



**Politecnico
di Torino**

POLITECNICO DI TORINO

Corso di Laurea Magistrale in
Ingegneria del Cinema e dei Mezzi di Comunicazione

Tesi di Laurea Magistrale

3D storytelling in Realtà Virtuale

Relatori

Andrea Sanna

Federico Manuri

Candidata

Valentina Carrescia

Anno Accademico 2022/2023

Sessione di Laurea Dicembre 2023

Sommario

La seguente tesi si propone di documentare lo sviluppo di un'applicazione basata sulla Realtà Virtuale per la creazione di storyboard.

Il testo inizia definendo i concetti chiave di storyboard e Realtà Virtuale, approfondendo quest'ultimo anche dal punto di vista della sua evoluzione storica.

Gli storyboard sono rappresentazioni visive utilizzate nella pre-produzione di film e progetti multimediali. Aiutano il regista, lo scenografo e altri professionisti a pianificare le riprese attraverso sequenze di disegni dettagliati. Questi disegni sono spesso accompagnati da informazioni come descrizioni, dialoghi, durata, ecc., e possono essere realizzati manualmente o digitalmente. L'uso degli storyboard risale alle prime fasi del cinema, con figure come George Méliès e Walt Disney che hanno contribuito al loro sviluppo.

La Realtà Virtuale (VR) può essere vista in due modi: in senso filosofico, come una fusione di elementi magici e reali, o da un punto di vista tecnico, come un'interfaccia uomo-computer che genera un ambiente realistico e interattivo. L'immersione rappresenta il coinvolgimento nel mondo virtuale mentre la presenza è la sensazione di appartenenza ad esso.

Per una VR efficace sono indispensabili aspetti come un ampio campo visivo, una rappresentazione realistica e feedback dati a diversi sensi. I dispositivi di tracciamento, noti come tracker, sono essenziali per monitorare i movimenti. Inoltre, il suono spazializzato riveste un ruolo importante nell'esperienza immersiva.

Nel 1929 Edwin A. Link inventò un simulatore per far provare ai passeggeri la sensazione del volo. Nel 1956 Morton Heilig inventò il Sensorama, un dispositivo che emulava l'esperienza di guida in motocicletta, inclusi odori e vento.

Nel 1965 il Dr. Sutherland teorizzò la "The Ultimate Display," che promuoveva

l'immersione e l'interazione in ambienti virtuali controllati dai computer. Nel 1973 venne creato il primo acceleratore grafico per immagini di sintesi. Nel 1978 il "Aspen Movie Map" preannunciò i sistemi di navigazione virtuale. Nel 1979 Eric Howlett sviluppò il LEEP, un piccolo display in grado di offrire un'ampia esperienza di visione. La VR è cresciuta dopo l'introduzione dei computer nelle case, GPU migliori, dispositivi di interazione economici e software.

Negli ultimi anni, l'interesse è rinato con dispositivi come Oculus Rift, HTC Vive e PlayStation VR, e modelli più recenti, come l'HTC VIVE XR Elite e il Meta Quest 3.

La tesi passa poi ad analizzare lo stato dell'arte sull'uso della Realtà Virtuale per creazione di elementi di pre-visualizzazione. Diversi strumenti sono stati esaminati, sia che siano stati descritti in articoli di ricerca sia che siano applicazioni commerciali. Questi strumenti consentono agli utenti di creare storyboard, mettere in posa personaggi, controllare le impostazioni della camera e altro ancora, il tutto indossando il visore. Gli utenti e i professionisti del settore hanno valutato positivamente l'utilità della VR nel processo di pre-produzione cinematografica.

Nella fase che ha preceduto lo sviluppo dell'applicazione sono stati definiti i requisiti basati sull'analisi dello stato dell'arte, su un'applicazione precedentemente sviluppata in versione desktop e su un questionario rivolto ad alcuni studenti del Politecnico di Torino. Il metodo MoSCoW (Must, Should, Could, Would) è stato utilizzato per classificare i requisiti in base alla loro importanza.

I risultati del questionario hanno portato a identificare i requisiti fondamentali, come l'inserimento di oggetti e personaggi nella scena, la possibilità di visualizzare lo storyboard completato e la gestione delle camere virtuali. Altri requisiti importanti includevano la gestione delle azioni dei personaggi e la generazione automatica di descrizioni delle vignette dello storyboard.

I requisiti sono stati successivamente classificati in base alla loro priorità e alla

complessità associata. La classifica finale ha guidato lo sviluppo dell'applicazione VR.

Diverse tecnologie sono state utilizzate. Tra queste, l'Oculus Rift, Unity (motore grafico multiplatforma), Microsoft Visual Studio come ambiente di sviluppo e il Mixed Reality Toolkit (MRTK), strumento Microsoft nato per facilitare lo sviluppo di applicazioni per Realtà Virtuale, Mista e Aumentata. Gli asset provengono da varie fonti, tra cui siti web di modelli 3D. Mixamo è utilizzato per personaggi e animazioni. Infine, l'API OpenAI è utilizzata per riformulare il testo nelle descrizioni delle vignette dello storyboard.

L'applicazione sviluppata si compone di due parti fondamentali. Nella prima sezione l'utente può inserire personaggi e oggetti su un tavolo virtuale e rinominarli, oltre a poter gestire le luci e decidere l'atmosfera. La seconda parte riguarda la fase di storyboard dove l'utente è libero di assegnare delle azioni e animazioni predefinite ai personaggi presenti sul tavolo, decidere su quale vignetta lavorare, settandone la durata, e posizionare le camere, decidendone la lunghezza focale per creare gli screenshot che rappresenteranno le vignette. Ogni vignetta dello storyboard finale, presentato all'utente sottoforma di documento HTML, conterrà quindi le informazioni di durata, lunghezza focale della camera oltre ad una descrizione, generata automaticamente durante l'utilizzo dell'applicazione. Ogni sezione di testo delle vignette del documento è modificabile dall'utente.

L'applicazione, prima di farla provare ad un gruppo di utenti, è stata fatta analizzare da tre esperti di dominio, che stanno svolgendo un dottorato o post-doc al Politecnico di Torino. È stato chiesto loro di valutare l'applicazione dando un punteggio da 1 a 5 per ogni Euristiche di Nielsen. L'applicazione ha ricevuto in media un punteggio generale di 4,47/5. Gli esperti hanno inoltre consigliato modifiche che sono state effettuate prima di proporre l'applicativo agli utenti per i test finali.

Gli utenti finali, un gruppo misto di studenti del Politecnico di Torino, hanno valutato molto positivamente l'applicazione. Dai questionari a loro somministrati,

il SUS per l'usabilità e il NASA-TLX per il carico di lavoro, sono emersi riscontri positivi. In particolare, l'applicazione ha ottenuto, in media, un punteggio di 85,31/100, eccellente, per il SUS e un valore di 23,37/100, carico di lavoro medio, per il NASA-TLX.

L'applicazione offre quindi un buon punto di partenza per essere ampliata in futuro aggiungendo ulteriori funzionalità a quelle già presenti, che a loro volta possono essere ancora modificate basandosi sui commenti degli utenti.

Indice

Sommario.....	I
Acronimi	VIII
Indice delle figure	XI
Indice delle tabelle	XIV
Indice dei grafici	XV
1. Introduzione.....	1
1.1. Storyboard.....	1
1.2. Realtà Virtuale	4
1.3. Realtà Virtuale: ieri e oggi	10
2. Stato dell'Arte	24
2.1. Empowering Creative People: Virtual Reality for Previsualization - articolo del 2018	24
2.2. VR as a Content Creation Tool for Movie Previsualisation - articolo del 2019	25
2.3. A Collaborative Previsualization Tool for Filmmaking in Virtual Reality - articolo del 2019	29
2.4. A Comparison of Immersive and Non Immersive VR for the Education of Filmmaking - articolo del 2022	32
2.5. Tvorì – Storyboard	34
2.6. Storyboarder.....	36
2.7. Gravity Sketch VR	37
2.8. Confronto delle caratteristiche dei software analizzati	39

3.	Requisiti.....	41
3.1.	Applicazione di partenza.....	41
3.2.	Metodo MoSCoW	43
3.3.	Questionario	43
3.4.	Analisi risultati questionario MoSCoW	58
4.	Tecnologie utilizzate	63
4.1.	Oculus Rift CV1.....	63
4.2.	Unity.....	65
4.3.	Microsoft Visual Studio	67
4.4.	Linguaggi di programmazione e formati.....	67
4.5.	MRTK	69
4.6.	Oculus Integration SDK.....	70
4.7.	Blender	70
4.8.	Asset.....	72
4.9.	Mixamo	72
4.10.	TextMeshPro	73
4.11.	API OpenAI	73
5.	Sviluppo dell'applicazione	74
5.1.	Pipeline di funzionamento.....	74
5.2.	Movimento nell'ambiente	75
5.3.	Menù iniziale.....	77
5.4.	Creazione ambiente	78
5.5.	Luci	80
5.6.	Simulazione.....	82
5.7.	Camere	89
5.8.	Controller	91

5.9. Interfaccia vocale	94
5.10. Feedback visivi e uditivi	94
5.11. Generazione storyboard	95
5.12. Palette colori	98
6. Test e analisi dei risultati	99
6.1. Primo caso d'uso	99
6.2. Secondo caso d'uso	101
6.3. Test con gli esperti	101
6.4. Test Utenti	105
6.5. Tempi ed errori - Utenti	106
6.6. Questionario SUS e domande generali	107
6.7. Questionario NASA-TLX	111
7. Conclusioni	115
7.1. Sviluppi futuri	115
Appendice	118
A.1 Questionario MoSCoW	118
A.2 Questionario SUS e domande generali	129
Bibliografia e sitografia	134

Acronimi

2D Two-dimensional

3D Three-dimensional

AI Artificial Intelligence

AMOLED Active-Matrix Organic Light-Emitting Diode

API Application Programming Interface

AR Augmented Reality

AV Augmented Virtuality

CG Computer Graphics

CSS Cascading Style Sheets

CSV Comma-Separated Values

DoF Degree Of Freedom

DRAM Dynamic Random Access Memory

EEVEE Extra Easy Virtual Environment Engine

FBX FilmBoX

FOV Field Of View

FPS Frame Per Second

GB GigaByte

GPU Graphic Processing Unit

HDMI High-Definition Multimedia Interface

HDRI High Dynamic Range Image

HMD Head Mounted Display

HTML HyperText Markup Language

IK Inverse Kinematics

IR Infrared Radiation

IVR Immersive Virtual Reality

JSON JavaScript Object Notation

LCD Liquid Crystal Display

MR Mixed Reality

MRTK Mixed Reality Toolkit

NIVR Non-Immersive Virtual Reality

OLED Organic Light-Emitting Diode

PBR Physically Based Rendering

PC Personal Computer

PPD Pixels per Degree

PPI Pixels per Inch

RGB Red Green Blue

ROM Read Only Memory

SDK Software Development Kit

SFX Sound Effects

UI User Interface

UI User Interface

USB Universal Serial Bus

VFX Visual Effects

VR Virtual Reality

XR Extended Reality

Indice delle figure

Figura 1.1. Esempio di una vignetta dello storyboard ufficiale del video musicale “willow” di Taylor Swift confrontata con l’inquadratura poi effettivamente realizzata. Storyboard di Vincent Lucido [2].	2
Figura 1.2. Prime 4 vignette di uno storyboard accompagnate a destra da informazioni aggiuntive.	3
Figura 1.3 Rappresentazione semplificata del VR Continuum.	5
Figura 1.4 Differenza tra 3-DoF (rotazione) e i 6-Dof (rotazione e traslazione) con in caschetto di realtà virtuale.	7
Figura 1.5 Il suono spazializzato nella VR.	9
Figura 1.6. Sensorama, il primo sistema di immersione virtuale, la tavola tecnica e le immagini mostrate durante la presentazione del prodotto sperimentale [10].	10
Figura 1.7 Il primo HMD, inventato da Ivan Sutherland [11].	11
Figura 1.8 Boom [12].	12
Figura 1.9 Eric Howlett, l’inventore del LEEP.	13
Figura 1.10 Nintendo Mattel Power Glove, modello americano.	14
Figura 1.11 Headset (caschetto) HTC Vive.	16
Figura 1.12 Controller HTC Vive e Vive Base Station.	16
Figura 1.13 Headset PlayStation VR.	17
Figura 1.14 PlayStation Camera per “Outside looking in”.	17
Figura 1.15 Controller di movimento PlayStation Move.	17
Figura 1.16 Visore Meta Quest Pro e controller.	19
Figura 1.17 Visore Meta Quest 3 e Controller Touch Plus [25].	19
Figura 1.18 HTC VIVE Cosmos XR e controller.	20
Figura 1.19 HTC VIVE Pro 2, Base Station e controller.	21
Figura 1.20 HTC VIVE Flow e controller da 3DoF.	21
Figura 1.21 HTC VIVE Focus 3 e controller.	22
Figura 1.22 HTC VIVE XR Elite e controller.	22

Figura 1.23 PlayStation VR2 e controller.....	23
Figura 2.1 La camera virtuale gialla del primo prototipo.	25
Figura 2.2 Le tre modalità presenti nell'applicativo.....	26
Figura 2.3 Raycast, Grab degli oggetti, menù sul controller, menù come 3D UI.	27
Figura 2.4 Modello in scala di un personaggio e camera.	30
Figura 2.5 Menù e tasti dei controller.....	30
Figura 2.6 “Shooting mode” nella versione IVR.....	33
Figura 2.7 Funzioni dei tasti sui controller in base alla modalità corrente.	34
Figura 2.8 Funzionamento di Tvorì per la creazione di storyboard [41].....	35
Figura 2.9 Possibile visuale mentre si utilizza l'applicazione in VR [44].....	36
Figura 2.10 Sezione dell'applicazione in versione Desktop dove poter aggiungere le informazioni aggiuntive alle vignette ottenute in VR [44].	37
Figura 2.11 Doug Fahl mostra come poter sfruttare Gravity Sketch per realizzare uno storyboard [47].....	38
Figura 3.1 Interfaccia grafica dell'applicazione per Desktop nella fase di costruzione della scena.	42
Figura 3.2 Interfaccia grafica dell'applicazione per Desktop nel momento in cui si vuole far eseguire un'azione ad un personaggio.....	42
Figura 4.1 Headset Oculus Rift CV1	63
Figura 4.2 Sensore Oculus Rift Constellation	64
Figura 4.3 Controller Oculus Touch.....	64
Figura 4.4 Tasti controller Oculus Touch [52].	65
Figura 5.1 Pipeline funzionamento applicazione.....	75
Figura 5.2 Alcune delle azioni disponibili interagendo con i tasti del controller sinistro.....	76
Figura 5.3 Alcune delle azioni disponibili interagendo con i tasti del controller destro.....	76
Figura 5.4 Menù presente all'avvio dell'applicazione.....	77
Figura 5.5 Visuale che si può presentare all'utente dopo aver premuto “ <i>Create New Scene</i> ”.....	78

Figura 5.6 Oggetti posizionati sul piano, sopra l'oggetto selezionato appare il nome e il tasto " <i>Rename</i> " per poterlo rinominare. A destra è visibile il menù contenente gli oggetti.....	79
Figura 5.7 Menù principale, sempre presente durante l'utilizzo dell'applicazione.	80
Figura 5.8 Luce con parametri modificabili.	81
Figura 5.9 Oggetto sole che comanda la posizione del sole nell'HDRI.	81
Figura 5.10 Possibile visuale dell'utente dopo aver premuto " <i>Start Storyboarding</i> ".	82
Figura 5.11 Storyboard Panels Menù.	83
Figura 5.12 Particelle colorate che evidenziano gli oggetti selezionati.....	84
Figura 5.13 Lista azioni, nome, tasti " <i>MOVE</i> ", " <i>STOP</i> " (o " <i>PLAY</i> ") riferiti al personaggio selezionato.	85
Figura 5.14 Animator del personaggio di tipo <i>man</i>	88
Figura 5.15 Tasto " <i>Get Control</i> ".	89
Figura 5.16 Camera e screenshot eseguiti.	90
Figura 5.17 Funzionalità dei tasti dei controller nella fase di creazione dell'ambiente.	93
Figura 5.18 Funzionalità dei tasti dei controller nella fase di simulazione.	93
Figura 5.19 Tastiera e tasto microfono.	94
Figura 5.20 Esempio di storyboard finale costituito da due vignette.	97
Figura 5.21 Colori utilizzati nell'applicazione per gli elementi della UI.	98
Figura 6.1 Storyboard fornito per il primo caso d'uso.	100

Indice delle tabelle

Tabella 2.1 Confronto caratteristiche dei software e strumenti analizzati.....	39
Tabella 6.1 Punteggio dato da ogni esperto all'applicazione usando le Euristiche di Nielsen, in una scala da 1 a 5.....	103
Tabella 6.2 Tabella per interpretare il valore SUS ottenuto [78].....	108
Tabella 6.3 Interpretazione risultati NASA-TLX.....	112

Indice dei grafici

Grafico 3.1 Risposte in % alla domanda 1 del questionario MoSCoW.....	45
Grafico 3.2 Risposte in % alla domanda 2 del questionario MoSCoW.....	45
Grafico 3.3 Risposte in % alla domanda 3 del questionario MoSCoW.....	45
Grafico 3.4 Risposte in % alla domanda 4 del questionario MoSCoW.....	46
Grafico 3.5 Risposte in % alla domanda 5 del questionario MoSCoW.....	46
Grafico 3.6 Risposte in % alla domanda 6 del questionario MoSCoW.....	46
Grafico 3.7 Risposte in % alla domanda 7 del questionario MoSCoW.....	47
Grafico 3.8 Risposte in % alla domanda 8 del questionario MoSCoW.....	47
Grafico 3.9 Risposte in % alla domanda 9 del questionario MoSCoW.....	47
Grafico 3.10 Risposte in % alla domanda 10 del questionario MoSCoW.....	48
Grafico 3.11 Risposte in % alla domanda 11 del questionario MoSCoW.....	48
Grafico 3.12 Risposte in % alla domanda 12 del questionario MoSCoW.....	48
Grafico 3.13 Risposte in % alla domanda 13 del questionario MoSCoW.....	49
Grafico 3.14 Risposte in % alla domanda 14 del questionario MoSCoW.....	49
Grafico 3.15 Risposte in % alla domanda 15 del questionario MoSCoW.....	49
Grafico 3.16 Risposte in % alla domanda 16 del questionario MoSCoW.....	50
Grafico 3.17 Risposte in % alla domanda 17 del questionario MoSCoW.....	50
Grafico 3.18 Risposte in % alla domanda 18 del questionario MoSCoW.....	50
Grafico 3.19 Risposte in % alla domanda 19 del questionario MoSCoW.....	51
Grafico 3.20 Risposte in % alla domanda 20 del questionario MoSCoW.....	51
Grafico 3.21 Risposte in % alla domanda 21 del questionario MoSCoW.....	51
Grafico 3.22 Risposte in % alla domanda 22 del questionario MoSCoW.....	52
Grafico 3.23 Risposte in % alla domanda 23 del questionario MoSCoW.....	52
Grafico 3.24 Risposte in % alla domanda 24 del questionario MoSCoW.....	52
Grafico 3.25 Risposte in % alla domanda 25 del questionario MoSCoW.....	53
Grafico 3.26 Risposte in % alla domanda 26 del questionario MoSCoW.....	53
Grafico 3.27 Risposte in % alla domanda 27 del questionario MoSCoW.....	53

Grafico 3.28 Risposte in % alla domanda 28 del questionario MoSCoW.....	54
Grafico 3.29 Risposte in % alla domanda 29 del questionario MoSCoW.....	54
Grafico 3.30 Risposte in % alla domanda 30 del questionario MoSCoW.....	54
Grafico 3.31 Risposte in % alla domanda 31 del questionario MoSCoW.....	55
Grafico 3.32 Risposte in % alla domanda 32 del questionario MoSCoW.....	55
Grafico 3.33 Risposte in % alla domanda 33 del questionario MoSCoW.....	55
Grafico 3.34 Risposte in % alla domanda 34 del questionario MoSCoW.....	56
Grafico 3.35 Risposte in % alla domanda 35 del questionario MoSCoW.....	56
Grafico 3.36 Risposte in % alla domanda 36 del questionario MoSCoW.....	56
Grafico 3.37 Risposte in % alla domanda 37 del questionario MoSCoW.....	57
Grafico 6.1 Numero di errori commessi da ogni utente.	106
Grafico 6.2 Punteggio SUS ottenuto per ogni utente.....	109
Grafico 6.3 Risultati questionario NASA-TLX.	113
Grafico 6.4 Media dei valori pesati assegnati dagli utenti alle 6 sotto-scale.....	114

1. Introduzione

L'obiettivo principale di questa tesi è quello di documentare le fasi di sviluppo di un'applicazione, che sfrutta le potenzialità della realtà virtuale per dare la possibilità a chi la utilizza di generare uno storyboard.

Nel primo capitolo, verranno esaminati i concetti fondamentali relativi a storyboard, realtà virtuale e la sua storia fino ad oggi.

In seguito, un capitolo sarà dedicato all'analisi dello stato dell'arte, concentrandosi sulle principali soluzioni esistenti in questo campo.

La sezione relativa alla descrizione degli obiettivi includerà dettagli su come l'applicazione è stata concepita, l'idea alla base, ciò che si intende realizzare, le scelte funzionali, i requisiti, il design e il processo di sviluppo.

Inoltre, il motivo dietro tali scelte verrà chiarito mediante l'analisi delle risposte fornite da un gruppo di studenti, di diversa esperienza nell'ambito, a un questionario somministrato come Modulo Google.

Seguirà un capitolo sulle tecnologie utilizzate.

Nel capitolo dedicato allo sviluppo, verrà spiegato come è stato realizzato il progetto, compresa la pipeline di funzionamento dell'applicazione.

Infine, la tesi concluderà con un'analisi dei risultati dei test condotti dagli utenti e una discussione sui possibili sviluppi futuri del progetto.

1.1. Storyboard

“Gli storyboard raccontano una storia visiva del film. Forniscono gli elementi chiave della scena affinché il regista e lo scenografo possano pianificare l'azione, la costruzione e la programmazione del film [1].” Lo storyboard è una sequenza di disegni che illustra inquadratura per inquadratura ciò che verrà girato sul set.

Viene preparato durante la fase di pre-produzione¹ di un film, di una pubblicità, di un video o di una qualsiasi altra forma di rappresentazione multimediale. Fa parte della tecnica definita pre-visualizzazione o previs. È uno strumento fondamentale per la comunicazione tra i membri del gruppo creativo, il regista, il produttore, il supervisore VFX gli scenografi e gli altri professionisti coinvolti nella realizzazione del progetto visivo. Le tavole dello storyboard solitamente sono disegnate a mano o digitalmente.



Figura 1.1. Esempio di una vignetta dello storyboard ufficiale del video musicale “willow” di Taylor Swift confrontata con l’inquadratura poi effettivamente realizzata. Storyboard di Vincent Lucido [2].

¹ La pre-produzione è la fase che precede le riprese.

Vicino alle singole vignette vengono spesso indicate informazioni aggiuntive, utili a comprendere meglio l'inquadratura. Si potrebbe trovare, quindi, il numero della scena e dell'inquadratura, la descrizione, i dialoghi, il tipo di obiettivo che si intende usare, la durata, l'aspect ratio², il tipo di inquadratura, la location³, i SFX, la luce o l'atmosfera che si vuole avere, i VFX, e a volte il costo dell'inquadratura. I movimenti della macchina da presa e degli attori vengono spesso indicati utilizzando delle frecce.

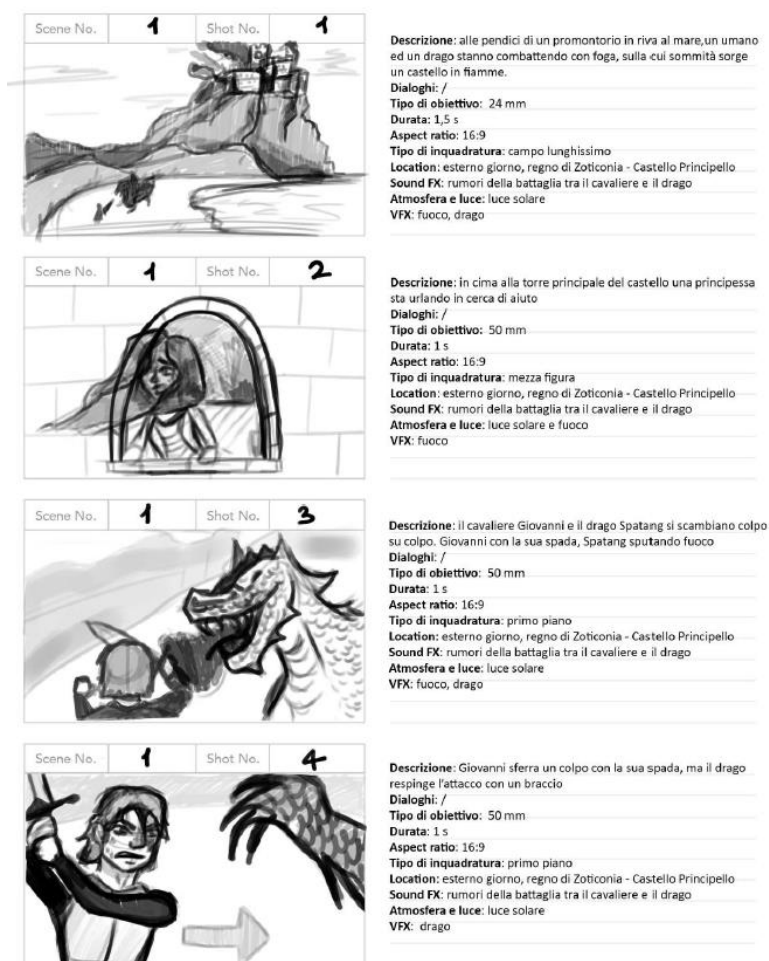


Figura 1.2. Prime 4 vignette di uno storyboard accompagnate a destra da informazioni aggiuntive.

² Il formato, cioè il rapporto tra la lunghezza e la larghezza di un'immagine.

³ Il luogo dove si effettuano le riprese.

Il primo regista ad utilizzare gli storyboard fu probabilmente George Méliès, mentre Walt Disney già dal 1927 sviluppò il processo di storyboarding nel modo in cui è conosciuto oggi [3].

Nella fase di pre-produzione, una volta completato lo storyboard, a partire dalle singole immagini di questo è possibile creare l'animatic o story reel: i disegni vengono montati in sequenza. Rispetto allo storyboard l'animatic contiene importanti informazioni in più come la quantità di inquadrature necessarie, il timing per indicare quando appare l'inquadratura e quanto permane sullo schermo e infine l'azione e quindi cosa succede nella vignetta. Spesso si ricorre all'uso di animatic elaborati che fanno l'uso della CG.

1.2. Realtà Virtuale

Nel 1994 il professore Paul Milgram e i ricercatori Haruo Takemura, Akira Utsumi, Fumio Kishino cercando di ridefinire il concetto di realtà aumentata con una definizione più coerente e meno ampia rispetto a *“arricchimento del feedback naturale per l'operatore con segnali simulati”* o *“una forma di realtà virtuale in cui il display indossato dal partecipante è trasparente, consentendo una visione chiara del mondo reale”*, si pongono due domande: *“Qual è la relazione tra realtà aumentata (AR) e realtà virtuale (VR)?”* e *“Il termine realtà aumentata dovrebbe essere limitato esclusivamente a display indossabili trasparenti?”* [4].

Concordano sul fatto che AR e VR siano correlati ed è valido considerare insieme i due concetti e, piuttosto che considerare i due concetti semplicemente come antitesi, è più conveniente vederli come situati agli opposti di un continuum. Elaborano, quindi, il concetto di Reality-Virtuality Continuum, rappresentato nella figura 1.3.

Il Reality-Virtuality Continuum è una scala continua che spazia tra il completamente virtuale e il completamente reale [5].

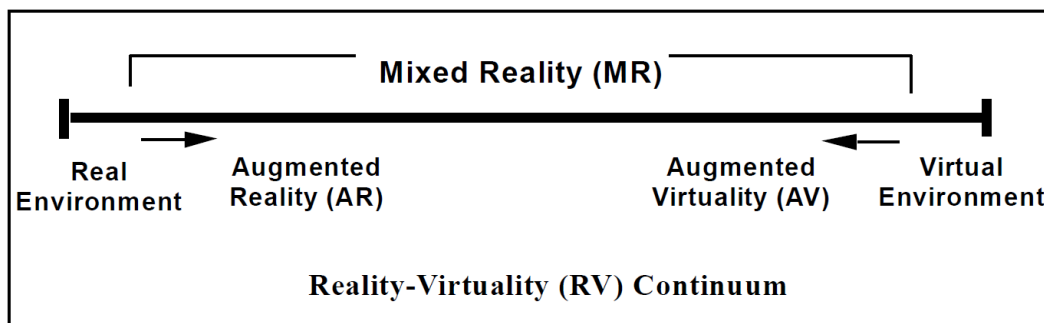


Figura 1.3 Rappresentazione semplificata del VR Continuum.

Il “*Real Environment*”, nell’estremo sinistro del diagramma “*definisce un ambiente composto esclusivamente da oggetti reali e include tutto ciò che potrebbe essere osservato quando si guarda una scena del mondo attraverso una sorta di finestra o tramite un display video.*”

Il “*Virtual Environment*”, nell’opposto destro, “*definisce ambienti composti esclusivamente da oggetti virtuali, tra cui simulazioni grafiche computerizzate convenzionali, basate su monitor o immersive.*” In questo segmento si colloca il concetto di VR.

All’interno di questo quadro, si definisce, inoltre, un ambiente generico di realtà mista (MR) dove gli oggetti del mondo reale e quelli del mondo virtuale sono presentati insieme in un unico display. All’interno della MR si collocano, quindi, spostata verso sinistra la AR e verso destra la AV.

La AR va quindi ad aggiungere elementi virtuali in un ambiente reale, mentre la AV aggiunge elementi reali, come ad esempio l’immagine reale di una persona, in un mondo virtuale.

Tornando, invece, al concetto di VR sono state date diverse definizioni nel corso degli anni.

Una prima definizione più filosofica della realtà virtuale è la seguente: “La realtà virtuale permette di vivere in un mondo verosimile, in cui si mescolano il magico e il reale”. Dove questo ambiente può essere considerato magico perché non è vincolato alle leggi fisiche tradizionali.

Una seconda definizione più tecnica invece è stata data da Gregor Burdea, professore alla Rutgers, l'Università statale del New Jersey: “La realtà virtuale è un'interfaccia uomo-computer di alto livello che comprende la simulazione in real-time di un mondo realistico e le molteplici interazioni con gli oggetti di tale ambiente attraverso canali sensoriali multipli.” [6].

È, quindi, una simulazione in cui la CG è utilizzata per creare un mondo che sia visivamente realistico. Poiché la realtà si basa su azioni, il mondo virtuale deve essere interattivo e rispondere modificandosi agli input dell'utente, che siano gesti, comandi verbali, interazione tramite dispositivi...; si ha allora un'interattività in tempo reale e istantanea come nella nostra realtà. Esistono poi le percezioni: l'immersione a livello sensoriale deve essere convincente e possibilmente coinvolgere tutti i sensi.

Ci sono, inoltre, altri due concetti fondamentali quando si parla di VR e questi sono l'immersione e la presenza.

L'immersione è l'estraniamento dal contesto reale e l'immersività misura la percezione di essere fisicamente presente nel mondo virtuale. Se l'immagine va a coprire tutto il campo visivo dell'uomo, è coerente con la posizione e orientamento dell'utente e vengono riprodotte le proprietà tattili e acustiche, allora si è realmente immersi. Inoltre, maggiore è il numero di canali sensoriali umani stimolati, maggiore è il senso di immersione. Questo ultimo concetto è però anche legato alla descrizione tecnica di cosa il sistema è in grado di fare. Esistono, infatti, principalmente due tipologie di gradi di libertà (DoF), cioè il numero di modi in cui un oggetto rigido può muoversi nello spazio tridimensionale della realtà virtuale. Esistono in totale 6 gradi di libertà che descrivono ogni possibile movimento di un oggetto: tre sono riferiti al movimento rotazionale attorno agli assi x, y, z (anche chiamati in inglese pitch, yaw e roll) e altri tre sono per i movimenti di traslazione lungo tali assi, che può essere pensato come movimento in avanti o indietro, a sinistra o a destra, e su o giù. I visori VR sono generalmente 3-DoF o 6-DoF [7].

