

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea

Progettazione e sviluppo di un'Applicazione per storyboarding in realtà estesa ed integrazione di funzionalità di Scene Understanding per interazioni con oggetti reali



Relatori

prof. Federico Manuri

prof. Andrea Sanna

firma dei relatori

.....
.....

Candidato

Federico Mafri

firma del candidato

.....

Anno Accademico 2024/2025

Sommario

Lo storyboard rappresenta uno strumento indispensabile nella fase di pre-produzione cinematografica, offrendo la possibilità di pre-visualizzare la storia definita nello script tramite una sequenza di inquadrature. L'applicativo proposto è stato sviluppato per il dispositivo Meta Quest 3 e consente con successo di realizzare e gestire tutti i passaggi necessari per completare uno storyboard. In particolare, il sistema permette di virtualizzare l'ambiente, stabilire la posa dei personaggi e infine realizzare le vignette che andranno a comporre lo storyboard.

L'utilizzo degli storyboard affonda le sue radici nei primi anni del 1900, con l'illusionista e regista Georges Méliès che li utilizzava per impostare in modo preliminare le scene chiave del prodotto cinematografico, affermandosi fino ad oggi come uno dei più importanti strumenti per la pre visualizzazione di un film. La sua realizzazione costituisce un processo impegnativo in termini di tempo e risorse economiche, poiché richiede competenze specifiche di natura sia artistica che tecnica. La creazione di uno storyboard richiede infatti concezioni cinematografiche per posizionare la macchina da presa e artistiche per avere l'abilità sufficiente a rappresentare le varie vignette. Oltre alle competenze richieste, è necessario un tempo minimo per completare il prodotto, poiché molto spesso il regista si interfaccia con un artista per realizzarlo. Sono quindi necessarie diverse revisioni per avvicinarsi alla visione del regista. A partire dall'avvento del digitale, sono stati sviluppati degli applicativi in grado di abbattere questa barriera di competenze specifiche richieste, così da rendere lo storyboarding accessibile a chiunque. Questo elaborato tratta della progettazione e dello sviluppo di un'applicazione di storyboarding in realtà estesa che permetta all'utente di usare l'ambiente fisico come location per gli storyboard. Tramite un approccio basato su modelli di machine learning per la segmentazione degli oggetti a partire da immagini, l'ambiente reale viene virtualizzato permettendo l'utilizzo di oggetti reali in animazioni digitali. Il progetto è stato realizzato utilizzando la piattaforma di sviluppo "Unity" nella sua ultima versione, la 6.0.0, e le sue librerie più recenti: XR Interaction Toolkit e AR Foundation. XR Interaction Toolkit è un framework operativo per realizzare applicazioni in realtà virtuale/mista compatibili con una vasta gamma di dispositivi. AR Foundation, invece, è una libreria specializzata nello sviluppo di applicazioni in realtà aumentata. Quest'ultima si è rivelata fondamentale nel progetto per implementare correttamente tutte le funzionalità di riconoscimento degli oggetti e dello spazio circostante. Per installare l'applicazione all'interno del visore, è stato utilizzato il software sviluppato da Meta, il Meta Quest Developer Hub. Questo strumento consente di installare la build prodotta con Unity.

L'applicazione si suddivide concettualmente in quattro fasi principali: La prima fase riguarda la configurazione dell'ambiente. L'utente si trova immerso in una stanza sintetica che emula quella reale in cui si trova. In questa stanza vengono inseriti tutti gli oggetti riconosciuti, ciascuno corredato un'etichetta. Successivamente, si verificano la correttezza delle etichette e il posizionamento degli oggetti riconosciuti, che possono essere eventualmente modificati. Si procede poi a posizionare gli oggetti digitali necessari alla rappresentazione dello script, tra cui personaggi protagonisti, o utilizzare quelli reali riconosciuti. Infine, una volta posizionati i personaggi, si avvia la fase di storyboarding permettendo a quest'ultimi di compiere azioni ed interagire con gli oggetti inseriti nella scena. In questa fase è possibile sfruttare una telecamera virtuale sincronizzata con la vista dell'utente per integrare la componente reale con quella virtuale, generando così le tavole dello storyboard. Inoltre, se si desidera realizzare lo storyboard in un ambiente completamente virtuale, è possibile sostituire gli oggetti reali segmentati nella scena con le loro versioni digitali. L'utente viene quindi reinserito nella stanza virtuale iniziale, che replica quella reale, permettendo di lavorare senza essere disturbati da oggetti esterni. Durante tutta la durata della simulazione, l'utente è accompagnato da un tutorial che lo guida in modo dettagliato. Il tutorial è consultabile in qualsiasi momento per risolvere dubbi e facilita l'utilizzo e l'apprendimento dell'applicazione.

Per valutare l'efficacia e l'utilità dell'applicazione sviluppata, sono stati realizzati due casi di studio basati su script selezionati da una raccolta di opere televisive e cinematografiche di grande successo. Gli script scelti appartengono al celebre film d'animazione *Rapunzel - L'intreccio della torre* e alla serie *Mercoledì*, prodotta da Netflix sotto la direzione artistica di Tim Burton. Da ciascuno script è stato estrapolato un passaggio significativo, selezionato con l'obiettivo di verificare l'efficienza e le funzionalità dell'applicazione nella creazione di storyboard.

Una volta definiti gli script, sono stati inseriti i personaggi e gli oggetti necessari per realizzare uno storyboard accurato. Inoltre, sono state programmate tutte le azioni e animazioni che i personaggi dovevano eseguire per rappresentare le dinamiche previste dallo script selezionato. Questo approccio ha consentito di testare non solo l'affidabilità tecnica dell'applicazione, ma anche la sua capacità di gestire e rappresentare scene complesse, riproducendo fedelmente le interazioni previste.

Al termine delle simulazioni, è stato richiesto agli utenti di compilare i questionari SUS e NASA-TLX, strumenti standard per valutare rispettivamente l'usabilità e il carico cognitivo associati all'utilizzo dell'applicazione. Questa analisi ha fornito dati preziosi sull'accessibilità dell'applicazione e sull'impatto percepito dagli utenti durante il suo utilizzo, contribuendo a identificare eventuali miglioramenti futuri.

Il presente progetto di tesi si è focalizzato sull'aggiunta di funzionalità avanzate per il riconoscimento degli oggetti e l'identificazione dello spazio in realtà aumentata, con l'obiettivo di migliorare l'esperienza di storyboarding. Tale approccio ha aperto diverse possibilità di sviluppo, molte delle quali possono essere ulteriormente ampliate ed esplorate al fine di incrementare la velocità e l'immersione che l'applicazione fornisce. Ad esempio, si potrebbero sfruttare i risultati ottenuti dalle ricerche del progetto ARIA, una piattaforma di ricerca finanziata e supportata da Meta, dedicata allo sviluppo di algoritmi egocentrici basati sull'intelligenza artificiale per migliorare ed approfondire le funzionalità di scene understanding presenti nell'applicativo. Si potrebbe altresì optare per un miglioramento

delle interfacce utente e le modalità di interazione con l'obiettivo di andare a rendere più accessibile l'applicazione a utenti inesperti. Infine, si potrebbe aggiungere un tool di produzione di animatic basandosi sul risultato dello storyboard.

Ringraziamenti

Credo che nella vita si possano osservare le cose con pessimismo o con gratitudine. Personalmente, ho sempre preferito cogliere ciò che di bello mi circonda per apprezzarne l'esistenza. Per questo vorrei dedicare questo spazio a ringraziare tutti coloro che, con il loro modo unico di essere, hanno reso questo mio viaggio un'esperienza straordinaria e indimenticabile.

Grazie innanzitutto, ai miei relatori Manuri Federico e Sanna Andrea, per l'incessante supporto e pazienza dimostrata in questi mesi di tesi, senza di voi non credo sarei mai riuscito a raggiungere i risultati a cui sono arrivato. Lavorare con voi è stato un piacere di cui sono molto grato.

A mio padre e mia madre, per il vostro costante e continuo supporto incondizionato. Grazie per i miei pasti preferiti al ritorno, per tutti "hai bisogno di questo quando risali o di quest'altro?", di tutte le volte che siete venuti su a trovarci se vedevate che le cose non andavano bene e di tutte quelle in cui ci riportavate a casa alla fine di una sessione o per le festività. Potrei elencare tutte le cose per cui ringraziarvi, ma forse non riuscirei mai ad esaurirle. Nonostante molte volte non capiate le strade che ho scelto di intraprendere, so che ci sarete sempre per me a prescindere da tutto. A voi va il mio grazie più sincero e profondo per essere sempre stati i genitori di cui avevo bisogno, quelli amorevoli ma anche quelli severi quando era necessario, ma soprattutto grazie per avermi amato come mai nessuno probabilmente farà mai.

A mio fratello, grazie per l'aiuto in questi anni, per aver pulito casa quando io ero in crisi, per i consigli nel primo periodo a Torino e per tutti i bei momenti che hanno arricchito questi anni insieme. Presto, purtroppo, il nostro periodo di convivenza terminerà e questo, nonostante le mie mille battute a riguardo, mi rende molto triste. Non sempre ci capiamo l'un l'altro e non sempre siamo d'accordo, ma su una cosa sono certo, ovvero che il nostro affetto ha radici profonde.

A mia nonna Rita, con le tue continue preoccupazioni e domande piene d'ansia riesci sempre a strapparmi un sorriso, grazie per preoccuparti tu per tutti e due, grazie per tutti i momenti passati insieme fin dalla mia infanzia, per tutte le partite a memory, rubamazzo e briscola in cui vincevo sempre. Non importa quanto tempo passerà, ti vorrò sempre un enorme bene.

A mio zio Sandro, grazie per essere un punto di riferimento nella mia vita. Grazie per avermi fatto scoprire la montagna e la libertà dello sci, per il prosciutto e le salsicce sempre pronti al mio ritorno, ma soprattutto per avermi insegnato a godermi le piccole cose della vita: una chiacchierata tra di noi, una giornata in montagna o una passeggiata

insieme possono sembrare piccole cose, ma sono invece ricordi che porterò con me per tutta la vita.

A mia nonna Graziella, grazie per tutte le chiamate e le chiacchierate infinite a casa tua, grazie per il tuo interessamento costante nella mia carriera e nella mia vita. Nonostante la distanza e la recente poca possibilità di vederci, la tua presenza all'interno della mia vita non è mai diminuita e di questo te ne sono grato.

A Giulia, grazie per tutto l'aiuto ed il supporto che mi hai dato in questi anni. A te va la mia sincera gratitudine per essere stata il mio porto sicuro in questo mare aperto che è la vita. Per tutti i messaggi e le mail che mi hai ricontrollato, per tutte le volte che mi hai aiutato con la scelta degli outfit, per tutte le volte che mi facevo prendere dal panico e mi hai aiutato a ritrovare la tranquillità. Tu, sei sempre stata e, spero, sempre sarai la compagna con cui affrontare le sfide e le gioie che la vita mi riserverà.

Ad Alberto, compagno di mille avventure, di strada dal primo liceo ne abbiamo fatta, di tempo ne è passato da quando fantasticavamo sul gestire un allevamento di struzzi o sull'aprire una pizzeria in Giappone (piano ancora possibile tra l'altro). Oggi viviamo in posti diversi, facciamo cose completamente opposte ed abbiamo orari e routine agli antipodi. Nonostante tutto questo, non c'è mai stata una singola volta in cui non potessi contare su di te. Per una chiacchierata, una sessione di DnD, una piccola seduta di gaming di 6 ore almeno, tu sei sempre stato lì per aiutarmi nei momenti difficili o per ridere in quelli felici. Nonostante tutti gli impegni, potrai sempre contare su di me, ti ringrazio per esserci.

Ai miei amici di "Giu", Daniele, Mirella, Paolo, Elisa, Marco, Aurora, Simone, Rachele, Cristina, Hooch, Diddi, grazie per essere stati il mio punto fisso ad ogni mio ritorno, grazie perché in una città, ed una vita, sempre di fretta voi siete sempre stati il mio respiro di sollievo. Alla fine di una sessione impegnativa, di un periodo stressante o prima di grandi sfide, lo svago ed il divertimento che ogni volta generiamo insieme mi ha permesso di affrontare tutto questo percorso con leggerezza. Spero che nel futuro, a prescindere da dove andrò e cosa farò, non smetteremo mai di divertirci in questo modo.

