAeroSpace.2018.8453612](https://doi.org/10.1109/MetroAeroSpace.2018.8453612) (cit. alle pp. 19, 20).
- [34] Mathilde R. Desselle, Ross A. Brown, Allan R. James, Mark J. Midwinter, Sean K. Powell e Maria A. Woodruff. «Augmented and Virtual Reality in Surgery». In: *Computing in Science Engineering* 22.3 (2020), pp. 18–26. DOI: [10.1109/MCSE.2020.2972822](https://doi.org/10.1109/MCSE.2020.2972822) (cit. a p. 22).
- [35] Gretchen Purcell Jackson e John L Tarpley. «How long does it take to train a surgeon?». In: *Bmj* 339 (2009) (cit. a p. 23).

- [36] John Seely Brown, Allan Collins e Paul Duguid. «Situated cognition and the culture of learning». In: *1989* 18.1 (1989), pp. 32–42 (cit. a p. 23).
- [37] Christian Hansen, Jan Wieferich, Felix Ritter, Christian Rieder e Heinz-Otto Peitgen. «Illustrative visualization of 3D planning models for augmented reality in liver surgery». In: *International journal of computer assisted radiology and surgery* 5 (2010), pp. 133–141 (cit. a p. 24).
- [38] Philip Pratt, Matthew Ives, Graham Lawton, Jonathan Simmons, Nasko Radev, Liana Spyropoulou e Dimitri Amiras. «Through the HoloLens™ looking glass: augmented reality for extremity reconstruction surgery using 3D vascular models with perforating vessels». In: *European radiology experimental* 2 (2018), pp. 1–7 (cit. a p. 24).
- [39] Alexandre Costa Henriques e Ingrid Winkler. «Automotive Marketing Research and Virtual Reality: A Systematic Literature Review». In: *2020 22nd Symposium on Virtual and Augmented Reality (SVR)*. 2020, pp. 339–345. DOI: [10.1109/SVR51698.2020.00057](https://doi.org/10.1109/SVR51698.2020.00057) (cit. a p. 25).
- [40] Antonino Gomes de Sá e Gabriel Zachmann. «Virtual reality as a tool for verification of assembly and maintenance processes». In: *Computers Graphics* 23.3 (1999), pp. 389–403. ISSN: 0097-8493. DOI: [https://doi.org/10.1016/S0097-8493\(99\)00047-3](https://doi.org/10.1016/S0097-8493(99)00047-3). URL: <https://www.sciencedirect.com/science/article/pii/S0097849399000473> (cit. a p. 25).
- [41] V.D. Lehner e T.A. DeFanti. «Distributed virtual reality: supporting remote collaboration in vehicle design». In: *IEEE Computer Graphics and Applications* 17.2 (1997), pp. 13–17. DOI: [10.1109/38.574654](https://doi.org/10.1109/38.574654) (cit. a p. 25).
- [42] Michele Fiorentino, Raffaele de Amicis, Giuseppe Monno e Andre Stork. «Spacedesign: A mixed reality workspace for aesthetic industrial design». In: *Proceedings. International Symposium on Mixed and Augmented Reality*. IEEE. 2002, pp. 86–318 (cit. a p. 26).
- [43] Wing Bun Lee, Chi Fai Cheung e JG Li. «Applications of virtual manufacturing in materials processing». In: *Journal of materials processing technology* 113.1-3 (2001), pp. 416–423 (cit. a p. 26).
- [44] Shiguang Qiu, Xiumin Fan, Dianliang Wu, Qichang He e Deji Zhou. «Virtual human modeling for interactive assembly and disassembly operation in virtual reality environment». In: *The International Journal of Advanced Manufacturing Technology* 69 (2013), pp. 2355–2372 (cit. a p. 26).
- [45] Simone Borsci, Glyn Lawson e Simon Broome. «Empirical evidence, evaluation criteria and challenges for the effectiveness of virtual and mixed reality tools for training operators of car service maintenance». In: *Computers in Industry* 67 (2015), pp. 17–26 (cit. a p. 26).
- [46] Lakshmi Jayendran e G Rejikumar. «A study on the potential benefits of applying virtual reality in travel and tourism for better destination image». In: *International Journal of Pure and Applied Mathematics* 118.5 (2018), pp. 669–672 (cit. a p. 26).

- [47] Timothy Jung, M Claudia tom Dieck, Natasha Moorhouse e Dario tom Dieck. «Tourists' experience of Virtual Reality applications». In: *2017 IEEE International Conference on Consumer Electronics (ICCE)*. IEEE. 2017, pp. 208–210 (cit. a p. 26).
- [48] Ade Bastian, Tri Ferga Prasetyo e Nanda Juanda Dipura Atmaja. «An Application of Virtual Reality with Android Device for Tourism Assistant». In: *2019 2nd International Conference of Computer and Informatics Engineering (IC2IE)*. 2019, pp. 1–5. DOI: [10.1109/IC2IE47452.2019.8940861](https://doi.org/10.1109/IC2IE47452.2019.8940861) (cit. a p. 26).
- [49] «SteamVR». In: URL: <https://store.steampowered.com/steamvr?l=italian> (cit. a p. 28).
- [50] «Unity». In: URL: <https://unity.com/> (cit. a p. 29).
- [51] *PACELAB WEAVR - USER MANUAL* (cit. a p. 31).
- [52] «WEAVR Simulation Hub». In: URL: <https://help.pace.de/xr/weavr/2.0.0/advanced-features/weavr-simulation-hub/index.html> (cit. a p. 35).
- [53] «Visual Studio». In: URL: <https://visualstudio.microsoft.com/it/#vs-section> (cit. a p. 36).
- [54] «X-Plane». In: URL: <https://www.x-plane.com/> (cit. a p. 36).
- [55] «X-Plane Connect». In: URL: <https://software.nasa.gov/software/ARC-17185-1> (cit. a p. 38).
- [56] «HTC Vive Pro». In: URL: <https://www.vive.com/eu/> (cit. a p. 38).
- [57] «What is a Standard Operating Procedure (SOP)?» In: URL: <https://www.flightdeckfriend.com/ask-a-pilot/what-is-an-sop/#:~:text=SOP%20is%20an%20abbreviation%20for,commercial%20aircraft%20safely%20and%20consistently.> (cit. a p. 42).
- [58] «FlyByWire Simulations Documentation». In: URL: <https://docs.flybywiresim.com/pilots-corner/SOP/> (cit. a p. 42).
- [59] «What is CBT in Aviation? (Computer-Based Training)». In: URL: <https://termaviation.com/what-is-cbt-in-aviation/> (cit. a p. 89).
- [60] Fabio Buttussi e Luca Chittaro. «A Comparison of Procedural Safety Training in Three Conditions: Virtual Reality Headset, Smartphone, and Printed Materials». In: *IEEE Transactions on Learning Technologies* 14.1 (2021), pp. 1–15. DOI: [10.1109/TLT.2020.3033766](https://doi.org/10.1109/TLT.2020.3033766) (cit. a p. 90).
- [61] «Rally». In: URL: <https://rally.co/it> (cit. a p. 91).
- [62] «NASA Task Load Index». In: URL: <https://humansystems.arc.nasa.gov/groups/tlx/> (cit. a p. 94).