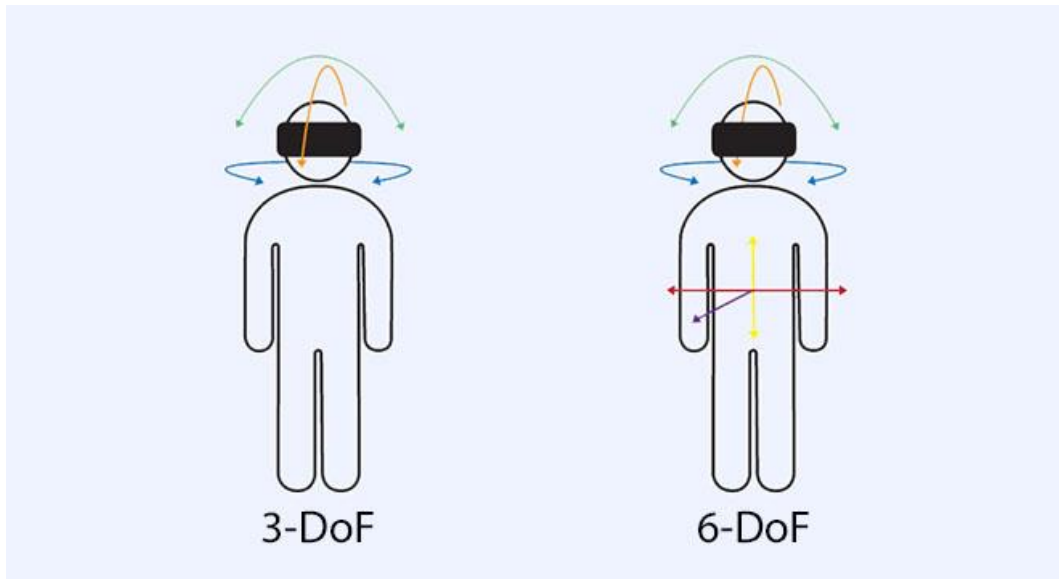


Figura 1.4 Differenza tra 3-DoF (rotazione) e i 6-Dof (rotazione e traslazione) con in caschetto di realtà virtuale.

La presenza è, invece, la sensazione di appartenenza al mondo virtuale che porta ad una immersione mentale e ad una volontaria “sospensione dell’incredulità” da parte dell’utente. L’immersione e la presenza richiedono una progettazione apposita e dipende molto dagli strumenti che si hanno a disposizione. Una VR immersiva con un visore VR risulterà diversa da una Desktop VR, di immersività scarsa, che fa uso di interfacce limitate come monitor, mouse e tastiera, le quali limitano i movimenti e portano a sistemi monoutente.

L’immaginazione umana, infine, completerà e supererà i limiti dati dalla tecnologia, che solo “approssima” il mondo reale.

La VR per raggiungere il suo obiettivo deve quindi rispettare certi vincoli. Nella VR immersiva un ampio FOV è fondamentale se si vuole ottenere un’esperienza coinvolgente. Con la visione binoculare, gli esseri umani hanno un campo visivo orizzontale di quasi 200°, di cui 120° di sovrapposizione binoculare. Il campo visivo verticale è di circa 130°. Il campo visivo minimo richiesto per un dispositivo VR è di 100° [8].

Il mondo in cui si è immersi deve avere una scala realistica con il mondo reale e l’utente deve ricevere un feedback multimodale (tattile, motorio, uditivo...) tramite interfacce specifiche per la manipolazione e controllo dell’ambiente.

I tracker, studiati per la realtà virtuale, sono corpi rigidi in uno spazio 3D con 6 DoF composti dai sensori, componenti di acquisizione, e da componenti di elaborazione che elaborano i dati dei sensori e forniscono l'input per il sistema. Ne esistono di diversi tipi, che lavorano con diverse tecnologie tra cui la meccanica, il magnetismo, l'acustica e l'ottica.

Vi sono diverse prestazioni da considerare per capire quali sono quelli più utili per il proprio sistema:

- La precisione o accuratezza, data dalla differenza tra la posizione attuale dell'oggetto e quella misurata dal sensore. L'area di lavoro del tracker è il massimo valore entro il quale l'accuratezza è accettabile.
- Il rumore o jitter è la variazione del valore misurato dal sensore quando l'oggetto è stazionario. Se c'è rumore la vista finisce per traballare di continuo oppure l'oggetto si muove continuamente.
- La deriva o drift, cioè l'aumento dell'errore fisso introdotto nella misura dal sensore in funzione del tempo. Dev'essere controllata ed eventualmente azzerata tramite l'utilizzo di un dispositivo secondario.
- La latenza, cioè il ritardo di risposta del sensore nel misurare una variazione di posizione o orientamento dell'oggetto. È un parametro dovuto alla catena di elaborazione dei dati e dipende dal sistema di tracking, dalla pipeline di rendering⁴, dal display e dal sistema di sincronizzazione. Una latenza sotto i 100 ms non è percepibile direttamente anche se il minimo accettabile sarebbe di 20 ms.
- La velocità di acquisizione dei dati o update rate, il numero di misurazioni al secondo del tracker, dipende dalla tecnologia del tracker.

⁴ Passaggi necessari per elaborare gli oggetti 3D in una scena.

Il suono gioca, infine, un ruolo fondamentale nell'aumentare il senso di immersione. Se spazializzato, cioè generato in modo che sembri provenire da una determinata posizione spaziale, il cervello umano riuscirà a localizzarlo e identificare la posizione di emissione del suono. Questo perché un suono genera nello spazio un'onda sonora che si propaga fino all'ascoltatore. Vi sono tre fattori percettivi che danno informazioni sulla direzione di emissione:

- Il fattore tempo dato dall'Interaural Time Delay, cioè il ritardo con cui il suono raggiunge prima un orecchio e poi l'altro, che deve essere al massimo di 0.7 ms.
- Il fattore volume legato alla differenza di intensità con cui il suono viene percepito definito Interaural Intensity Difference.
- Il fattore suono che viene filtrato dall'interazione acustica con torso, testa e orecchio esterno.

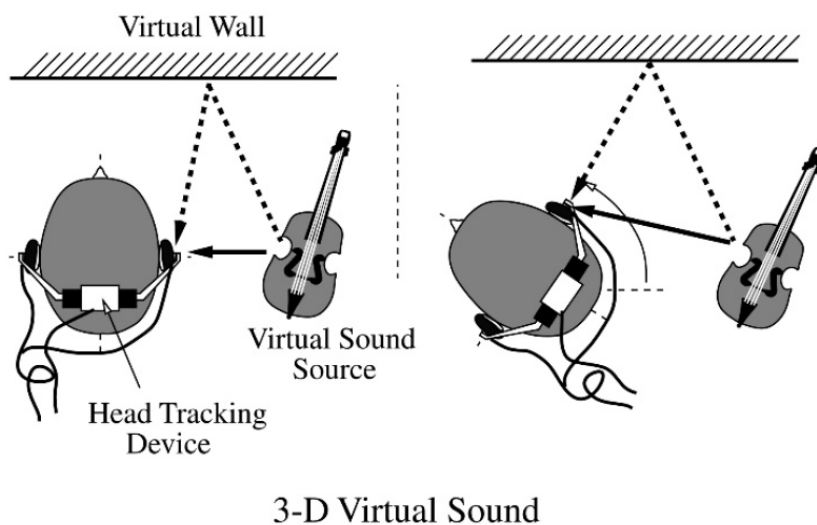


Figura 1.5 Il suono spazializzato nella VR.

1.3. Realtà Virtuale: ieri e oggi

Nel 1929, Edwin A. Link inventò un tipo di simulatore di volo per far provare ai passeggeri la sensazione del volo. Fu il primo tentativo in cui gli esseri umani simularono o emularono la realtà fisica.

Nel 1956, Morton Heilig inventò il Sensorama, un emulatore di motocicletta che permetteva all'utente di avere "l'impressione" di guidare per le strade di Manhattan e percepire anche l'odore dei gas di scarico delle auto e il vento sul viso. Mostrava immagini reali riprese con due telecamere e presentate all'utente con un visore stereoscopico, era dotato di un sedile e manubrio che vibravano riproducendo il movimento, audio stereofonico, phon che simulavano il vento a diverse velocità e uno spray per gli odori. Brevettò quindi il Sensorama Simulator nel 1962, ma mancavano ancora due aspetti importanti: l'interazione e il mondo virtuale [9].

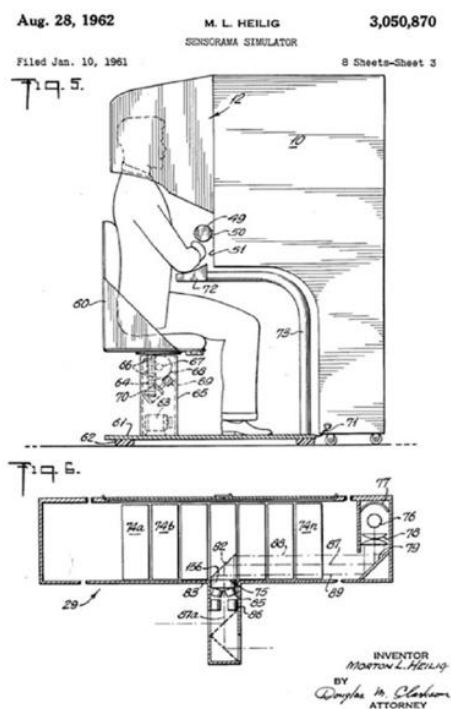


Figura 1.6. Sensorama, il primo sistema di immersione virtuale, la tavola tecnica e le immagini mostrate durante la presentazione del prodotto sperimentale [10].

Nel 1965, il Dr. Sutherland, fondatore della grafica computerizzata, pubblicò un saggio dal titolo "The Ultimate Display". In questo saggio, teorizzò una tecnologia di visualizzazione dove gli osservatori avrebbero potuto immergersi in un ambiente virtuale controllato da un computer, simile alla vita quotidiana nel mondo reale. Allo stesso tempo, suggerì che gli osservatori avrebbero potuto interagire con gli oggetti nell'ambiente virtuale in modo naturale, utilizzando il tatto e il controllo di oggetti virtuali, tra le altre cose. Dal punto di vista della visualizzazione tramite computer e dell'interazione uomo-macchina, il saggio di Sutherland introdusse il concetto di simulazione del mondo reale, contribuendo così allo sviluppo della grafica. Inoltre, spinse la ricerca su nuovi dispositivi di interazione uomo-macchina come il visore montato sulla testa (HMD) e i guanti dati. Nel 1966, Sutherland iniziò a sviluppare il primo HMD e successivamente integrò nel sistema un dispositivo di feedback di forza per la simulazione e una sensazione di tocco. Il dispositivo era sospeso al soffitto data la sua pesantezza e perché l'asta rilevava i movimenti della testa [9].

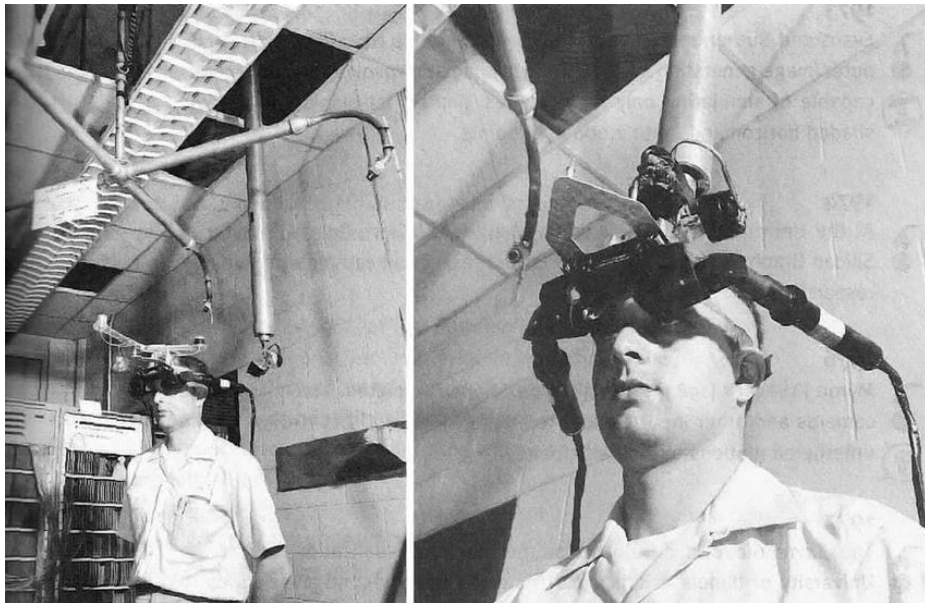


Figura 1.7 Il primo HMD, inventato da Ivan Sutherland [11].

Nel 1968 venne sviluppato dalla NASA's Ames Researcher Center il Boom, un dispositivo montato su un braccio meccanico, utile a sostenerlo e a mandare informazioni sulla posizione, dotato di due schermi.



Figura 1.8 Boom [12].

Nel 1973 nacque il primo “acceleratore” grafico per la produzione di immagini di sintesi, dall’azienda statunitense Evans & Sutherland (400 poligoni a 20 FPS).

Nel 1977 i ricercatori Daniel J. Sandin e Thomas Defanti, presso il Laboratorio di Visualizzazione Elettronica dell'Università dell'Illinois a Chicago, inventarono il Sayre Glove, un guanto in grado di percepire e misurare la flessione delle dita. Questo guanto utilizzava tubi conduttori di luce per trasmettere quantità variabili di luce proporzionali al grado di flessione delle dita, stimando così la configurazione della mano dell'utente [11].

Nel 1978 ci fu il primo tentativo di simulazione interattiva di un ambiente. Il progetto si chiamava “Aspen Movie Map” e fu sviluppato presso l'Architecture Machine Group del MIT (Massachusetts Institute of Technology); il direttore del progetto fu Andrew Lippman. Si può definire come un progetto antenato dell'attuale Google Street View, si poteva selezionare la stagione e navigare per le strade di Aspen, cittadina statunitense, attraverso una serie di 50.000 frame⁵, contenuti su un videodisco [13].

Nel 1979, Eric Howlett sviluppò il LEEP (Large Expanse Extra Perspective), un piccolo display in grado di offrire un'ampia esperienza di visione. Il LEEP rimase la base di influenza tecnica per i successivi display VR [14].



Figura 1.9 Eric Howlett, l'inventore del LEEP.

⁵ Fotogramma, singola immagine che compone un video.

Nel 1979 nacque il Polhemus 3SPACE, il primo tracker non a contatto a 6 gradi di libertà. Era un tracker magnetico che soffriva la presenza di strutture metalliche, in quanto le interferenze elettromagnetiche portavano ad elevato rumore e bassa precisione.

Nel 1984 venne fondata la Virtual Programming Languages (VPL Research) da Jaron Lanier, l'uomo che conierà il termine realtà virtuale. Il laboratorio VIEW della NASA stipulerà contratti con la VPL Research per lavorare sui DataGlove, guanti-sensori, ed i EyePhones, HMD che sfruttano ottiche LEEP [9].

Fino a quel momento ad usare i dispositivi inventati erano solo le grandi industrie americane. Un grosso passo avanti fu rendere disponibili questi dispositivi a livello commerciale e la prima compagnia a fare ciò fu la VPL Inc. commercializzando prima i VPL DataGlove nel 1987 e successivamente gli EyePhones. I DataGlove erano molto costosi, non avevano un feedback⁶ tattile ed era difficile farli adattare alle diverse configurazioni della mano. Gli EyePhones avevano una risoluzione di 360x240 pixel⁷, erano molto costosi e pesanti.

Nel 1989 la Nintendo Co. Ltd. presenta i Mattel PowerGlove, a \$75 (equivalenti a \$184 nel 2023). Durarono un anno perché furono abbandonati per mancanza di giochi che li sfruttassero e per la bassa precisione [15].



Figura 1.10 Nintendo Mattel Power Glove, modello americano.

⁶ Retroazione, processo per cui il risultato dall'azione di un sistema si riflette sul sistema stesso per correggerne o modificarne il comportamento.

⁷ Componente elementare di un'immagine.

Il primo sistema integrato fu l'RB2 Model 2, sempre della VPL, il quale incorporava: EyePhone, DataGlove, un sistema di tracking per l'HMD, una workstation di controllo e progetto, collegamenti con una workstation esterna di rendering SGI 4D/310 VGX, collegamento con un sistema opzionale di gestione del suono 3D.

In quegli anni il punto debole rimaneva il software, poiché doveva essere sviluppato da zero a seconda dell'applicazione. Solo nei primi anni '90 si svilupparono i primi software e API che facilitavano lo sviluppo di applicazioni di realtà virtuale.

Ci furono alte aspettative tra gli anni '80 e '90 sulla realtà virtuale, ma questa non era ancora all'altezza. I tre motivi principali, secondo Lanier, furono: i problemi legati alle elevate capacità di calcolo richieste, la mancanza di interfacce ad alta qualità, display ad alta risoluzione e infine la mancanza di software adeguati alla progettazione e lo sviluppo.

La rinascita della VR si ebbe solo dopo l'introduzione dei computer nelle case, i miglioramenti delle GPU grazie all'industria dei videogiochi, la diffusione di dispositivi di interazione a basso costo e all'ampia disponibilità di software per la creazione di contenuti 2D e 3D, il processamento dell'audio e per la gestione di ambienti complessi.

Negli ultimi dieci anni l'argomento è stato nuovamente oggetto di risonanza. Infatti, è il 2014 quando Facebook (oggi Meta) compra Oculus VR per 2 miliardi di dollari [16]; mentre nel 2015 al Mobile World Congress di Barcellona HTC e Valve Corp. annunciano insieme il sistema VR HTC Vive e i controller [17].

Il 28 marzo 2016 viene pubblicato commercialmente l'Oculus Rift CV1, sviluppato da Oculus VR. Le versioni per gli sviluppatori già erano disponibili nel 2012, due modelli sono stati rilasciati agli sviluppatori: Oculus VR DK1 (Development Kit 1) e Oculus VR DK2 (Development Kit 2) [18]. Il dispositivo verrà descritto nelle specifiche nel capitolo 4 "Tecnologie utilizzate".

Il 5 aprile 2016 esce sul mercato L'HTC Vive, progettato da Valve in collaborazione con HTC [19]. È dotato di due display AMOLED e di una risoluzione di 1080x1200 pixel per occhio, un refresh rate⁸ di 90 Hz e un FoV di 110°. Ha un microfono integrato. Ha bisogno di diverse connessioni al computer per poter funzionare e permette di gestire uno spazio virtuale di approssimativamente 3.5m x 3.5m [20].



Figura 1.11 Headset (caschetto) HTC Vive.

Utilizza 2 Vive Base Station che consentono di identificare il caschetto e i controller nello spazio. Questo perché vengono usati dei tracker ottici, che usano delle telecamere e che si basano sui segnali luminosi, di tipo “Outside looking in”. La tecnica consiste nel posizionare delle telecamere nell’ambiente, le base station, che riprendono l’oggetto e i marker attivi o passivi, in base a se emettono luce o la riflettono, posizionati su di esso.



Figura 1.12 Controller HTC Vive e Vive Base Station.

⁸ Frequenza di aggiornamento, numero di volte al secondo di aggiornamento dell'immagine sullo schermo.

Il 13 ottobre 2016 la Sony lancia sul mercato la PlayStation VR, progettata per essere compatibile con la Playstation 4 [21]. È dotata di display OLED e di una risoluzione di 1920 x RGB x 1080 pixel (960 x RGB x 1080 pixel), un refresh rate di 90 Hz o 120Hz e un FoV di circa 100°. Ha un microfono integrato, accelerometro, e giroscopio. Utilizza connessioni HDMI e USB [22].



Figura 1.13 Headset PlayStation VR.



Figura 1.14 PlayStation Camera per “Outside looking in”.



Figura 1.15 Controller di movimento PlayStation Move.

Dal 2016 ad oggi sono stati presentati diversi modelli di visori.

Meta Inc. ha commercializzato Oculus Go nel 2017, il primo dispositivo completamente standalone⁹, creato in collaborazione con Qualcomm e Xiaomi. Il sistema operativo è Android, personalizzato da Oculus per la VR. Viene venduto insieme ad un unico controller esterno. Ha una risoluzione totale di 2560 x 1440 pixel e un refresh rate massimo di 72 Hz. Utilizza un giroscopio, un accelerometro e un magnetometro per tracciare solamente la rotazione della testa, non permettendo il movimento nello spazio.

Oculus Rift S, l'evoluzione del primo Rift, viene creato nel 2019 in collaborazione con Lenovo. Anche questo dispositivo ha bisogno di essere collegato ad un PC per funzionare. Non necessita di tracker esterni per tracciare l'area di gioco poiché monta camere e sensori direttamente sul visore, permettendo i 6 DoF. Questa tecnologia prende il nome di "Inside looking out" o "Inside-out tracking". Con un nuovo schermo LCD ha una risoluzione di 2560 x 1440 pixel e un refresh rate di 80 Hz. La produzione dei Rift cessa nell'aprile del 2021.

Lo stesso anno viene presentato l'Oculus Quest, capostipite di una nuova famiglia di visori Oculus, più economica rispetto ai Rift. È un visore stand-alone basato su sistema Android e Qualcomm Snapdragon 835, che a differenza del predecessore Oculus Go permette i 6 DoF. La risoluzione dello schermo, nuovamente OLED, è di 1660 x 1440 pixel a occhio con un massimo di refresh rate di 72 Hz [23].

L'anno successivo esce il Quest 2, con prestazioni migliori rispetto al Quest grazie alla piattaforma Snapdragon XR2 Gen1 e i 6 GB di DRAM, con un display 1832 x 1920 pixel per occhio, 773 PPI e 20 PPD, fino ad un massimo di 120 Hz. FoV di 90° verticali e orizzontali.

⁹ Oggetto o software capace di funzionare da solo.

Nell'ottobre del 2022 viene presentato il Quest Pro, molto più costoso dei precedenti e destinato agli ambienti di lavoro. Dotato di Snapdragon XR2+ Gen1 e 12 GB di DRAM, FoV di 106° orizzontali e 96° verticali, refresh rate di 90 Hz e risoluzione 1800 x 1920 pixel, 1058 PPI e 22 PPD. Viene introdotto il tracking¹⁰ degli occhi e del viso e la realtà mista: grazie ad una scala di colori 1.3 volte più grande e ad una risoluzione 4 volte superiore rispetto al Quest 2, l'integrazione del mondo virtuale con il mondo reale risulta essere più fluida.



Figura 1.16 Visore Meta Quest Pro e controller.

Nell'autunno del 2023 viene presentato quello che è ad oggi il visore più recente di casa Meta: il Meta Quest 3. Con Snapdragon XR2 Gen2, 8 GB di DRAM, display da 2064 x 2208 pixel, 1218 PPI e 25 PPD, refresh rate di massimo 120 Hz, FoV di 110° orizzontali e 96° verticali, l'azienda lo definisce come “*Il nostro visore più potente di sempre, con prestazioni e grafica di nuova generazione* [24].”



Figura 1.17 Visore Meta Quest 3 e Controller Touch Plus [25].

¹⁰ Tracciamento.

L'8 gennaio 2018 HTC presenta HTC VIVE Pro, una versione aggiornata del predecessore. Dotato di schermi AMOLED da 1440 x 1600 pixel di risoluzione per occhio, 615 PPI, refresh rate di 90 Hz, FoV di 110°. Usa la porta DisplayPort al posto della HDMI [26].

Il VIVE Focus è invece il primo modello a non aver bisogno di un computer per funzionare, essendo standalone. Ha all'inizio, però, solo 3 DoF.

L'anno successivo presenta HTC VIVE Pro Eye in cui viene aggiunto il tracciamento oculare che monitora i movimenti oculari per determinare dove il soggetto sta guardando [27]. E successivamente il VIVE Focus Plus, un aggiornamento del Focus che ora supporta i 6 DoF. Lo stesso anno viene commercializzato il Vive Cosmos, che similmente all'Oculus Quest e Rift S usa l'Inside-out Tracking, avendo 6 camere sul caschetto al posto delle Base Station. Ha bisogno quindi di diversi controller rispetto agli altri modelli [28].

Nel 2020 vengono presentati il Cosmos Elite, simile alla serie Pro che ritorna ad utilizzare le Base Station, il Cosmos XR, che aggiunge un campo visivo passante quasi completo, per visualizzare i contenuti virtuali sul mondo reale, e il Cosmos Play, una versione più economica con solo 4 telecamere progettato per gli utenti che si avvicinano per la prima volta alla VR [29].



Figura 1.18 HTC VIVE Cosmos XR e controller.

Nel 2021 esce il VIVE Pro 2, aggiornamento della serie Pro, con schermi LCD, risoluzione 2448×2448 pixel per occhio, refresh rate di massimo 120 Hz, fino a 120° di FoV orizzontali [30].



Figura 1.19 HTC VIVE Pro 2, Base Station e controller.

A ottobre esce il Vive Flow, dal design differente rispetto agli altri modelli. Supporta l'Inside-out Tracking e può usare lo smartphone come controller [31].



Figura 1.20 HTC VIVE Flow e controller da 3DoF.

Ad oggi l'ultimo modello della serie Focus è il VIVE Focus 3, uno standalone headset per professionisti. Ha le stesse specifiche del Pro 2, ma usa la tecnologia Inside-out Tracking.



Figura 1.21 HTC VIVE Focus 3 e controller.

A gennaio 2023 viene annunciato il VIVE XR Elite. Dotato di una risoluzione di 1920 x 1920 pixel per occhio (3840 x 1920 pixel combinati), 6 DoF inside-out tracking grazie alle 4 camere a colori, refresh rate di 90 Hz, fino a 110° di FoV, 28 GB ROM, 12 GB RAM. Ha le lenti con le diottrie regolabili e utilizza gli stessi controller del Focus 3 [32].



Figura 1.22 HTC VIVE XR Elite e controller.

A inizio 2023 PlayStation aggiorna il suo precedente visore con il PlayStation VR2, dotato di schermi OLED, risoluzione di 2000 x 2040 pixel per occhio, refresh rate fino a 120 Hz, FoV di circa 110°, 4 telecamere incorporate per il rilevamento del visore e del controller, telecamera IR per occhio per il rilevamento dello sguardo, vibrazione del visore. Funziona solo con la console PlayStation 5 [33].



Figura 1.23 PlayStation VR2 e controller.

Apple Inc. all'Apple WorldWide Developers Conference, conferenza annuale dell'azienda tenuta in California, di giugno 2023 ha presentato il suo primo visore: Apple Vision Pro. Il visore è "cieco", ma riproduce gli occhi dell'utente verso l'esterno attraverso la tecnologia chiamata EyeSight.

Introduce alcune novità piuttosto interessanti rispetto ai concorrenti come Meta o HTC. Non appena si indossa il visore la prima cosa che si osserva è l'ambiente esterno; il sistema si popola poi di tutte le proprie applicazioni, che si possono spostare nell'ambiente "virtuale" (nel senso di ambiente reale, ma proiettato nell'interfaccia). Non viene prodotto un controller fisico dedicato: si fa tutto tramite occhi e mani. Sarà disponibile commercialmente a partire dal 2024 [34].

2. Stato dell'Arte

Lo stato dell'arte sull'utilizzo della realtà virtuale per generare Storyboard ed elementi di pre-visualizzazione è in continua evoluzione. In questo capitolo verranno analizzati i casi più rilevanti relativi all'argomento, sia che siano strumenti di ricerca sia applicazioni commerciali.

2.1. Empowering Creative People: Virtual Reality for Previsualization - articolo del 2018

Gli autori di questo articolo [35] descrivono le caratteristiche dei propri prototipi e riportano la valutazione di professionisti non tecnici del settore cinematografico, dell'animazione e del teatro che hanno provato l'applicativo. I risultati dimostrano che gli utenti sono stati in grado di interagire con scene tridimensionali complesse, dopo una breve fase di familiarizzazione, e hanno valutato la realtà virtuale utile per il loro lavoro professionale, in quanto permette loro di eseguire da soli compiti di previs in 3D senza dover comunicare le loro idee al personale tecnico.

Hanno sviluppato due prototipi: uno focalizzato sul lavoro della camera per la realizzazione delle riprese e uno per esplorare il design del palcoscenico e sperimentare con l'equipaggiamento teatrale in un teatro.

Per l'implementazione sono stati usati il motore grafico Unity, l'HTC Vive, i controller ed un'area di tracking di 4m x 4m.

In entrambi i prototipi per muoversi gli utenti possono muovere, traslare e scalare il mondo virtuale attorno a loro premendo un tasto dei controller. Gli oggetti possono essere afferrati quando un controller si trova vicino ad essi e viene

premuto il grilletto. Premendolo nuovamente questi vengono rilasciati nella nuova posizione.

Nel primo prototipo la camera virtuale, l'elemento di colore giallo nella figura sottostante, è spostabile come gli altri oggetti e attraverso un movimento di swipe si può aggiustare lo zoom. L'anteprima di ciò che la camera visualizza e gli screenshot fatti seguono il controller sinistro.



Figura 2.1 La camera virtuale gialla del primo prototipo.

Nel secondo prototipo viene riprodotto un accurato modello virtuale del teatro Landestheater Linz in Austria, costituito da 20 parti e 3 figure umanoidi. Ogni elemento può essere riposizionato dagli utenti senza considerare le leggi fisiche per consentire più creatività e sperimentazione agli utenti.

2.2. VR as a Content Creation Tool for Movie Previsualisation - articolo del 2019

Gli autori propongono uno strumento in VR [36] per poter realizzare una pre-visualizzazione 3D di quello che poi si andrà a girare in un film. Lo strumento copre le principali fasi nella pre-produzione cinematografica: la creazione di storyboard, la realizzazione di animazioni in 3D e la preparazione degli aspetti tecnici della ripresa. Hanno valutato lo strumento con utenti professionali per

raccogliere le prospettive degli esperti sui vantaggi dell'utilizzo della realtà virtuale in questo campo.

L'applicazione ruota attorno a tre modalità principali. La modalità Storyboard dove gli utenti possono posizionare gli oggetti e definire le pose dei personaggi. È utile per pianificare visivamente come apparirà la scena, stabilendo la disposizione degli elementi principali. La modalità Previs dove vengono animati i personaggi e gli oggetti attraverso i keyframe¹¹. Infine, la modalità Tecnica dove vengono manipolate le complesse camere e le luci. Vengono sfruttati i 6 DoF ma ogni specifico compito ha dei vincoli specifici sulla libertà dell'utente per poterlo guidare nell'interazione.

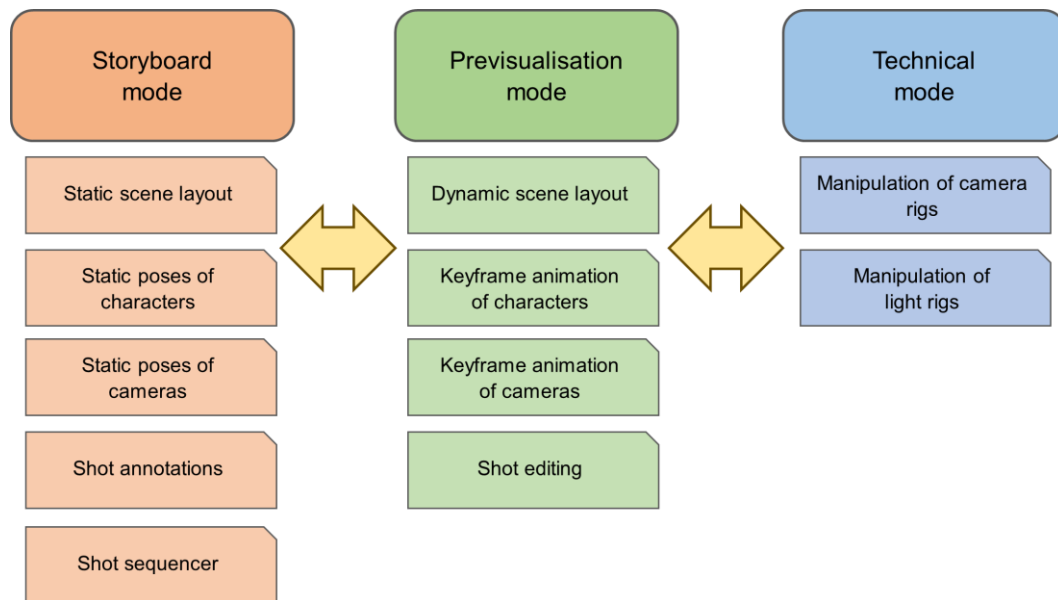


Figura 2.2 Le tre modalità presenti nell'applicativo.

Sono stati usati il motore grafico Unity, l'HTC Vive e le tecniche di selezione, manipolazione, navigazione e controllo delle applicazioni presenti nella letteratura.

Il controller destro viene utilizzato per editare la scena. Ogni oggetto può essere manipolato (azione di Grab) nello spazio premendo il grilletto e rilasciandolo.

¹¹ Fotogramma chiave, base delle animazioni computerizzate.

Il raycast¹² viene utilizzato per posizionare un nuovo oggetto quando viene creato.