Ad Alberto, Alice, Tina e Mich, perché gli amici di "Giu" non sono solo quelli sopra, a voi va il mio grazie per le serate giochi da tavolo ma anche quelle a sapere e potere o anche quelle passate a giocare a beach volley, le grigliate estive, i palloni a fiume, le serate al McDonald's. Con voi ho scoperto la passione per la pallavolo che mi accompagna tutt'ora. La mia gratitudine va a voi anche per essere sempre rimasti nella mia vita nonostante la lontananza e per aver arricchito questi anni con tanti fantastici ricordi.

A Kevin, il mio compagno di viaggio, negli anni la vita ci ha avvicinato e allontanato a più riprese, ma mai una volta ci ha diviso. A te va il mio grazie più profondo per essere stato quell'amico che non ti abbandona mai. Grazie per avermi spronato a fare meglio quando potevo e per avermi insegnato a non mollare mai. Tu, che testardo come il marmo riesci in ogni impresa, sei stato per me fonte di grande ispirazione a non arrendermi mai alle difficoltà di ogni giorno. Grazie anche e soprattutto per tutti i momenti felici che abbiamo condiviso: dai tuoi compleanni, alle giornate con "Macchietta" al parco, ai pomeriggi di studio. Questi ricordi saranno sempre con me ovunque andrò.

Ai miei amici di Torino, Pietro, Davide, Noemi, Dario, Sante, Salvo Tilocca e, perché no, anche Salvo Tartaglione, forse non ve l'ho mai detto, ma di ritorno da Torino dopo il covid avevo perso i contatti con quasi tutti i miei amici storici. Ecco, riguardo a questo,

grazie per avere reso questa fredda città in cui tornavo un caldo ed accogliente ritrovo, grazie per tutte le dispense, appunti ed audio spiegazioni per ogni corso, grazie per ogni serata di giochi da tavolo senza la quale non avrei mai superato i periodi più stressanti che il Poli ci ha propinato. Non so ancora se rimarrò a Torino, so però che ho avuto la fortuna di trovare dei colleghi che ora sono orgoglioso di chiamare amici.

A Pietro, mio fratello fiorentino, deh, grazie per la C aspirata, per tutte le tue squallide freddure e per i momenti insieme a Firenze. Sei stato il compagno che mi ha affiancato in ogni corso della mia magistrale, un collega eccezionale, un compagno di progetto intraprendente ma anche, e forse la qualità migliore, un amico straordinario. Purtroppo, anche se in questi ultimi mesi sei tornato in quel paesino che è Firenze, il mio affetto per te non è mai diminuito. Sono sempre felice di incontrarti quando vengo e sono davvero grato per tutto il tempo passato insieme e per quello che avremo ancora modo di passare.

A Fabio, l'uomo dalle mille risposte. Abbiamo affrontato moltissimi corsi insieme e tu hai alleggerito ogni materia noiosa e laboratori inutili con la tua calma e la tua gioia. Grazie per non avermi mai mandato a fanculo nonostante ti abbia scritto, anche – le ho contate – 15 volte in un solo giorno per farmi spiegare concetti mai compresi a fondo.

A Gioele, grazie per tutto il supporto che mi hai dato in quest'ultimo anno da quando ti ho conosciuto. Hai reso quest'ultima, caldissima estate, molto più leggera grazie alla settimana di vacanza ad Alassio. Sei stato all'inizio un collega onesto e trasparente ed ora un amico prezioso. Infine, grazie per aver ascoltato tutte le mie paure in questo ultimo periodo della tesi. Sei stato per me fonte di grande ispirazione e gioia.

A Fausto, il mio insegnante di factorio personale, grazie per tutte le ore di divertimento degli ultimi mesi, senza la quale probabilmente sarei già stato rinchiuso in manicomi. Sei un amico davvero prezioso per me.

Ad Andreino, da tanti anni a questa parte un amico sincero e vero, grazie per le chiacchierate su discord per le sessioni di gioco a giochi che nemmeno ti piacevano solo per poter parlare. I momenti per vederci sono sempre pochissimi ma ogni volta lo stare insieme è per me una gioia infinita.

A Mauro, grazie per avermi insegnato a rallentare, ad accontentarmi e a gioire delle cose di ogni giorno. Senza questi insegnamenti probabilmente non sarei riuscito a completare questo percorso con la positività e la gioia che mi ha sempre accompagnato.

Infine, anche se forse è un po' cringe, grazie a me stesso, perché nonostante tantissime difficoltà ho avuto la fede di non mollare mai la presa e di continuare a navigare. Giunto alla fine di questo viaggio che è stata la mia laurea, sono orgoglioso di poter dire che, seppur l'inizio non sia stato dei migliori, ho avuto la forza di abbandonarmi al fiume della vita e di scoprire in che posto meraviglioso ciò mi avrebbe portato. Non avrei mai pensato che avrei avuto la fortuna di incontrare così tante persone meravigliose e non vedo l'ora di scoprire in quali altri posti arriverò.

Indice

Elenco delle tabelle	12
Elenco delle figure	13
1 Introduzione	15
1.1 Obiettivo	15
1.2 Preproduzione di un film	16
1.3 Storyboard	17
1.3.1 Cenni storici	17
1.3.2 Caratteristiche ed elementi di uno storyboard	19
1.4 La realtà aumentata	20
1.4.1 Rapporto tra realtà aumentata e virtuale:il reality-virtuality continuum	20
1.4.2 Dispositivi di realtà aumentata	21
1.4.3 Ambiti di applicazione	23
1.5 Reti neurali	23
1.5.1 Ambiti di utilizzo delle reti neurali	24
1.5.2 Funzionalità delle reti	24
1.5.3 Limitazioni dei modelli attualmente disponibili	25
2 Stato dell'arte	27
2.1 2.1 Software di storyboarding 2D attualmente disponibili	27
2.2 Generazione di storyboard	28
2.3 Storyboarding in VR e AR	28
2.4 2.2.2 Augmented reality storyTelling: story creatAR	30
2.5 Animazione in realtà aumentata	31
2.5.1 Authoring animations of virtual objects in augmented reality-based 3D space	31
2.6 AR animator	31
2.7 Previsualizzazione in realtà mista	32
2.8 Analisi dei requisiti applicativi	34

3	Tecnologie	35
3.1	Unity	35
3.1.1	Interfaccia di unity	36
3.2	Unity XR interaction toolkit	37
3.3	Rider	37
3.4	Github	38
3.5	Meta Quest	38
3.6	Specifiche	39
3.7	Funzionalità	39
3.8	Meta Quest Developer Hub	40
3.9	Oculus Link	40
3.10	Pacchetti aggiuntivi e librerie	41
4	Implementazione	43
4.1	Comportamento generale della simulazione	43
4.2	Manipolazione degli oggetti	44
4.3	configurazione dell'utente	47
4.4	Gestione delle interazioni all'interno della simulazione	47
4.5	Utilizzo delle funzionalità di ancoraggio	49
4.6	Gestione del piano di seduta e piazzamento oggetti	49
4.7	Animazione dei personaggi	50
4.8	Scansione dell'ambiente automatica effettuata all'avvio dell'applicazione e configurazione delle bounding box	51
4.9	Simulation manager	53
4.10	Character animation manager	54
4.11	Descrizione testuale delle azioni	55
4.12	Funzionalità di cattura foto	55
4.13	Generazione storyboarding	56
4.14	Interfaccia utente	56
4.15	Funzioni di accessibilità	59
5	Test	61
5.1	Casi di studio	61
5.2	Struttura dei test	63
5.3	Tipologia dei questionari	64
5.3.1	NASA-TLX	64
5.3.2	SUS	64
6	Sviluppi Futuri	67
6.1	progetto ARIA e posa dei personaggi	67
6.2	Miglioramento delle interfacce per la manipolazione degli oggetti	69
6.3	Animatic	69

Elenco delle tabelle

Elenco delle figure

1.1	Esempio di storyboard del famoso corto Plane's crazy	18
1.2	grafico di transizione dalla realtà alla realtà virtuale	21
1.3	Sistema a realtà aumentata basato su un dispositivo mobile	22
1.4	Sistema realtà aumentata basato su proiettore	22
2.1	Applicativo StoryboardThat	27
2.2	Struttura della fase di Storyboarding	29
2.3	Creazione di uno storyboard mediante aggiunta di carte	29
2.4	Funzionamento generale di StoryCreatAR	31
2.5	Immagine panoramica dello storyboard	32
2.6	Funzionamento di AR animator	33
2.7	Pipeline di lavoro di MrPreViz	33
3.1	Viewport di Unity	36
3.2	dispositivo Meta Quest	38
3.3	Interfaccia MetaQuest Developer Hub	40
4.1	Campi del Componente SnapToPlane	44
4.2	Gerarchia del componente xr origin	47
4.3	Albero delle animazioni di un personaggio	50
4.4	Esempio di Bounding box generica	52
4.5	Bounding box con piano di seduta/interazione abilitato	53
4.6	Pannello di preview del frame dello storyboard	55
4.7	menu istanziazione dei personaggi	56
4.8	Menu consultazione dello script	57
4.9	Pannello di preview del frame dello storyboard	58
4.10	Pannello collezione degli screenshot	58
4.11	Pannello di preview del frame dello storyboard	59
5.1	Estratto dello script del film Rapunzel	62
5.2	Estratto dello script del film Rapunzel	63
6.1	Occhiali virtuali progetto ARIA	68

*If you cannot understand my
argument, and declare
it's Greek to me
you are quoting Shakespeare.*

[B. LEVIN, Quoting Shakespeare]

Capitolo 1

Introduzione

Il presente elaborato ha l'obiettivo di illustrare il funzionamento, le potenzialità e i risultati ottenuti da un applicativo sviluppato per la creazione di uno storyboard in realtà aumentata a grandezza naturale.

Il primo capitolo mira a fornire al lettore le nozioni essenziali per comprendere il dominio di interesse dell'applicativo, evidenziando le principali esigenze nel contesto della pre-produzione cinematografica. In particolare, si descriveranno le motivazioni e le scelte progettuali che hanno guidato lo sviluppo di un applicativo capace di supportare concretamente le attività preliminari alla produzione cinematografica.

Nei capitoli successivi, verranno presentati lo stato dell'arte relativo alla produzione di storyboard, le tecnologie utilizzate per lo sviluppo dell'applicativo, e una descrizione generale dell'implementazione, al fine di delineare il funzionamento del progetto. A conclusione dell'elaborato, si proporranno i risultati ottenuti attraverso test condotti su un caso d'uso preparato ad hoc e una discussione delle possibili evoluzioni future del progetto, con particolare attenzione ai potenziali sviluppi che potranno contribuire all'ulteriore miglioramento dell'applicativo.

1.1 Obiettivo

Lo storyboard rappresenta uno strumento fondamentale nella fase di pre-produzione e pre-visualizzazione cinematografica. Consiste in una sequenza di disegni, scena per scena, tratti dalla sceneggiatura, con l'obiettivo di fornire una visione dettagliata e predefinita delle inquadrature del film. Questi disegni concettuali arricchiscono la narrazione e agevolano l'organizzazione delle complesse azioni richieste dalla sceneggiatura, consentendo al team di produzione di pianificare accuratamente l'aspetto visivo del film prima delle riprese [5].

La realizzazione di uno storyboard è un processo spesso essenziale ma dispendioso, sia in termini di tempo sia di risorse economiche, poiché richiede competenze specifiche nel campo artistico e tecnico. La possibilità di disporre di un software in grado di generare uno storyboard in maniera rapida e a basso costo, riducendo o eliminando la necessità

di competenze avanzate, rappresenta un'opportunità significativa per l'ambito della produzione cinematografica.

Il presente progetto di tesi si propone di aggiornare un'applicazione di storyboarding già esistente, integrando nuove funzionalità e tecnologie avanzate di riconoscimento degli oggetti e di "scene understanding". Tali miglioramenti mirano a rendere l'esperienza di storyboarding più immersiva e interattiva, permettendo una verifica immediata della validità dello storyboard mediante l'utilizzo della realtà aumentata.