Il controller sinistro viene utilizzato per i menù e per la navigazione che avviene tramite la tecnica del teletrasporto e lo spostamento tramite la pressione di un tasto dedicato. I menù, sul controller o come elementi 3D della UI, vengono utilizzati per passare da una modalità all'altra e per dare l'accesso a specifiche funzioni di ogni modalità, come la possibilità di aggiungere ed eliminare gli oggetti, controllare il tempo, aggiungere i keypoint della camera...

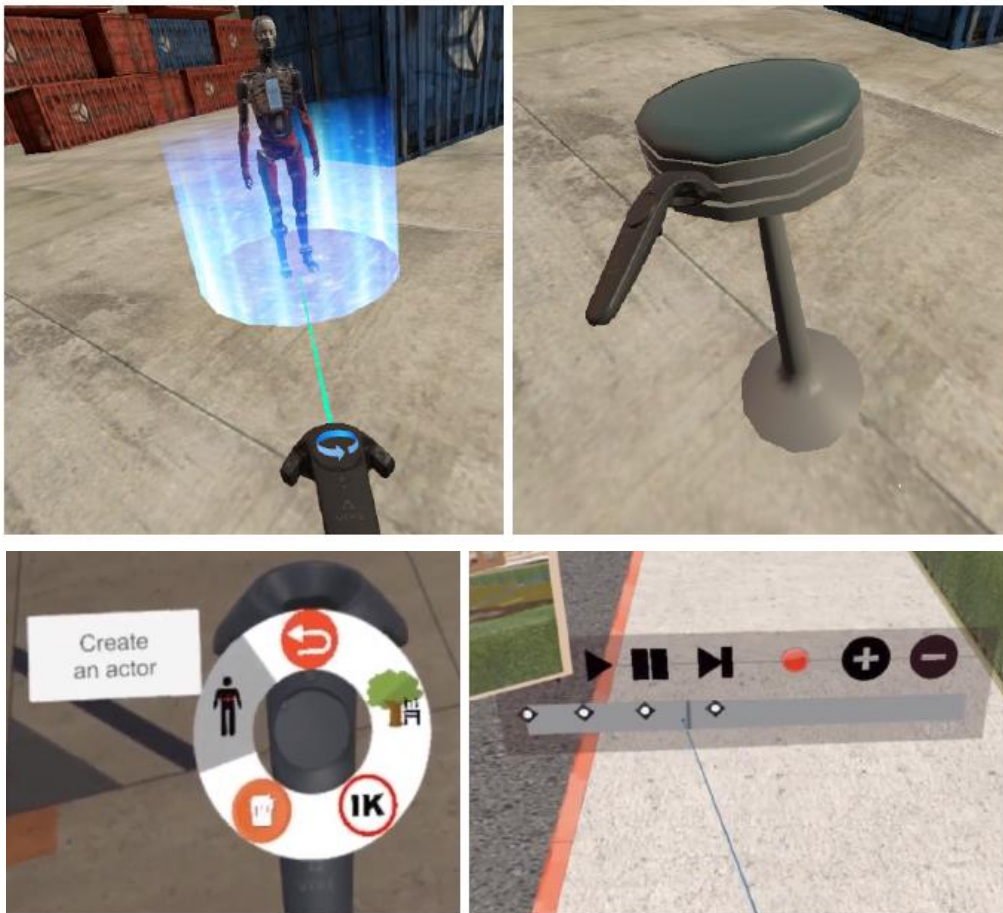


Figura 2.3 Raycast, Grab degli oggetti, menù sul controller, menù come 3D UI.

¹² Raggio che parte dai controller e collide con gli elementi 3D della scena.

I personaggi possono essere messi in posa manipolando i nodi delle armature, nello stesso modo in cui vengono manipolati gli altri oggetti.

Una o più oggetti camera 3D possono essere messi in scena, sono fluttuanti e manipolabili come gli altri oggetti. Proiettano quello che inquadrano su uno schermo spostabile nello spazio. Si usano uno slider per gestire la focale e pulsanti della 3D UI per fare screenshot, creare nuove camere e aggiungere sopra all'inquadratura il tipo di movimento di camera indicandolo con delle frecce. Queste camere possono poi essere programmate nei movimenti, facendo seguire loro ad esempio delle rotaie posizionate dall'utente.

I personaggi possono essere animati attraverso diverse modalità: con i keyframe, attraverso la tuta di motion capture¹³ che indossa l'utente, registrando i movimenti dei controller e editando le animazioni una volta create. Un pannello attaccato ad ogni oggetto animato mostra la lista di keyframe.

Si possono controllare complesse attrezzature per le camere, come ad esempio il crane, e le luci, delle quali si possono modificare la posizione, l'orientamento, l'apertura e l'intensità.

I risultati dei test condotti su principianti e professionisti dagli sviluppatori evidenziano che lo strumento potrebbe essere integrato nella pipeline di produzione cinematografica senza un forte impatto in termini di formazione, ovvero garantendo una buona facilità d'utilizzo dello strumento.

Gli autori, in conclusione, sottolineano una mancanza di stabilizzazione nei movimenti della camera virtuale e una mancanza di tecniche di montaggio più complesse emerse dai test.

¹³ Tecnica per la registrazione digitale dei movimenti di un corpo.

2.3. A Collaborative Previsualization Tool for Filmmaking in Virtual Reality - articolo del 2019

Lo strumento [37] nasce per essere utilizzato principalmente online e in modalità collaborativa. Il sistema è stato costruito utilizzando Unity come motore grafico, HTC VIVE Pro come visore per la VR, 2 controller, i 6DoF e un'area di 7m x 7m.

Il sistema è diviso in tre fasi: la preparazione della scena, l'animazione in tempo reale e l'esportazione del video. La prima fase è quella offline dove il mondo viene popolato di oggetti 3D. La fase successiva è invece online e simula le azioni che avvengono realmente in un set cinematografico: movimenti dei personaggi e dialoghi, i movimenti di camera, i cambi di ottiche (lunghezza focale e apertura). L'ultima fase è quella conclusiva in cui ciò che la camera virtuale ha registrato viene esportato in formato video.

Non vengono usati i keyframe, ma tutte le animazioni vengono create in tempo reale tracciando i dati dei controller. Viene adottato un paradigma in miniatura in cui gli utenti dirigono modelli in scala dei personaggi e delle telecamere. Questo design consente una configurazione molto più semplice e meno costosa. Utilizzando modelli in scala ridotta, lo spazio d'azione nella scena può essere aumentato, e il sistema può simulare una vasta gamma di inquadrature come dolly, gru e riprese in elicottero, oltre a una maggiore varietà di movimenti dei personaggi.

I passaggi di animazione sono incrementali in quanto l'utente registra incrementalmente prima il movimento di ciascun personaggio e successivamente quello di ciascuna telecamera. Durante la registrazione di ciascun oggetto, le animazioni precedentemente registrate su altri oggetti vengono riprodotte in tempo reale. Ciò consente la creazione incrementale di scene complesse che includono decine di personaggi e comparse.

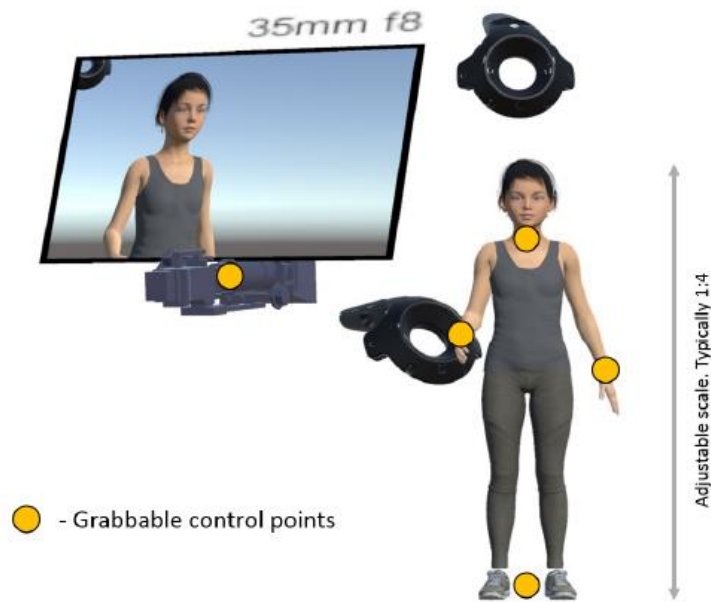


Figura 2.4 Modello in scala di un personaggio e camera.

L'interfaccia utente consiste di menù fluttuanti vicino al controller sinistro e diverse azioni collegate ai tasti dei controller.

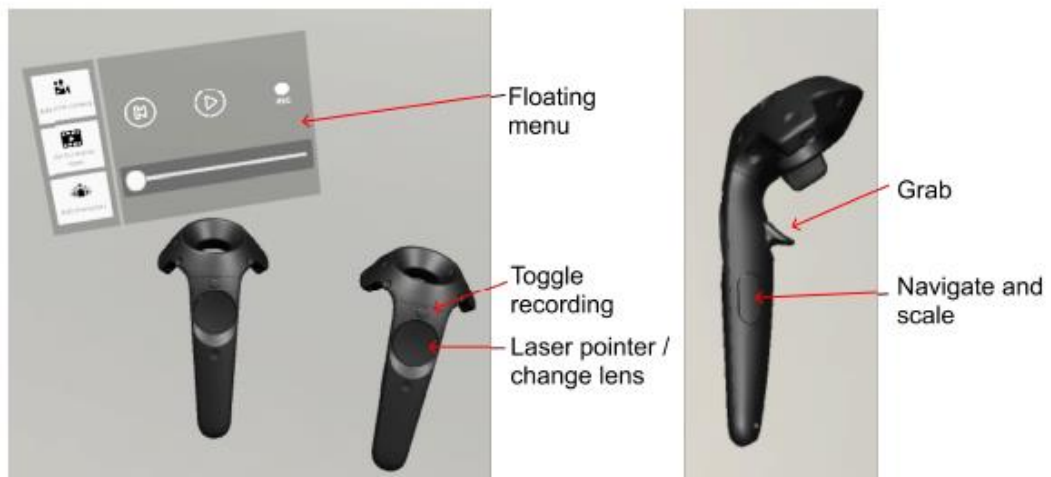


Figura 2.5 Menù e tasti dei controller.

La navigazione avviene attraverso lo spostamento e lo scalamento del mondo da parte dell'utente attraverso i controller.

Il laser parte dal controller solo nel momento in cui viene premuto un tasto sui controller.

L'animazione dei personaggi avviene afferrandoli dai punti di controllo presenti sui modelli. Quando viene afferrato il nodo radice i movimenti di camminata e corsa del personaggio vengono generati automaticamente in base alla velocità e alla direzione dei controller. L'animazione delle braccia e della direzione della testa avviene usando la cinematica inversa¹⁴.

Le camere, dotate di uno schermo, possono essere afferrate, spostate e dai menù possono essere modificati i parametri come la lunghezza focale.

La timeline si trova sul menù del controller sinistro e permette di tornare indietro, riprodurre, registrare e controllare lo scorrere del tempo.

Le registrazioni fatte possono essere editate selezionando il tempo sulla timeline e andando a prendere una nuova camera e iniziando una nuova registrazione che verrà unita alla precedente.

È, infine, presente una “sala cinema” virtuale dotata di un grande schermo dove l'utente può vedere la visuale da tutte le camere presenti in scena. Sono presenti una timeline, il tasto di play e per tornare indietro. Questa stanza virtuale è stata pensata per la fase collaborativa, per mostrare a tutti i partecipanti la stessa visuale e poterne discutere.

I 20 filmmaker a cui è stato presentato l'applicativo, l'hanno valutato positivamente indicando che la realtà virtuale è utile per i processi produttivi di pre-visualizzazione collaborativi e distribuiti.

¹⁴ Animazione in base al posizionamento delle sole estremità dell'oggetto.

2.4. A Comparison of Immersive and Non Immersive VR for the Education of Filmmaking - articolo del 2022

L'articolo [38] si concentra principalmente nel fare il confronto tra la VR immersiva (IVR, quella dei visori) e la VR non immersiva (NIVR, la Desktop VR) utilizzate per insegnare l'arte della produzione cinematografica. Per confrontarle è stato sviluppato uno stesso contenuto per entrambe le tecnologie.

La trama utilizzata in entrambe le applicazioni VR arriva da Afandi's Stories, un cartone in stop-motion¹⁵ prodotto dallo Shanghai Animation Film Studio negli anni '80. Sono stati presi 3 minuti dal secondo episodio e trasformati in un'animazione in 3D. L'applicativo è stato sviluppato su Unity per Oculus Rift e HTC VIVE, per la IVR.

Entrambe le versioni posseggono le seguenti funzioni:

- Movimento libero della telecamera: gli utenti possono spostare e ruotare una telecamera liberamente nello spazio;
- Lenti zoom: gli utenti possono modificare la lunghezza focale e capire come influisce su prospettiva, profondità di campo...
- Controllo dell'esposizione: gli utenti possono regolare i parametri, ad esempio l'apertura dimensione, velocità dell'otturatore e valore ISO¹⁶, per modificare l'esposizione;
- Controllo del tempo: gli utenti possono far partire la riproduzione, mettere in pausa, avanzare rapidamente nella sequenza, riavvolgere rapidamente e saltare all'inizio/fine;

¹⁵ Tecnica di ripresa cinematografica consistente nello scattare una serie di fotogrammi di un oggetto o di un disegno bidimensionale cambiando ogni volta la sua posizione nello spazio.

¹⁶ La sensibilità del sensore della macchina fotografica, ovvero la quantità di luce che una pellicola riesce ad assorbire in un intervallo di tempo.

- Montaggio non lineare: gli utenti possono rivedere le parti finite e modificarle.

Entrambe le versioni, inoltre, sono costituite da quattro modalità. Nella fase 1, l'utente può guardare l'animazione completa da diverse angolazioni per pianificare le inquadrature. Può, successivamente, passare alle fasi 2 e 3 per modificare i parametri della camera, fare nuove inquadrature e rivedere quelle già fatte. Per modificarle basta tornare a un punto precedente sulla linea del tempo e poi sovrascrivere il contenuto esistente.

Quando un utente è soddisfatto del risultato, può passare alla fase 4 per terminare la sessione. In IVR l'utente può passare da una fase all'altra premendo il tasto trigger sul controller sinistro.

Per i fini della tesi, ci si concentrerà a descrivere la IVR.

Le funzioni dei pulsanti del controller cambiano automaticamente in base alla modalità corrente.

Nella "Shooting mode" la camera virtuale è direttamente controllata nei movimenti e nelle rotazioni dal controller destro. Lo schermo della camera è posizionato nell'angolo in alto a sinistra, in modalità non diegetica.



Figura 2.6 "Shooting mode" nella versione IVR.

Nella “Editing Mode” l'utente scorre le funzioni e utilizza il pulsante di trigger per eseguirle. Può agire sulla linea del per sovrascrive ciò che è già sulla timeline.

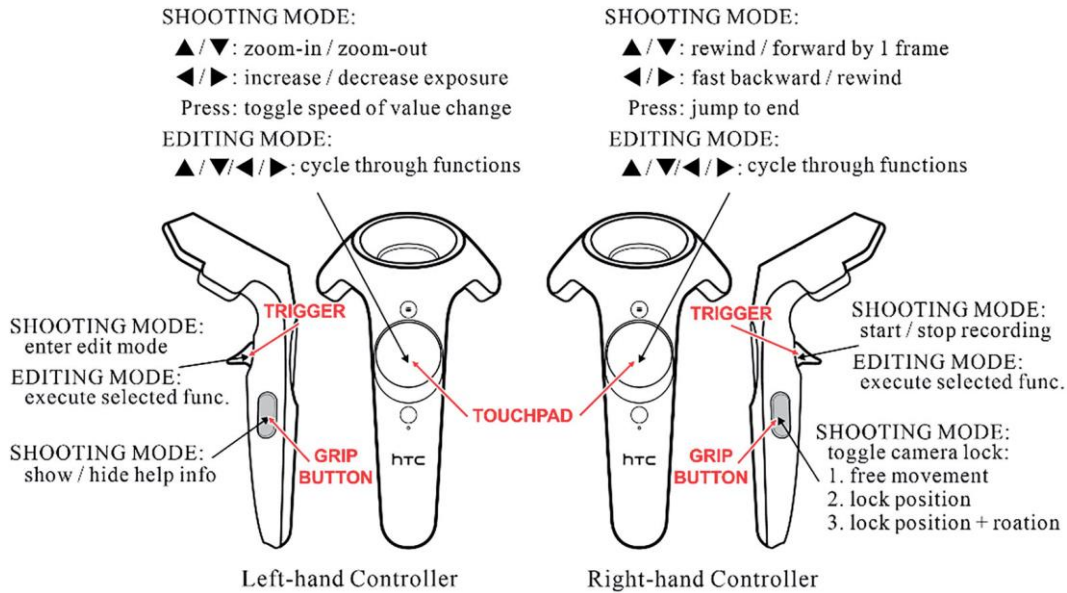


Figura 2.7 Funzioni dei tasti sui controller in base alla modalità corrente.

In conclusione, entrambe le tecnologie, IVR e INVR, sono risultate idonee a supportare efficacemente l'apprendimento della cinematografia.

2.5. Tвори – Storyboard

Tвори è un software VR commerciale con cui si possono creare rapidamente animatic, previs, prototipi di app XR e giochi e realizzare un intero film animato [39].

Si può animare in tempo reale o con i keyframe, importare modelli di oggetti, immagini e suoni. Si può creare una esperienza VR, esportare video in 2D e in 360. Il software è ancora in fase di sviluppo ed è in formato Beta, ma esiste anche una versione Pro Beta per le aziende con maggiori funzioni [40].

Tvori, come menzionato, è in grado di aiutare a progettare più elementi di pre-visualizzazione, tra queste funzionalità è presente anche quella per realizzare degli storyboard mentre si è immersi nel mondo virtuale, presentata a fine marzo 2021.

La creazione dell'ambiente avviene trascinando in scena degli oggetti pre-visualizzati in miniatura, vi è la possibilità di scolarli e ruotarli utilizzando i tasti dei controller. Si possono mettere in posa i personaggi utilizzando la cinematica inversa e i punti di controllo posti su questi. L'utente può inoltre importare la propria sceneggiatura sotto forma di pdf in scena e sfogliarla. Per aggiungere le camere bisogna selezionarle dall'elenco degli oggetti e trascinarle in scena. Ogni camera è dotata di uno schermo, slider per la focale e un tasto per ottenere lo screenshot dell'inquadratura. Ottenute le foto l'utente può aggiungere in scena un template¹⁷ per storyboard e decidere l'ordine in cui posizionarle al suo interno, non è quindi presente una timeline. A quel punto lo storyboard può essere esportato.



Figura 2.8 Funzionamento di Tvori per la creazione di storyboard [41].

¹⁷ Modello predefinito.

2.6. Storyboarder

Storyboarder è un software di nicchia, gratuito, sviluppato dallo studio statunitense Wonder Unit nel 2021 [42].

Lo strumento può essere utilizzato sia in versione NIVR e quindi sul PC, sia con un visore di realtà virtuale. Si descriverà la versione fruibile con il visore, le stesse funzioni sono disponibili in versione desktop.

Utilizzando un'applicazione per smartphone che permetta di scansionare gli ambienti in 3D, come ad esempio Polycam [43], ed esportando il file ottenuto in formato .glb¹⁸ si potrà importare nel software l'ambiente o location su cui si andrà a realizzare il proprio storyboard.

Per muoversi nell'ambiente in scala reale si possono usare i tasti dei joystick sui controller o camminare. Sul controller destro sono presenti i menù per poter interagire con l'applicazione. L'utente può aggiungere camere virtuali, luci, oggetti semplici e personaggi predefiniti, quest'ultimi possono essere modificati nell'aspetto fisico attraverso il menù che appare dopo averli selezionati. Possono essere messi in posa attraverso le loro armature sfruttando la cinematica inversa.



Figura 2.9 Possibile visuale mentre si utilizza l'applicazione in VR [44].

¹⁸ Rappresentazione in formato file binario di modelli 3D.

In generale non è presente la gravità e le camere sono dotate di un proprio display. Il parametro della lunghezza focale è modificabile dal menù sul controller destro, come lo sono quelli riferiti alla luce.

Come output¹⁹ si ottiene lo storyboard o volendo un animatic. Tutti gli screenshot fatti vengono salvati nell'applicazione sul computer, dove si potranno aggiungere informazioni come durata dell'inquadratura, i dialoghi, le azioni e altre note.

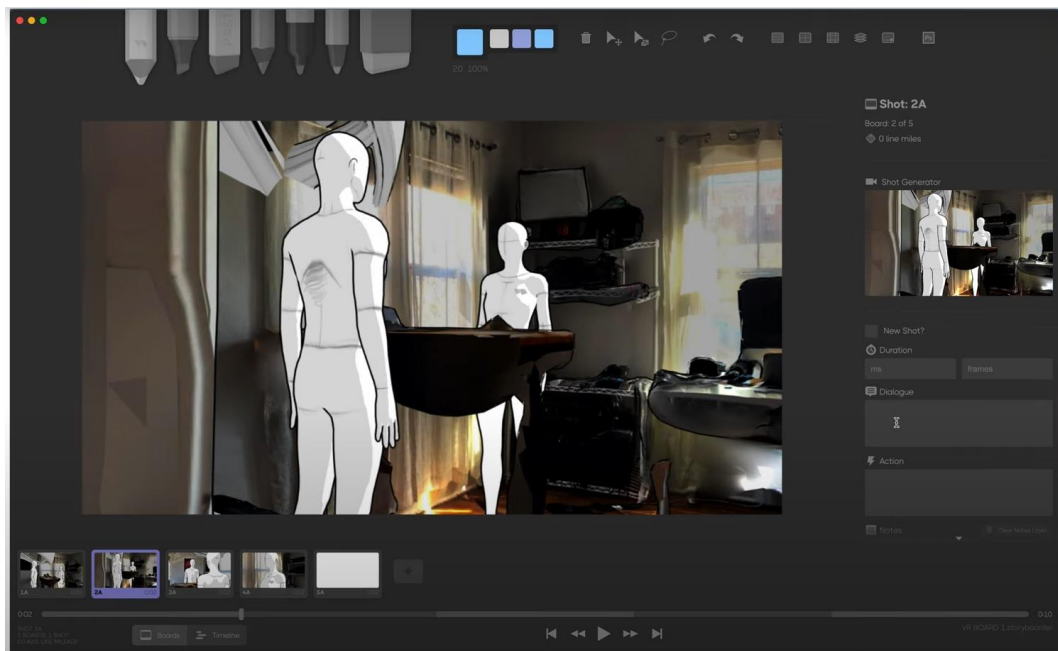


Figura 2.10 Sezione dell'applicazione in versione Desktop dove poter aggiungere le informazioni aggiuntive alle vignette ottenute in VR [44].

2.7. Gravity Sketch VR

Gravity Sketch, prodotto sviluppato da Gravity Sketch Limited, è uno strumento di design 3D e collaborazione disponibile per iPad, VR e Desktop [45]. Sono disponibili sia una versione business collaborativa a pagamento sia una gratuita individuale.

¹⁹ Il risultato finale.

Anche se Gravity Sketch è uno strumento nato per l'industria del design, viene utilizzato anche in diverse aree.

Nel 2018 l'attore e regista Doug Fahl ha condiviso un video su YouTube dove mostrava come usare il software per ottenere uno storyboard in VR [47].

Nel video mostra come sfruttare al meglio l'applicazione. Si può, infatti, disegnare e modellare velocemente usando i controller e delle primitive. Tutto ciò che è presente in scena può essere scalato per avere una visione sia dall'alto sia all'interno del mondo. Vi è la possibilità di importare nell'applicazione i propri modelli creati utilizzando altri programmi.

Si possono aggiungere in scena e mettere in posa, sfruttando la cinematica inversa, dei personaggi stilizzati. Vicino ai controller ci sono dei menù per interagire con l'applicazione. Sono presenti dei preset di illuminazione dell'ambiente e un oggetto a forma di torcia che emana luce e proietta le ombre.

È possibile aggiungere una camera in scena, accompagnata dal proprio display e dagli slider e tasti a lei associati per poter fare delle foto che saranno salvate nel PC.



Figura 2.11 Doug Fahl mostra come poter sfruttare Gravity Sketch per realizzare uno storyboard [47].

2.8. Confronto delle caratteristiche dei software analizzati

	1	2	3	4	5	6	7
<i>In miniatura</i>	x		x	x			x
<i>In scala reale</i>	x	x			x	x	x
<i>Creazione ambiente</i>	x	x	x		x	x	x
<i>Animazione personaggi con IK o keyframe</i>		x	x		x	x	x
<i>Animazione personaggi con motion capture</i>		x					
<i>Camere con display</i>	x	x	x	x	x	x	x
<i>Storyboard</i>	x	x			x	x	x
<i>Animatic</i>		x	x	x		x	
<i>Controller visibili</i>	x	x	x	x	x	x	x
<i>Raggio dai controller</i>		x	x			x	
<i>Menù sul controller</i>	x	x	x			x	x
<i>Teletrasporto</i>		x					
<i>Mondo spostabile, ruotabile e scalabile</i>	x		x				
<i>Luci</i>		x				x	x
<i>Movimenti di camera con attrezzature (es. crane, dolly)</i>		x	x				
<i>Impostazioni camera modificabili da UI</i>		x	x	x	x	x	x
<i>Impostazioni camera modificabili attraverso gestures</i>	x						
<i>Timeline</i>		x	x	x			
<i>Collaborativo</i>			x				x

Tabella 2.1 Confronto caratteristiche dei software e strumenti analizzati.

1 = Empowering Creative People: Virtual Reality for Previsualization

2 = VR as a Content Creation Tool for Movie Previsualisation

3 = A Collaborative Previsualization Tool for Filmmaking in Virtual in Virtual Reality

4 = A Comparison of Immersive and Non Immersive VR for the Education of Filmmaking

5 = Tвори

6 = Storyboarder

7 = Gravity Sketch VR

Gli strumenti e i software analizzati posseggono diverse funzionalità e caratteristiche in comune, come evidenziato nella Tabella 2.1. Alcune di queste, come le camere virtuali con display, la possibilità di creare l'ambiente a proprio piacimento, i controller sempre visibili in scena, una UI disponibile sotto forma di menù vicino ai controller, le si ritrovano in tutti gli applicativi o quasi. Altre funzionalità sono invece meno presenti, ma comunque importanti per lo scopo dei singoli strumenti.

Partendo da questa analisi e da quanto verrà descritto nel seguente capitolo sono stati definiti i requisiti dell'applicazione sviluppata per questa tesi.

3. Requisiti

Durante la fase che ha preceduto lo sviluppo dell'applicazione sono stati definiti i requisiti. Questi sono dipesi dall'analisi dello stato dell'arte, da un applicativo già esistente, sviluppato nel 2022 per una tesi di Laurea Magistrale, e dalle risposte ad un questionario da parte di un gruppo di studenti, dove si chiedeva di classificare dei requisiti in base al metodo MoSCoW [48].

3.1. Applicazione di partenza

L'applicazione sviluppata per questa tesi ha come punto di partenza una versione Desktop della stessa [49]. Quest'ultima, sviluppata utilizzando il motore grafico Unity, permette di:

- popolare il proprio ambiente di personaggi e oggetti selezionabili da una lista di oggetti;
- gestire gli stati e le possibili azioni che questi possono compiere, attraverso delle animazioni, per interagire con ciò che li circonda;
- gestire la camera, modificandone posizione, orientamento, lunghezza focale per effettuare gli screenshot;
- gestire luci di scena, delle quali si può modificare l'intensità;
- gestire la durata e la sequenzialità delle azioni attraverso una barra del tempo;
- generare automaticamente la descrizione delle vignette dello storyboard che si crea.

Per interagire con l'applicativo vengono usati i tasti e gli slider presenti nell'interfaccia grafica, la tastiera, il mouse e il microfono.

L'output che si ottiene è un file .html contenente lo storyboard con campi di testo editabili e una bozza di animatic, il quale mostra a tempo e in successione le vignette generate, in base alla durata di ognuna.



Figura 3.1 Interfaccia grafica dell'applicazione per Desktop nella fase di costruzione della scena.



Figura 3.2 Interfaccia grafica dell'applicazione per Desktop nel momento in cui si vuole far eseguire un'azione ad un personaggio.

3.2. Metodo MoSCoW

Il metodo MoSCoW, noto anche come analisi MoSCoW, è una tecnica di definizione delle priorità inventata da Dai Clegg nel 1994.

È una tecnica usata in diversi ambienti, tra cui lo sviluppo di software, che permette ai destinatari di un servizio o applicazione di esprimere l'importanza che essi attribuiscono a ciascun requisito presentato dagli sviluppatori.

I requisiti vengono quindi divisi in quattro aree chiave:

- *Must (Deve)*: requisiti indispensabili, decisivi per il corretto funzionamento del sistema/applicazione.
- *Should (Dovrebbe)*: requisiti che sono considerati importanti, ma non essenziali.
- *Could (Potrebbe)*: requisiti che potrebbero essere utili ma che hanno un impatto meno significativo.
- *Would (Piacerebbe)*: requisiti con minima importanza e minimo valore aggiunto, da riconsiderare solo dopo aver sviluppato tutti gli altri. La “W” può anche essere interpretata con “*Won't have*”.

Il metodo è stato usato per chiedere agli studenti, a cui è stato sottoposto il questionario, di valutare la priorità dei vari requisiti.

3.3. Questionario

Il questionario A.1, creato attraverso l'applicazione Moduli di Google, è stato somministrato ad un gruppo di studenti dei corsi di Laurea Magistrale in Ingegneria Del Cinema E Dei Mezzi Di Comunicazione e in Ingegneria Informatica con orientamento Grafica e Multimedia del Politecnico di Torino. Ogni persona poteva rispondere una sola volta.

Le domande sono state ragionate in base alle caratteristiche all'applicazione di partenza e a quelle presenti nei software analizzati nel capitolo "Stato dell'Arte".

Il modulo è stato suddiviso in tre sezioni:

1. Nella prima si chiedeva, oltre all'età, il livello di conoscenza degli argomenti storyboard e realtà virtuale.
2. La seconda sezione si concentrava ad utilizzare il metodo MoSCoW, dando la possibilità di rispondere con Must, Should, Could e Would ad ogni domanda. È stata suddivisa a sua volta in sottosezioni in base al tipo di requisito.
3. Nell'ultima sezione era presente un'unica domanda aperta, facoltativa, in cui si chiudeva di indicare eventuali requisiti non considerati, ma ritenuti dall'utente importanti.

Gli studenti che hanno partecipato al questionario sono stati 35 in totale, di una età compresa tra i 22 e i 28 anni.

Le domande presentate nel questionario e le relative risposte, rappresentate in grafici a torta, sono riportate nelle pagine che seguono.

Prima sezione:

1. Quante volte ti è capitato di realizzare degli storyboard?

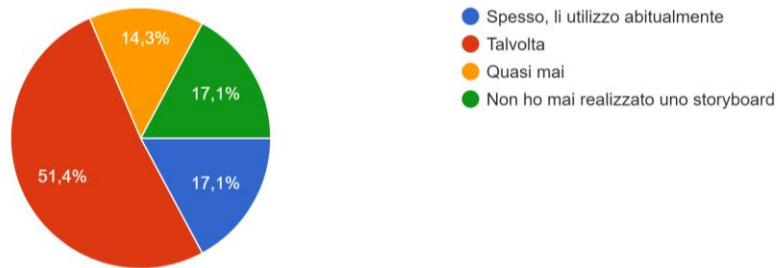


Grafico 3.1 Risposte in % alla domanda 1 del questionario MoSCoW.

2. Quali strumenti utilizzi solitamente per realizzare uno storyboard?

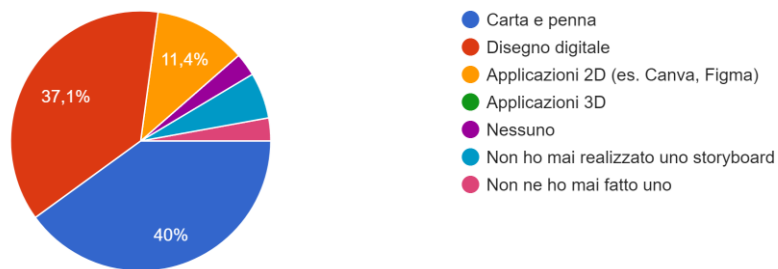


Grafico 3.2 Risposte in % alla domanda 2 del questionario MoSCoW.

3. Conosci la realtà virtuale (VR)?

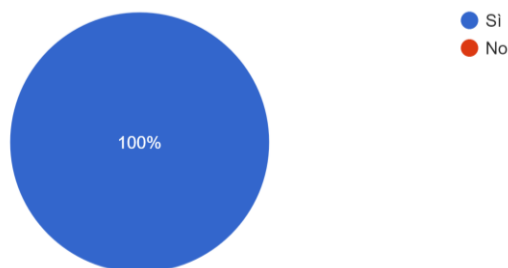


Grafico 3.3 Risposte in % alla domanda 3 del questionario MoSCoW.