Per raggiungere questo obiettivo, si è scelto di utilizzare le funzionalità offerte dal visore Meta Quest 3 e dalla piattaforma di sviluppo "Unity", un ambiente per la creazione di applicazioni 2D e 3D in tempo reale, particolarmente adatto per lo sviluppo di applicativi in realtà aumentata.

L'applicazione consente di posizionare oggetti digitali e di riconoscere un set specifico di oggetti fisici presenti nell'ambiente reale, con i quali è possibile interagire. Al completamento della scena, l'utente può catturare uno screenshot del frame selezionato, generare una descrizione automatica e salvare il tutto in un file HTML.

1.2 Preproduzione di un film

Preproduzione di un Film

Le produzioni cinematografiche sono comunemente gestite suddividendo il progetto in quattro fasi principali [12]:

- Sviluppo: In questa fase iniziale, una serie di idee vengono esaminate per identificare quella più idonea alla realizzazione di un film.
- Preproduzione: Durante questa fase, si definiscono i costi e si reperiscono i fondi necessari per avviare la produzione. Inoltre, vengono selezionati i membri della troupe e il cast, pianificando ogni aspetto in preparazione delle riprese.
- Riprese: Questa è la fase di cattura effettiva di tutti i materiali multimediali necessari per la creazione del film.
- Postproduzione: Include una serie di processi riguardanti sia l'aspetto visivo sia quello sonoro. I materiali raccolti sul set vengono montati, sincronizzati e doppiati per generare il prodotto cinematografico finale.

La fase di particolare interesse è quella della preproduzione, che si articola in quattro principali sottosezioni [13]:

- Sviluppo della storia: Questa fase preliminare si concentra su una revisione iterativa dello script, mirata a eliminare eventuali scelte narrative o sezioni non efficaci. Al termine di questa fase, la sceneggiatura dovrebbe essere solida e coesa, pronta per la fase di produzione. Contemporaneamente, viene stilata una lista di tutti i materiali necessari per la realizzazione del film.
- Design dei costumi e della produzione: Qui si compiono tutte le scelte stilistiche per garantire una coerenza estetica nel prodotto finale. Questa fase comprende

sia il design dei costumi sia gli elementi scenografici, al fine di creare un prodotto cinematografico stilisticamente uniforme.

- **Budgeting e finanziamento:** Questa fase prevede la definizione di una lista dettagliata delle spese necessarie alla realizzazione del film, nonché la ricerca di finanziatori disposti a supportare economicamente il progetto.
- **Preparazione alle riprese:** In questo momento, si procede alla formazione del cast e all'assunzione del team di produzione. Una volta confermato il personale, viene creato lo storyboard e una "shot list", ovvero una lista dei fotogrammi chiave, che viene rivista e criticata dal team per garantire una coerenza visiva e narrativa. È possibile, inoltre, creare un'animatic, una versione animata dello storyboard, che include informazioni sulle tempistiche di ogni fotogramma chiave.

Concluso lo storyboard, si procede alla definizione dei tempi necessari per registrare e catturare ogni momento chiave del racconto visivo.

Come si può osservare, la fase di storyboarding costituisce l'ultimo passaggio della fase di preproduzione; tuttavia, riveste un'importanza cruciale per conferire una forma concreta al prodotto finale e per permettere una revisione approfondita della narrazione e delle sequenze di ripresa.

1.3 Storyboard

Come evidenziato in precedenza, lo storyboard rappresenta il principale strumento di preproduzione e pre-visualizzazione, ideato per fornire una sequenza di disegni ordinati in base alla successione delle inquadrature e basati direttamente sulla sceneggiatura delle riprese. Questo strumento si compone di disegni concettuali che non solo illustrano, ma arricchiscono la narrazione della sceneggiatura, permettendo all'intero team di produzione di pianificare e coordinare in anticipo tutte le complesse azioni richieste prima di iniziare le riprese, assicurando la coerenza visiva e narrativa del prodotto finito.

Uno storyboard si avvale di una serie di elementi essenziali che facilitano la visualizzazione e la costruzione delle scene chiave di un film. Tra questi elementi figurano la posizione della telecamera, la descrizione degli attori coinvolti e degli oggetti di scena, nonché la durata della ripresa. Questi componenti costituiscono alcune delle informazioni fondamentali trasmesse attraverso lo storyboard, contribuendo a una gestione precisa e dettagliata della realizzazione delle scene.

1.3.1 Cenni storici

Il primo esempio documentato di utilizzo degli storyboard risale alla Francia e, in particolare, al celebre regista e illusionista Georges Méliès, che impiegò una forma rudimentale di storyboarding per realizzare i suoi iconici film, tra cui il celebre "Le Voyage dans la Lune" (Viaggio nella Luna)[8].

L'uso degli storyboard si diffuse successivamente anche nel mondo dei fumetti, in particolare nelle Sunday Comics [5], dove troviamo esempi significativi come la serie di strisce di Gertie the Dinosaur.



Figura 1.1. Esempio di storyboard del famoso corto Plane's crazy

Un ultimo, importante riferimento riguarda gli storyboard prodotti durante la cosiddetta "Era Disney" a partire dagli anni '30. Nel 1930, infatti, i Walt Disney Studios furono i primi a intuire l'importanza dello storyboard nella produzione di lungometraggi. A partire dal corto Plane Crazy, Disney implementò uno specifico dipartimento per la realizzazione di storyboard, ridefinendo gli standard di creazione e utilizzo di questo strumento. L'approccio di Disney ispirò ben presto altri studi cinematografici, come la Warner Brothers, fino a giungere al 1939, anno in cui *Gone with the Wind* (Via col vento) divenne il primo film live action prodotto utilizzando in modo sistematico lo storyboard. Da quel momento, la realizzazione dello storyboard divenne una prassi essenziale e imprescindibile

per la produzione di lungometraggi.

1.3.2 Caratteristiche ed elementi di uno storyboard

Pur mantenendo una struttura flessibile per adattarsi a diverse esigenze, uno storyboard presenta solitamente una serie di elementi essenziali che ne garantiscono la funzionalità e la coerenza narrativa. Di seguito vengono elencati e descritti brevemente tali elementi [4]:

- **Frame o Vignetta:** Disegno o schizzo che rappresenta visivamente la scena, includendo i personaggi e l'ambiente necessari per descriverla.
- **Numero della scena e dell'inquadratura:** Codice numerico che permette di ordinare le vignette in sequenza logica. Solitamente si utilizza una struttura del tipo 1.1, dove "1" indica la scena e "1" l'inquadratura all'interno di essa.
- **Tipo di Inquadratura:** Indica la posizione della fotocamera rispetto alla scena. Questo elemento costituisce uno strumento creativo del regista, consentendogli di scegliere cosa e come mostrare, ad esempio se focalizzarsi su oggetti, paesaggi, o su specifici elementi della scena, per orientare l'attenzione dello spettatore.
- **Tipo di Obiettivo:** Specifica la lunghezza focale utilizzata, determinando l'ampiezza del campo visivo catturato. Una lunghezza focale maggiore restringe l'angolo di campo, mentre una focale più corta lo amplia.
- **Durata dell'Inquadratura:** Indica il tempo in cui una determinata inquadratura è mantenuta. Questo dato è particolarmente rilevante per le animatic, in cui il ritmo temporale è scandito dal montaggio stesso.
- **Descrizione della Scena:** Breve descrizione testuale delle azioni e degli eventi che dovrebbero avvenire nella ripresa.
- **Movimenti dei Personaggi:** Utilizzo di frecce direzionali per indicare i movimenti che i personaggi dovrebbero compiere all'interno della scena.
- **Movimenti della Macchina da Presa:** Anche in questo caso si utilizzano frecce direzionali o, meno frequentemente, annotazioni testuali per descrivere movimenti della telecamera, come "zoom in," "zoom out," o movimenti dolly.
- **Luogo:** Informazioni riguardanti l'ambientazione della scena, inclusi dettagli su luce, ora del giorno e altri elementi contestuali che aiutano a visualizzare il set.
- **Cambio di Inquadratura:** Può avvenire all'interno della stessa vignetta o nella transizione tra due riprese.
- **Dialoghi:** Trascrizione testuale delle interazioni verbali tra i personaggi.
- **Suoni:** Annotazione degli effetti sonori considerati fondamentali per la scena.
- **VFX (Effetti Visivi):** Indicazione di eventuali effetti speciali richiesti per la scena.

- Costo: Stima del budget necessario per realizzare l'inquadratura. Questi elementi costituiscono la struttura basilare di uno storyboard, fornendo tutte le informazioni essenziali per la pianificazione visiva, tecnica e narrativa delle riprese

1.4 La realtà aumentata

Con il termine realtà aumentata (AR) si fa riferimento a un insieme di tecnologie che sovrappongono contenuti digitali a elementi del mondo reale. Negli ultimi anni, sono stati sviluppati diversi metodi per interagire e ricevere informazioni digitali, il che ha portato alla creazione di numerosi termini spesso utilizzati in modo ambiguo o senza una definizione precisa. Lo scopo di questo paragrafo è quindi di elencare e descrivere in maniera dettagliata tali termini per chiarirne le differenze.

1.4.1 Rapporto tra realtà aumentata e virtuale: il reality-virtuality continuum

Immaginando di tracciare una linea ed inserendo il mondo reale ad un estremo ed una simulazione completamente virtuale all'altro, si ottiene il cosiddetto Reality-Virtuality Continuum (inserire riferimento all'immagine). Questa linea consente di visualizzare la realtà e la realtà virtuale non come opposti, ma come estremi di un continuum, lungo il quale si collocano diversi tipi di esperienze digitali. All'interno di questa classificazione, trovano il loro posto i concetti di realtà mista (Mixed Reality, MR), realtà aumentata (Augmented Reality, AR) e virtualità aumentata (Augmented Virtuality, AV).

La principali differenze tra queste tipologie risiede nel grado di integrazione delle informazioni reali all'interno della simulazione:

Mixed Reality (MR): In questa categoria, informazioni provenienti dal mondo reale e informazioni sintetiche della simulazione si uniscono in un'unica esperienza interattiva. La realtà mista permette, quindi, una combinazione dinamica tra mondo reale e contenuti virtuali ed è la classificazione più ampia che può essere fatta all'interno del continuum (come si può vedere in figura). **Augmented Reality (AR):** Il termine si riferisce ad una simulazione in cui oggetti digitali vengono sovrapposti al mondo reale, arricchendo l'esperienza sensoriale e cognitiva dell'utente. L'AR mira a integrare contenuti virtuali che completano la percezione dell'ambiente circostante. **Augmented Virtuality (AV):** Questo concetto si avvicina alla realtà virtuale, poiché si tratta di una simulazione in un mondo totalmente digitale. Tuttavia, all'interno di questo mondo virtuale vengono inseriti elementi provenienti dalla realtà, consentendo l'interazione con oggetti reali all'interno di un contesto virtuale. Queste categorie rappresentano le diverse modalità di interazione tra il mondo reale e quello digitale, ciascuna caratterizzata da un diverso grado di fusione tra elementi reali e virtuali.