4. Quanto spesso utilizzi applicazioni VR?

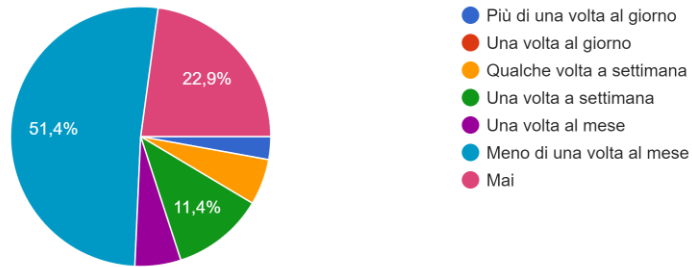


Grafico 3.4 Risposte in % alla domanda 4 del questionario MoSCoW.

5. Quale dispositivo usi più spesso per applicazioni VR? Se è un dispositivo dedicato, indica il modello nel campo "altro" (es. Meta Quest 2)

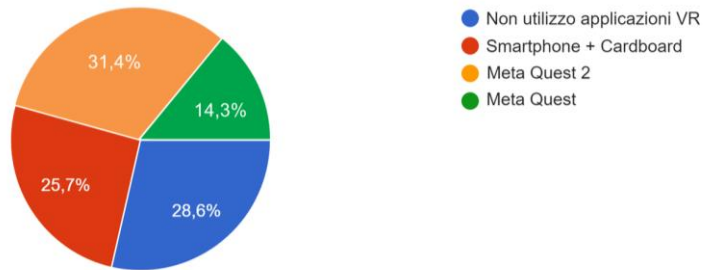


Grafico 3.5 Risposte in % alla domanda 5 del questionario MoSCoW.

6. Hai utilizzato/stai utilizzando la realtà virtuale per un corso universitario?

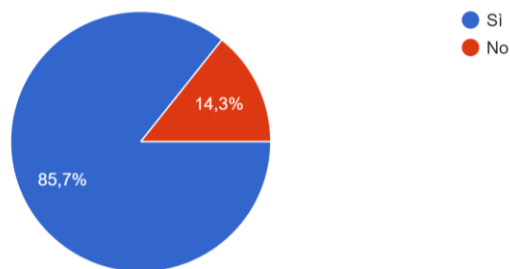


Grafico 3.6 Risposte in % alla domanda 6 del questionario MoSCoW.

Seconda sezione:

Requisiti per la creazione/modifica delle scene:

7. Inserire oggetti e personaggi nella scena, scegliendoli da una lista, e definirne posizione e orientamento.

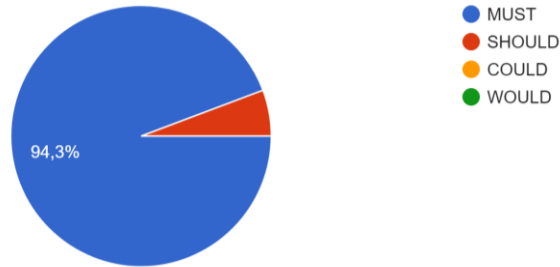


Grafico 3.7 Risposte in % alla domanda 7 del questionario MoSCoW.

8. Rinominare i personaggi e gli oggetti in scena.

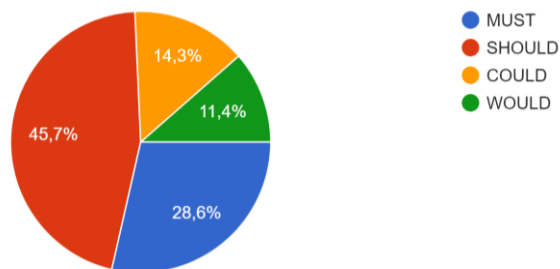


Grafico 3.8 Risposte in % alla domanda 8 del questionario MoSCoW.

9. Eliminare oggetti o personaggi da una scena.

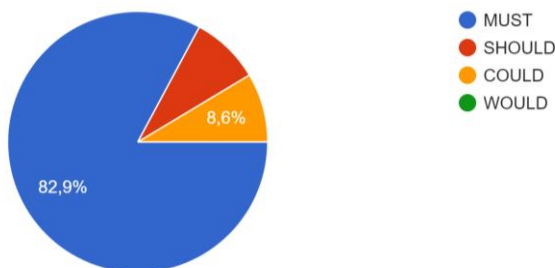


Grafico 3.9 Risposte in % alla domanda 9 del questionario MoSCoW.

10. Possibilità di poter salvare una scena creata e poterla ricaricare in un secondo momento.

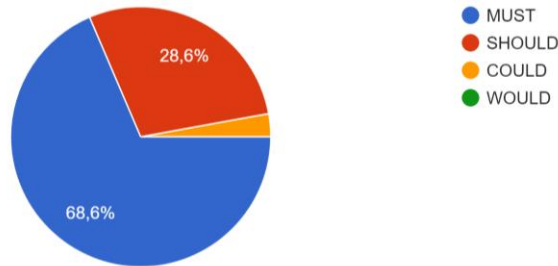


Grafico 3.10 Risposte in % alla domanda 10 del questionario MoSCoW.

Requisiti per la gestione dei personaggi:

11. Possibilità di definire le pose dei personaggi manualmente, modificando l'armatura del modello.

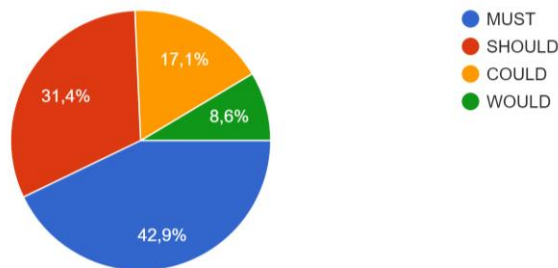


Grafico 3.11 Risposte in % alla domanda 11 del questionario MoSCoW.

12. Possibilità di animare un personaggio applicando al modello delle pose predefinite, selezionabili da una libreria.

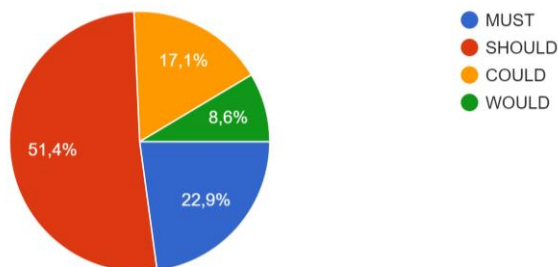


Grafico 3.12 Risposte in % alla domanda 12 del questionario MoSCoW.

13. Possibilità di controllare i personaggi della scena muovendoli nello spazio e animandoli con dei comandi, come all'interno di un videogioco.

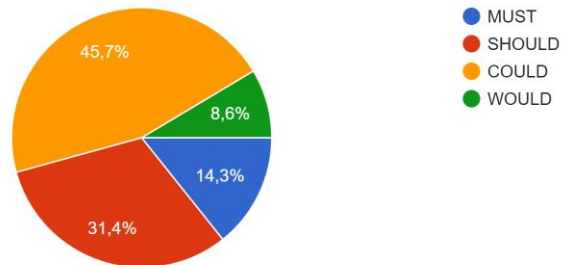


Grafico 3.13 Risposte in % alla domanda 13 del questionario MoSCoW.

14. Possibilità di interazione tra personaggio e personaggio.

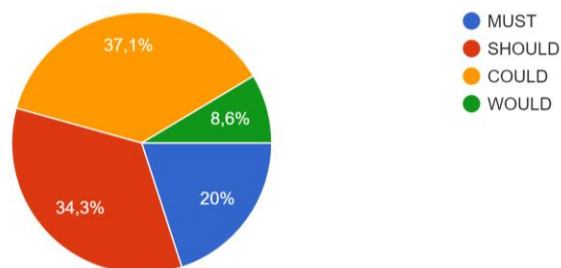


Grafico 3.14 Risposte in % alla domanda 14 del questionario MoSCoW.

15. Possibilità di interazione tra un personaggio e gli altri oggetti presenti nella scena.

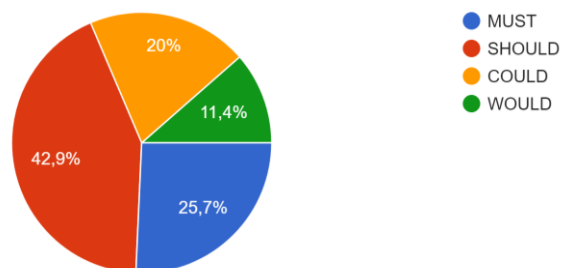


Grafico 3.15 Risposte in % alla domanda 15 del questionario MoSCoW.

Requisiti per la gestione delle camere virtuali:

16. L'applicazione deve permettere l'aggiunta e il movimento di camere virtuali per inquadrare la scena da diversi punti nello spazio ed eseguire gli screenshot che costituiscono le vignette dello storyboard.

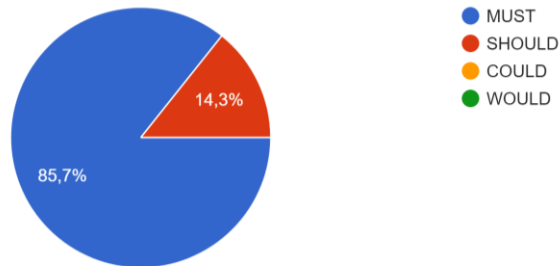


Grafico 3.16 Risposte in % alla domanda 16 del questionario MoSCoW.

17. Permettere di visualizzare ciò che è inquadrato da una camera virtuale all'interno dell'applicazione con una preview in tempo reale.

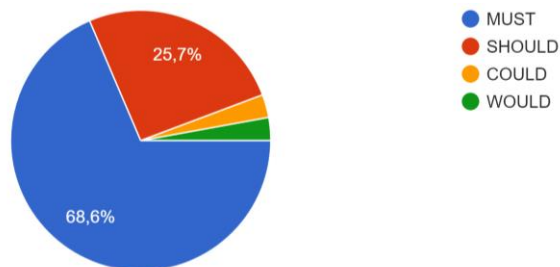


Grafico 3.17 Risposte in % alla domanda 17 del questionario MoSCoW.

18. Per ogni camera deve essere possibile settare i parametri (fov, lunghezza focale, sensor type...).

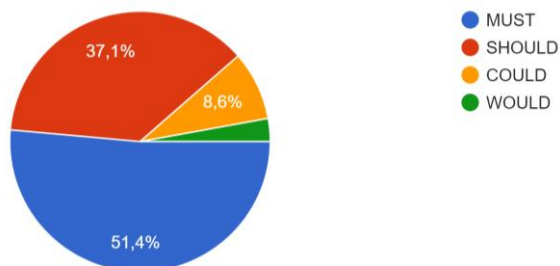


Grafico 3.18 Risposte in % alla domanda 18 del questionario MoSCoW.

Requisiti riguardanti le luci:

19. Possibilità di inserire nella scena delle luci virtuali.

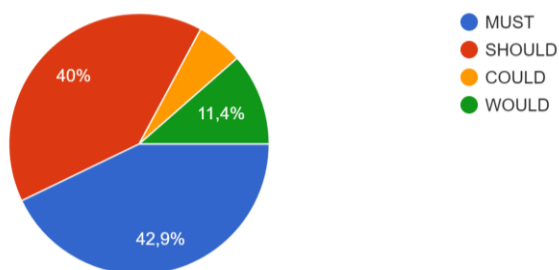


Grafico 3.19 Risposte in % alla domanda 19 del questionario MoSCoW.

20. Modificare i parametri delle luci (potenza, area, colore).

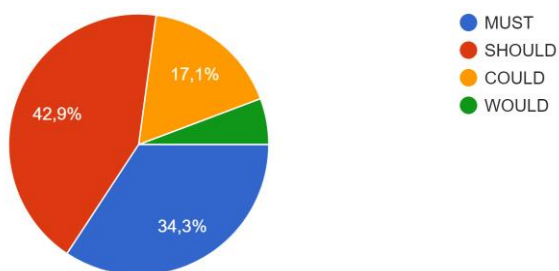


Grafico 3.20 Risposte in % alla domanda 20 del questionario MoSCoW.

21. Possibilità di avere un preset di illuminazione per ambienti esterni.

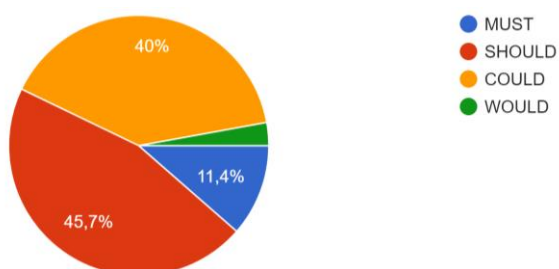


Grafico 3.21 Risposte in % alla domanda 21 del questionario MoSCoW.

Requisiti riguardanti la creazione dello storyboard o di una pre-visualizzazione:

22. Creare degli screenshot della scena che rappresentano le vignette dello storyboard.

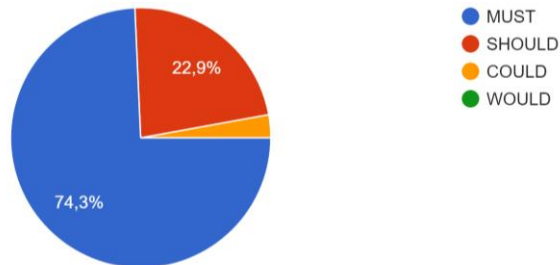


Grafico 3.22 Risposte in % alla domanda 22 del questionario MoSCoW.

23. Dare la possibilità di definire la durata di un'azione o di un'inquadratura.

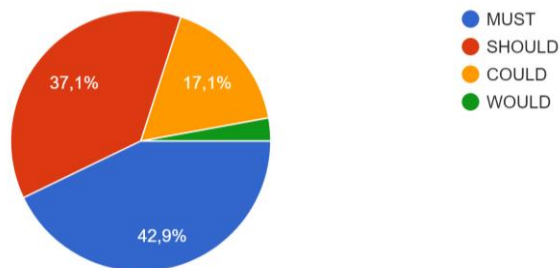


Grafico 3.23 Risposte in % alla domanda 23 del questionario MoSCoW.

24. Modificare lo storyboard finale riordinando le immagini e cambiando le descrizioni.

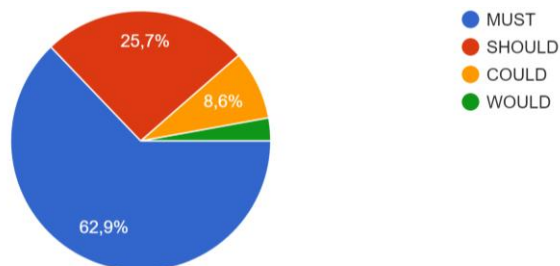


Grafico 3.24 Risposte in % alla domanda 24 del questionario MoSCoW.

25. Prevedere uno o più livelli di UNDO per tornare alla condizione dello “screenshot precedente”.

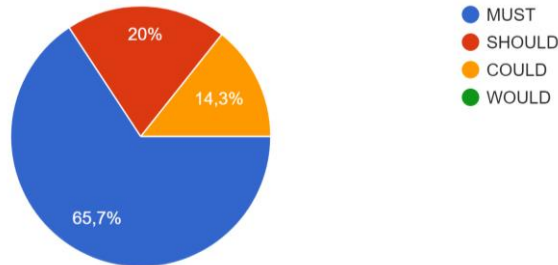


Grafico 3.25 Risposte in % alla domanda 25 del questionario MoSCoW.

26. Possibilità di definire la durata di un’azione rappresentata da una vignetta dello storyboard.

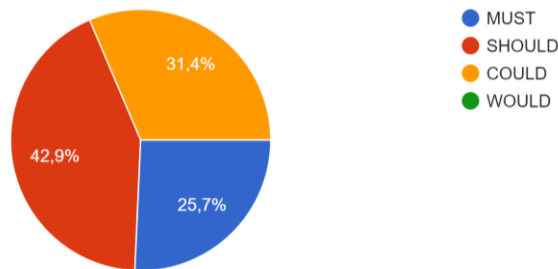


Grafico 3.26 Risposte in % alla domanda 26 del questionario MoSCoW.

27. Rendere possibile la visualizzazione dello storyboard prima del completamento.

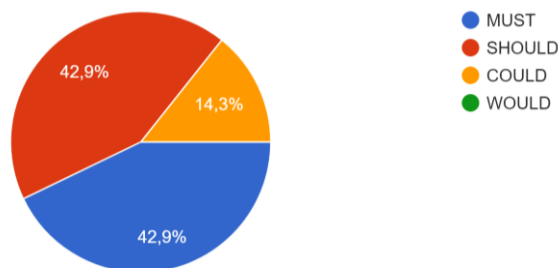


Grafico 3.27 Risposte in % alla domanda 27 del questionario MoSCoW.

28. Rendere possibile la visualizzazione dello storyboard una volta completato.

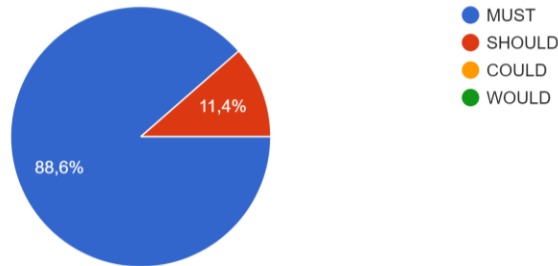


Grafico 3.28 Risposte in % alla domanda 28 del questionario MoSCoW.

29. Creare una pre-visualizzazione delle azioni svolte nell'applicazione attraverso un video.

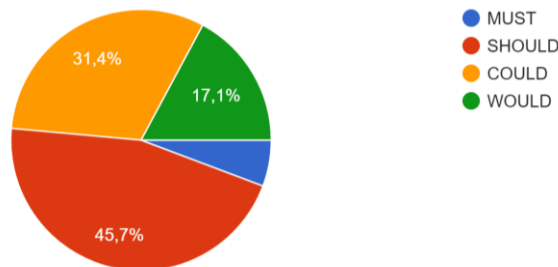


Grafico 3.29 Risposte in % alla domanda 29 del questionario MoSCoW.

30. Avere una descrizione, generata automaticamente, che accompagna ogni inquadratura dello storyboard.

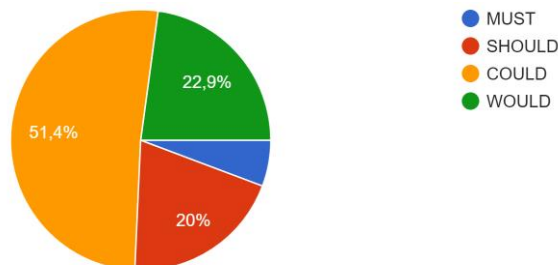


Grafico 3.30 Risposte in % alla domanda 30 del questionario MoSCoW.

31. Possibilità per l'utente di aggiungere all'interno dell'inquadratura delle indicazioni che specifichino i movimenti di camera e dei personaggi (es. Freccette che indichino le direzioni del movimento, zoom in/out ...).

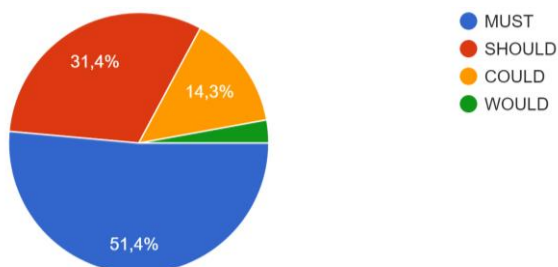


Grafico 3.31 Risposte in % alla domanda 31 del questionario MoSCoW.

Requisiti riguardanti il sistema di interazione e visualizzazione dell'utente:

32. L'applicazione deve supportare un'interfaccia vocale.

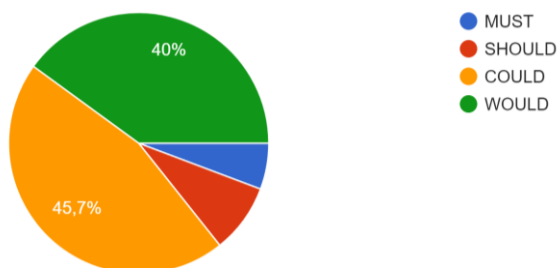


Grafico 3.32 Risposte in % alla domanda 32 del questionario MoSCoW.

33. L'applicazione deve supportare un'interfaccia gestuale.

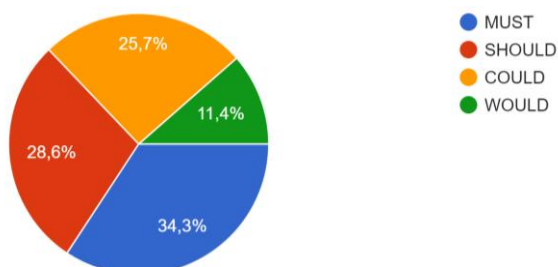


Grafico 3.33 Risposte in % alla domanda 33 del questionario MoSCoW.

34. L'utente deve avere la possibilità di visualizzare e modificare la scena in scale differenti (tabletop, scala 1:1 in prima persona).

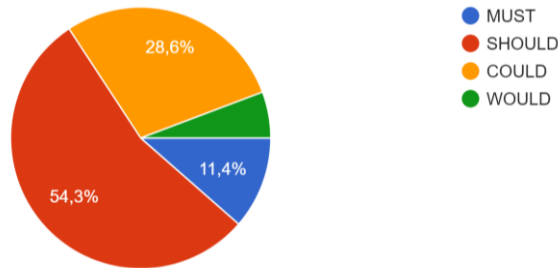


Grafico 3.34 Risposte in % alla domanda 34 del questionario MoSCoW.

Requisiti riguardanti la libertà dell'utente:

35. Il sistema permette massima libertà all'utente di poter scegliere qualsiasi azione tra quelle disponibili da far eseguire ai personaggi, in qualsiasi momento a prescindere dal contesto.

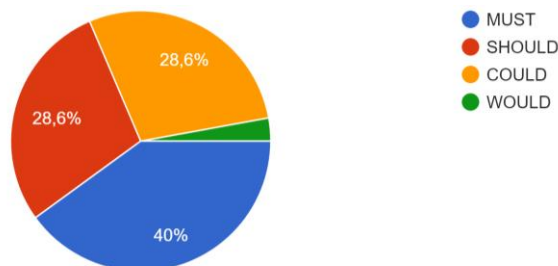


Grafico 3.35 Risposte in % alla domanda 35 del questionario MoSCoW.

36. Il sistema, in base al contesto, permette di selezionare solo determinate azioni che sono coerenti ad esso.

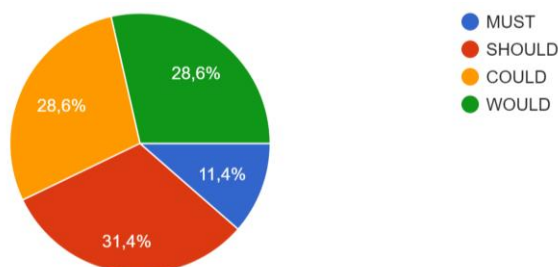


Grafico 3.36 Risposte in % alla domanda 36 del questionario MoSCoW.

37. Il sistema deve verificare la coerenza temporale delle azioni tra vignette consecutive.

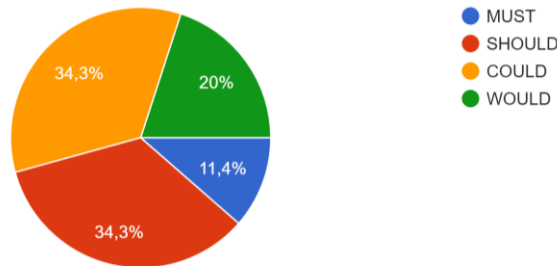


Grafico 3.37 Risposte in % alla domanda 37 del questionario MoSCoW.

Terza sezione:

38. Ci sono eventuali requisiti che non abbiamo considerato e che secondo te sarebbe necessario aggiungere? Se sì indica nel campo sottostante quali sono e che priorità assegneresti loro secondo il metodo MoSCoW.

Una delle risposte più significative è stata:

“La possibilità di poter associare ad ogni personaggio il proprio character sheet "carta d'identità del personaggio" poiché il suo modo di essere influenza direttamente il suo comportamento. MUST

La possibilità di associare ad ogni inquadratura una o più reference che siano in formato testuale, immagine o video. SHOULD

Utilizzare strumenti di AI per realizzare preset su cui poter iniziare il lavoro, attraverso una descrizione testuale. WOULD

La possibilità di poter navigare su internet dall'applicazione per avere la possibilità di avere delle reference durante il processo creativo. COULD”

3.4. Analisi risultati questionario MoSCoW

Il contenuto delle 35 risposte a ciascuna domanda della seconda sezione del questionario è stato inserito in un file Excel al fine di condurre un'analisi per poter classificare i requisiti secondo il metodo MoSCoW.

Sul foglio Excel ogni riga rappresenta una diversa domanda, mentre ogni colonna contiene una delle 35 risposte date.

Le domande, in primo luogo, sono state riordinate in ordine decrescente in base alla percentuale di Must ricevuti. Per poter assegnare ogni requisito nella sezione corrispondente (Must (M), Should (S), Could (C) o Would(W)) si è, in seguito, posta la seguente condizione ad ogni riga:

IF {[M% ≥ 50%] OR [(M% >S%) AND (M% + S% ≥ 50%)]} => Must

IF (M% + S% ≥ 50%) => Should

IF (C% ≥ W%) => Should

IF (W% > C%) => Would

In base a queste condizioni i risultati, ordinati, sono stati i seguenti:

MUST

1. Inserire oggetti e personaggi nella scena, scegliendoli da una lista, e definirne posizione e orientamento.
2. Rendere possibile la visualizzazione dello storyboard una volta completato.
3. L'applicazione deve permettere l'aggiunta e il movimento di camere virtuali per inquadrare la scena da diversi punti nello spazio ed eseguire gli screenshot che costituiscono le vignette dello storyboard.
4. Eliminare oggetti o personaggi da una scena.
5. Creare degli screenshot della scena che rappresentano le vignette dello storyboard.

6. Permettere di visualizzare ciò che è inquadrato da una camera virtuale all'interno dell'applicazione con una preview in tempo reale.
7. Possibilità di poter salvare una scena creata e poterla ricaricare in un secondo momento.
8. Prevedere uno o più livelli di UNDO per tornare alla condizione dello "screenshot precedente".
9. Modificare lo storyboard finale riordinando le immagini e cambiando le descrizioni.
10. Per ogni camera deve essere possibile settare i parametri (fov, lunghezza focale, sensor type...).
11. Possibilità per l'utente di aggiungere all'interno dell'inquadratura delle indicazioni che specifichino i movimenti di camera e dei personaggi (es. Freccette che indichino le direzioni del movimento, zoom in/out ...).
12. Dare la possibilità di definire la durata di un'azione o di un'inquadratura.
13. Possibilità di definire le pose dei personaggi manualmente, modificando l'armatura del modello.
14. Possibilità di inserire nella scena delle luci virtuali.
15. Il sistema permette massima libertà all'utente di poter scegliere qualsiasi azione tra quelle disponibili da far eseguire ai personaggi, in qualsiasi momento a prescindere dal contesto.
16. L'applicazione deve supportare un'interfaccia gestuale.

SHOULD

1. Rendere possibile la visualizzazione dello storyboard prima del completamento.
2. Modificare i parametri delle luci (potenza, area, colore).
3. Rinominare i personaggi e gli oggetti in scena.
4. Possibilità di interazione tra un personaggio e gli altri oggetti presenti nella scena.
5. Possibilità di definire la durata di un'azione rappresentata da una vignetta dello storyboard.

6. Possibilità di animare un personaggio applicando al modello delle pose predefinite, selezionabili da una libreria.
7. Possibilità di interazione tra personaggio e personaggio.
8. L'utente deve avere la possibilità di visualizzare e modificare la scena in scale differenti (tabletop, scala 1:1 in prima persona).
9. Possibilità di avere un preset di illuminazione per ambienti esterni.
10. Creare una pre-visualizzazione delle azioni svolte nell'applicazione attraverso un video.

COULD

1. Possibilità di controllare i personaggi della scena muovendoli nello spazio e animandoli con dei comandi, come all'interno di un videogioco.
2. Il sistema deve verificare la coerenza temporale delle azioni tra vignette consecutive.
3. Il sistema, in base al contesto, permette di selezionare solo determinate azioni che sono coerenti ad esso.
4. Avere una descrizione, generata automaticamente, che accompagna ogni inquadratura dello storyboard.
5. L'applicazione deve supportare un'interfaccia vocale.

Non sono presenti ***WOULD***.

La classifica dei requisiti è stata, infine, analizzata ponendola a confronto con i requisiti già presenti nell'applicazione di partenza e in base al grado di difficoltà degli stessi. Inoltre, alcuni requisiti, come ad esempio *“Il sistema permette massima libertà all'utente di poter scegliere qualsiasi azione tra quelle disponibili da far eseguire ai personaggi, in qualsiasi momento a prescindere dal contesto”* e *“Il sistema, in base al contesto, permette di selezionare solo determinate azioni che sono coerenti ad esso”*, erano in opposizione tra loro, quindi alcune scelte escludevano altre.

La condizione di ordinamento data dalla sezione di appartenenza (Must, Should, Could) ha comunque influito maggiormente sulla classifica finale dei requisiti, dove quelli fondamentali, per l'applicazione che si voleva andare a sviluppare, precedono i meno significativi.

La classifica finale è risultata essere la seguente:

1. Inserire oggetti e personaggi nella scena, scegliendoli da una lista, e definirne posizione e orientamento.
2. Rendere possibile la visualizzazione dello storyboard una volta completato
3. Eliminare oggetti o personaggi da una scena.
4. Creare degli screenshot della scena che rappresentano le vignette dello storyboard.
5. Permettere di visualizzare ciò che è inquadrato da una camera virtuale all'interno dell'applicazione con una preview in tempo reale.
6. Dare la possibilità di definire la durata di un'azione o di un'inquadratura.
7. Per ogni camera deve essere possibile settare i parametri (fov, lunghezza focale, sensor type...).
8. L'applicazione deve permettere l'aggiunta e il movimento di camere virtuali per inquadrare la scena da diversi punti nello spazio ed eseguire gli screenshot che costituiscono le vignette dello storyboard.
9. Possibilità di poter salvare una scena creata e poterla ricaricare in un secondo momento.
10. Possibilità di inserire nella scena delle luci virtuali.
11. Prevedere uno o più livelli di UNDO per tornare alla condizione dello "screenshot precedente".
12. Modificare lo storyboard finale riordinando le immagini e cambiando le descrizioni.
13. Rinominare i personaggi e gli oggetti in scena.
14. Possibilità di interazione tra un personaggio e gli altri oggetti presenti nella scena.
15. Possibilità di interazione tra personaggio e personaggio.
16. Modificare i parametri delle luci (potenza, area, colore).

17. Possibilità di definire la durata di un'azione rappresentata da una vignetta dello storyboard.
18. Possibilità di avere un preset di illuminazione per ambienti esterni.
19. Rendere possibile la visualizzazione dello storyboard prima del completamento.
20. Possibilità di animare un personaggio applicando al modello delle pose predefinite, selezionabili da una libreria.
21. Possibilità di controllare i personaggi della scena muovendoli nello spazio e animandoli con dei comandi, come all'interno di un videogioco.
22. Il sistema, in base al contesto, permette di selezionare solo determinate azioni che sono coerenti ad esso.
23. Avere una descrizione, generata automaticamente, che accompagna ogni inquadratura dello storyboard.
24. L'applicazione deve supportare un'interfaccia vocale.
25. Possibilità per l'utente di aggiungere all'interno dell'inquadratura delle indicazioni che specifichino i movimenti di camera e dei personaggi (es. Frece che indichino le direzioni del movimento, zoom in/out ...).
26. Creare una pre-visualizzazione delle azioni svolte nell'applicazione attraverso un video.
27. L'utente deve avere la possibilità di visualizzare e modificare la scena in scale differenti (tabletop, scala 1:1 in prima persona).
28. Il sistema deve verificare la coerenza temporale delle azioni tra vignette consecutive.
29. Il sistema permette massima libertà all'utente di poter scegliere qualsiasi azione tra quelle disponibili da far eseguire ai personaggi, in qualsiasi momento a prescindere dal contesto.
30. L'applicazione deve supportare un'interfaccia gestuale.
31. Possibilità di definire le pose dei personaggi manualmente, modificando l'armatura del modello.

In base a queste condizioni è stata sviluppata l'applicazione per la VR, che verrà descritta nel capitolo "Sviluppo dell'applicazione". In particolare, sono stati sviluppati i primi 24 punti dell'elenco.

4. Tecnologie utilizzate

Per lo sviluppo dell'applicativo sono state usate differenti tecnologie che verranno approfondite in questo capitolo.