Mixed Reality

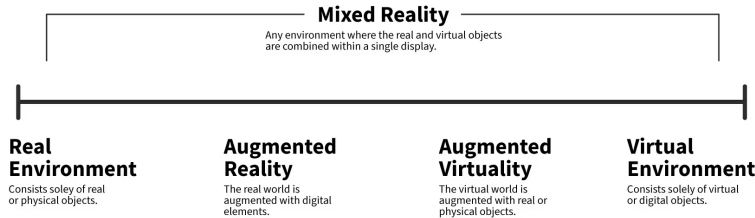


Figura 1.2. grafico di transizione dalla realtà alla realtà virtuale

1.4.2 Dispositivi di realtà aumentata

Negli ultimi anni, il settore della realtà virtuale ed aumentata si è sempre più orientato verso il mercato consumer. I dispositivi destinati a questo mercato cercano di mantenere un prezzo relativamente accessibile, concentrandosi su specifiche modalità di realtà mista, descritte nel dettaglio di seguito:

- **Head Mounted Display (HMD) o visori immersivi:** Si tratta di dispositivi indossabili dotati di lenti posizionate direttamente nel campo visivo dell'utente. Gli HMD si dividono in due principali tipologie: **Optical See-Through:** Visori con lenti trasparenti che permettono di sovrapporre i contenuti digitali alla visione del mondo reale. **Video See-Through:** Dispositivi con lenti opache integrate con display interni, che riproducono un'immagine aumentata dell'ambiente circostante. Un esempio innovativo di questa seconda categoria è il Meta Quest 3, il dispositivo utilizzato per lo sviluppo di questa tesi. Questo visore sfrutta telecamere e sensori integrati per generare una mappa di punti visualizzata attraverso le lenti dell'utente. Tale mappatura fornisce una consapevolezza spaziale degli oggetti nell'ambiente e localizza con precisione le interazioni dell'utente.
- **Display:** Questo tipo di sistema utilizza un display come interfaccia tra l'utente e il mondo virtuale, e può essere suddiviso in due categorie: Sistemi con un display statico e una telecamera mobile, che riprende l'ambiente. Sistemi con telecamera e display integrati, come nei moderni tablet, che permettono una maggiore mobilità, consentendo all'utente di muoversi nello spazio e adattare l'esperienza virtuale alle diverse angolazioni di visione.
- **Projector-Based Systems:** Questa tipologia sfrutta proiettori per arricchire l'ambiente reale con contenuti digitali, adattando le proiezioni alla geometria dello spazio circostante. Un esempio di questi sistemi è rappresentato da proiettori che modificano il contenuto visivo in base ai dislivelli e alle caratteristiche del piano di proiezione.



Figura 1.3. Sistema a realtà aumentata basato su un dispositivo mobile



Figura 1.4. Sistema realtà aumentata basato su proiettore

Questi diversi approcci alla realtà aumentata e virtuale dimostrano la varietà e l'accessibilità crescente delle soluzioni tecniche orientate al mercato consumer.

1.4.3 Ambiti di applicazione

Negli ultimi anni, il crescente aumento di potenza dei componenti hardware ha reso i dispositivi di realtà aumentata e virtuale più accessibili e diffusi. Grazie a questi sviluppi, la realtà aumentata ha trovato applicazione in numerosi settori, tra cui i seguenti:

- **Intrattenimento:** Nel settore videoludico, dispositivi come il Meta Quest e l'HTC Vive hanno favorito l'introduzione di esperienze di realtà virtuale e aumentata nelle case dei videogiocatori. Inoltre, con dispositivi AR basati su display, sono emerse applicazioni di realtà aumentata per dispositivi mobili, come il popolare videogioco Pokémon GO della Nintendo. Quest'ultimo sfrutta il GPS e la telecamera dei cellulari, permettendo agli utenti di esplorare il mondo reale per sbloccare ricompense e incontrare creature virtuali.
- **Industriale:** In ambito industriale, la realtà aumentata e virtuale trova impiego nell'addestramento del personale per operare in ambienti o situazioni ad alto rischio, dove un errore potrebbe avere conseguenze gravi. Attraverso simulazioni che ricreano ambienti potenzialmente pericolosi, gli utenti vengono guidati nella risoluzione di problemi specifici, ricevendo correzioni e istruzioni senza che vi siano rischi reali.
- **Medicina:** Il settore medico rappresenta uno dei campi di applicazione più promettenti. Grazie alle applicazioni di realtà aumentata, è possibile utilizzare dispositivi AR per proiettare immagini di organi o vasi sanguigni, supportando i chirurghi durante le procedure e guidandoli in tempo reale. Inoltre, le simulazioni in realtà aumentata sono utilizzate per addestrare il personale medico, offrendo un ambiente sicuro per l'apprendimento di procedure complesse.
- **Turismo:** Nel settore turistico, la realtà aumentata consente esperienze interattive che coinvolgono gli utenti, mostrando particolarità del territorio e delle attrazioni locali. Nei musei, ad esempio, la realtà aumentata permette di effettuare visite guidate interattive, offrendo approfondimenti sui contenuti esposti e arricchendo l'esperienza culturale.
- **Marketing:** Numerose aziende stanno adottando applicazioni AR per consentire ai clienti di visualizzare i prodotti nel contesto domestico. Un esempio significativo è IKEA Place, un'applicazione che permette di posizionare virtualmente i mobili IKEA negli spazi di casa, aiutando i clienti a valutare l'adeguatezza degli arredi rispetto al proprio ambiente. Questi settori dimostrano il potenziale della realtà aumentata e virtuale nel migliorare l'interazione, la formazione e l'esperienza dell'utente in contesti diversi, favorendo un'adozione sempre più ampia di queste tecnologie.

1.5 Reti neurali

Durante lo studio delle possibili soluzioni per implementare un sistema di riconoscimento, è stata valutata anche l'opzione di utilizzare una rete neurale, integrandola nel flusso video del dispositivo di realtà aumentata e virtuale. Una rete neurale è un modello di

machine learning che prende decisioni in modo simile al cervello umano, attraverso processi che imitano il funzionamento dei neuroni biologici, lavorando in sinergia per identificare fenomeni, valutare opzioni e giungere a conclusioni [1].

1.5.1 Ambiti di utilizzo delle reti neurali

Negli ultimi anni, le reti neurali hanno trovato ampia applicazione in diversi settori, tra cui:

Sistemi Esperti: Questo è uno degli ambiti in cui l'intelligenza artificiale è maggiormente impiegata. Un sistema esperto opera in un dominio limitato, in cui l'IA acquisisce competenze paragonabili a quelle di un esperto umano. L'IA in questo contesto si articola in due componenti principali: una base di conoscenza, che raccoglie le informazioni rilevanti per il dominio specifico, e un motore inferenziale, che applica queste conoscenze per risolvere problemi specifici. **Medicina:** L'applicazione delle reti neurali in ambito medico risale al 1972, quando fu sviluppato il sistema MYCIN, un programma di intelligenza artificiale progettato per analizzare referti medici e fornire diagnosi accurate. MYCIN era inoltre in grado di spiegare le ragioni delle sue scelte, rappresentando un primo esempio di intelligenza artificiale trasparente e interpretabile nel campo della diagnosi medica. **Automotive:** Nel settore automobilistico, l'impiego dell'intelligenza artificiale, in particolare per il riconoscimento degli oggetti, è sempre più comune. Tali tecnologie assistono i conducenti durante la guida, contribuendo a evitare incidenti attraverso l'individuazione di ostacoli e la gestione automatica di situazioni potenzialmente pericolose. **Robotica:** In questo settore, le reti neurali permettono ai robot di prendere decisioni in tempo reale, basandosi su dati acquisiti in tempo reale dall'ambiente circostante. Questa capacità decisionale immediata è cruciale per applicazioni in cui la rapidità e la precisione delle risposte sono essenziali (riferimento sito web BuiltIn). Gli ambiti di applicazione delle reti neurali sono pressoché illimitati; qui ci siamo soffermati sui principali, ma i potenziali sviluppi continuano ad ampliare il campo d'azione di queste tecnologie.

1.5.2 Funzionalità delle reti

Le funzionalità offerte dalle reti neurali, valutate per questo progetto, si concentrano principalmente sui campi del riconoscimento degli oggetti (object recognition), della comprensione della scena (scene understanding) e della stima della posa (pose estimation). È utile approfondire questi concetti prima di spiegare perché questa soluzione si è rivelata non ideale nella fase di sviluppo del progetto.

Comprensione della scena (Scene Understanding): La scene understanding è una disciplina nell'ambito delle reti neurali e della visione artificiale che si occupa di analizzare e interpretare scene visive complesse. L'obiettivo è consentire al modello di riconoscere e comprendere non solo gli oggetti presenti in un'immagine o un video, ma anche le loro relazioni spaziali, semantiche e contestuali. Questo tipo di analisi mira a fornire al sistema una visione completa e contestualizzata della scena, migliorando l'interazione tra il mondo reale e quello virtuale. **Stima della Posa (Pose Estimation):** La pose estimation consente di stimare la "posa" di un oggetto, ovvero la combinazione di informazioni relative alla sua posizione e orientamento nello spazio. Questa funzione è essenziale per localizzare

con precisione un oggetto in una scena e per determinare come esso sia orientato rispetto ad altri elementi, facilitando l'interazione precisa con contenuti virtuali. Riconoscimento degli Oggetti (Object Recognition): Il riconoscimento degli oggetti è la capacità di una rete neurale di identificare e classificare vari tipi di oggetti, assegnando loro delle etichette. Questo campo si suddivide in due modalità: Riconoscimento bidimensionale: La capacità di riconoscere oggetti in immagini bidimensionali, come fotografie o altre rappresentazioni senza profondità. Riconoscimento tridimensionale: La capacità di riconoscere oggetti in rappresentazioni che includono informazioni sulla profondità, come video o acquisizioni 3D, in cui l'algoritmo può rilevare e classificare oggetti anche in presenza di dati spaziali completi. Queste funzionalità rappresentano ambiti chiave nella visione artificiale, ma si sono rivelate non del tutto adatte per gli obiettivi specifici del progetto, come verrà analizzato a seguire.

1.5.3 Limitazioni dei modelli attualmente disponibili

Sebbene le funzionalità di queste reti neurali siano estremamente utili e, ad oggi, esistono modelli all'avanguardia scaricabili e utilizzabili gratuitamente, vi sono significative limitazioni che hanno impedito il loro impiego in questo progetto.

Durante la fase di implementazione è emerso il problema che le reti capaci di realizzare le funzionalità descritte – come l'object recognition, la scene understanding e la pose estimation – richiedono dati di profondità e varie altre metriche che devono essere raccolte da sensori specifici, ai quali non è possibile accedere direttamente sulla maggior parte dei dispositivi disponibili in commercio. Sebbene questi dati siano sfruttati dai dispositivi stessi, raramente sono disponibili funzioni che consentano l'accesso a tali informazioni. La mancanza di dati adeguati avrebbe compromesso il funzionamento corretto delle reti.

Un ulteriore impedimento significativo riguardava la disponibilità di dati per l'addestramento delle reti neurali. La raccolta di dataset sufficientemente ampi e rappresentativi per adattare le reti alle esigenze specifiche del progetto avrebbe richiesto un'enorme quantità di tempo e sforzi, ben oltre le possibilità di una singola persona.

Per superare queste limitazioni, si è scelto di adottare le innovative funzionalità offerte dal visore Meta Quest. Pur non consentendo l'accesso diretto e completo ai dati raccolti dai vari sensori, il visore elabora queste informazioni internamente mediante reti neurali predefinite e offre accesso ai risultati finali. In particolare, è possibile eseguire una scansione dell'ambiente circostante, ottenendo una serie di bounding box di oggetti automaticamente etichettati. Nel capitolo dedicato all'implementazione dell'applicazione verranno esaminati in dettaglio i modi in cui questi dati sono stati sfruttati per realizzare un'interazione efficace tra oggetti virtuali e reali.

Capitolo 2

Stato dell'arte

2.1 2.1 Software di storyboarding 2D attualmente disponibili

Attualmente nel mercato la grande maggioranza di software disponibili si concentrano principalmente sulla creazione di storyboard 2D, inserendo assets composti da immagini che possono andare a corredare una vignetta ed eventualmente animare questi oggetti. In questo senso il software più comunemente usato è l'applicativo storyboardThat [14] che realizza proprio la funzione sopra descritta e permette di creare una serie di vignette corredate di personaggi, ambientazioni e descrizioni testuali. Queste soluzioni hanno il grande vantaggio di non richiedere dispositivi hardware eccessivamente potenti e di mantenersi molto aderenti al concetto di storyboarding tradizionale introdotto nei primi capitoli.