4.1. Oculus Rift CV1

L'Oculus Rift CV1 [50], noto come Oculus Rift, è stato annunciato a gennaio 2016 e rilasciato a marzo dello stesso anno. È dotato di due display OLED, di risoluzione 1080 x 1200 pixel per occhio, un refresh rate di 90 Hz, un FoV di 110°, accelerometro, giroscopio e di cuffie integrate per l'audio spazializzato.



Figura 4.1 Headset Oculus Rift CV1

Per funzionare ha bisogno di essere collegato ad un potente PC dotato di almeno 3 porte USB 3.0 (2 per i sensori e 1 per l'headset) e 1 porta HDMI nella scheda grafica.

I sensori chiamati “*Oculus Rift Constellation sensor*” consentono i 6 DoF con bassa latenza e altissima precisione sfruttando la tecnologia ottica a infrarossi.



Figura 4.2 Sensore Oculus Rift Constellation

Nella sua presentazione iniziale il sistema veniva utilizzato solo per tracciare l’headset e bastava quindi un singolo sensore. Veniva venduto, inoltre, con l’Oculus Remote fino all’introduzione a fine anno dei controller Oculus Touch.



Figura 4.3 Controller Oculus Touch.

La mezza luna in ciascun controller contiene un insieme di LED infrarossi e consente ai controller di essere tracciati dal sistema di tracciamento Constellation. Ogni controller è alimentato da una pila alcalina di tipo AA [51].

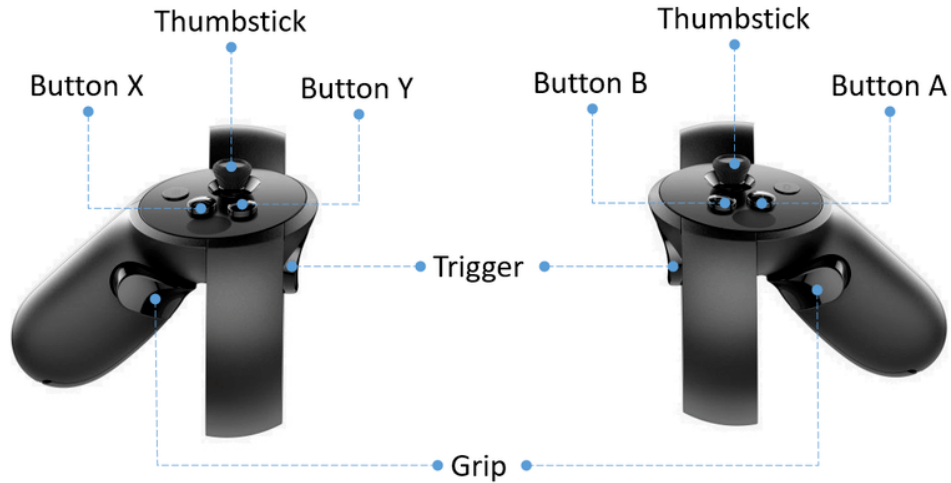


Figura 4.4 Tasti controller Oculus Touch [52].

Nel momento in cui si utilizzano solo due sensori il corpo dell'utente può fare da barriera, occultando a questi le informazioni ottiche. Per evitare il problema è consigliato l'utilizzo di tre o quattro sensori che consentono una rotazione a 360° senza rischio di perdita del tracking.

4.2. Unity

Unity è un motore grafico multipiattaforma sviluppato da Unity Technologies [53]. Può essere utilizzato per sviluppare videogiochi e altri contenuti interattivi in 3D o 2D. La prima versione risale al 2005.

L'editor di Unity è supportato su Windows, macOS e sulla piattaforma Linux, mentre il motore stesso attualmente supporta la creazione di giochi per oltre 19 piattaforme diverse, tra cui mobile, desktop, console e realtà virtuale.

Nel contesto 2D, Unity consente l'importazione di sprite²⁰ e un avanzato motore di rendering 2D. Per le applicazioni 3D permette di specificare la compressione delle texture, il MIP map²¹, per aumentare la velocità di rendering e di ridurre gli artefatti, e le impostazioni di risoluzione per ogni piattaforma supportata dal motore del gioco. Fornisce, inoltre, il supporto per il bump mapping²², il reflection mapping²³, il parallax mapping²⁴, l'ambient occlusion²⁵ e le ombre dinamiche mediante shadow mapping²⁶.

È possibile scegliere tra due pipeline di rendering: High Definition Render Pipeline (HDRP) e Universal Render Pipeline (URP, precedentemente noto come LWRP), oltre alla pipeline di rendering incorporata.

L'interfaccia è costituita da diversi pannelli, posizionabili a proprio piacimento sullo schermo:

- *Hierarchy*: contenente la lista di GameObject (oggetti) presenti nella scena;
- *Scene*: in cui appaiono gli oggetti, sui quali si può interagire posizionandoli nell'ambiente;
- *Game*: pannello di output che mostra il gioco in esecuzione nel momento in cui si preme play;
- *Inspector*: dove si possono trovare tutte le informazioni riguardanti il GameObject selezionato, sono presenti gli script ad esso collegato e le sue proprietà definite "componenti";
- *Project*: contenente tutti i file del progetto;

²⁰ Immagine in grafica raster, generalmente bidimensionale, che fa parte di una scena più grande e che può essere spostata in maniera indipendente rispetto ad essa. Può essere sia statica che dinamica.

²¹ Raccolta ottimizzata di immagini bitmap precalcolate associate ad una texture principale.

²² Tecnica di rendering dei materiali che aumenta la complessità degli oggetti realizzati senza effettivamente aumentare il numero di poligoni che compongono l'oggetto di partenza.

²³ Tecnica di lighting per approssimare l'aspetto di una superficie riflettente per mezzo di una texture precalcolata.

²⁴ Tecnica che permette di avere un'illusione di profondità negli oggetti grazie ad effetti di parallasse che seguono il cambio di prospettiva.

²⁵ Metodo di ombreggiatura che contribuisce a conferire realismo ai modelli di riflessione locale in quanto tiene conto dell'attenuazione luminosa in prossimità di volumi occlusi.

²⁶ Processo con il quale vengono calcolate e renderizzate le ombre.

- *Console*: altro pannello di output utile per notare errori e correggere bug.

Sono inoltre presenti altri pannelli che possono essere aggiunti e rimossi a proprio piacimento, come ad esempio *Lighting*, per gestire le luci, o *Asset Store* che collega alla pagina web [54] dove poter scaricare e caricare modelli di oggetti, audio, templates e Add-Ons²⁷ gratuitamente o a pagamento.

Per lo sviluppo di questa applicazione è stata usata la versione 2021.3.19f1, una versione LTS, Long Term Support, cioè una versione stabile che viene mantenuta per un periodo di tempo più lungo rispetto alle altre.

4.3. Microsoft Visual Studio

Microsoft Visual Studio [55], o più comunemente Visual Studio, è un ambiente di sviluppo integrato sviluppato da Microsoft che supporta la creazione di progetti per varie piattaforme, tra cui Unity. Impostandolo come External Script Editor nelle impostazioni del motore grafico è infatti possibile aprire in automatico gli script su Visual Studio.

La prima versione risale al 1997. Supporta diversi linguaggi di programmazione, tra cui C++, C#, Visual Basic .Net, C++, Java e JavaScript.

È stata usata la versione Visual Studio Community 2019 per lo sviluppo dell'applicazione.

4.4. Linguaggi di programmazione e formati

C# [56] è un linguaggio di programmazione multi-paradigma che supporta tutti i concetti della programmazione orientata agli oggetti. Esiste dal 2000. Unity

²⁷ Piccoli programmi che ampliano le funzioni del programma principale.

utilizza questo linguaggio per programmare; gli script nel momento in cui vengono creati hanno infatti l'estensione .cs.

Uno script deve essere aggiunto come componente a un GameObject presente nella scena per poter essere chiamato da Unity.

Il linguaggio utilizza variabili, funzioni e classi. Le variabili contengono valori e riferimenti agli oggetti e per convenzione i loro nomi devono iniziare con una lettera minuscola. Le funzioni sono raccolte di codice che confrontano e manipolano queste variabili e iniziano con una lettera maiuscola. Le classi, invece, sono un modo per strutturare il codice e racchiudere insieme raccolte di variabili e funzioni. Il nome della classe deve corrispondere al nome del file .cs per poter funzionare [57].

Nel momento in cui viene creato uno script su Unity la classe prende il nome assegnato al file .cs e in automatico viene fatta ereditare dalla classe MonoBehaviour della libreria UnityEngine.

Esistono numerose funzioni che vengono eseguite automaticamente all'interno di Unity, tra queste ci sono:

- Awake() viene chiamata solo una volta quando viene istanziato il GameObject con quel componente. Se un GameObject è inattivo, non verrà chiamato finché non verrà reso attivo. Tuttavia, Awake viene chiamato anche se GameObject è attivo ma il componente non è abilitato.
- Start() è come Awake () e viene chiamata se un GameObject è attivo, ma solo se il componente è abilitato.
- Update() viene chiamata una volta per frame.

Nel progetto sviluppato sono stati usati, inoltre, altri linguaggi di programmazione.

È stato utilizzato il linguaggio JavaScript al fine di poter generare un file HTML come output dell'applicazione, contenente lo storyboard finale.

JavaScript è un linguaggio di programmazione multi-paradigma orientato agli eventi, utilizzato sia nella programmazione lato client web che lato server [58].

Per formattare il file HTML è stato invece usato CSS [59], un linguaggio nato nel 1996 usato per definire la formattazione di pagine web.

Per formattare le informazioni riguardanti lo stato dei personaggi e degli oggetti all'interno di un file .txt è stato usato il formato JSON, un formato per lo scambio dati basato sul linguaggio di programmazione JavaScript [60].

È stato, inoltre, utilizzato il formato file CSV basato su file di testo per indicare le azioni che i personaggi possono compiere su se stessi, su altri personaggi e su oggetti, per poter tornare allo stato di screenshot precedente e per salvare e ricaricare la scena che l'utente crea posizionando gli oggetti e i personaggi. Questi argomenti verranno approfonditi nel capitolo "Sviluppo dell'applicazione".

4.5. MRTK

Mixed Reality Toolkit [61], MRTK, è un kit di sviluppo software (SDK) open-source²⁸ sviluppato da Microsoft nel 2016 per lo sviluppo di applicazioni software di MR, VR e AR. Consiste in una raccolta di componenti e funzionalità utili per facilitare lo sviluppo di applicazioni.

Nel dicembre 2017, quando è stato rilasciato al pubblico, era conosciuto come HoloToolkit in quanto nato per lo sviluppo di software per gli allora nuovi Microsoft HoloLens, dispositivo per AR.

Fornisce il sistema di input multiplatforma ed elementi precostituiti per le interazioni spaziali e l'UI. Consente la prototipazione rapida tramite simulazione nell'editor che consente di vedere immediatamente le modifiche. Funziona come un framework estensibile che offre agli sviluppatori la possibilità di sostituire i componenti principali. Può supportare diversi plug-in²⁹: Unity OpenXR, Windows XR, Oculus XR, ARCore XR e ARKit XR.

²⁸ Codice sorgente reso disponibile gratuitamente per possibili modifiche e ridistribuzioni.

²⁹ Componenti di un software che aggiungono delle funzionalità.

Oltre a diversi dispositivi supportati: Microsoft HoloLens, Microsoft HoloLens 2, Windows Mixed Reality headset, Meta Quest, dispositivi in esecuzione su SteamVR tramite OpenXR, dispositivi Android e iOS tramite AR Foundation. Per poter funzionare richiede Windows SDK, Unity 2018/2019/2020 LTS e Visual Studio 2019 [62]. Per il progetto è stata usata la versione MRTK 2.8.3.0.

4.6. Oculus Integration SDK

L'Oculus Integration SDK è una risorsa di funzionalità, componenti, script e plugin utili a facilitare e migliorare il processo di sviluppo delle applicazioni per Oculus in Unity e in altre piattaforme. Comprende diversi SDK, come il pacchetto di utilità, Interaction SDK, Voice SDK, Platform SDK e molte altre funzionalità [63].

Il pacchetto è stato installato solo per permettere al componente MixedRealityToolkit di MRTK di riconoscere in automatico l'Oculus Rift e i controller. Per lo sviluppo, invece, sono stati usati componenti MRTK. Per il progetto è stata usata la versione 47.0.

4.7. Blender

Blender è un software gratis e open-source, nato nel 1994, utilizzato per modellazione 3D, UV mapping³⁰, texturing³¹, rigging³², animazione, simulazione

³⁰ Mappatura UV è una tecnica di texture mapping che permette di applicare efficacemente e correttamente le texture su un modello 3D.

³¹ Tecnica che permette di proiettare una o più texture sulla superficie di un modello 3D.

³² Il rigging serve a costruire una serie di ossa (o scheletro) per il modello 3D. È una fase che precede l'animazione.

di fluidi, fumo, particelle e soft body³³, rendering, montaggio video e composizione³⁴.

Ad oggi è disponibile per Microsoft Windows, macOS e Linux.

Blender ha la caratteristica di permettere la modellazione sfruttando i modificatori, strumenti per la modifica non distruttiva, quindi reversibile, degli oggetti presenti in scena. Possiede un sistema a nodi per la creazione e manipolazione di geometrie in modo non distruttivo e procedurale.

Blender include diverse modalità, tra queste le due principali sono la *Object Mode*, utile per manipolare gli oggetti nello spazio, e la *Edit Mode*, utilizzata per modificare la mesh³⁵ di un oggetto.

Possiede tre motori di rendering: Workbench, EEVEE e Cycles.

- Workbench è motore di rendering in real-time utile per avere una preview del render finale durante la modellazione e l'animazione.
- EEVEE è un motore di rendering di tipo physically based (PBR), un approccio che cerca di riprodurre le immagini in modo da modellare le luci e le superfici con l'ottica nel mondo reale [64].
- Cycles è un motore di rendering di tipo path tracing. Questa tecnica, utile a simulare l'illuminazione realistica, segue i raggi di luce mentre si diffondono e interagiscono con gli oggetti nella scena, tenendo conto di riflessioni, rifrazioni, ombre e altri effetti luminosi. Supporta il rendering con la GPU [65].

Blender è stato usato per modificare la modellazione di alcuni oggetti scaricati da Internet. Gli oggetti venivano esportati da Blender utilizzando il formato FBX e importati successivamente su Unity.

È stata usata la versione 3.3.5.

³³ Oggetti deformabili.

³⁴ Disposizione degli elementi in un'immagine o in una scena.

³⁵ Reticolo che definisce un oggetto nello spazio, composto da vertici, spigoli e facce.

4.8. Asset

Gli oggetti visibili nell'applicazione hanno diverse fonti.

Il piano del tavolo è presente tra gli asset di Oculus Integration SDK. Alcuni asset sono stati presi dall'applicazione di partenza per Desktop, mentre altri sono stati scaricati da siti online come Sketchfab [66], TURBOSQUID [67] [68], Free3D [69] e da Unity Asset Store [70]. Questi siti sono una raccolta principalmente di assets 2D e 3D in vari formati caricati dagli utenti, i quali decidono se condividerli gratuitamente o a pagamento.

4.9. Mixamo

Mixamo è un'azienda, fondata nel 2008, di tecnologia di computer grafica 3D con sede a San Francisco. L'azienda sviluppa e vende servizi basati sul web per l'animazione di personaggi 3D. È stata acquisita da Adobe Systems nel 2015 [71].

Sul sito ufficiale [72] permettono agli utenti, previa registrazione, di scaricare gratuitamente personaggi e animazioni in diversi formati, compatibili con le applicazioni di Adobe, Unity, Unreal Engine e Blender.

I personaggi 3D spaziano dallo stile realistico a quello dei cartoni animati, dal fantasy alla fantascienza. Ogni personaggio viene fornito già texturizzato e riggato.

Le animazioni disponibili sul sito sono create con l'utilizzo della motion capture, tecnologia che registra il movimento di persone o oggetti. Una volta catturato tecnologicamente il movimento, i dati sono trasferiti su un programma al computer e le animazioni vengono ripulite nei movimenti dagli animatori.

I personaggi e le animazioni presenti nell'applicazione sono stati scaricati da Mixamo.

4.10. TextMeshPro

TextMeshPro [73], noto anche come TMP, è un pacchetto di Unity dedicato agli elementi testuali. Utilizza tecniche di rendering avanzate del testo insieme a una serie di shader personalizzati che portano a miglioramenti della qualità visiva, offrendo, inoltre, ai programmatori varietà di scelta di stile.

TextMeshPro è stato installato e utilizzato per tutti i testi presenti nell'applicazione.

4.11. API OpenAI

OpenAI [74] è un laboratorio di ricerca, nato nel 2015 e con sede principale a San Francisco, che si impegna nello sviluppo di tecnologie e modelli di intelligenza artificiale avanzati. OpenAI offre diverse API che consentono agli sviluppatori di integrare le tecnologie di intelligenza artificiale nelle proprie applicazioni.

Per utilizzare OpenAI nell'ambiente di sviluppo Unity è stata installata una libreria creata da una comunità di sviluppatori, segnalata sul sito ufficiale di OpenAI. La libreria, scaricabile da GitHub, prende il nome di “com.openai.unity” [75]. È stata utilizzata la versione 5.0.11.

Si tratta di un pacchetto per il motore di gioco Unity per poter utilizzare i modelli OpenAI attraverso le loro API. Richiede la versione di Unity 2021.3 LTS o superiori. In particolare, è stato utilizzato l'endpoint API³⁶ “Edits” per riformulare il testo presente nelle descrizioni delle vignette dello storyboard. Il modello utilizzato da questo endpoint è *text-davinci-edit-001*.

³⁶ Posizione dell'API in cui un sistema interagisce con un'API web. È il punto di comunicazione tra due sistemi.

5. Sviluppo dell'applicazione

Lo sviluppo dell'applicazione, come indicato nei capitoli precedenti, è dipeso dall'applicazione di partenza per Desktop e dall'analisi dei requisiti, oltre ad essere influenzata da una serie di decisioni relative sia alla sua logica di funzionamento che alla sua interfaccia grafica. Questi aspetti saranno esaminati nel dettaglio all'interno di questo capitolo.

5.1. Pipeline di funzionamento

L'applicazione è costituita da quattro step principali:

1. il menù iniziale dove poter scegliere se accedere al tutorial (tasto "*Tutorial*"), se creare un nuovo progetto (tasto "*Create New Scene*") o se partire da una scena precedentemente salvata (tasto "*Load Scene*");
2. la scena in cui poter popolare un tavolo virtuale di oggetti e personaggi in miniatura, scegliendoli da una lista. Orientare la luce cinematografica, aggiungerne di nuove e decidere la posizione del sole;
3. la fase di simulazione in cui animare i personaggi assegnando loro azioni, scegliendole da una lista, e scattare foto con le camere virtuali;
4. lo storyboard finale costituito dalle foto scattate visualizzabile e modificabile sul PC.

L'utente ha la possibilità di muoversi tra le fasi intermedie a proprio piacimento premendo sui tasti "*Start Storyboarding*" per passare al terzo step e "*Stop Storyboarding*" per tornare al secondo.

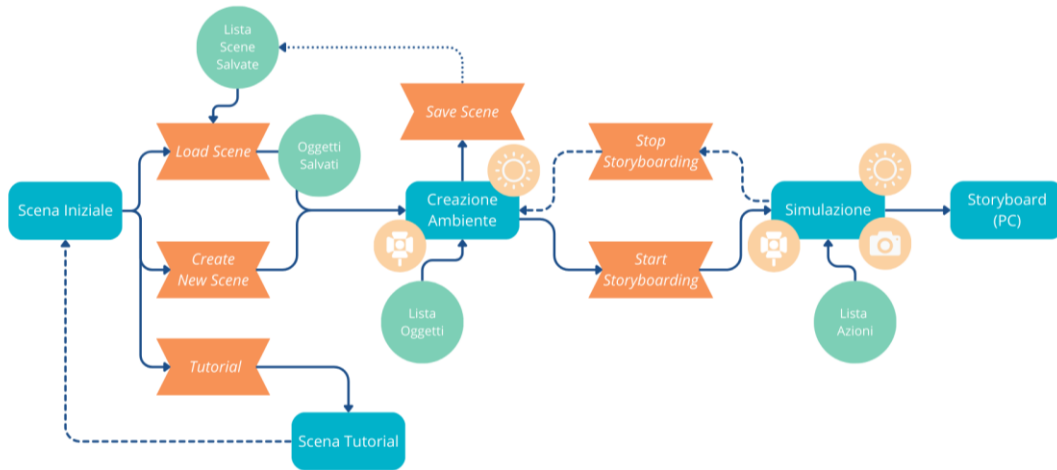


Figura 5.1 Pipeline funzionamento applicazione.

Dal punto di vista tecnico è composta solo da tre scene Unity: la scena iniziale, il tutorial e quella di creazione dell'ambiente e simulazione. Le seconde scene vengono caricate in modo additivo rispetto alla prima e i vari GameObject vengono attivati o disabilitati via script a seconda delle necessità. È presente, inoltre, una pagina HTML che si apre in locale sul PC, contenente lo storyboard.

5.2. Movimento nell'ambiente

L'utente ha la possibilità di muoversi nell'ambiente camminando e muovendo la testa, grazie ai 6 DoF, oppure utilizzando i thumbstick presenti sui controller.

Il componente MRTK *MixedRealityToolkit* dà la possibilità di personalizzare l'interazione dei controller nel pannello *Input>Controllers>Controllers Definitions*.

Le azioni dei tasti *Grip* e *Trigger* di entrambi i controller sono quelle default assegnate da MRTK. Verranno descritte nel dettaglio nel paragrafo 5.8.

Le azioni dei *Thumbstick* sono state invece modificate. Con il *Thumbstick* sinistro è possibile muoversi avanti, indietro, a destra e a sinistra nell'ambiente attraverso la *Navigation Action*. Con il *Thumbstick* destro e l'azione *Rotation And Height Action* è, invece, possibile ruotare lo sguardo lungo l'asse verticale e aumentare o diminuire l'altezza in modo da regolare a proprio piacimento la posizione degli

elementi presenti in scena. All'avvio dell'applicazione il piano virtuale presente in scena è posizionato, rispetto al pavimento, all'altezza standard di un tavolo.

Entrambe le azioni sono state definite nello script *Movement*. Questo script si occupa, infatti, di gestire le velocità dei vari movimenti.

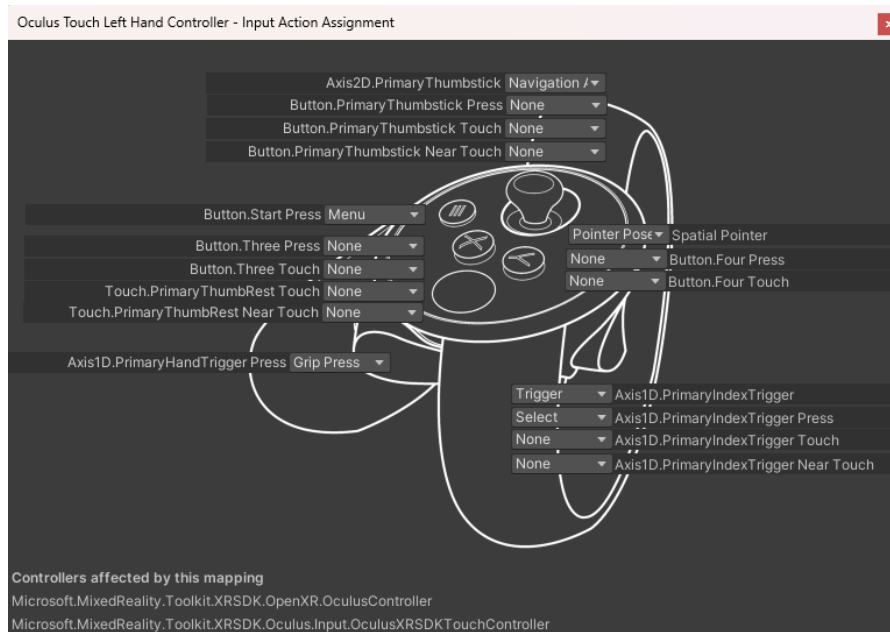


Figura 5.2 Alcune delle azioni disponibili interagendo con i tasti del controller sinistro.

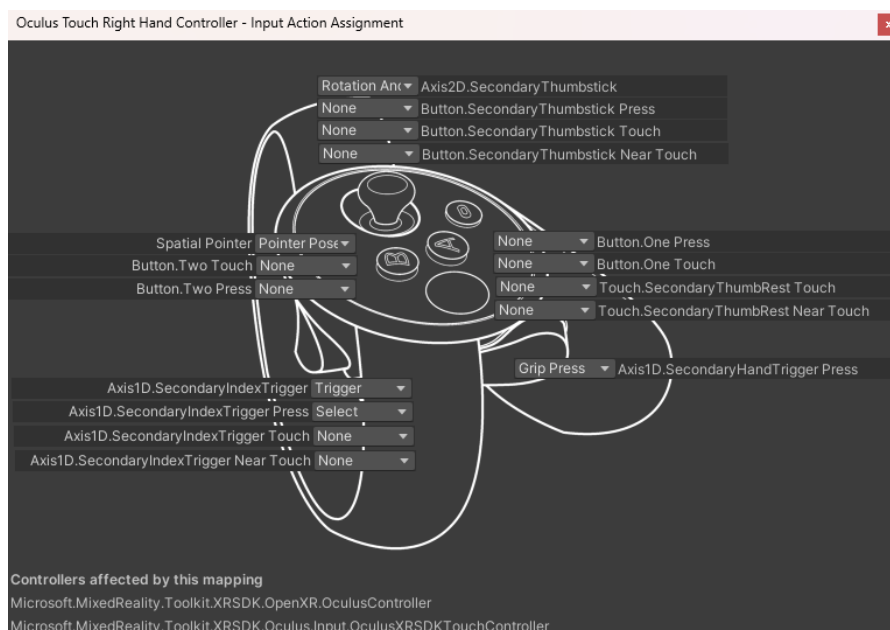


Figura 5.3 Alcune delle azioni disponibili interagendo con i tasti del controller destro.

5.3. Menù iniziale

Nel momento in cui si avvia l'applicazione l'utente si trova davanti un menù dove ha la possibilità di scegliere se creare una nuova scena dall'inizio, se caricarne una precedentemente salvata, scegliendola dalla lista, o se accedere al tutorial.



Figura 5.4 Menù presente all'avvio dell'applicazione.

Il tasto *Tutorial* porta alla rispettiva scena, dove l'utente viene guidato attraverso dei pannelli che spiegano a step il funzionamento dell'applicazione e i tasti dei controller. Lo script *Tutorial* gestisce l'attivazione e gli spostamenti degli elementi della scena. Il tasto "*Create New Scene*" permette di creare una nuova scena e il tasto "*Load Scene*" caricherà sul piano una serie di oggetti salvati dall'utente in una precedente sessione di utilizzo dell'applicazione. Lo script *StartingScene* si occupa di gestire il caricamento delle varie scene Unity.

Per poter caricare le scene salvate viene utilizzato lo script *SaveLoadStage* che si occupa di salvare e caricare gli oggetti presenti in scena, scrivendo e leggendo i dati, quali nome, posizione, orientamento, scala e tipologia di oggetto, nel file *saved_stages.csv*. Ogni riga del file rappresenta una scena salvata, il cui nome è il primo elemento. Per ogni oggetto oltre al nome viene salvata la tipologia di oggetto, in quanto il nome può essere modificato a proprio piacimento dall'utente.

5.4. Creazione ambiente

La scena successiva a quella del menù iniziale si presenta con un tavolo posto davanti all'utente, una lista di oggetti pre-renderizzati in un menù fluttuante situato alla destra del tavolo, un menù alla sinistra e due pannelli che ricordano all'utente il funzionamento dell'applicazione e la funzione dei tasti dei controller, in base al contesto in cui si trovano.

Ogni menù presente in scena è manipolabile dall'utente: può essere spostato nello spazio e scalato afferrandolo dalla barra superiore contenente il nome del menù. Ogni menù ruota lungo gli assi X e Y in modo da essere sempre rivolto verso l'utente, grazie allo script MRTK *Billboard*.



Figura 5.5 Visuale che si può presentare all'utente dopo aver premuto “Create New Scene”.

In questa fase l'utente ha la possibilità di selezionare gli oggetti presenti nel pannello “Objects”, i quali compariranno sul piano, e di posizionarli, ruotarli e scalarli a proprio piacimento usando i controller. Nel momento in cui si seleziona un oggetto presente in scena, sopra questo compare il suo nome e un tasto che permette di rinominare l'oggetto. Premendo il tasto “Rename” comparirà una tastiera, la *NonNativeKeyboard* di MRTK, con cui l'utente può digitare il proprio testo o dirlo a voce, attivando il microfono.

Lo script *ObjectManagerVR* si occupa principalmente di far apparire gli oggetti sul piano, attraverso la funzione *CreateObject()*, di poter permettere la loro eliminazione attraverso *DeleteObject()*, di rinominare gli oggetti e assegnare un nome alla scena e di conseguenza allo storyboard; oltre a continuare il lavoro di salvataggio e caricamento dello script *SaveLoadStage*.



Figura 5.6 Oggetti posizionati sul piano, sopra l'oggetto selezionato appare il nome e il tasto "Rename" per poterlo rinominare. A destra è visibile il menù contenente gli oggetti.

Il "Main Menù" posto a sinistra contiene vari tasti:

- il tasto "Help", con un "?" come icona, che permette di far apparire nuovamente due pannelli di istruzioni, se chiusi precedentemente dall'utente premendo il tasto *Close* (X);
- il tasto "Rename Scene" che apre la tastiera e permette di rinominare la scena, il cui nome comparirà alla sinistra di "Scene Name:", dopo aver premuto "Save Scene", tasto che si aggiungerà nel menù dopo che l'utente avrà premuto su *Enter* nella tastiera;
- i tasti "Use Case 1" e "Use Case 2" che aprono dei pannelli contenenti le sceneggiature da riprodurre per la fase di test dell'applicazione, i cui risultati verranno descritti nel capitolo 6;

- il tasto “*Start Storyboarding*” che permette di passare alla fase di successiva e che chiuderà alcuni menù, facendone comparire altri. Nel “*Main Menù*” appariranno altri tasti, tra cui quello per generare lo storyboard finale una volta che si è conclusa l’esperienza di utilizzo dell’applicazione nell’ambiente virtuale.

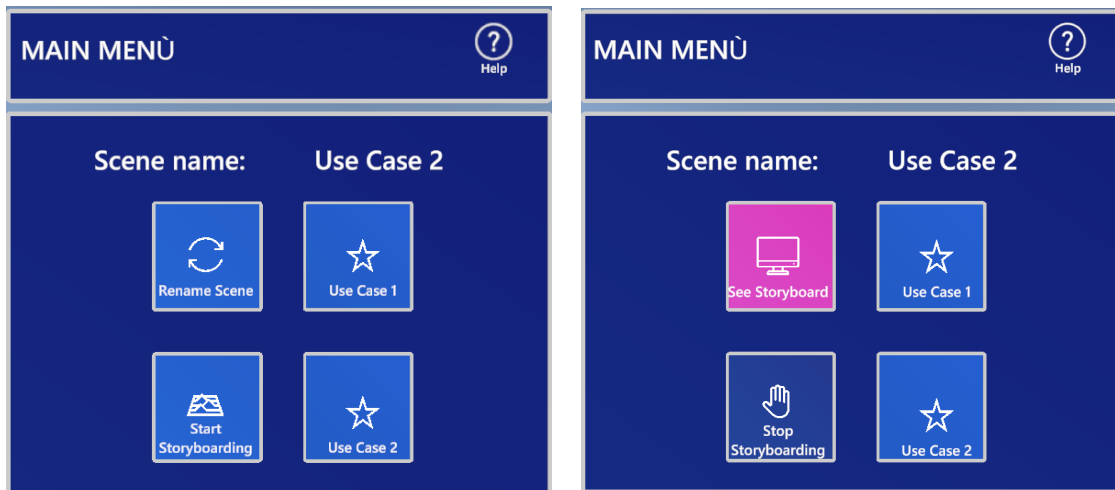


Figura 5.7 Menù principale, sempre presente durante l’utilizzo dell’applicazione.

5.5. Luci

Nella scena di creazione dell’ambiente è, inoltre, possibile posizionare a proprio piacimento l’oggetto a forma di faro cinematografico che emana luce, della quale si possono modificare colore e intensità tramite il componente *Color Picker* e relativo script di MRTK. Premendo il tasto *A* sul controller destro si può aggiungere in scena, nella posizione e orientamento del proprio viso, una nuova luce identica a quella già presente. Lo script *LightManager* si occupa della gestione degli elementi collegati alla luce, come il tasto per aprire il *Color Picker*; mentre la funzione *NewLight()* nel *SimulationManager* si occupa di consentire l’aggiunta di nuove luci.