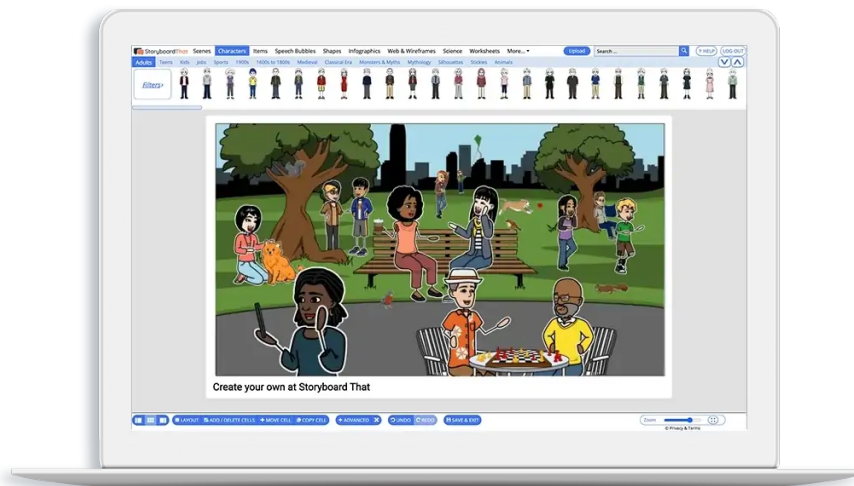


Figura 2.1. Applicativo StoryboardThat

2.2 Generazione di storyboard

Lo svantaggio delle soluzioni sopra descritte però è quello di non permettere di poter osservare gli oggetti nello spazio 3D ed inoltre gli asset usati non hanno il concetto di prospettiva. Si viene quindi a creare una certa differenza tra le immagini rappresentate nello storyboard e quelle poi effettivamente catturate in fase di ripresa. Un'altra mancanza di questi applicativi è la difficoltà nel farsi un'idea della relazione e della disposizione degli oggetti nello spazio prima che questi vengano effettivamente inseriti nella scena. Per questa ragione oggi diversi software in realtà aumentata si stanno diffondendo, aiutati anche dal crescente interesse e dall'arrivo di soluzioni dal costo relativamente basso. L'integrazione di un caschetto o di un dispositivo a realtà aumentata o virtuale permette un'esperienza di storyboarding più immersa e interattiva rispetto ai software tradizionali. Sfruttare la realtà Virtuale nella creazione di storyboarding permette poi una manipolazione molto più libera e svincolata dalla fisica della telecamera, inoltre è possibile pre visualizzare una serie di elementi per comprendere a pieno il loro potenziale o la loro efficacia nella scena. Se invece si opta per un'applicativo che si basa sulla realtà aumentata è possibile sfruttare la zona circostante come ambientazione per la realizzazione del proprio storyboarding.

2.3 Storyboarding in VR e AR

AR Storyboard [10] è un software per la creazione di storyboard in realtà aumentata. Lo strumento proposto è facile da usare e fornisce un'interfaccia intuitiva per la composizione della scena e il controllo della posa e del movimento della telecamera grazie a questa struttura anche utenti non esperti sono in grado di realizzare uno storyboard.

AR Storyboard si basa sull'utilizzo di marker definiti come "Item Blocks" che corrispondono a dei fogli di carta spessi dove l'informazione visiva è stampata. Il sistema (raffigurato nella figura) si costituisce da un computer per l'elaborazione dei dati, da una videocamera per inquadrare la scena ed una serie di item Blocks che l'utente può piazzare per creare la vignetta dello storyboard. Ogni item block identifica un personaggio o un oggetto di scena anche se per comodità e per non avere un numero smisurato di marker si è scelto di permettere all'utente di regolare una serie di impostazioni per ogni oggetto. Ad esempio per un personaggio piazzato può essere scelto il sesso, l'età o l'altezza in modo da non dover creare diversi item Block per ogni singola variazione.

Una scena per lo storyboarding viene quindi creata inquadrando uno o più item blocks per inserire oggetti di scena e personaggi, infine si aggiungono delle azioni da svolgere. Si sfrutta poi la telecamera per effettuare degli screenshot e salvare il tutto.

Lo svantaggio di questa soluzione sta nel fatto che, pur sfruttando la realtà aumentata non facendo uso di un caschetto l'immersività è limitata al campo di visibilità della telecamera che si usa per inquadrare la scena

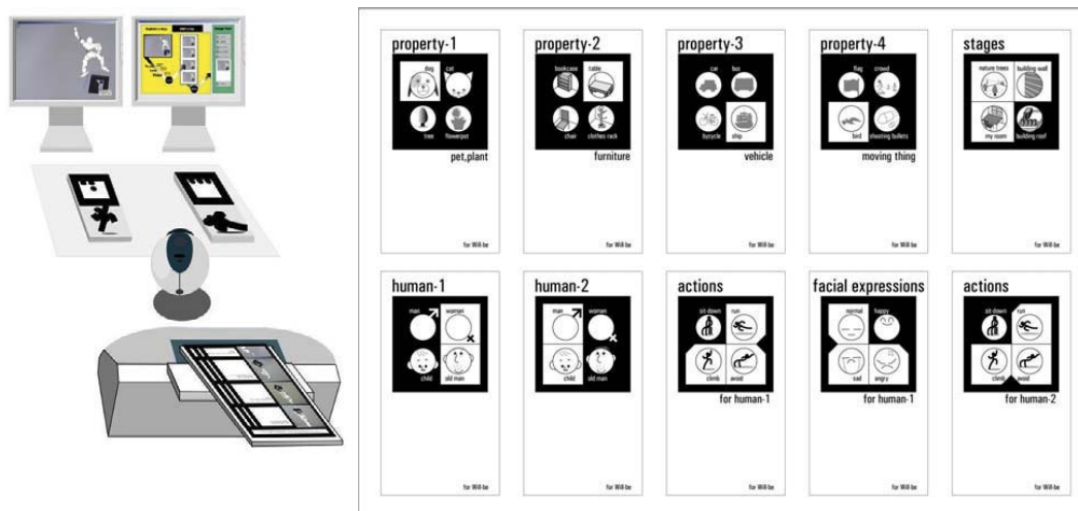


Figura 2.2. Struttura della fase di Storyboarding

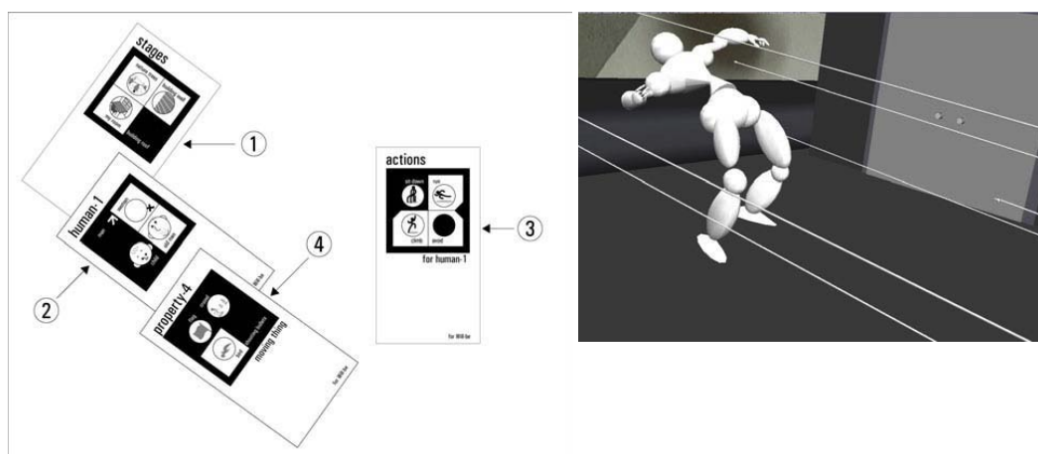


Figura 2.3. Creazione di uno storyboard mediante aggiunta di carte

2.4 2.2.2 Augmented reality storyTelling: story creatAR

Story CreatAR [16] è stato progettato per supportare gli autori nell'analisi, nella sperimentazione e nella riflessione sulle relazioni spaziali tra gli elementi narrativi e tra la storia e l'ambiente circostante. Il progetto è stato validato tramite test su casi di studio specifici, volti a verificare l'efficacia dell'applicativo.

Gli utenti possono utilizzare un visore immersivo (Head-Mounted Display, HMD) per visualizzare personaggi e oggetti in scala reale, come se questi fossero realmente presenti nel campo visivo (Field of View, FoV) dell'utente, pur rimanendo limitati dal FoV del display. Questo approccio differisce dall'AR portatile descritto nel sottoparagrafo precedente, dove lo spettatore, tenendo in mano il dispositivo, vede i contenuti vincolati al FoV del dispositivo stesso, rendendo più difficile l'interazione con contenuti "a grandezza naturale", come avatar umani, specialmente quando lo spettatore si avvicina a tali elementi.

Story CreatAR permette inoltre di acquisire una conoscenza di base dell'ambiente circostante e consente all'utente di:

- Definire regole di posizionamento, specificando come disporre gli elementi dello spazio in base a caratteristiche spaziali (ad esempio, complessità visiva, apertura e integrazione visiva).
- Stabilire regole di attraversamento, assegnando comportamenti di navigazione ai personaggi in base alla distanza, all'urgenza e alle caratteristiche dell'ambiente.
- Associare elementi interattivi dell'attività a determinate caratteristiche spaziali.
- Raggruppare gli elementi in modo che possano essere collocati insieme e creare raggruppamenti sovrapposti per rappresentare la sequenza narrativa di una storia.
- Definire regole di formazione, affinando la disposizione dei gruppi di elementi e utilizzando i confini dell'ambiente per migliorare la coerenza visiva.

Story CreatAR supporta una varietà di paradigmi narrativi, tra cui la narrazione lineare e non lineare, concentrandosi principalmente su una struttura di tipo "character-driven", dove la storia è guidata e sviluppata da un personaggio principale che interagisce attivamente con il mondo. L'utente può scegliere di eseguire una simulazione in realtà aumentata, scansionando l'ambiente circostante, oppure in realtà virtuale, creando una versione digitalizzata dello spazio. In entrambi i casi, le informazioni spaziali raccolte sono fondamentali per un posizionamento coerente degli oggetti. Tramite un'interfaccia intuitiva, l'utente può inserire elementi digitali e, ad esempio, aggiungere personaggi con animazioni predefinite o creare animazioni personalizzate utilizzando il software integrato in Unity, "Movebox". In questo modo, è possibile costruire progressivamente lo storyboard.

2.5 Animazione in realtà aumentata

2.5.1 Authoring animations of virtual objects in augmented reality-based 3D space

Dopo aver analizzato i principali software di storyboarding 3D, è utile spostare leggermente l'attenzione dallo storyboarding per esaminare più in dettaglio gli approcci all'animazione dei personaggi digitali nell'ambito della realtà aumentata. A questo proposito, è stato considerato uno studio condotto da Manabu Eitsuka e Masahito Hirakawa [9], che utilizza dispositivi mobili e una realtà aumentata di tipo “Display-Based” per animare in modo semplice e intuitivo personaggi digitali, consentendo all'utente di visualizzare la scena completa e catturare l'inquadratura desiderata. Il software impiegato nello studio si basa sull'uso di marker, in modo simile a AR Storyboard, e sfrutta il framework Vuforia, un noto software per il riconoscimento di marker che consente la generazione e l'interazione in tempo reale con modelli digitali. Per rilevare il movimento delle dita o della mano, il sistema utilizza la telecamera del dispositivo mobile per calcolare lo spostamento delle dita e la variazione della loro dimensione.



Figura 2.4. Funzionamento generale di StoryCreatAR

L'interfaccia utente, progettata per essere utilizzata principalmente con un solo dito, è estremamente intuitiva e rende facile cambiare inquadratura semplicemente spostando la posizione del dispositivo e continuando a inquadrare il soggetto. Le animazioni sono realizzate modificando la posa del personaggio nei frame chiave di riferimento, e successivamente interpolando questi frame per ottenere un'animazione fluida.

Questo sistema presenta il vantaggio di essere intuitivo e semplice da utilizzare, rendendolo accessibile anche a utenti privi di esperienza specifica nella modellazione e animazione dei personaggi.

2.6 AR animator

Il secondo caso di studio analizzato nell'ambito dell'animazione di personaggi in realtà aumentata è stato quello di AR Animator [6]. AR animator software che si basa sull'uso di



Figura 2.5. Immagine panoramica dello storyboard

un cellulare abilitato all'AR come controllore per l'animazione di personaggi basati sulla performance nell'AR mobile (Figura 2.6). L'idea chiave di AR Animator è quella di imitare il processo di manipolazione tipico delle bambole per la narrazione: gli utenti controllano un telefono cellulare e proprio questi movimenti impartiscono delle corrispondenti azioni al personaggio che quindi in simultanea esegue gesti di movimento per recuperare le animazioni esistenti da un archivio di animazione dei personaggi. Si tratta di un processo intuitivo anche per gli utenti occasionali, dal momento che l'interazione emula il comportamento di una bambola oggetto con cui la maggior parte di noi durante l'infanzia ha avuto familiarità. I risultati raggiunti da AR animato sono molto positivi in quanto riescono in pieno nel loro obiettivo di creare un sistema altamente accessibile ed al tempo stesso intuitivo. La limitazione principale di AR Animator sta però nel fatto che, l'accuratezza delle animazioni e delle riprese si basa innanzitutto proprio sulla qualità della cattura video. Avendo però integrato i movimenti del personaggio a dei gesti che vanno molto spesso a muovere bruscamente la telecamera ci sarà sempre un certo grado di distorsione delle traiettorie ideali per un personaggio.

2.7 Previsualizzazione in realtà mista

Come accennato nel Capitolo 1, oltre allo storyboard, un altro strumento di pre-produzione utile è l'animatic, che permette di visualizzare con maggiore chiarezza le informazioni temporali relative a una sequenza video. Un software specificamente progettato per questo scopo è Mr-PreViz, un'applicazione che consente di combinare oggetti digitali con ambienti reali per creare animatic, fornendo così una visione preliminare del risultato finale della scena, anche in presenza di effetti CGI.

Il funzionamento di Mr-PreViz si articola in cinque fasi principali, illustrate nella Figura [X.X]:

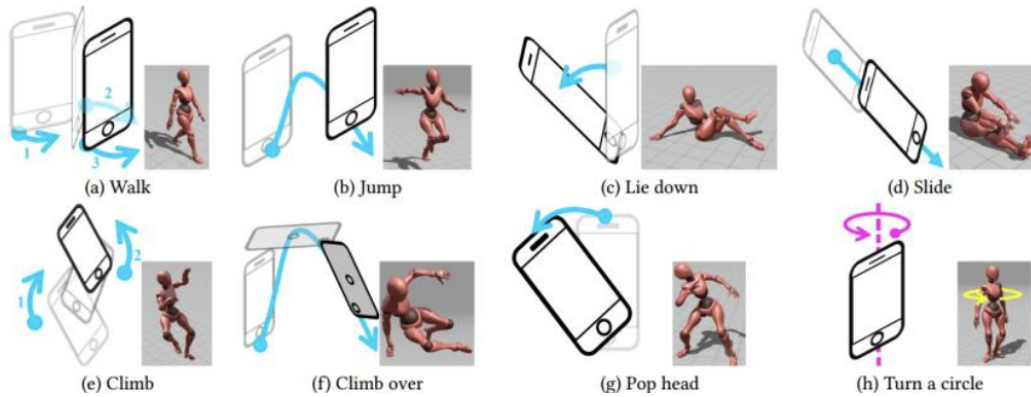


Figura 2.6. Funzionamento di AR animator

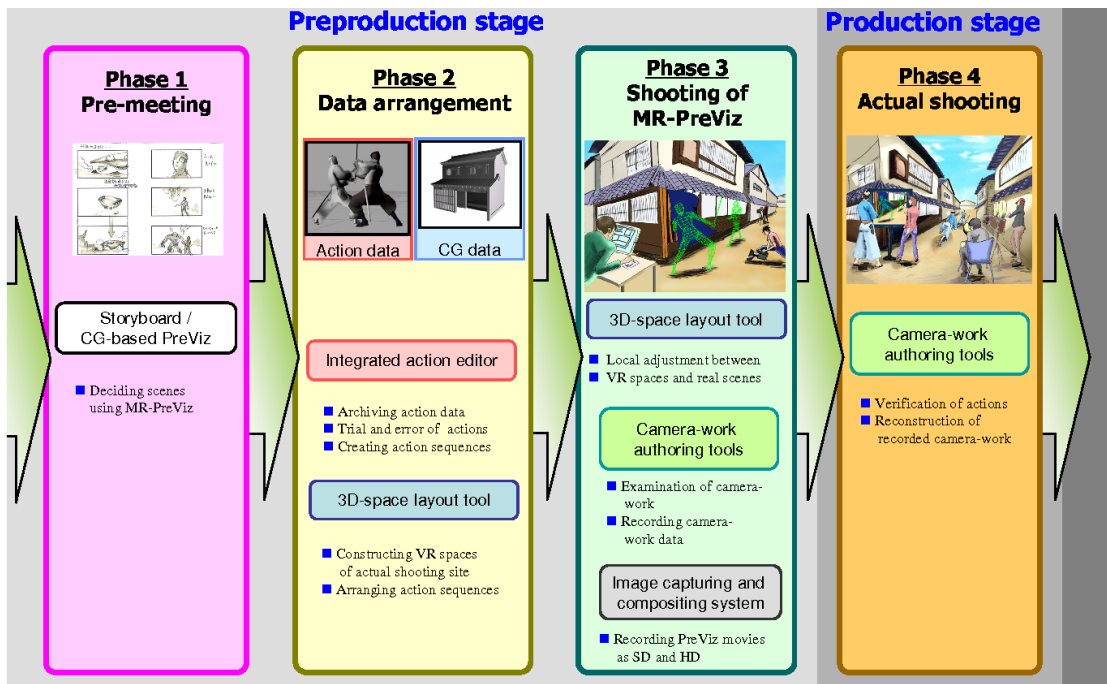


Figura 2.7. Pipeline di lavoro di MrPreViz

- Fase 1: Gli utenti selezionano le scene che richiedono l'uso di Mr-PreViz, partendo da una previsione approssimativa (PreViz) della sequenza CGI completa. Non tutte le scene richiedono necessariamente il software; alcune possono essere riprese

normalmente.

- Fase 2: Vengono raccolti i dati di computer grafica e delle azioni, che possono derivare da fonti diverse, come la motion capture o l'animazione in CG, per essere utilizzati nelle fasi successive.
- Fase 3: Gli utenti progettano lo spazio misto realtà (MR) posizionando, tramite un PC, gli oggetti in computer grafica e i dati di azione raccolti e modificati nella fase 2.
- Fase 4: Lo spazio MR viene visualizzato onsite, facendo riferimento ai dati di layout impostati nella fase 3 e utilizzando i dati per guidare l'inquadratura e i movimenti.
- Fase 5: I filmati e i dati elaborati nella fase 4 vengono integrati con le riprese effettive tramite l'uso del browser MRP.

Per realizzare queste fasi, il sistema è dotato di diversi strumenti: un editor di azioni per creare sequenze di movimento dei personaggi, uno strumento di layout per progettare lo spazio 3D come descritto nella fase 3, ed infine uno strumento di configurazione della videocamera per esaminare i movimenti e le inquadrature della scena.

Mr-PreViz rappresenta un supporto prezioso per l'analisi preliminare delle riprese che includono computer graphics, anche se non può sostituire il processo di ripresa tradizionale. Un vantaggio significativo di questo approccio è la possibilità di effettuare riprese direttamente onsite e in ambienti esterni, mantenendo una coerenza visiva con l'ambiente reale.

2.8 Analisi dei requisiti applicativi

Il presente progetto di tesi si sviluppa come estensione di un precedente lavoro realizzato dall'ingegnere Sara Piumatti. L'obiettivo consiste nell'eseguire un porting del progetto iniziale, originariamente concepito per Hololens 2, adattandolo al visore MetaQuest. La scelta di utilizzare questo dispositivo è motivata dall'integrazione di innovative funzionalità di scene understanding, che consentono di ampliare il ventaglio di funzionalità dell'applicazione di storyboarding. Grazie a tale scelta, è stato possibile aggiungere funzionalità di riconoscimento e interazione con oggetti reali, prima inaccessibili. Maggiori dettagli saranno forniti nei capitoli relativi alle tecnologie e all'implementazione.

Considerando l'esistenza di un'applicazione preesistente, le sfide principali si sono concentrate sul mantenimento di una struttura di interazione simile a quella dell'applicazione originale, aggiornando e, se necessario, modificando gli aspetti ottimizzati dai nuovi framework rilasciati nel frattempo. Un'ulteriore sfida ha riguardato il mantenimento di tutte le caratteristiche già presenti nella versione precedente, integrando al contempo le nuove funzionalità in modo coerente e fluido.

Infine, è stata condotta un'analisi comparativa sulle principali soluzioni software per lo storyboarding in realtà aumentata (AR), oltre a quelle sviluppate all'interno di questo progetto di tesi, con l'obiettivo di individuare i requisiti essenziali da implementare. Questo processo ha comportato una revisione di alcune scelte progettuali precedenti, allineandole ai criteri emersi dall'analisi.

Capitolo 3

Tecnologie

In questo capitolo verrà data una spiegazione generale di tutte le tecnologie che sono necessarie per la corretta implementazione del progetto di cui si parlerà in seguito.

3.1 Unity

Per la realizzazione dell'applicativo, è stato selezionato il motore grafico Unity [18]. Unity è un ambiente di sviluppo multiplatforma per la creazione di applicazioni interattive 2D e 3D. Questo motore grafico offre numerosi vantaggi e rappresenta una delle principali scelte degli sviluppatori per la produzione di applicazioni interattive. In particolare, i principali punti di forza di Unity risiedono nella sua natura multiplatforma: consente infatti di generare build per diverse piattaforme, riducendo i costi e il tempo necessari per effettuare porting.

Inoltre, Unity offre un modello di utilizzo gratuito che permette agli sviluppatori di lavorare e pubblicare applicazioni senza costi iniziali, accedendo a tutte le funzionalità del software. Superate determinate soglie di guadagno, Unity richiede una percentuale sui profitti, ma rimane una soluzione accessibile e completa. Il software è altamente compatibile con altri strumenti di terze parti comunemente utilizzati nello sviluppo di applicazioni, come Blender [citazione] e Maya [citazione].

Un altro vantaggio di Unity è la sua compatibilità con i principali sistemi operativi, quali Windows, Mac e Linux. Originariamente sviluppato per il settore dei videogiochi, Unity si è evoluto per supportare una vasta gamma di applicazioni e oggi trova ampio impiego in settori quali ingegneria, automobilismo, cinematografia, realtà virtuale e realtà mista.

Dal punto di vista tecnico, il funzionamento di Unity si basa sulla creazione di "script", ovvero programmi in linguaggio C sharp associati a oggetti di gioco per definirne i comportamenti e le interazioni. Unity consente inoltre di eseguire simulazioni direttamente all'interno dell'editor, una possibilità utile per verificare il corretto funzionamento dell'applicazione prima della produzione della build definitiva, e offre pratiche soluzioni di debugging per ottimizzare e risolvere eventuali problemi.