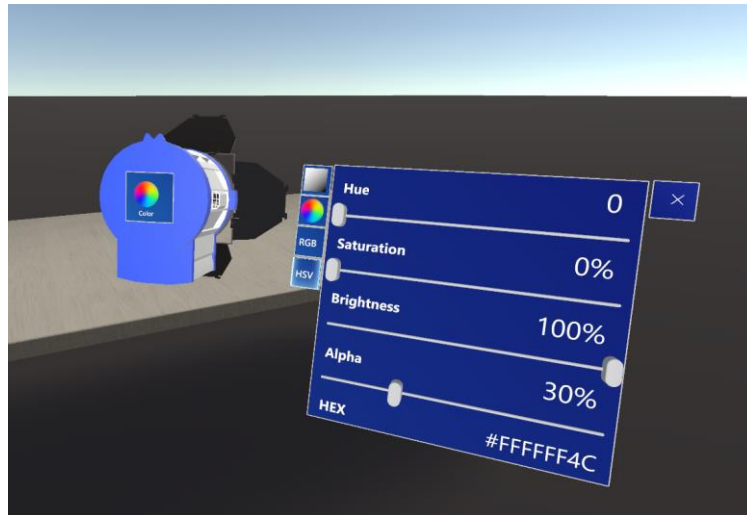


Figura 5.8 Luce con parametri modificabili.

L'oggetto a forma di sole può essere invece ruotato per modificare l'altezza del sole presente nel cielo, modificando le ombre proiettate dagli oggetti sul tavolo e definendo così l'atmosfera e i colori da far assumere al cielo in fase di storyboard.

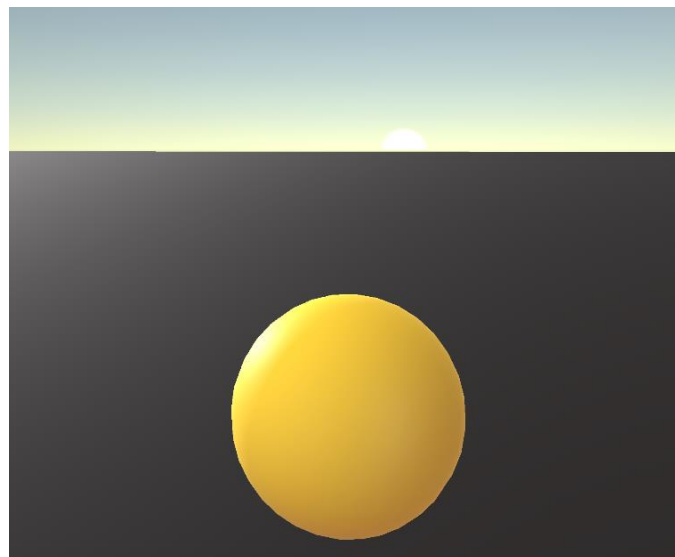


Figura 5.9 Oggetto sole che comanda la posizione del sole nell'HDRI.

5.6. Simulazione

Premendo sul tasto “*Start Storyboarding*” si passa alla fase di simulazione. Il menù con gli oggetti da poter inserire in scena si disattiva, mentre appaiono nuovi menù e un oggetto camera.



Figura 5.10 Possibile visuale dell'utente dopo aver premuto “*Start Storyboarding*”.

L'utente ha la possibilità di interagire con il “*Storyboard Panels Menù*” per poter decidere su quale vignetta lavorare, passando alla successiva premendo il tasto “*Next Panel*”, impostare la durata totale dell'inquadratura e se le azioni che farà compiere ai personaggi sono contemporanee o consecutive tra loro. Questi input sono gestiti da delle funzioni dello script *SimulationManager* e dallo script *ActionDurationScript*.

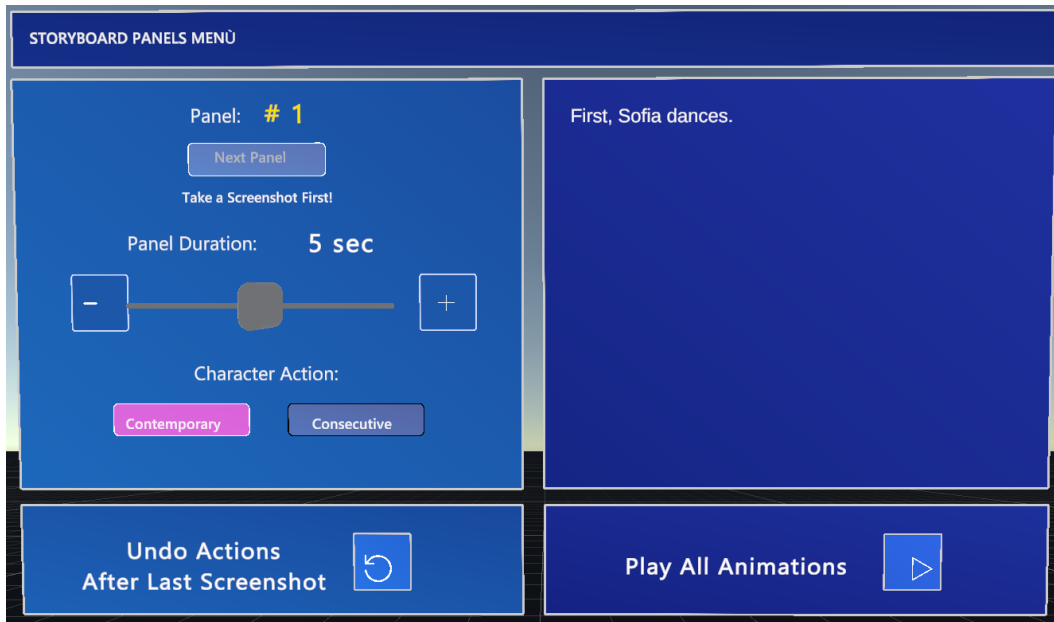


Figura 5.11 Storyboard Panels Menù.

Selezionando i personaggi presenti in scena, si possono assegnare loro delle azioni da compiere. In automatico verrà generata una frase, visibile nella parte destra del menù, che descriverà la situazione. Questi testi definiranno le descrizioni delle vignette dello storyboard finale.

Ogni personaggio possiede due tipologie di azioni: quelle che compie su sé stesso e le azioni che compie su un altro personaggio o oggetto inanimato. Le prime compaiono nell'elenco azioni posto al di sopra del personaggio, nel momento in cui lo si seleziona. Le seconde appaiono sempre sopra al personaggio, ma dopo aver selezionato un altro personaggio o oggetto.

Il primo personaggio selezionato, che corrisponde al soggetto della frase, viene evidenziato da un cerchio di particelle in movimento, poste attorno all'oggetto, sui toni del rosso. Il complemento oggetto, invece, viene evidenziato da uno sui toni del verde. Il colore dei tasti con le azioni cambia a seconda che le azioni si riferiscano al soggetto o al complemento oggetto. Quelle sul soggetto riprendono il colore delle particelle che lo evidenziano; mentre quelle del complemento si colorano come il secondo cerchio.

La creazione e distruzione delle particelle è eseguita dalle funzioni *CreateParticleActive()*, *CreateParticleComplement()*, *DestroyParticles()*, *DestroyParticlesComplement()* presenti nello script *SimulationManager*.



Figura 5.12 Particelle colorate che evidenziano gli oggetti selezionati.

L'azione "TALK TO", che appare selezionando prima un personaggio e poi un altro, fa aprire la tastiera in cui digitare la frase che il primo personaggio deve pronunciare al secondo e che verrà aggiunta nella descrizione della vignetta corrispondente tra virgolette basse. È presente un'ulteriore azione denominata "OTHER", che permette, facendo aprire la tastiera, di definire un'azione non presente in elenco, sia riferita al soggetto sia compiuta dal soggetto sul complemento oggetto.

Il pannello contenente i tasti con le azioni da assegnare appare sopra al personaggio selezionato, ma può essere spostato nello spazio dall'utente. Nel momento in cui si seleziona una seconda volta il personaggio o si prende il controllo di un secondo personaggio, il pannello si posiziona nuovamente sopra al personaggio attivo. Il pannello, come l'etichetta contenente il nome del personaggio ruotano in automatico lungo gli assi X e Y in modo da essere sempre rivolto verso l'utente, grazie allo script MRTK *Billboard*.

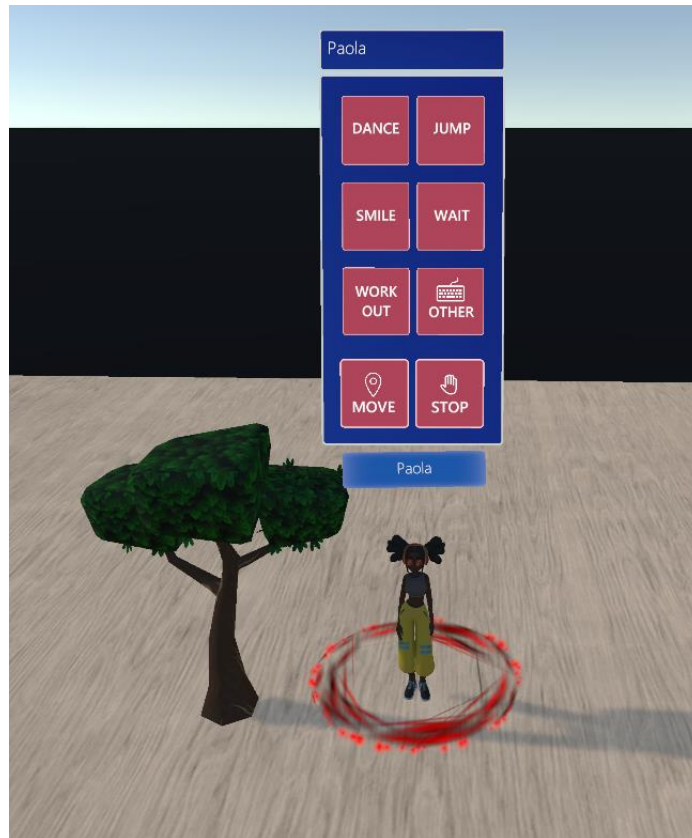


Figura 5.13 Lista azioni, nome, tasti “MOVE”, “STOP” (o “PLAY”) riferiti al personaggio selezionato.

L’elenco delle possibili azioni è contenuto nel file *actions.csv*. Ogni riga del file contiene un soggetto, un complemento (“self”, se l’azione è del soggetto su sé stesso, oppure il tipo di personaggio o oggetto) e l’elenco delle azioni separate da una virgola. Un esempio di contenuto del file è il seguente:

```
man-self,smile,wait,jump,stand up,stop play,dance,stop dance,clap  
woman-bench,sit
```

Gli stati sono definiti nel file di testo *states.txt*, scritto in formato JSON, dove ogni personaggio, definito dal proprio tipo, possiede la lista di possibili transizioni da uno stato ad un altro attraverso una specifica azione e la lista chiamata *S_A* che riporta, per ogni stato, la lista delle azioni esclusive di quel determinato stato.

Di seguito è riportato l'esempio della struttura dati per il personaggio di tipo *man*.

```
"name": "man",  
"transitions": [  
    "standing>sit>sitting",  
    "sitting>stand up>standing",  
    "standing>play>playing",  
    "playing>stop play>standing",  
    "standing>dance>dancing",  
    "dancing>stop dance>standing"  
],  
"S_A": [  
    "sitting:stand up",  
    "standing:sit,play,dance",  
    "playing:stop play",  
    "dancing:stop dance"  
]
```

Le azioni e gli stati sono assegnati attraverso gli script *State* e *ActionsDataBase*. I due script vengono richiamati dalla classe *AnimaPersonaggio* che si occupa di mostrare le azioni sotto forma di tasti e di gestire la parte grafica e di interazione.

Premendo sul tasto “*MOVE*” e indicando con il controller la destinazione finale, l'utente può far camminare il personaggio selezionato. Questo può essere bloccato in una determinata posa, durante qualsiasi azione e animazione, premendo il tasto “*STOP*”. Selezionando su “*PLAY*” il personaggio riprende l'animazione da dove si era interrotta.

L'utente può, inoltre, decidere di sbloccare in contemporanea dalla posa tutti i personaggi in scena premendo sul tasto “*Play All Animations*” presente nel “*Storyboard Panels Menù*”.

Queste funzionalità vengono gestite dallo script *AnimaPersonaggio*.

La classe *CharacterManager* si occupa di gestire la camminata, insieme al componente *NavMeshAgent* che ha come area *NavMesh*, su cui potersi muovere, la superficie del tavolo virtuale.

Ogni personaggio animabile è dotato di un componente *Animator*, un'interfaccia per controllare il sistema delle animazioni.

L'animazione di *idle* è lo stato in cui rimane il personaggio nel momento in cui non sta svolgendo alcuna azione. Da questo stato, a seconda dell'azione assegnata dall'utente, il personaggio passa ad un'animazione successiva per poi tornare in stato di *idle*.

Se l'azione non fa cambiare lo stato del personaggio, l'animazione avverrà tre volte e in automatico si tornerà in stato di *idle* al termine delle animazioni, in quanto le variabili “*Has Exit Time*” delle transizioni sono poste a “*true*”; al contrario, per le animazioni poste nella parte inferiore della Figura 5.14 (suonare, camminare, ballare e sedersi), che fanno variare lo stato del personaggio, sono le variabili booleane *playing*, *walking*, *dancing* e *sitting* a far tornare in stato di *idle* il personaggio, quando queste assumono il valore “*false*”, nel momento in cui l'utente preme un'azione del tipo “*Stop action*”, dall'elenco delle azioni. Le azioni di tipo “*Stop action*” assumono una colorazione più scura rispetto alle altre per essere maggiormente visibili.

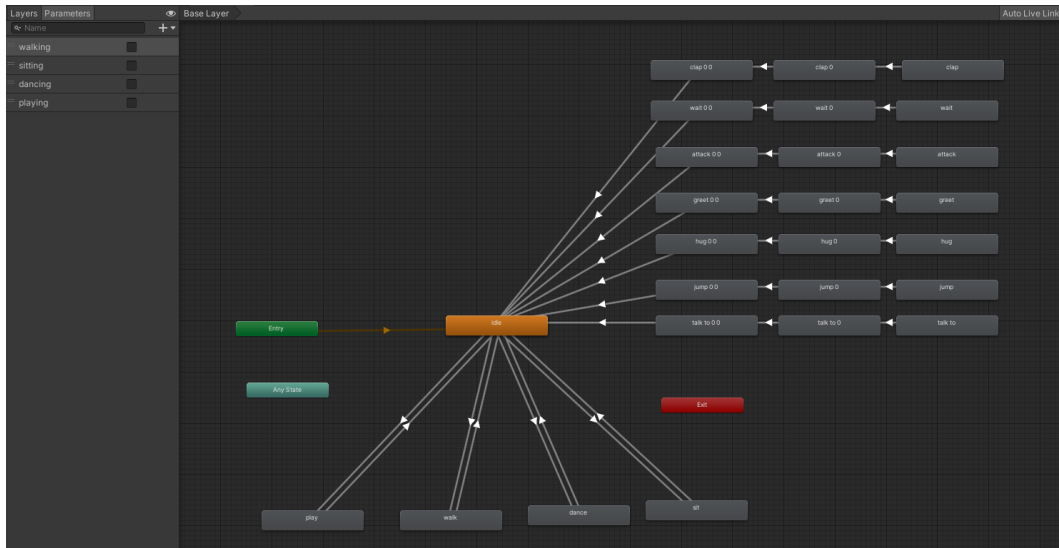


Figura 5.14 Animator del personaggio di tipo *man*.

Nel caso in cui sia attivo il tasto “*Contemporary*” nel “*Storyboard Panles Menù*” e il personaggio stia eseguendo un’azione che gli ha fatto cambiare stato, nel momento in cui l’utente seleziona un’altra azione questa interromperà momentaneamente l’azione in loop e l’animazione della nuova azione verrà eseguita tre volte; successivamente il personaggio tornerà a compiere l’azione precedente.

Al contrario se il tasto attivo è “*Consecutive*”, l’azione successiva interromperà la precedente facendo tornare il personaggio in stato di *idle* senza che l’utente sia obbligato ad assegnargli l’azione di tipo “*Stop action*”.

Quando si sceglie un secondo personaggio, dopo averne già selezionato uno, oltre a poter definire le azioni del primo sul secondo, si ha la possibilità di prendere il controllo del secondo personaggio per farlo diventare il soggetto e non più il complemento oggetto. Il tasto “*Get Control*” che appare sopra al secondo personaggio ha questa funzione.



Figura 5.15 Tasto “Get Control”.

5.7. Camere

Nel momento in cui si è nella fase di simulazione appare davanti al tavolo una prima camera virtuale, dotata di schermo, uno slider per impostare la lunghezza focale, modificabile anche con i tasti + e -, e un tasto per effettuare gli screenshot.

Lo schermo, che proietta in tempo reale ciò che la camera inquadra poiché dotato di una *RenderTexture* come elemento *albedo* del proprio materiale, può essere ingrandito e spostato nell'ambiente. Se spostato fuori dalla camera lo schermo smetterà di seguire i movimenti di quest'ultima, se riposizionato al suo interno o avvicinato alla camera tornerà ad essere spostato nello spazio con essa. Questo funzionamento avviene grazie alle funzioni *OnTriggerEnter()*, *OnTriggerStay()*, *OnTriggerExit()* di Unity presenti nello script *SecondCameraManager*. Funzioni che agiscono se entrambi i *GameObject* (Camera e Display in questo caso) hanno il componente *Collider*. È necessario che in uno dei due *GameObject* *Collider.isTrigger* sia abilitato e sia presente il componente *Rigidbody*.

Quando l'utente preme sul tasto "Screenshot", posto sopra la camera, creerà un nuovo screenshot. Lo screenshot eseguito andrà a sostituire il precedente fino a quando l'utente non premerà "Next Panel" nel "Storyboard Panels Menù". Gli screenshot finali di ogni pannello rimarranno nel pannello "Panels", aggiornato in tempo reale.

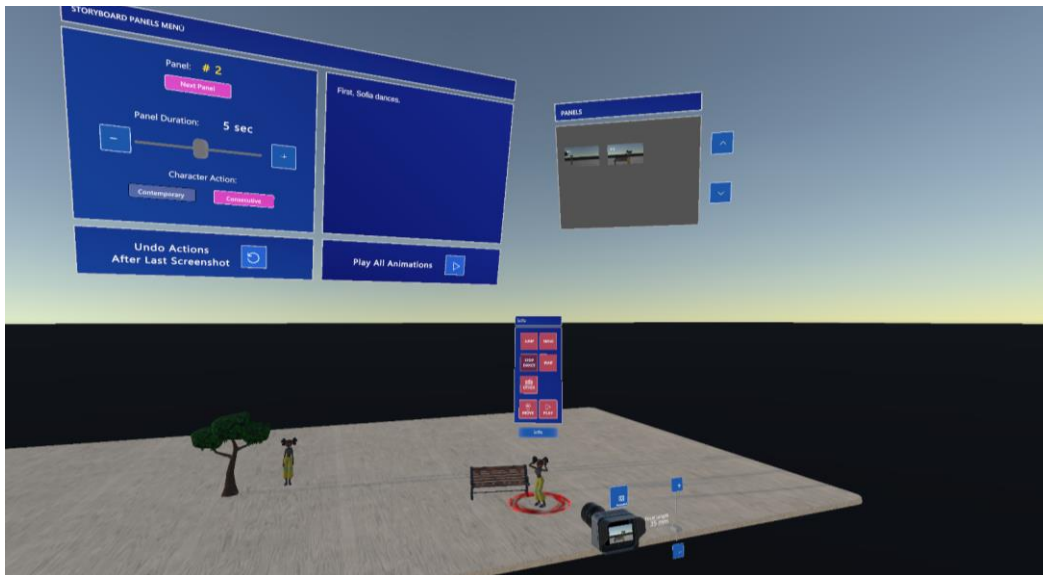


Figura 5.16 Camera e screenshot eseguiti.

Si può inserire in scena una nuova camera, identica alla precedente, posta nella stessa direzione e posizione del viso dell'utente, premendo il tasto *B* del controller destro.

Gli script *SecondCameraManager*, *CameraManager*, *ShowFocalLengthValue* si occupano di gestire la camera e l'inserimento di nuove camere, del funzionamento del tasto *Screenshot*, dello schermo e la sua *RenderTexture*, dello slider *Focal Length* e dei vari screenshot che appaiono come immagini nell'ambiente virtuale.

Ogni screenshot eseguito viene salvato nel progetto in formato *code_imgX.png*, con *code* un numero random di 6 cifre che viene generato ad ogni utilizzo dell'applicazione. Le immagini della stessa sessione avranno quindi lo stesso codice; *X* invece è una variabile che viene incrementata ogni volta che l'utente preme "Next Panel".

Le funzioni *CompletedAction()* e *UndoLastAction()* dello script *SimulationManager* si occupano rispettivamente di scrivere e di leggere i dati contenuti nel file *screenshotActions.csv*. Nel momento in cui si esegue uno screenshot vengono salvati i valori della *Focal Length* e della *Duration* presi dagli slider e la posizione e rotazione della camera e di tutti gli oggetti presenti sul tavolo. L'utente può ritornare alla configurazione, a livello di posizione degli oggetti sul tavolo, del loro stato, della camera e degli slider, dell'ultimo screenshot eseguito premendo il tasto con il simbolo di una freccia e la descrizione "*Undo Actions After Last Screenshot*" presente nel "*Storyboard Panels Menu*". Il testo generato successivamente allo screenshot verrà cancellato in quanto ogni frase viene salvata in un buffer di stringhe in formato *X_TestoGenerato_Y*, con *X* corrispondente al numero della vignetta e *Y* al numero dello screenshot. *Y* si aggiorna ogni volta in cui l'utente esegue uno screenshot con una camera. La funzione *DeleteSentencesUndo()* del *SimulationManager*, che restituisce la lista di stringhe, prende dal buffer ogni stringa e confronta il valore *Y* di ogni frase con il valore intero *screenshotCountUndo* che tiene traccia del numero di screenshot eseguiti; se i valori corrispondono le frasi vengono eliminate.

5.8. Controller

I controller sono l'interfaccia fisica con cui l'utente ha la possibilità di interagire con l'applicazione. Questi sono rappresentati virtualmente nell'esatta posizione di quelli presenti nel mondo reale. Premendo un tasto sul controller reale, sul controller virtuale si avrà un riscontro vedendo il rispettivo tasto animarsi.

Da ogni controller parte un raggio, che si accorcia nel momento in cui collide con i *GameObject* presenti in scena, che questi siano menù o oggetti della scena.

Alla fine di ogni raggio è presente un cursore, che cambia grafica in base all'azione che si sta compiendo, grazie alla classe *MRTK CursorContextObjectManipulator*. Nel momento in cui l'utente ha un controller

vicino ad un oggetto il raggio si disattiva e al suo posto appare, vicino al controller, un cursore rotondo, fondamentale per interagire con gli oggetti.

Gli script MRTK *ObjectManipulator*, e *NearInteractionGrabbable* permettono di manipolare gli oggetti da lontano, indicandoli e tenendo premuto il tasto *Trigger*, o da vicino, premendo il tasto *Grip*. Ogni oggetto manipolabile, che sia un menù, una camera o un oggetto sul tavolo, non è soggetto alla gravità. Questi possono essere ruotati, spostati nello spazio e, usando i due controller in contemporanea sullo stesso oggetto, scalati. Quando dalla fase di creazione dell'ambiente si passa a quella di simulazione entrambi gli script vengono disattivati per gli oggetti che popolano la scena sul tavolo.

Nella fase di simulazione non è infatti più possibile spostare gli oggetti della scena da inquadrare, ma solo assegnare delle azioni utilizzando i menù e i tasti che appaiono sopra ogni personaggio nel momento in cui vengono selezionati, attraverso lo script MRTK *Interactable*. Per selezionare un personaggio da lontano, bisogna premere il tasto *Trigger*. Per selezionarlo da vicino è sufficiente toccarlo con il controller, grazie allo script MRTK *NearInteractionTouchable*.

Per interagire con i tasti della UI il funzionamento è lo stesso.

Le funzioni dei vari tasti dei controller vengono ricordate all'utente attraverso un pannello che cambia immagine in base alla fase in cui ci si trova.

Nella fase di creazione dell'ambiente sono attivi il tasto *X* che permette di eliminare l'ultimo oggetto selezionato, il tasto *A* per aggiungere una nuova luce, i *Thumbstick* per potersi muovere nell'ambiente e i tasti di *Trigger* e *Grip* per selezionare e afferrare.

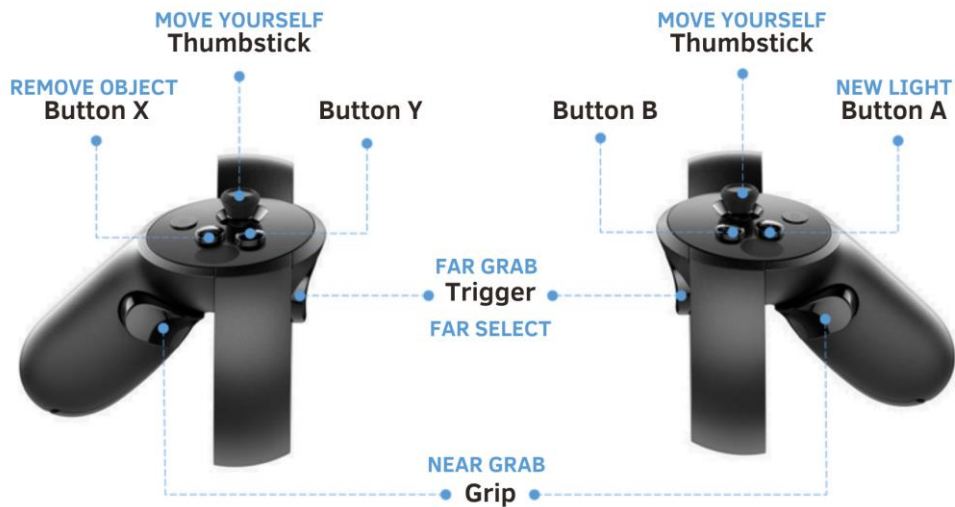


Figura 5.17 Funzionalità dei tasti dei controller nella fase di creazione dell'ambiente.

Nella fase di simulazione i *Thumbstick*, e i tasti *Trigger* e *Grip* continuano ad avere le stesse funzionalità, a cui si aggiungono il tasto *Y* per annullare le ultime azioni dopo l'ultimo screenshot e il tasto *B* per aggiungere in scena una nuova camera; mentre il tasto *X* rimane inutilizzato.

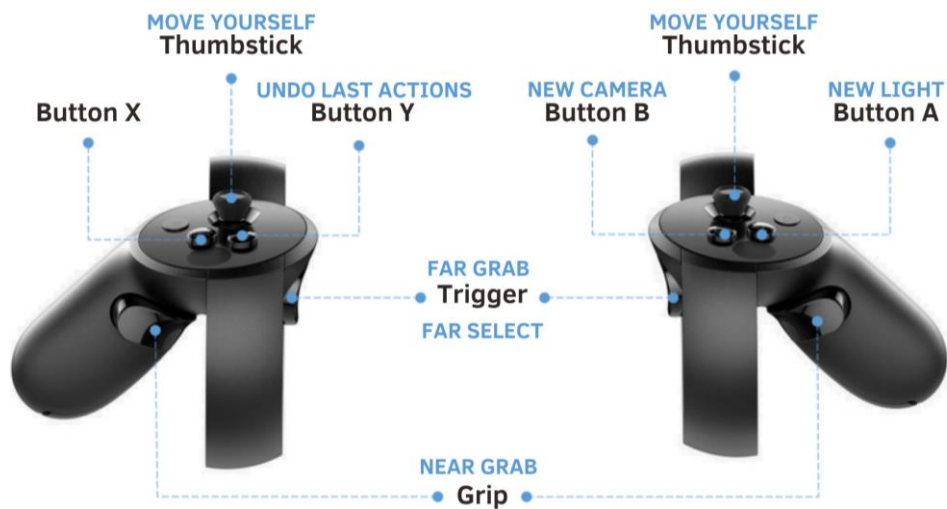


Figura 5.18 Funzionalità dei tasti dei controller nella fase di simulazione.

5.9. Interfaccia vocale

L'utente, oltre ad interagire con i controller, ha la possibilità di usare la propria voce, nel momento in cui preme il tasto con il simbolo del microfono presente sulla tastiera. Può quindi dettare al posto di digitare con la tastiera, grazie alla funzione *Dictate()* dello script *KeyboardKeyFunc*.

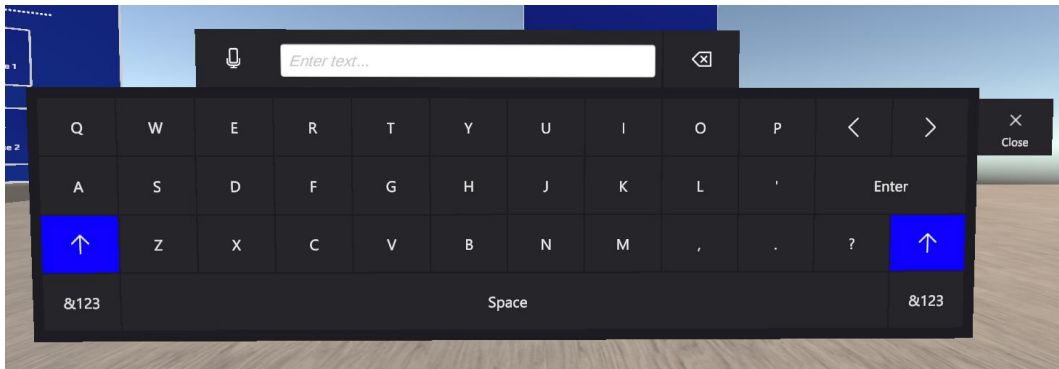


Figura 5.19 Tastiera e tasto microfono.

5.10. Feedback visivi e uditivi

Ogni azione di input compiuta dall'utente genera una risposta immediata da parte del sistema. I tasti cambiano colore, quando l'utente ci passa sopra il cursore e quando li preme. Nel momento in cui l'utente preme "See Storyboard", appare un pannello che indica che la generazione dello storyboard è in corso, accompagnato da un simbolo di caricamento.

Quando l'utente aggiunge in scena una nuova camera o una nuova luce viene notificato attraverso un pannello che gli appare davanti per pochi secondi e che lo invita a fare un passo indietro per vedere il nuovo oggetto aggiunto.

I riscontri uditivi sono altrettanto importanti per notificare l'utente che qualcosa è successo. Diversi suono spazializzati o in stereo notificano infatti l'utente quando quest'ultimo preme un tasto, sia della UI virtuale sia dei controller.

5.11. Generazione storyboard

Durante l'utilizzo dell'applicazione, quando si assegnano le azioni ai personaggi, in automatico viene generata una frase visibile nel "Storyboard Panels Menù". Tutte le frasi generate appartenute alla stessa vignetta verranno raggruppate nella descrizione del rispettivo screenshot. Lo script *PhraseGenerator* si occupa di gestire la generazione delle frasi. In base a se l'azione è contemporanea o consecutiva alla precedente viene scelto in modo casuale l'avverbio temporale da una lista di stringhe presente nel codice. Nel caso della prima azione eseguita l'avverbio è sempre "*First*".

La funzione *GenerateSimplePhrase()* di *PhraseGenerator*, richiamata da *AnimaPersonaggio* riceve come parametri il nome e il tipo del soggetto, l'azione, il nome e il tipo dell'oggetto e se l'azione il soggetto la svolge su se stesso o no. Genera quindi la frase in inglese, aggiungendo la "s" della terza persona, "the" se il tipo coincide con il nome, quindi, se il personaggio non è stato rinominato, e la forma "-ing" se il personaggio smette di compiere un'azione che gli aveva fatto cambiare stato, come ad esempio ballare.

Nel caso in cui l'azione premuta sia "*TALK TO*", viene chiamata la funzione *GenerateDialogue()* che aggiunge il testo inserito dall'utente tra virgolette.

La funzione *GenerateMovementPhrase()* gestisce le frasi riguardanti i movimenti che i personaggi compiono nel momento in cui l'utente preme il tasto "*MOVE*" e definisce un punto di arrivo sul piano. Nel caso in cui il personaggio si stia muovendo verso un oggetto viene generata una frase in cui si indica che il personaggio si sta muovendo in direzione dell'oggetto.

La funzione *ClearBuffer()* si occupa di prendere il buffer contenente le frasi e di eliminare le frasi ripetute, se consecutive. Questo per eliminare principalmente le frasi di camminata ripetute, in quanto ogni volta in cui l'utente seleziona un punto nel piano viene generata una frase di movimento.

La classe *OutputGenerator* ha il compito di scrivere il file HTML finale in base ai parametri presi dagli altri script. Viene quindi definito il nome dello storyboard in base al nome dato dall'utente quando ha rinominato la scena. Per ogni vignetta viene generata una nuova sezione colorata di azzurro contenente la foto, la descrizione e i parametri impostati dall'utente durante l'utilizzo dell'applicazione.

Una volta generato, lo storyboard finale viene aperto sul PC. L'utente può modificare i campi della descrizione, della lunghezza focale e della durata delle vignette e aggiungerne altri a suo piacimento.

Sotto ogni vignetta sono presenti due descrizioni: l'originale generata dallo script *PhraseGenerator* dell'applicazione e una riformulata dall'intelligenza artificiale.

Il task *Rephrase()* presente nell'*OutputGenerator* si occupa di effettuare la richiesta e di ricevere la risposta. Per utilizzare la versione gratuita ogni richiesta è intervallata da una pausa di 20 secondi in quanto OpenAI impone il limite di 3 richieste al minuto o in base al numero di caratteri, a seconda di quale condizione si verifica prima.

Le descrizioni sono state mantenute entrambe per dare la possibilità all'utente di scegliere la migliore, cancellando l'altra se necessario.

Use Case 2 Storyboard

Panels are editable! Click on them to make adjustments.

Go to 'print' and select 'save as pdf' to download your storyboard!

Shot #1



Original: First, Paola dances. Then, Paola jumps. At the same time, Sofia works out.

AI edit: Sofia works out while Paola dances and jumps.

Shot #2



Original: Following, Sofia stops working. Sofia walks. Sofia walks towards the bench. After that, Sofia sits on the bench.

AI edit: Following, Sofia stops working and she walks towards the bench. After that, Sofia sits on the bench.

Figura 5.20 Esempio di storyboard finale costituito da due vignette.

5.12. Palette colori

Dal punto di vista grafico, per la UI, è stata utilizzata la seguente palette di colori:

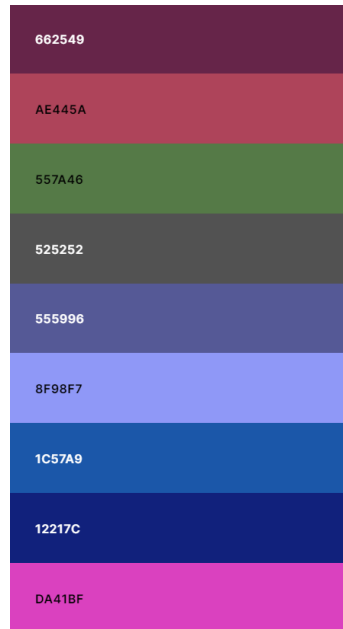


Figura 5.21 Colori utilizzati nell'applicazione per gli elementi della UI.

I colori **#525252** e **#12217C** sono quelli di default dei componenti MRTK e sono stati mantenuti per i menù. La maggior parte dei tasti ha il colore **#1C57A9**. Il tasto “Rename” e i tasti delle azioni che si riferiscono al soggetto hanno il colore **#AE445A**. Le azioni di tipo “Stop action” assumono la colorazione più scura **#662549**. I tasti che si riferiscono al complemento oggetto assumono invece la colorazione **#557A46**. I tasti “Next Panel”, “Get Control” e “See Storyboard” sono colorati di **#DA41BF**. Il tasto attivo tra “Contemporary” e “Consecutive” è colorato di **#DA41BF**, mentre quello non attivo di **#555996**. Quando un qualsiasi tasto viene indicato dal puntatore di un controller assume la colorazione **#12217C**; quando viene premuto si colora di **#8F98F7** per pochi secondi.

Nel tutorial un tasto disattivo assume la colorazione **#525252**. Tutte le scritte sono bianche, quelle dei tasti disattivi sono **#525252**. I colori vengono assegnati ai componenti attraverso i materiali e i temi creati con i colori precedentemente indicati.

6. Test e analisi dei risultati

Per poter collaudare l'applicazione sono stati definiti diversi casi d'uso. Sono stati svolti in una prima fase i test di usabilità con tre utenti esperti; successivamente è stato chiesto ad un gruppo di studenti del Politecnico di Torino di provare l'applicazione in modo da poterne valutare l'usabilità e il carico cognitivo in base alle risposte da loro fornite a due questionari.

Per i casi d'uso viene fornita all'utente una scena con gli oggetti utili già posizionati sul piano e rinominati. L'utente testa quindi l'applicazione dalla fase di simulazione.

6.1. Primo caso d'uso

Per il primo caso d'uso è stata creata una scena utilizzando un personaggio di tipo *woman* rinominato *Anna*, un personaggio di tipo *man* rinominato *Luca*, un oggetto di tipo *piano*, un oggetto di tipo *chair* e tre oggetti di tipo *wall*. La scena è stata rinominata "*Use Case 1*" e salvata. L'utente poteva quindi caricare la scena attraverso il tasto "*Load Scene*".

All'utente viene fornito uno script, riportato nell'applicazione in un pannello apribile premendo il tasto "*Use Case 1*" presente nel "*Main Menù*", in cui è presente una descrizione dettagliata della scena e una sequenza di 6 vignette accompagnate da alcune indicazioni, quali tipo di inquadratura, durata e descrizione.

La descrizione della scena presentata all'utente è la seguente:

“Anna e Luca si trovano nella loro casa.

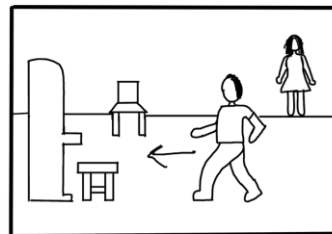
Luca si avvicina al pianoforte e comincia a suonare, nel mentre Anna si siede su una sedia accanto a lui.

A un certo punto Anna si alza dalla sedia e inizia a ballare.

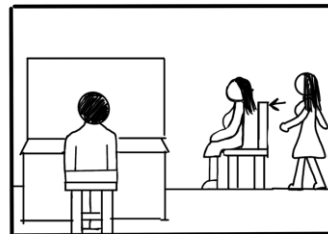
Successivamente Luca smette di suonare e si avvicina ad Anna.

Luca applaude ad Anna.

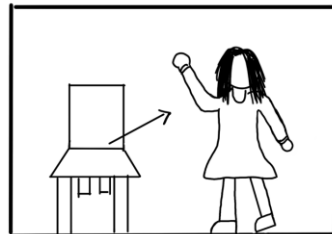
Anna smette di ballare e gli sorride.”



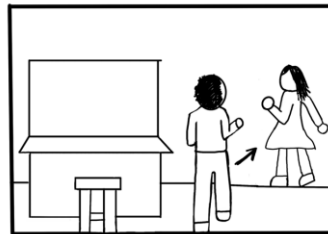
Tipo di inquadratura: Totale
Durata: ≈ 2 sec
Descrizione: Luca si avvicina al pianoforte.



Tipo di inquadratura: Figura intera
Durata: ≈ 4 sec
Descrizione: Luca inizia a suonare il pianoforte. Nel mentre Anna cammina verso una sedia vicino a lui e si siede.



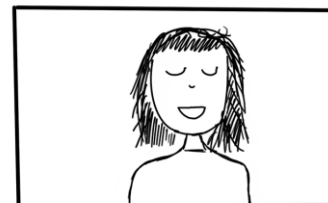
Tipo di inquadratura: Figura intera
Durata: ≈ 3 sec
Descrizione: Ad un certo punto Anna si alza e inizia a ballare.



Tipo di inquadratura: Totale
Durata: ≈ 5 sec
Descrizione: Luca smette di suonare e cammina verso Anna.



Tipo di inquadratura: Mezza figura
Durata: ≈ 4 sec
Descrizione: Luca applaude ad Anna



Tipo di inquadratura: Primo piano
Durata: ≈ 2 sec
Descrizione: Anna smette di ballare e sorride.

Figura 6.1 Storyboard fornito per il primo caso d'uso.

6.2. Secondo caso d'uso

Per il secondo caso d'uso è stata creata una scena utilizzando due personaggi di tipo *woman* rinominati *Sofia* e *Paola*, un oggetto di tipo *tree* e uno di tipo *bench*. La scena è stata rinominata “*Use Case 2*” e salvata.

All'utente in questo caso veniva fornita solo la descrizione della scena, lasciandogli quindi più libertà rispetto al primo caso d'uso:

“Sofia e Paola sono al parco.

Sofia è intenta a fare degli esercizi fisici di fianco a una panchina. Nello stesso momento Paola, che si trovava dall'altra parte del parco, inizia a camminare verso la panchina.

Una volta arrivata vicino alla panchina Paola saluta Sofia.

Sofia invita Paola ad unirsi a lei dicendole "Alleniamoci insieme!"

Paola inizia ad allenarsi con Sofia.

Dopo un po' di tempo Paola si siede sulla panchina.”

6.3. Test con gli esperti

Sono stati contattati tre esperti di domino, che stanno svolgendo un dottorato o post-doc al Politecnico di Torino, per poter avere un parere sull'applicazione prima di farla provare agli utenti e per individuare eventuali bug ed elementi da rivedere o sistemare. Tutti gli esperti hanno conoscenze sull'utilizzo di dispositivi di VR.

Agli esperti è stata fatta una breve introduzione sull'argomento ed è stato chiesto di svolgere il tutorial per intero, di caricare un caso d'uso a propria scelta e provare l'applicazione per tutto il tempo a loro necessario, senza avere il vincolo

di eseguire per intero il caso d'uso. Gli esperti hanno provato l'applicazione mentre erano seduti.

Gli esperti potevano fare commenti a voce durante l'utilizzo dell'applicazione e chiedere aiuto e informazioni in qualsiasi momento.

Per valutare l'usabilità del sistema, in questa fase, sono state usate le 10 Euristiche di Nielsen [76]. Queste regole, dette euristiche, sono state definite nel 1994 da Jakob Nielsen e ancora oggi sono fondamentali per valutare la UX dei sistemi:

1. *Visibilità dello stato del sistema*: i progetti dovrebbero tenere gli utenti informati su ciò che sta accadendo, attraverso un feedback appropriato e tempestivo.
2. *Corrispondenza tra il sistema e il mondo reale*: il design deve parlare la lingua degli utenti. Usare parole, frasi e concetti familiari all'utente.
3. *Controllo e libertà dell'utente*: gli utenti possono eseguire azioni per errore. Hanno bisogno di una "uscita di emergenza" chiaramente contrassegnata per lasciare l'azione indesiderata.
4. *Consistenza e standard*: gli utenti non dovrebbero chiedersi se parole, situazioni o azioni diverse significano la stessa cosa.
5. *Prevenzione dell'errore*: i messaggi di errore sono importanti, ma i migliori design prevengono accuratamente che si verifichino problemi.
6. *Riconoscimento anziché ricordo*: Ridurre al minimo il carico di memoria dell'utente rendendo visibili elementi, azioni e opzioni. Evitare di fare in modo che gli utenti ricordino le informazioni.
7. *Flessibilità ed efficienza d'uso*: le scorciatoie, nascoste agli utenti inesperti, possono velocizzare l'interazione per l'utente esperto.
8. *Design estetico e minimalista*: le interfacce non devono contenere informazioni irrilevanti. Ogni unità di informazione in più in un'interfaccia compete con le unità di informazione pertinenti.
9. *Riconoscimento, diagnosi e ripristino dagli errori*: i messaggi di errore devono essere espressi in un linguaggio semplice, indicare con precisione il problema e suggerire in modo costruttivo una soluzione.

10. *Aiuto e documentazione*: è meglio se il design non ha bisogno di ulteriori spiegazioni. Tuttavia, potrebbe essere necessario fornire la documentazione per aiutare gli utenti a completare le proprie attività.

Al termine di ogni sessione di utilizzo dell'applicazione è stato chiesto ad ogni esperto di valutare l'applicazione dando un punteggio da 1 a 5, in base alla scala Likert, per ogni Euristiche di Nielsen.

	<i>Esperto 1</i>	<i>Esperto 2</i>	<i>Esperto 3</i>	<i>MEDIA</i>
<i>1 - Visibilità dello stato del sistema</i>	4	4	5	4,33
<i>2 - Corrispondenza tra il sistema e il mondo reale</i>	5	5	5	5
<i>3 - Controllo e libertà dell'utente</i>	5	5	4	4,67
<i>4 - Consistenza e standard</i>	4	5	5	4,67
<i>5 - Prevenzione dell'errore</i>	4	3	4	3,67
<i>6 - Riconoscimento anziché ricordo</i>	5	4	5	4,67
<i>7 - Flessibilità ed efficienza d'uso</i>	5	5	4	4,67
<i>8 - Design estetico e minimalista</i>	4	5	4	4,33
<i>9 - Riconoscimento, diagnosi e ripristino dagli errori</i>	4	3	4	3,67
<i>10 - Aiuto e documentazione</i>	5	5	5	5
<i>MEDIA</i>	4,5	4,4	4,5	

Tabella 6.1 Punteggio dato da ogni esperto all'applicazione usando le Euristiche di Nielsen, in una scala da 1 a 5.

Tutti e tre gli esperti hanno valutato positivamente l'applicazione in generale, la media dei punteggi dati per ogni euristica da ogni esperto è maggiore di 4,4.

Le euristiche riguardanti gli errori, *prevenzione dell'errore e riconoscimento, diagnosi e ripristino dagli errori*, sono quelle che meno sono emerse nell'applicazione, avendo ricevuto un punteggio in media di 3,67/5. Seguono *visibilità dello stato del sistema e design estetico e minimalista* con punteggio 4,33/5. Tutte le altre, a parte *corrispondenza tra il sistema e il mondo reale e aiuto e documentazione* che hanno ottenuto punteggio 5/5, hanno avuto punteggio in media di 4,67/5.

I principali commenti e suggerimenti fatti dagli esperti sono stati i seguenti:

1. Presenza di troppi pannelli nell'interfaccia.
2. Mancanza di un avviso nel caso in cui si passa alla vignetta successiva premendo il tasto "Next Panel" e non è stato effettuato uno screenshot in quella precedente.
3. Il tasto *Y* del controller, per annullare le ultime azioni successive all'ultimo screenshot eseguito, sarebbe meglio averlo anche come tasto fisico sulla UI.
4. I tasti del tipo "Stop Action" per tornare nello stato di *idle*, dopo aver assegnato un'azione che fa cambiare stato al personaggio, sono poco evidenti in mezzo agli altri tasti delle azioni.
5. I movimenti nell'ambiente eseguiti con i thumbstick possono causare motion sickness.
6. Possibilità di non fare seguire al display della camera i movimenti della camera.
7. Far orientare sempre tutti i pannelli verso l'utente.
8. In modalità *Consecutive* le azioni devono stoppare le precedenti azioni che hanno fatto cambiare stato, far tornare il personaggio in stato di *idle* senza obbligare l'utente ad assegnare l'azione del tipo "Stop Action". In modalità *Contemporary* le azioni che non cambiano stato, eseguite mentre il personaggio non è in stato di *idle*, non devono far tornare in stato di *idle* il

personaggio, ma lasciarlo nel suo stato a meno che l'azione sia del tipo "Stop Action".

L'applicazione è stata migliorata in base ai commenti fatti dagli esperti, prima di presentarla per i test finali. La versione dell'applicazione descritta nel Sviluppo dell'applicazione è quella con le ultime modifiche effettuate dopo i test con gli esperti.

6.4. Test Utenti

Per i test con gli utenti è stato ripreso il primo caso d'uso proposto agli esperti e modificato. In particolare, si è deciso di fornire solo la descrizione della scena, come per il secondo caso d'uso, per concentrare l'attenzione dell'utente sull'utilizzo dell'applicazione e non sul dover riprodurre perfettamente lo storyboard fornito. Agli utenti, dopo una breve introduzione sui motivi dello sviluppo dell'applicazione di ricerca, è stato chiesto di prendere confidenza con la stessa svolgendo per intero il tutorial e successivamente di caricare la scena contenente il primo caso d'uso e la descrizione dello storyboard da realizzare. Si è consigliato di produrre tra le 5 e le 7 vignette e di rimanere seduti su una sedia per tutta la durata del test. Durante il tutorial gli utenti erano liberi di porre domande, è stata richiesta invece più autonomia in fase di creazione dello storyboard per il caso d'uso assegnato.

Durante lo svolgimento dei test si è tenuta traccia dei tempi di svolgimento del caso d'uso e dei principali errori commessi da ogni utente.

Gli obiettivi dei test sono stati quelli di verificare l'usabilità, il carico di lavoro percepito e infine l'esperienza di utilizzo dell'applicazione, attraverso delle domande aperte.

6.5. Tempi ed errori - Utenti

Per completare il caso d'uso in media gli utenti hanno impiegato 13:50 minuti. La deviazione standard è di 0,10.

Gli utenti hanno commesso o 1 errore o nessun errore, come evidenziato nel Grafico 6.1.

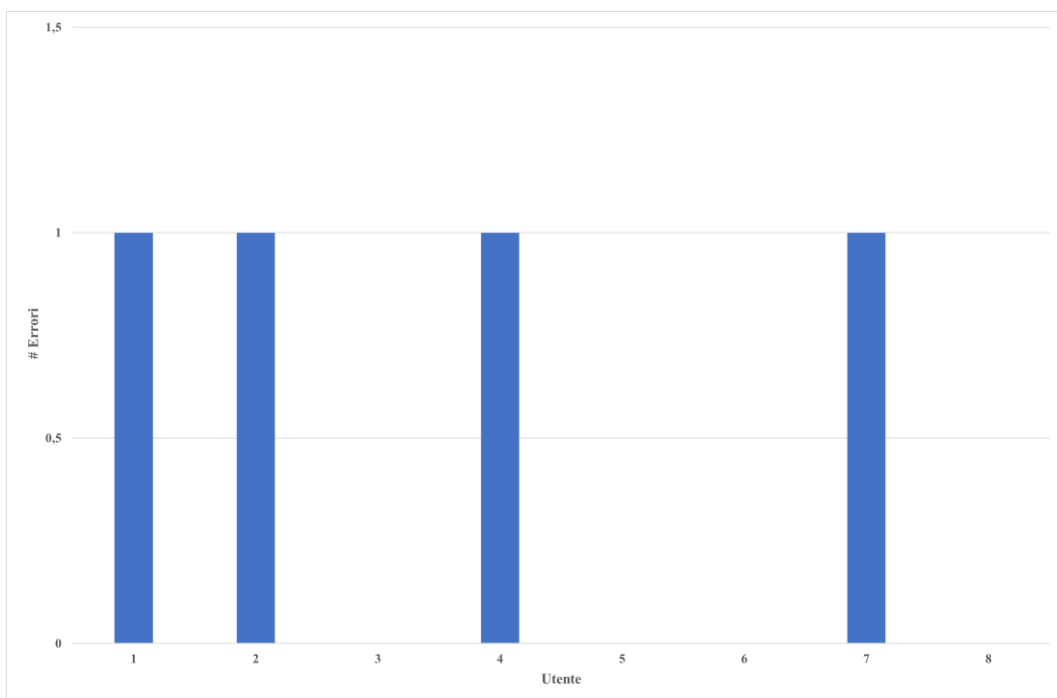


Grafico 6.1 Numero di errori commessi da ogni utente.

In media sono stati commessi 0,5 errori.

Gli errori commessi riguardano la scelta della durata della vignetta, l'ordine di assegnazione delle azioni ai personaggi, il passare alla vignetta successiva prima di assegnare le azioni e il fatto di dimenticarsi di posizionare le luci prima di effettuare lo screenshot e passare alla vignetta successiva. In generale, comunque, gli utenti sono riusciti a portare a termine il proprio compito, producendo a testa dalle 5 alle 7 vignette per storyboard.

6.6. Questionario SUS e domande generali

Il questionario “System Usability Scale” (SUS) [77] è stato creato nel 1986 da John Brooke e consiste di 10 affermazioni e 5 opzioni di risposta per ognuna per permettere agli utenti di indicare attraverso la scala Likert quanto si è d’accordo o in disaccordo con l’affermazione riguardante l’usabilità del sistema. Il questionario può essere utilizzato per valutare diverse tipologie di prodotti, tra cui hardware, software, siti e applicazioni ed è diventato un questionario standard per l’industria.

Le affermazioni originali in lingua inglese sono:

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

I punteggi dati alle singole affermazioni non sono significativi individualmente, ma è necessario calcolare il punteggio SUS.

Alle affermazioni dispari bisogna sottrarre 1 al punteggio attribuito. Per le affermazioni pari bisogna sottrarre a 5 il valore attribuito dall'utente. Infine, si possono sommare i valori ottenuti e moltiplicare per 2,5. Si otterrà un valore tra 0 e 100 confrontabile con la seguente figura.

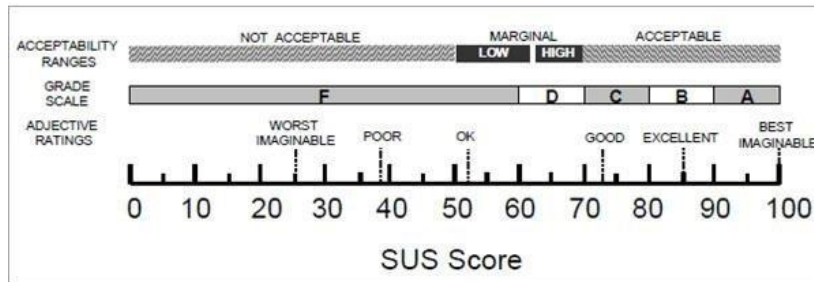


Tabella 6.2 Tabella per interpretare il valore SUS ottenuto [78].

Agli utenti è stato proposto un modulo Google, A.2, composto da tre sezioni. Nella prima parte sono presenti domande personali, come ad esempio età e corso di studi, e domande generali per capire il livello di conoscenza dell'utente sugli argomenti di Storyboard e VR, alcune di queste domande sono le stesse presenti nel questionario MoSCoW proposto ad un gruppo di studenti prima dello sviluppo dell'applicazione. Nella seconda sezione è presente il questionario SUS; mentre nella terza vengono proposte all'utente le seguenti domande aperte:

1. Quali funzionalità dell'applicazione ti sono piaciute maggiormente?
2. Cosa ti è piaciuto di meno o hai avuto difficoltà a fare?
3. Quali funzionalità aggiungeresti?
4. Che cosa cambieresti dell'applicazione?

Ai test hanno partecipato 8 studenti del Politecnico di Torino, dei quali 5 sono studenti a fine carriera del corso di laurea magistrale in ingegneria del cinema e dei mezzi di comunicazione, 1 studente è in fase di iscrizione al medesimo corso, 1 studente è del corso di laurea magistrale in ingegneria informatica orientamento grafica e multimedia e 1 studente è iscritto alla triennale in ingegneria informatica.

L'età dei partecipanti va dai 19 ai 39 anni, con una media di 25,5 e una varianza di 37,43; di questi il 50% è donna e il 50% è uomo.

Tutti gli utenti hanno conoscenze sul concetto di realtà virtuale e la maggior parte degli utenti non realizza storyboard abitualmente, uno tra questi afferma di non aver mai realizzato storyboard e di non aver mai indossato un visore di realtà virtuale prima del test. Nonostante queste premesse le risposte date da quest'ultimo utente ai questionari sono concordi a quelle date dagli altri utenti più esperti sull'argomento. In base a queste considerazioni si è deciso di trattare i dati in blocco e di non analizzarli suddividendoli in sottocategorie.

Dai risultati del questionario si è potuto calcolare per ogni utente il punteggio SUS riportato nel Grafico 6.2. Tutti i valori rientrano nel range "Acceptable". In media il punteggio SUS risulta essere 85,31, un punteggio "Excellent" che viene associato al voto B. La deviazione standard è di 8,60.

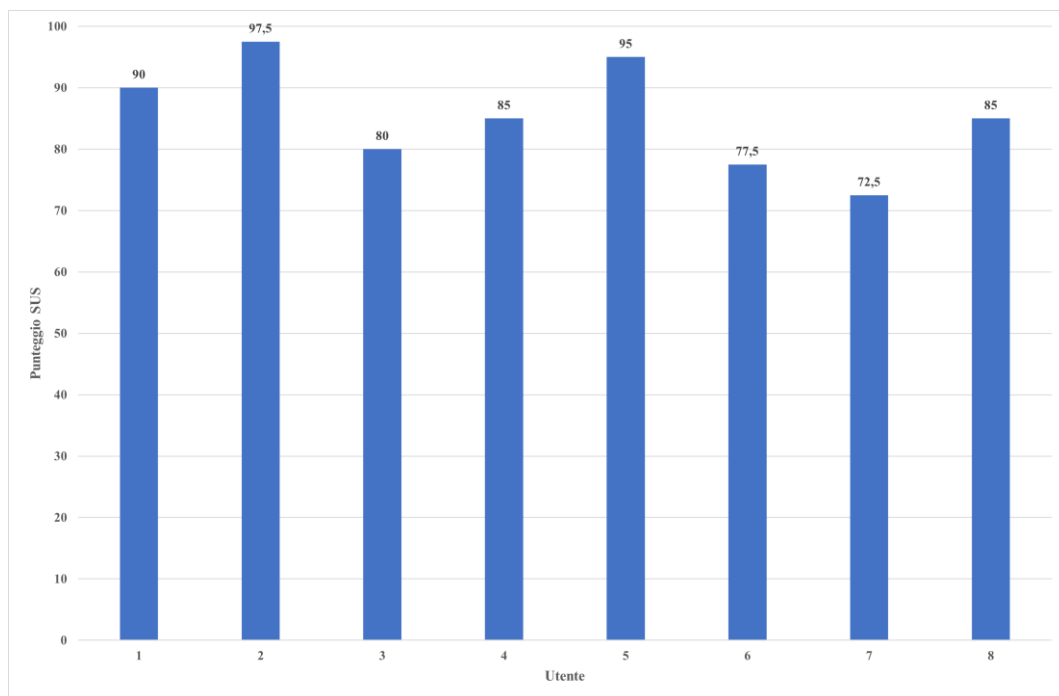


Grafico 6.2 Punteggio SUS ottenuto per ogni utente.

Le funzionalità maggiormente apprezzate dagli utenti sono state la camera con il display scalabile e posizionabile a proprio piacimento nello spazio, la possibilità

di poter posizionare le luci a proprio piacimento e la modalità di interazione con i personaggi. Gli utenti segnalano invece come funzionalità difficile da compiere quella di muoversi agevolmente nello spazio, che ad un utente, inoltre, creava motion sickness. Alcuni hanno avuto difficoltà a leggere le scritte e lamentavano il fatto di dover essere obbligati ad avvicinarsi i pannelli, operazione non sempre facile da compiere. Un utente non ha apprezzato la creazione delle camere nella stessa posizione del proprio volto.

Tra le funzionalità che gli utenti aggiungerebbero risultano:

1. Presenza di una bolla vicino alla camera e possibilità di decidere quali elementi mettere a fuoco.
2. Presenza di un maggior numero di azioni assegnabili ai personaggi.
3. Possibilità di salvare una scena anche in modalità storyboarding per completare il lavoro successivamente.
4. Possibilità di far corrispondere la vista della camera con la posizione dell'utente in modo da poter visualizzare in prima persona la scena, dal punto di vista della camera.
5. Esportazione di un floor plan basato sulla posizione degli oggetti in scena.
6. Presenza di template per poter scegliere la disposizione dei pannelli in scena che più aggrada.
7. Rotazione sul posto dei personaggi in scena.

Tra le funzionalità che gli utenti modificherebbero, invece, risultano:

1. Adozione di un font diverso, come ad esempio EasyReadingPro.
2. Il sistema di movimento dell'utente nella scena.
3. Il design della UI.
4. Scelta della vignetta su cui lavorare, con possibilità di tornare a modificare le precedenti vignette anche dopo aver premuto "*Next Panel*".

Osservando gli utenti durante l'utilizzo dell'applicazione si è notato che alcune funzionalità come il poter posizionare il sole, poter settare la durata della vignetta e spesso il posizionamento delle luci e l'aggiunta di nuove camere non venivano sempre utilizzate durante il processo creativo di creazione dello storyboard.

6.7. Questionario NASA-TLX

Il Task Load Index (TLX) ufficiale della NASA [79] è uno strumento che consente agli utenti di eseguire valutazioni soggettive del carico di lavoro. È stato sviluppato originariamente da Sandra Hart dell'Ames Research Center (ARC) della NASA negli anni '80, il TLX della NASA è diventato lo standard di riferimento per la misurazione del carico di lavoro soggettivo in un'ampia gamma di applicazioni. Incorporando una procedura di valutazione multidimensionale, NASA-TLX ricava un punteggio complessivo del carico di lavoro su una scala di 100 punti basato su una media ponderata delle valutazioni su sei sotto-scale:

- **Mental Demand** (Richiesta Mentale) (low/high): how much mental and perceptual activity was required? Was the task easy or demanding, simple or complex, forgiving or exacting?
- **Physical Demand** (Richiesta Fisica) (low/high): how much physical activity was required? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?
- **Temporal Demand** (Richiesta Temporale) (low/high): how much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?
- **Performance** (Prestazione) (good/poor): how successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?
- **Effort** (Sforzo) (low/high): How hard did you have to work (mentally and physically) to accomplish your level of performance?
- **Frustration Level** (Frustrazione) (low/high): how insecure, discouraged, irritated, stressed, and annoyed versus secure, gratified, content, relaxed, and complacent did you feel during the task?

La somministrazione prevede due fasi:

1. Valutazione del carico di lavoro per ogni sotto-scala attraverso la scheda di valutazione in cui l'utente può indicare il valore in una scala da 0 a 100, divisa in 20 intervalli.
2. 15 confronti a coppie delle sei sotto-scale per poter definire i pesi relativi ad ognuna, che varieranno tra 0 e 5.

Per ognuna delle 6 sotto-scale si ottiene quindi il peso, dalla seconda fase, e la valutazione grezza, dalla prima. Per calcolare la valutazione ponderata si moltiplica il peso per la valutazione grezza. Si sommano allora i 6 valori ottenuti e si divide il risultato per 15 per ottenere il carico di lavoro complessivo per ogni utente. Valori alti indicano un carico di lavoro elevato, al contrario valori bassi non indicano un eccessivo sforzo.

Il questionario è stato somministrato usando un software non ufficiale per computer. Il sito ufficiale propone il questionario in versione cartacea o in versione applicazione per dispositivi Apple.

Score	Category
0-9	Low
10-29	Medium
30-49	Somewhat High
50-79	High
80-100	Very High

Tabella 6.3 Interpretazione risultati NASA-TLX

In media il valore ottenuto è di 23,37, il quale rientra nel range “*Medium*”. La deviazione standard è di 11,44. Il task richiede quindi all’utente un carico di lavoro medio.

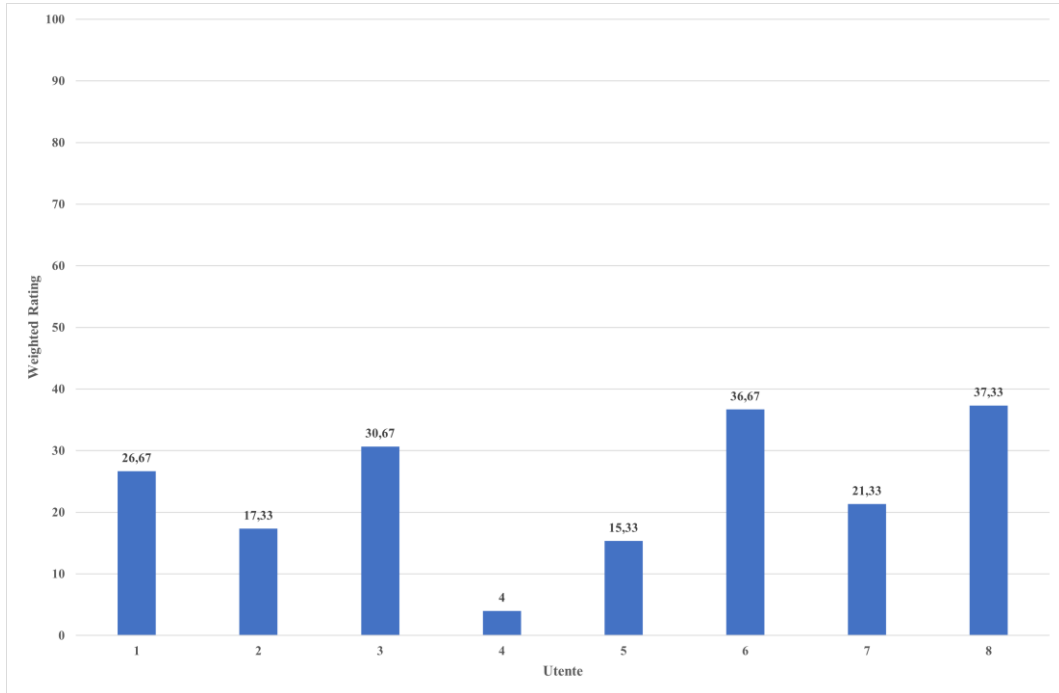


Grafico 6.3 Risultati questionario NASA-TLX.