3.1.1 Interfaccia di unity

Data la complessità del software, l'interfaccia di Unity si è arricchita nel tempo di una serie di funzionalità essenziali. Prima di proseguire, è utile soffermarsi brevemente su alcuni dei pannelli principali, i quali svolgono un ruolo fondamentale nel processo di sviluppo:

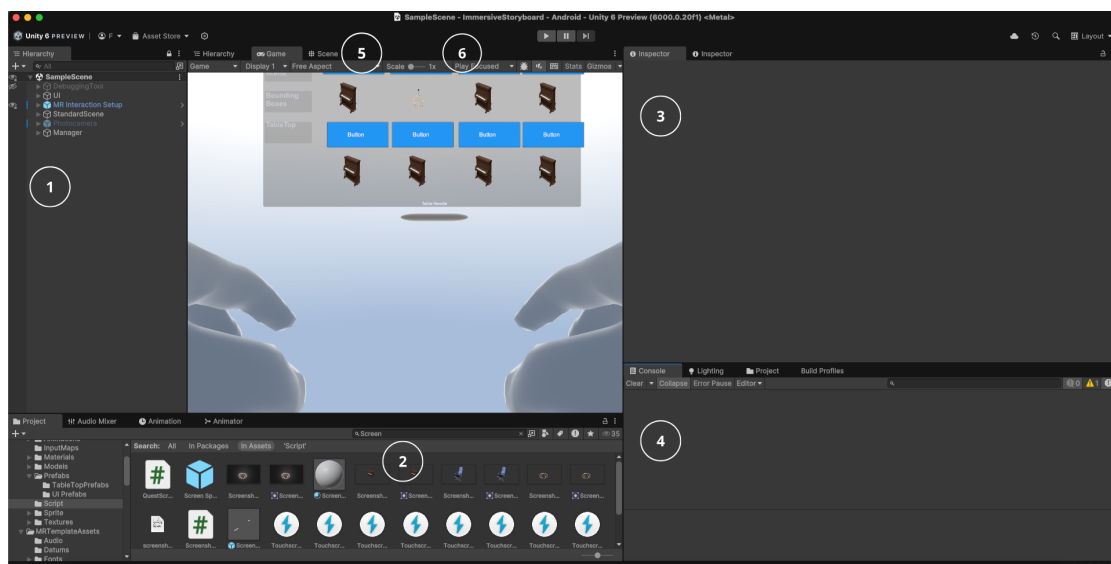


Figura 3.1. Viewport di Unity

1. Pannello delle Gerarchie: questo pannello elenca tutti gli oggetti presenti nella scena, organizzati in una struttura gerarchica. È possibile annidare oggetti al suo interno, in modo che qualsiasi modifica di posizione applicata a un oggetto "padre" si rifletta automaticamente sugli oggetti "figli".
2. Pannello del Progetto: il pannello Project contiene tutti i file necessari per il corretto funzionamento dell'applicazione, ma che non sono direttamente inclusi nella scena. Organizzato come un comune file system, consente di creare e gestire cartelle e file di vario tipo in una struttura ad albero.
3. Inspector: l'Inspector mostra tutte le proprietà dell'oggetto selezionato. Questo pannello permette di visualizzare, ad esempio, gli script associati a un oggetto di scena, oltre a informazioni quali materiale, posizione, orientamento e altre caratteristiche fondamentali per la gestione dell'oggetto.

4. Console: la Console, visibile all'interno del viewport, fornisce informazioni sui processi in corso, eventuali errori o eccezioni che si verificano durante l'avvio della simulazione. Inoltre, segnala errori di sintassi negli script che possono impedire la compilazione e fornisce aggiornamenti sullo stato di produzione della build dell'applicazione.
5. Pannello Scena/Videogioco(6): questi pannelli rappresentano lo spazio della simulazione. Il pannello Scena consente di posizionare oggetti, aggiungere script e modificarne le proprietà; il pannello Game, invece, permette di avviare la simulazione dall'editor, mostrando l'aspetto che l'applicazione avrà una volta generata la build.

L'integrazione delle informazioni fornite da questi pannelli ha reso possibile la realizzazione dell'applicativo oggetto di questo progetto.

3.2 Unity XR interaction toolkit

Oltre ai vantaggi precedentemente esposti, Unity è stato scelto anche per un ulteriore motivo: il rilascio di un avanzato framework per lo sviluppo di applicazioni in realtà mista. L'XR Interaction Toolkit di Unity offre un set completo di componenti che consente di implementare rapidamente la manipolazione di oggetti, sia tramite controller che attraverso l'uso delle mani.

In particolare, il framework fornisce una serie di scene preconfigurate che permettono di avviare il progetto con una struttura di base già impostata, comprensiva della gestione della telecamera e della cattura di gesti e azioni fondamentali. Inoltre, sono disponibili numerosi tutorial e video informativi che agevolano l'apprendimento del framework, rendendolo accessibile anche a sviluppatori meno esperti. Un'altra funzionalità vantaggiosa è la presenza di un simulatore che consente di testare l'applicazione direttamente in fase di sviluppo nell'editor, senza necessità di generare la build e trasferirla sul visore. Processo che richiederebbe svariati minuti per ogni test.

Un ulteriore vantaggio del framework è la sua compatibilità con diversi tipi di visori, il che apre la possibilità futura di esportare il progetto verso dispositivi differenti senza dover riadattare completamente il lavoro iniziale; ad eccezione di alcuni aggiornamenti, il progetto risulterà generalmente compatibile.

Infine, il framework offre una serie variabile di funzionalità in base al dispositivo per il quale è configurato. Nel nostro caso, le funzionalità sfruttate includono il riconoscimento di oggetti e la funzionalità di "Scene Understanding", le quali verranno illustrate più nel dettaglio nella sezione dedicata all'implementazione.

3.3 Rider

Rider è l'editor visuale scelto per la scrittura degli script dell'applicazione. Questo editor offre un'integrazione diretta con GitHub, che consente di caricare il codice sulla piattaforma web facilitando la gestione delle versioni e la collaborazione sul progetto. Inoltre, Rider dispone di avanzate funzionalità di debugging: è possibile collegarlo a Unity e posizionare dei breakpoint all'interno del codice per analizzare il flusso di esecuzione dell'applicazione, facilitando l'identificazione e la risoluzione di eventuali anomalie.

3.4 Github

GitHub è una piattaforma di hosting per il controllo di versione e la collaborazione sul codice, basata su Git, un sistema di controllo di versione distribuito. La piattaforma è progettata per supportare i programmatori nella gestione collaborativa dei progetti software, consentendo loro di organizzare, condividere e revisionare il codice in modo efficiente.

Ogni progetto su GitHub è archiviato in un repository, ovvero una cartella che contiene il codice, la cronologia delle modifiche e la documentazione del progetto. Grazie a questa struttura, GitHub permette di tracciare facilmente gli sviluppi del codice e ripristinare versioni precedenti in caso di corruzione o perdita di dati. GitHub è particolarmente apprezzato dai programmatori per la capacità di facilitare il lavoro in team e consentire a sviluppatori di tutto il mondo di collaborare in modo ordinato e organizzato.

In questa tesi, GitHub è stato uno strumento fondamentale sia per proteggere il progetto da eventuali errori o malfunzionamenti del sistema su cui è stato sviluppato, sia per condividere il progetto con i relatori e aggiornare rapidamente lo stato di avanzamento dei lavori.

3.5 Meta Quest

Nei capitoli precedenti è stato brevemente introdotto il dispositivo utilizzato per questa tesi, ovvero l'HMD Meta Quest 3. Questo dispositivo è stato selezionato per le sue avanzate funzionalità di object detection e scene understanding, caratteristiche che lo rendono particolarmente adatto a sviluppi innovativi nel campo della realtà aumentata e mista.



Figura 3.2. dispositivo Meta Quest

Rispetto alla versione precedente, che supportava esclusivamente la realtà virtuale, il Meta Quest 3 integra una serie di telecamere, giroscopi e sensori (descritti in dettaglio nella sezione specifiche) che consentono di delineare la composizione dello spazio fisico circostante in cui l'utente si trova. Prima di avviare la simulazione, l'utente è infatti sottoposto a una procedura di mappatura della stanza e dell'area di gioco, con l'obiettivo di individuare i principali oggetti di medie e grandi dimensioni che potrebbero costituire un ostacolo durante l'esperienza.

Il visore utilizza queste informazioni per generare confini virtuali, che, se superati, interrompono la simulazione. Questi dati sono inoltre accessibili agli sviluppatori, i quali possono sfruttarli per implementare funzionalità avanzate all'interno delle proprie applicazioni.

3.6 Specifiche

Di seguito sono riportate le specifiche tecniche principali del dispositivo Meta Quest 3:

Display: due schermi LCD da 2064 x 2208 pixel ciascuno, con una struttura simile a quella del Quest Pro. Campo visivo: 110° orizzontale e 96° verticale. Regolazione della messa a fuoco: dotato di una nuova rotella esterna per regolare la posizione delle lenti, al fine di trovare il punto di messa a fuoco ottimale. SoC: Qualcomm Snapdragon XR2 Gen 2. RAM: 8 GB. Archiviazione: disponibile con capacità di 128 o 512 GB. Autonomia: dichiarata simile a quella del Quest 2, con una durata variabile tra 90 e 180 minuti a seconda del tipo di attività. Fotocamere: doppia fotocamera RGB frontale che offre un miglioramento significativo nel passthrough video e consente l'uso del visore sia in modalità VR pura sia in modalità mista, sovrapponendo il mondo reale a quello olografico. Design: più compatto rispetto al Quest 2, ma leggermente più pesante (515 g contro 503 g). Controller: nuovi controller Touch Plus, dotati di tracciamento con sensori a infrarossi e miglior feedback aptico.

3.7 Funzionalità

Per facilitare lo sviluppo di applicazioni, Meta ha reso disponibile un proprio SDK (Software Development Kit) che consente di esplorare le funzionalità offerte dal visore. Questo strumento è stato fondamentale nella fase preliminare di studio per verificare se il dispositivo fosse in grado di soddisfare le specifiche esigenze del progetto.

Inoltre, Meta ha messo a disposizione alcuni asset preconfigurati per accelerare la creazione di applicazioni, sfruttando le principali funzionalità del visore. Questi asset, denominati Building Blocks, possono essere facilmente integrati installando l'All-in-One SDK direttamente dall'Asset Store di Unity.

Sebbene nel progetto applicativo non sia stato impiegato l'SDK ufficiale di Meta, in quanto ancora in una fase iniziale di sviluppo, esso si è rivelato utile per valutare le potenzialità del visore.

3.8 Meta Quest Developer Hub

Dopo la creazione dei primi prototipi dell'applicazione e i test iniziali all'interno dell'editor, si è resa necessaria una fase di testing direttamente sul visore per assicurarsi che le prestazioni fossero fluide. Inoltre, prevedendo una fase di testing con un gruppo di utenti, è stato ritenuto importante poter installare e avviare l'applicazione sul dispositivo in modo diretto.

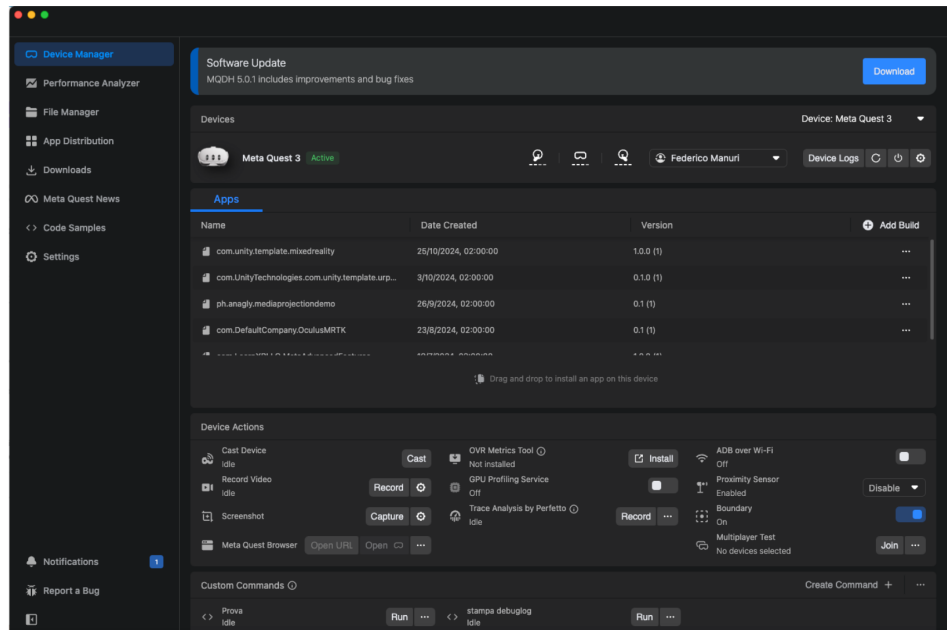


Figura 3.3. Interfaccia MetaQuest Developer Hub

A tal fine, Meta mette a disposizione un'applicazione chiamata MetaQuest Developer Hub, che consente di riconoscere e configurare un visore collegato a un account, nonché di caricare e gestire le build. In questo progetto, sono state utilizzate build incrementali di Unity per dispositivi Android, una scelta che permette di velocizzare il processo di compilazione. La build incrementale modifica solo le parti aggiornate del progetto, riducendo i tempi di creazione della build finale.