Come dimostra il Grafico 6.4, che riporta in media per ogni sotto-scala il valore pesato ottenuto dai questionari posti agli utenti, il parametro che più ha influito sul carico di lavoro dell’utente è stato il “*Mental Demand*”, seguono “*Effort*”, “*Performance*”, “*Physical Demand*”, “*Temporal Demand*” e infine “*Frustration Level*”.

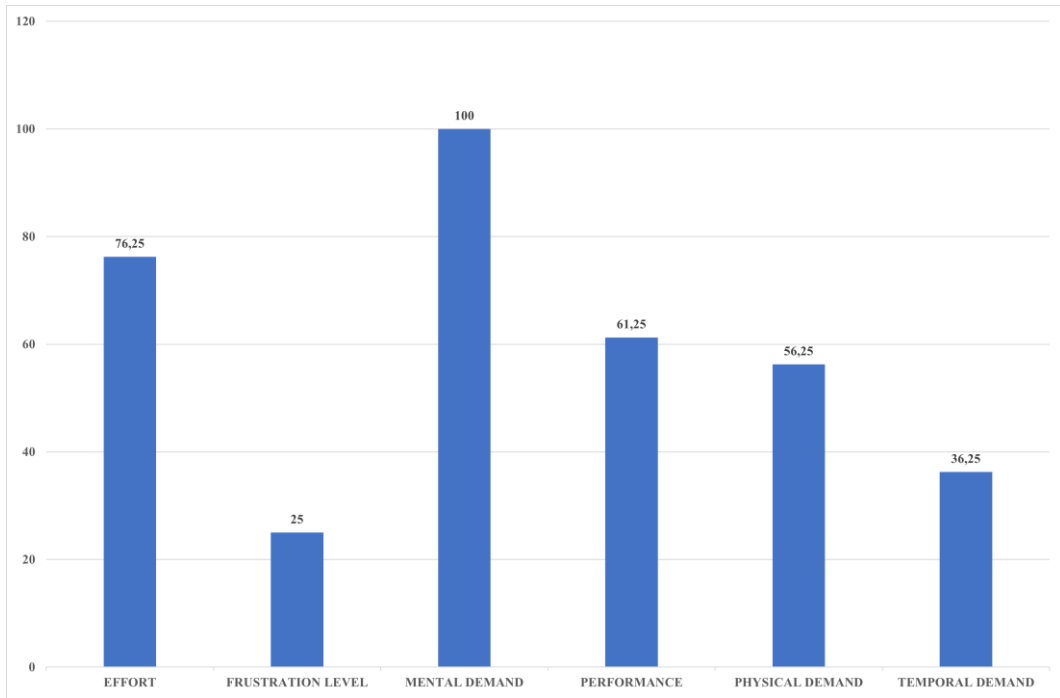


Grafico 6.4 Media dei valori pesati assegnati dagli utenti alle 6 sotto-scale.

7. Conclusioni

L'applicazione sviluppata per questa tesi ha avuto come principale obiettivo quello di utilizzare la tecnologia della VR per la creazione di storyboard; un metodo alternativo a quello classico di creazione storyboard eseguito disegnando a mano libera o utilizzando applicazioni 2D. Alcune funzionalità dell'applicazione si basano su quelle già presenti nello stato dell'arte, altre come l'assegnazione di animazioni predefinite, generazione delle descrizioni delle vignette in modo automatico e la creazione di camere nello stesso punto dello sguardo dell'utente possono essere definite come funzionalità di ricerca.

7.1. Sviluppi futuri

Il progetto creato per questa tesi può essere, in futuro, ulteriormente arricchito di funzionalità e migliorato a livello di quantità di animazioni e oggetti virtuali inseribili in scena.

L'applicazione è stata sviluppata per l'Oculus Rift ma con alcune modifiche può essere adattata a diversi dispositivi VR, anche più recenti.

Una funzionalità interessante, implementabile in futuro, è poter dare la possibilità all'utente di interagire con una versione in cui i vari oggetti, che l'utente inserisce per la creazione dell'ambiente, siano in scala 1:1 e non più posti sopra un tavolo; oppure al contrario che l'utente possa scegliere di essere delle dimensioni degli oggetti e ritrovarsi anche lui nella scena, ora soltanto in miniatura.

Una seconda funzionalità potrebbe essere quella di consentire all'utente di creare le camere e le luci nuove non solo nel punto del proprio viso ma anche in altri punti dello spazio, decisi dall'utente attraverso, ad esempio, la posizione dei propri controller.

Le azioni assegnate ai personaggi dall'utente nel momento in cui preme "OTHER" al momento non contengono alcuna animazione collegata, ma generano solo la parte testuale di descrizione della vignetta. Come sviluppo futuro si potrebbe integrare una funzionalità che vada a prendere da un database di animazioni online, come ad esempio Mixamo, l'animazione che più si avvicina a quella definita dall'utente.

Un ulteriore sviluppo riguardante le animazioni potrebbe essere quello di permettere l'assegnazione di animazioni contemporanee che vadano a riferirsi solo a determinate ossa delle armature dei personaggi; permettendo ad esempio al personaggio di poter salutare anche mentre è seduto, senza l'obbligo per l'utente di farlo alzare.

Una ulteriore aggiunta potrebbe essere quella di consentire la creazione di un altro elemento fondamentale della fase di pre-visualizzazione: l'animatic. Questo potrebbe essere ottenuto registrando quello che le camere virtuali inquadrano, a discrezione dell'utente.

Una funzionalità, richiesta da un utente in fase di test, è quella di poter ottenere dall'applicativo, come secondo output, l'esportazione di un floor plan, altro elemento redatto in fase di pre-produzione, basato sulla posizione degli oggetti in scena.

Infine, un'ultima funzionalità integrabile potrebbe essere quella di dare la possibilità agli utenti di poter aggiungere i propri oggetti virtuali e le proprie sceneggiature nell'ambiente virtuale o via computer, prima di iniziare ad utilizzare l'applicazione, o in una sezione apposita dell'applicazione, collegata al web o alle cartelle del computer.

Appendice

A.1 Questionario MoSCoW

VR Tabletop Storyboarding

Nelle prime fasi della produzione, gli storyboard vengono utilizzati per descrivere visivamente la storia e la sceneggiatura tramite l'utilizzo di disegni e immagini per capire in che modo organizzare e filmare le scene. La realizzazione di uno storyboard è essenziale per la creazione di un prodotto cinematografico ma risulta essere anche molto lunga e laboriosa: in particolare ogni inquadratura che si vuole realizzare deve essere disegnata da zero e questo fa sì che i tempi di realizzazione diventino molto lunghi. Nella creazione degli storyboard in 2D inoltre può risultare complesso definire la posizione effettiva della camera e dei personaggi, è complesso capire come e dove posizionare le luci e ogni errore o modifica necessita di cancellare e ridisegnare l'intera inquadratura.

Nel progetto di tesi "Un approccio in VR al tabletop storyboarding" ci poniamo l'obiettivo di realizzare un'applicazione che permetta la creazione di uno storyboard in maniera più semplice e immersiva. **Abbiamo bisogno del tuo aiuto** per valutare quali sono i requisiti essenziali per l'applicazione.

L'idea è quella di creare un ambiente 3D composto da tutti gli elementi della scena come oggetti e attori e di poter prendere il controllo dei personaggi muovendoli nello spazio e permettendo loro di compiere delle azioni su altri personaggi e oggetti. Una volta realizzato l'ambiente e deciso che azioni far compiere ai personaggi l'utente può creare lo storyboard realizzando degli screenshot della scena, accompagnati da delle descrizioni che possono essere generate automaticamente.

Questo progetto prevede la creazione di un'applicazione di storyboarding in realtà virtuale, con un'interazione più naturale rispetto ad un'applicazione desktop ma i vantaggi del digitale rispetto ad uno storyboard su carta. La realtà virtuale permetterà all'utente di allestire un set virtuale e generare uno storyboard in maniera intuitiva ed immersiva. Inoltre, stiamo valutando lo sviluppo di due modalità di utilizzo:

- nella modalità tabletop l'utente visualizza su di una scrivania virtuale un set virtuale in miniatura;
- nella modalità scala 1:1, l'utente visualizza la scena come in un videogioco in prima persona a dimensioni reali, con un elevato livello di immersione e facendo immedesimare l'utente nei panni del regista sul set

Trattamento dei dati

I dati da Lei forniti non saranno comunicati a terzi e verranno distrutti o resi anonimi subito dopo la raccolta. La comunicazione o la diffusione dei risultati delle rilevazioni può avvenire soltanto in forma anonima o aggregata, in modo che i dati elaborati nell'ambito del sondaggio non possano essere associati ad interessati identificati o identificabili.

Età *

La tua risposta

Quante volte ti è capitato di realizzare degli storyboard? *

- Spesso, li utilizzo abitualmente
- Talvolta
- Quasi mai
- Non ho mai realizzato uno storyboard

Quali strumenti utilizzi solitamente per realizzare uno storyboard? *

- Carta e penna
- Disegno digitale
- Applicazioni 2D (es. Canva, Figma)
- Applicazioni 3D
- Altro:

Conosci la realtà virtuale (VR)? *

- Sì
- No

Quanto spesso utilizzi applicazioni VR? *

- Più di una volta al giorno
- Una volta al giorno
- Qualche volta a settimana
- Una volta a settimana
- Una volta al mese
- Meno di una volta al mese
- Mai

Quale dispositivo usi più spesso per applicazioni VR? Se è un dispositivo dedicato, indica il modello nel campo "altro" (es. Oculus Quest 2) *

- Non utilizzo applicazioni VR
- Smartphone + Cardboard
- Altro:

Hai utilizzato/stai utilizzando la realtà virtuale per un corso universitario? *

- Sì
- No

QUESTIONARIO MoSCoW

Per il progetto di tesi avremmo bisogno di valutare quali sono i requisiti essenziali per l'applicazione. Chiediamo il tuo contributo in quanto possibile utilizzatore della nostra applicazione di storyboarding.

La prioritizzazione MoSCoW, nota anche come metodo MoSCoW o analisi MoSCoW, è una tecnica di **prioritizzazione** popolare per la gestione dei requisiti. Permette ai destinatari di un servizio o applicazione di esprimere l'**importanza** che essi attribuiscono al raggiungimento di ciascun **requisito** descritto nel questionario.

La scala di valori è definita da **4 categorie**:

- **MUST - DEVE**: requisiti indispensabili, decisivi per il corretto funzionamento del sistema/applicazione
- **SHOULD - DOVREBBE**: requisiti che sono considerati importanti ma non essenziali
- **COULD - POTREBBE**: requisiti che potrebbero essere utili ma che hanno un impatto meno significativo
- **WOULD - DESIDERATA**: requisiti con minima importanza e minimo valore aggiunto, da riconsiderare solo dopo aver sviluppato tutti gli altri, in un secondo momento

*Per avere una visione generale ti consigliamo di **leggere tutti i requisiti prima e poi votarli**, oppure di **rileggerli finito il questionario per valutare eventuali modifiche** alle priorità già assegnate.*

REQUISITI PER LA CREAZIONE/MODIFICA DELLE SCENE

Inserire oggetti e personaggi nella scena, scegliendoli da una lista, e definirne posizione e orientamento *

- MUST
- SHOULD
- COULD
- WOULD

Rinominare i personaggi e gli oggetti in scena *

- MUST
- SHOULD
- COULD
- WOULD

Eliminare oggetti o personaggi da una scena *

- MUST
- SHOULD
- COULD
- WOULD

Possibilità di poter salvare una scena creata e poterla ricaricare in un secondo momento *

- MUST
- SHOULD
- COULD
- WOULD

REQUISITI PER LA GESTIONE DEI PERSONAGGI

Possibilità di definire le pose dei personaggi manualmente, modificando l'armatura del modello *

- MUST
- SHOULD
- COULD
- WOULD

Possibilità di animare un personaggio applicando al modello delle pose predefinite, selezionabili da una libreria *

- MUST
- SHOULD
- COULD
- WOULD

Possibilità di controllare i personaggi della scena muovendoli nello spazio e animandoli con dei comandi, come all'interno di un videogioco *

- MUST
- SHOULD
- COULD
- WOULD

Possibilità di interazione tra personaggio e personaggio *

- MUST
- SHOULD
- COULD
- WOULD

Possibilità di interazione tra un personaggio e gli altri oggetti presenti nella scena *

- MUST
- SHOULD
- COULD
- WOULD

REQUISITI PER LA GESTIONE DELLE CAMERE VIRTUALI

L'applicazione deve permettere l'aggiunta e il movimento di camere virtuali per inquadrare la scena da diversi punti nello spazio ed eseguire gli screenshot che costituiscono le vignette dello storyboard *

- MUST
- SHOULD
- COULD
- WOULD

Permettere di visualizzare ciò che è inquadrato da una camera virtuale all'interno dell'applicazione con una preview in tempo reale *

- MUST
- SHOULD
- COULD
- WOULD

Per ogni camera deve essere possibile settare i parametri (fov, lunghezza focale, sensor type...)

- MUST
- SHOULD
- COULD
- WOULD

REQUISITI RIGUARDANTI LE LUCI

Possibilità di inserire nella scena delle luci virtuali *

- MUST
- SHOULD
- COULD
- WOULD

Modificare i parametri delle luci (potenza, area, colore) *

- MUST
- SHOULD
- COULD
- WOULD

Possibilità di avere un preset di illuminazione per ambienti esterni *

- MUST
- SHOULD
- COULD
- WOULD

REQUISITI RIGURDANTI LA CREAZIONE DELLO STORYBOARD O DI UNA PRE-VISUALIZZAZIONE

Creare degli screenshot della scena che rappresentano le vignette dello storyboard *

- MUST
- SHOULD
- COULD
- WOULD

Dare la possibilità di definire la durata di un'azione o di un'inquadratura *

- MUST
- SHOULD
- COULD
- WOULD

Modificare lo storyboard finale riordinando le immagini e cambiando le descrizioni *

- MUST
- SHOULD
- COULD
- WOULD

Prevedere uno o più livelli di UNDO per tornare alla condizione dello "screenshot precedente" *

- MUST
- SHOULD
- COULD
- WOULD

Possibilità di definire la durata di un'azione rappresentata da una vignetta dello storyboard *

- MUST
- SHOULD
- COULD
- WOULD

Rendere possibile la visualizzazione dello storyboard prima del completamento *

- MUST
- SHOULD
- COULD
- WOULD

Rendere possibile la visualizzazione dello storyboard una volta completato *

- MUST
- SHOULD
- COULD
- WOULD

Creare una pre-visualizzazione delle azioni svolte nell'applicazione attraverso un video *

- MUST
- SHOULD
- COULD
- WOULD

Avere una descrizione, generata automaticamente, che accompagna ogni inquadratura dello storyboard *

- MUST
- SHOULD
- COULD
- WOULD

Possibilità per l'utente di aggiungere all'interno dell'inquadratura delle indicazioni che specificano i movimenti di camera e dei personaggi (es. Freccie che indichino le direzioni del movimento, zoom in/out ...)

- MUST
- SHOULD
- COULD
- WOULD

REQUISITI RIGUARDANTI IL SISTEMA DI INTERAZIONE E VISUALIZZAZIONE DELL'UTENTE

L'applicazione deve supportare un'interfaccia vocale *

- MUST
- SHOULD
- COULD
- WOULD

L'applicazione deve supportare un'interfaccia gestuale *

- MUST
- SHOULD
- COULD
- WOULD

L'utente deve avere la possibilità di visualizzare e modificare la scena in scale differenti (tabletop, scala 1:1 in prima persona) *

- MUST
- SHOULD
- COULD
- WOULD

REQUISITI RIGUARDANTI LA LIBERTÀ DELL'UTENTE

Il sistema permette massima libertà all'utente di poter scegliere qualsiasi azione tra quelle disponibili da far eseguire ai personaggi, in qualsiasi momento a prescindere dal contesto *

- MUST
- SHOULD
- COULD
- WOULD

Il sistema, in base al contesto, permette di selezionare solo determinate azioni che sono coerenti ad esso *

- MUST
- SHOULD
- COULD
- WOULD

Il sistema deve verificare la coerenza temporale delle azioni tra vignette consecutive *

- MUST
- SHOULD
- COULD
- WOULD

COMMENTI

Ci sono eventuali requisiti che non abbiamo considerato e che secondo te sarebbe necessario aggiungere? Se sì indica nel campo sottostante quali sono e che priorità assegneresti loro secondo il metodo MoSCoW.

La tua risposta

A.2 Questionario SUS e domande generali

Questionario finale - Storyboard Maker VR

Informativa trattamento dati personali

I dati da Lei forniti non saranno comunicati a terzi e verranno resi anonimi subito * dopo la raccolta. I dati verranno archiviati offline e vi avranno accesso i soli responsabili di questa ricerca scientifica. La comunicazione o la diffusione dei risultati delle rilevazioni può avvenire soltanto in forma anonima o aggregata, in modo che i dati elaborati nell'ambito del sondaggio non possano essere associati ad interessati identificati o identificabili. I dati raccolti verranno impiegati al solo scopo di valutare l'applicazione proposta e poterla confrontare ad altri strumenti di storyboarding.

Autorizzo il trattamento dei miei dati personali ai sensi del Decreto Legislativo 30 giugno 2003, n. 196 e del GDPR (Regolamento UE 2016/679).

Consenso informato

I responsabili della ricerca scientifica in oggetto mi hanno informat* circa gli * obiettivi conoscitivi dello studio, l'impatto rispetto ai risultati attesi, le forme e modalità di coinvolgimento dei partecipanti, gli argomenti oggetto dei questionari, le tecnologie usate per lo svolgimento del test ed eventuali rischi sulla salute.

Fornisco il consenso informato

Età *

La tua risposta

Genere *

- Donna
- Uomo
- Preferisco non specificarlo

Corso di studi *

La tua risposta

Quante volte ti è capitato di realizzare degli storyboard? *

- Spesso, li utilizzo abitualmente
- Talvolta
- Quasi mai
- Non ho mai realizzato uno storyboard

Quali strumenti utilizzi solitamente per realizzare uno storyboard? *

- Carta e penna
- Disegno digitale
- Applicazioni 2D (es. Canva, Figma)
- Applicazioni 3D
- Altro:

Conosci la Realtà Virtuale (VR)? *

- Sì
- No

Quanto spesso utilizzi applicazioni VR? *

- Più di una volta al giorno
- Una volta al giorno
- Qualche volta a settimana
- Una volta a settimana
- Una volta al mese
- Meno di una volta al mese
- Mai

Quale dispositivo usi più spesso per applicazioni VR? Se è un dispositivo dedicato, indica il modello nel campo "altro" (es. Meta Quest 2) *

- Non utilizzo applicazioni VR
- Smartphone + Cardboard
- Altro:

Hai utilizzato/stai utilizzando la realtà virtuale per un corso universitario? *

- Sì
- No

Domande

Rispondi alle domande che seguono, in base alla tua esperienza di utilizzo dell'applicazione, in base a quanto sei d'accordo o meno con ciascuna affermazione dando un punteggio da 1 (Molto in disaccordo) a 5 (Molto d'accordo).

Penso che mi piacerebbe usare questo sistema frequentemente. *

- | | | | | | | |
|---------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Molto in disaccordo | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Molto d'accordo |

Ho trovato il sistema inutilmente complesso. *

- | | | | | | | |
|---------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Molto in disaccordo | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Molto d'accordo |

Ho trovato il sistema semplice da usare. *

- | | | | | | | |
|---------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Molto in disaccordo | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Molto d'accordo |

Penso che avrei bisogno del supporto di una persona già in grado di utilizzare il sistema. *

- | | | | | | | |
|---------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------|
| | 1 | 2 | 3 | 4 | 5 | |
| Molto in disaccordo | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | Molto d'accordo |

Ho trovato le varie funzionalità del sistema ben integrate. *

	1	2	3	4	5	
Molto in disaccordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molto d'accordo

Ho trovato incoerenze tra le varie funzionalità del sistema. *

	1	2	3	4	5	
Molto in disaccordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molto d'accordo

Penso che la maggior parte delle persone potrebbe imparare ad utilizzare *
il sistema facilmente.

	1	2	3	4	5	
Molto in disaccordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molto d'accordo

Ho trovato il sistema molto macchinoso da utilizzare. *

	1	2	3	4	5	
Molto in disaccordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molto d'accordo

Ho avuto molta confidenza con il sistema durante l'uso. *

	1	2	3	4	5	
Molto in disaccordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molto d'accordo

Ho avuto bisogno di imparare molti processi prima di riuscire ad utilizzare al *
meglio il sistema.

	1	2	3	4	5	
Molto in disaccordo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Molto d'accordo

Domande aperte

Quali funzionalità dell'applicazione ti sono piaciute maggiormente? *

La tua risposta

Cosa ti è piaciuto di meno o hai avuto difficoltà a fare? *

La tua risposta

Quali funzionalità aggiungeresti? *

La tua risposta

Che cosa cambieresti dell'applicazione? *

La tua risposta

Bibliografia e sitografia

- [1] Okun J. A., Zwerman S., *The VES Handbook of Visual Effects. Industry Standard VFX Practices and Procedures.*, 3 ed., Routledge, 2020, cap 3, pag 140.
- [2] Swift T., “Taylor Swift - willow (dancing witch version) - storyboards from the willow music video”, YouTube, 2020
<https://youtu.be/3CUM7Z3TsmY?feature=shared>
- [3] “Storyboard”, Wikipedia <https://it.wikipedia.org/wiki/Storyboard>
- [4] Milgram P., Takemura H., Utsumi A., Kishino F., “Augmented reality: A class of displays on the reality-virtuality continuum” in *Telemanipulator and Telepresence Technologies*, v. 2351, 1994.
- [5] “Reality–virtuality continuum”, Wikipedia
https://en.wikipedia.org/wiki/Reality%E2%80%93virtuality_continuum
- [6] Burdea G., Coiffet P., *Virtual Reality Technology*, Wiley, New Jersey, 2003.
- [7] Barnard D., “Degrees of Freedom (DoF): 3-DoF vs 6-DoF for VR Headset Selection”, VirtualSpeech <https://virtualspeech.com/blog/degrees-of-freedom-vr>
- [8] Chang C., Bang K., Wetzstein G., Lee B., Gao L., “Toward the next-generation VR/AR optics: a review of holographic near-eye displays from a human-centric perspective,” in *Optica*, v. 7, n. 11, 1563-1578, 2020.
- [9] Zhao Q., “A survey on virtual reality.” in *Sci. China Ser. F-Inf. Sci.* 52, 348–400, 2009.
- [10] Basso A. “Advantages, Critics and Paradoxes of Virtual Reality Applied to Digital Systems of Architectural Prefiguration, the Phenomenon of Virtual Migration.” in *Proceedings*, 1(10):915, 2017.
- [11] Basu A., “A brief chronology of Virtual Reality”, Emory University, Atlanta, Georgia, United States, 2019.

- [12] “*Virtual reality headset and wired gloves, Ames Research Center*”, Wikipedia Commons
https://commons.wikimedia.org/wiki/File:Virtual_reality_headset_and_wired_gloves,_Ames_Research_Center.jpg
- [13] “*Aspen Movie Map*”, Wikipedia
https://it.wikipedia.org/wiki/Aspen_Movie_Map
- [14] “*About The People Behind LeapVR*”, LEEPVR
<http://www.leepvr.com/about.php>
- [15] “*Power Glove*”, Wikipedia https://en.wikipedia.org/wiki/Power_Glove
- [16] “*Facebook to Acquire Oculus*”, Meta, 2014
<https://about.fb.com/news/2014/03/facebook-to-acquire-oculus/>
- [17] Ottolina P., “*Quando la realtà virtuale ti lascia senza fiato: la nostra prova di Htc Vive*”, Corriere della Sera, 2015.
<https://www.corriere.it/tecnologia/mobile-world-congress-2015/notizie/htc-vive-prova-recensione-realta-virtuale-61d180b8-c317-11e4-9a3c-d1424c2aada1.shtml>
- [18] “*Oculus VR*”, Wikipedia https://it.wikipedia.org/wiki/Oculus_VR
- [19] “*HTC Vive*”, Wikipedia https://it.wikipedia.org/wiki/HTC_Vive
- [20] “*HTC Vive*”, VIVE <https://www.vive.com/eu/product/vive/>
- [21] “*PlayStation VR*”, Wikipedia https://it.wikipedia.org/wiki/PlayStation_VR
- [22] “*PlayStation VR Specifiche tecniche*”, PlayStation
<https://www.playstation.com/it-it/ps-vr/tech-specs/>
- [23] “*L’evoluzione di Oculus Visore VR*”, airlapp
<https://airlapp.com/blog/evoluzione-di-oculus-visore-vr/>
- [24] “*Confronta il tuo Meta Quest*”, Meta <https://www.meta.com/it/quest/compare/>
- [25] “*Meta Quest 3*”, Meta <https://www.meta.com/it/quest/quest-3/>
- [26] “*HTC VIVE Pro*”, VIVE <https://www.vive.com/eu/product/vive-pro/>
- [27] “*HTC VIVE Pro Eye*”, VIVE
<https://www.vive.com/sea/product/vive-pro-eye/overview/>
- [28] “*HTC VIVE*”, Wikipedia https://en.wikipedia.org/wiki/HTC_Vive

- [29] Henderson R., “*Le cuffie HTC Vive Cosmos Play, Cosmos Elite e Cosmos XR si aggiungono alla gamma VR, oltre ai frontalini da aggiornare*”, Pocket-lint, 2020 <https://www.pocket-lint.com/it-it/ar-vr/notizie/htc/151136-htc-vive-cosmos-play-cosmos-elite-e-cosmos-xr-headset-aggiunti-alla-gamma-vr-piu-frontalini-da-aggiornare/>
- [30] “*VIVE Pro 2*”, VIVE <https://www.vive.com/eu/product/vive-pro2-full-kit/specs/>
- [31] “*VIVE Flow*”, VIVE <https://www.vive.com/us/product/vive-flow/overview/>
- [32] “*VIVE XR Elite*”, VIVE <https://www.vive.com/eu/product/vive-xr-elite/overview/>
- [33] “*PlayStation VR2*”, PlayStation <https://www.playstation.com/it-it/ps-vr2/ps-vr2-tech-specs/>
- [34] Francesco, “*Apple Vision Pro ufficiale alla WWDC 2023*”, HDBLOG, 2023 <https://www.hdblog.it/indossabili/articoli/n570505/apple-vision-pro-visore-prezzi-specifiche/>
- [35] Muender T., Fröhlich T., Malaka R., “*Empowering creative people: Virtual reality for previsualization.*” in *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, p. 1-6
- [36] Galvane Q., Lin I. S., Argelaguet F., Li T. Y., Christie M., “*Vr as a Content Creation Tool for Movie Previsualisation*” in *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, Osaka, France, 2019. p. 303-311.
- [37] Ardal D., Alexandersson S., Lempert M., Abelho Pereira A. T., “*A Collaborative Previsualization Tool for Filmmaking in Virtual Reality*” in *Proceedings of the 16th ACM SIGGRAPH European Conference on Visual Media Production*. 2019. p. 1-10.
- [38] Liu Z., Jin Y., Ma M., Li J., “*A Comparison of Immersive and Non-Immersive VR for the Education of Filmmaking*” *International Journal of Human-Computer Interaction*, 2022, 39.12: 2478-2491.
- [39] “*Storyboards: become a full-fledged director of your scenes and prototypes. Communicate your ideas and prototype in a storyboard format*”, Tвори, 2021 <https://tvori.co/storyboard-tvori-create-and-animate-scenes-in-vr>

- [40] “*What is Tvorì*”, Learn Tvorì <https://learn.tvorì.co/>
- [41] ShapesXR, “*Storyboards in VR*”, YouTube, 2021
<https://www.youtube.com/watch?v=pL4a-mCmaj0&t=54s>
- [42] “*Storyboarder*”, wonder unit <https://wonderunit.com/storyboarder/>
- [43] Polycom <https://poly.cam/>
- [44] Sakurai S., “*How to STORYBOARD in VR*”, YouTube, 2022
<https://www.youtube.com/watch?v=TAJlb-orh2A>
- [45] Gravity Sketch <https://www.gravitysketch.com/>
- [46] Antunes J., “*Gravity Sketch: set design and storyboarding in VR*”, PROVIDEOCOALITION, 2021 <https://www.provideocoalition.com/gravity-sketch-set-design-and-storyboarding-in-vr/>
- [47] Augmented Actor, “*Set Design Using Gravity Sketch VR - How to Storyboard and Concept Your Set*”, YouTube, 2018
<https://www.youtube.com/watch?v=4Y1SOjjNh9M>
- [48] “*MoSCoW method*”, Wikipedia
https://en.wikipedia.org/wiki/MoSCoW_method
- [49] Scarzello M., Sanna A., Manuri F., “*Advanced Storyboard: Generazione automatica di storyboard mediante il controllo diretto dei personaggi*”, Tesi Corso di Laurea Magistrale in Ingegneria Del Cinema E Dei Mezzi Di Comunicazione, 2022
- [50] “*Oculus Rift CV1*”, Wikipedia https://en.wikipedia.org/wiki/Oculus_Rift_CV1
- [51] “*Oculus Touch*”, Wikipedia https://en.wikipedia.org/wiki/Oculus_Touch
- [52] Chen S., Miranda F., Ferreira N., Lage M., Doraiswamy H., Brenner C., Defanti C., Koutsoubis M., Wilson L., Perlin K., Silva C., “*UrbanRama: Navigating Cities in Virtual Reality*” in *IEEE Transactions on Visualization and Computer Graphics*, 2021, p. 1-1.
- [53] “*Unity (game engine)*”, Wikipedia
[https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine))
- [54] Unity Asset Store <https://assetstore.unity.com/>
- [55] “*Microsoft Visual Studio*”, Wikipedia
https://it.wikipedia.org/wiki/Microsoft_Visual_Studio
- [56] “*C#*”, Wikipedia https://it.wikipedia.org/wiki/C_sharp

- [57] “*Coding in C# in Unity for beginners*”, Unity <https://unity.com/how-to/learning-c-sharp-unity-beginners>
- [58] “*JavaScript*”, Wikipedia <https://it.wikipedia.org/wiki/JavaScript>
- [59] “*CSS*”, Wikipedia <https://it.wikipedia.org/wiki/CSS>
- [60] “*Introducing JSON*”, JSON <https://www.json.org/json-en.html>
- [61] “*Mixed Reality Toolkit*”, Wikipedia https://en.wikipedia.org/wiki/Mixed_Reality_Toolkit
- [62] “*What is the Mixed Reality Toolkit*”, github <https://github.com/microsoft/MixedRealityToolkit-Unity#what-is-the-mixed-reality-toolkit>
- [63] “*Oculus Integration SDK, Import Entire SDK Package*”, Meta <https://developer.oculus.com/documentation/unity/unity-import/>
- [64] “*Physically based rendering*”, Wikipedia https://en.wikipedia.org/wiki/Physically_based_rendering
- [65] “*Blender*”, Wikipedia [https://en.wikipedia.org/wiki/Blender_\(software\)](https://en.wikipedia.org/wiki/Blender_(software))
- [66] re1mosen, “*Cinema Camera, modello 3D*”, Sketchfab <https://sketchfab.com/3d-models/cinema-camera-e1fec87c8ae0487f964d20de2559aa62>
- [67] Team Black Cat, “*Luce rotonda del film, modello 3D*”, TURBOSQUID <https://www.turbosquid.com/it/3d-models/movie-light-3d-1495147>
- [68] HuNtEr_3DdD, “*Vecchia panchina PBR a basso numero di poligoni, modello 3D*”, TURBOSQUID <https://www.turbosquid.com/it/3d-models/3d-model-old-park-bench-lowpoly-pbr-2064824>
- [69] garleth93, “*Piano Bench, modello 3D*”, Free3D <https://free3d.com/3d-model/piano-bench-762312.html>
- [70] Infima Games, “*Animated Loading Icons*”, Unity Asset Store <https://assetstore.unity.com/packages/2d/gui/icons/animated-loading-icons-47844>
- [71] “*Mixamo*”, Wikipedia <https://en.wikipedia.org/wiki/Mixamo>
- [72] Mixamo <https://www.mixamo.com/#/>
- [73] “*TextMeshPro*”, Unity <https://docs.unity3d.com/Manual/com.unity.textmeshpro.html>

- [74] “OpenAI” <https://openai.com/>
- [75] RageAgainstThePixel, “*com.openai.unity 5.0.11*”, GitHub
<https://github.com/RageAgainstThePixel/com.openai.unity>
- [76] “10 Usability Heuristics for User Interface Design”, Nielsen Norman Group
<https://www.nngroup.com/articles/ten-usability-heuristics/>
- [77] Brooke J., “*SUS: A quick and dirty usability scale.*” in *Usability Eval. Ind.*, 189, 1995.
- [78] Bangor A., Kortum P.T., Miller J.T., “*Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale*” in *Journal of Usability Studies*, 4(3), 114-123, 2009.
- [79] “*NASA TLX: TASK LOAD INDEX*”, NASA
<https://humansystems.arc.nasa.gov/groups/tlx/index.php>