3.9 Oculus Link

La seconda applicazione utilizzata è Oculus Link, necessaria per abilitare la funzionalità di Virtual Desktop. Questa funzione consente di sviluppare in Unity direttamente all'interno del visore, sfruttando il sistema di input del dispositivo stesso anziché il simulatore. Questo approccio permette di ridurre significativamente i tempi di sviluppo e testing, offrendo un'esperienza di sviluppo più fluida e immediata.

3.10 Pacchetti aggiuntivi e librerie

- **AR Foundation:** Per implementare le funzionalità di riconoscimento degli oggetti, dei piani delle mura, del pavimento e dei soffitti, è stata utilizzata la libreria AR Foundation. Questo insieme di strumenti consente di sviluppare interazioni avanzate con il mondo reale. In particolare, è stata impiegata la versione 6.0.3, introdotta di recente insieme alla versione di Unity 6, poiché è l'unica in grado di supportare tutte le funzionalità di scene understanding integrate nel visore.
- **WordNet:** WordNet è un database lessicale per la lingua inglese, sviluppato per facilitare la ricerca e l'elaborazione del linguaggio naturale. Creato presso la Princeton University, WordNet organizza parole come sostantivi, verbi, aggettivi e avverbi in gruppi chiamati synset (insiemi di sinonimi), che riuniscono termini con significati affini. In questo progetto è stato utilizzato per generare descrizioni testuali delle vignette in modo chiaro e comprensibile.
- **Nuovo Sistema di Input:** Per interpretare correttamente gli input dei controller e i gesti delle mani, è stato utilizzato il nuovo sistema di Input di Unity. Questo sistema permette di creare una mappa di comandi, facilitando l'esecuzione delle azioni previste e migliorando l'esperienza di interazione dell'utente.
- **OpenXR e XR Core Utilities:** Il Unity XR Interaction Toolkit si basa fortemente su OpenXR, uno standard aperto per l'accesso a piattaforme e dispositivi di realtà virtuale (VR) e realtà aumentata (AR). Sviluppato dal gruppo Khronos, OpenXR mira a semplificare lo sviluppo di applicazioni di realtà estesa (XR), fornendo un'interfaccia unica e interoperabile tra dispositivi e piattaforme.
- **Asset Esterni e Animazione (Sketchfab e TurboSquid):** per ragioni di praticità, tutti gli asset grafici e le animazioni utilizzate sono stati scaricati da Sketchfab [17] e Mixamo [11]. Per quanto riguarda i pannelli dell'interfaccia utente (UI), sono stati utilizzati quelli presenti nelle scene d'esempio dell'Interaction Toolkit, resi disponibili gratuitamente per applicazioni non a scopo di lucro.

Capitolo 4

Implementazione

4.1 Comportamento generale della simulazione

L'applicazione è costituita essenzialmente da quattro moduli principali, ciascuno gestito da un menu di aiuto che segue l'utente durante la simulazione e che può essere opzionalmente nascosto se necessario.

La prima fase consiste nell'avvio dell'applicazione e nella familiarizzazione dell'utente con lo spazio sintetico. In questa fase, l'utente è immerso in un ambiente virtuale popolato da tutti gli oggetti rilevati durante la scansione iniziale dell'ambiente fisico. Nella seconda fase l'utente, aiutato da un tutorial che lo guida passo passo, ha la possibilità di visualizzare gli oggetti riconosciuti, modificare la posizione delle *bounding box* se risultano imprecise e correggere eventuali etichette assegnate in modo errato. Inoltre, è possibile definire un piano di interazione o di seduta che si attiva in base al tipo di oggetto in fase di configurazione. Ad esempio, qualora si stia configurando una sedia, la *bounding box* sarà dotata di un piano corrispondente all'altezza della seduta della sedia stessa. Durante questa fase, è inoltre possibile specificare l'orientamento del piano, permettendo di definire l'asse di direzione *forward*. Terminata la configurazione dello spazio, viene attivata la funzionalità di *passthrough*, che consente di passare dalla simulazione in realtà virtuale a quella in realtà aumentata. In questa modalità, l'utente può visualizzare sia gli oggetti reali sia le *bounding box* configurate. A questo punto compaiono tre menu di supporto: uno per generare nuovi oggetti nello spazio virtuale, uno per configurare le funzionalità di *storyboarding* e visualizzare gli *screenshot* effettuati, e infine un menu, accessibile alzando il pollice della mano, che consente di configurare le opzioni generali della simulazione. Durante questa fase, l'utente può posizionare nuovi oggetti o personaggi nello spazio virtuale prima di passare alla fase di *storyboarding*. L'ultima fase è dedicata allo *storyboarding* vero e proprio, durante la quale l'utente può selezionare personaggi, oggetti o *bounding box* introdotti nelle fasi precedenti per generare uno *storyboard*. L'utente può far eseguire azioni o interazioni agli elementi selezionati, generando descrizioni testuali che, una volta selezionata l'opzione *save storyboard*, vengono associate alle vignette nel *file* prodotto alla fine della simulazione.

Nei prossimi paragrafi verranno illustrati nel dettaglio il funzionamento dell'applicazione ed i principali script e funzioni responsabili del comportamento generale appena descritto.

4.2 Manipolazione degli oggetti

Prima di procedere con la descrizione dettagliata della struttura dell'applicazione, è opportuno chiarire come avviene la manipolazione degli oggetti nello spazio virtuale.

Il componente principale per la manipolazione degli oggetti, denominato “SnapToPlane,” eredita le funzionalità di base dal componente XR Grab Interactable appartenente al *framework Unity XR Interaction Toolkit*, estendendone le capacità per rispondere alle specifiche esigenze applicative. Il componente XR Grab Interactable (come illustrato in Figura 4.1) offre una serie di funzionalità essenziali per la logica dell'applicazione, elencate di seguito:

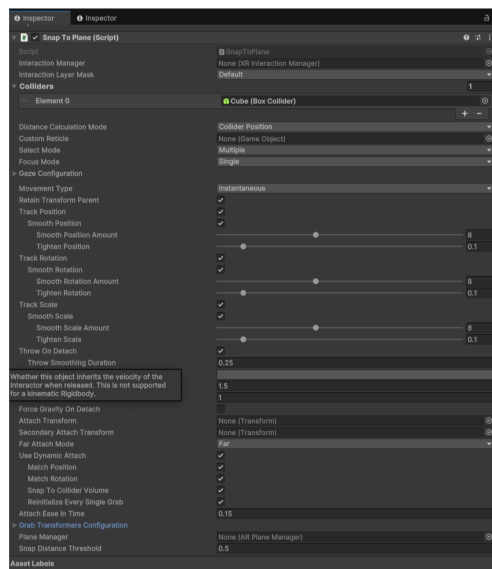


Figura 4.1. Campi del Componente SnapToPlane

- **Movement Types:** Definisce la modalità di movimento dell'oggetto afferrato, con le seguenti opzioni:
- **Kinematic:** l'oggetto si sposta direttamente alla posizione e orientamento del controller, mantenendo una dinamica kinematica.
- **Velocity Tracking:** l'oggetto si muove utilizzando velocità e accelerazioni, per una simulazione più realistica del movimento.

- **Instantaneous:** l'oggetto viene teletrasportato alla posizione e rotazione del controller senza interpolazione.
- **Attach Transform - Attach Points:** Consente di specificare un punto di attacco personalizzato, per controllare dove l'oggetto si collega alla mano o al controller, includendo posizione e rotazione.
- **Throwing Physics - Throw on Detach:** Permette di lanciare l'oggetto quando viene rilasciato, usando velocità e rotazione del controller per determinare direzione e forza del lancio.
- **Throw Velocity Scale e Throw Angular Velocity Scale:** consentono di regolare la velocità e la rotazione per ottenere effetti di lancio personalizzati e realistici.
- **Snap Behavior:** attacca automaticamente l'oggetto al controller o la mano che lo ha afferrato.
- **Smooth Position and Rotation:** opzioni per interpolare posizione e rotazione dell'oggetto, garantendo una transizione fluida tra posizione iniziale e di attacco.
- **Snap Threshold Distance e Snap Threshold Angle:** stabiliscono soglie per lo snapping immediato o graduale di posizione e rotazione.
- **Multiple Attachments:** Supporta interazioni multiple, consentendo a più utenti di afferrare lo stesso oggetto in contesti multi-user.
- **Secondary Interactors:** Permette l'uso di un secondo interattore per manipolare l'oggetto, utile per una presa a due mani o modifiche simultanee da più fonti.
- **Movement Constraints:** limita l'oggetto a specifici assi di movimento o rotazione, per funzioni come manopole o slider.
- **Rotation Constraints:** specifica limiti di rotazione, evitando rotazioni indesiderate.
- **Force Grab:** Consente di afferrare l'oggetto a distanza, tramite raycast o altri metodi di interazione a distanza.
- **Interaction Events:**
- **On Select Entered/Exited:** eventi attivabili quando l'oggetto viene afferrato o rilasciato.
- **On Activated/Deactivated:** eventi per interazioni avanzate, come attivazioni momentanee o azioni di rilascio.
- **Hover Events:** attiva eventi quando un interattore è nelle vicinanze, evidenziando visivamente l'oggetto.
- **Layer Mask:** specifica i layer su cui l'oggetto può essere afferrato, controllando chi può interagire con chi.

- Interaction Layer Mask: limita l'interazione a determinati tipi di interattori, utile per interazioni diversificate.
- Auto Drop e Detach Behavior: Opzioni per rilasciare automaticamente l'oggetto se l'interattore si allontana troppo o se non vengono soddisfatte condizioni specifiche.

Queste funzionalità offrono un ampio controllo sul comportamento e sulla fisica degli oggetti afferrabili, facilitando la creazione di interazioni naturali e reattive in contesti AR/VR.

Il componente `SnapToPlane` eredita tutte le funzionalità sopra descritte e aggiunge una caratteristica: la funzione `SnapToClosestPlane`, progettata per agevolare il controllo dell'oggetto nello spazio virtuale. La funzione viene richiamata ogni volta che, nel ciclo di `Update`, viene rilevato un piano vicino all'oggetto selezionato. Come illustrato nel codice, quando l'oggetto viene rilasciato questo viene allineato in modo tale che la normale sia coerente con il piano a lui più vicino e che la posizione non compenetri con il piano di riferimento.

Listing 4.1. Example C# Code

```
private void SnapToClosestPlane()
{
    if (closestPlane == null)
        return;
    Vector3 planePosition = closestPlane.transform.position;
    Vector3 planeNormal = closestPlane.transform.up;
    Vector3 objectPosition = transform.position;
    if (Vector3.Dot(planeNormal, Vector3.down) > 0.5f && transform.name == "Chandelier")
    {
        planeNormal = -planeNormal; // Invertiamo la normale
    }
    float distance = Vector3.Dot(planeNormal, objectPosition - planePosition);
    transform.position = objectPosition - distance * planeNormal;

    // Ci assicuriamo che l'oggetto sia allineato
    transform.rotation = Quaternion.LookRotation(transform.forward, planeNormal);

    if (Mathf.Abs(distance) < snapDistanceThreshold)
    {
        Debug.Log("Oggetto" + this.name + " aggranciato al piano.");
        isSnapping = false; // Impostiamo snapping su false per continuare a manipolare l'oggetto
    }
}
```

Infine, per una manipolazione corretta, ogni oggetto deve includere un `Rigidbody`, che viene automaticamente aggiunto quando si associa lo script all'oggetto, e deve inoltre disporre di un `Collider`.

4.3 configurazione dell'utente

I componenti descritti nel paragrafo precedente costituiscono il primo blocco essenziale per l'interazione, e vengono denominati *interactables* (tradotto come "oggetti interagibili"). Per consentire la manipolazione degli oggetti, è necessario definire anche un *Interactor* (o interattore). Nel nostro caso, l'interattore è rappresentato dall'utente stesso, identificato attraverso un oggetto denominato *XR Origin* (illustrato in Figura 4.2), che è composto da una serie di elementi vuoti (*empty*) e da script associati. Questi elementi consentono di rilevare e gestire correttamente tutti gli input provenienti dal controller o dalle mani dell'utente, configurando al contempo la telecamera per operare in modalità passthrough o virtual reality.

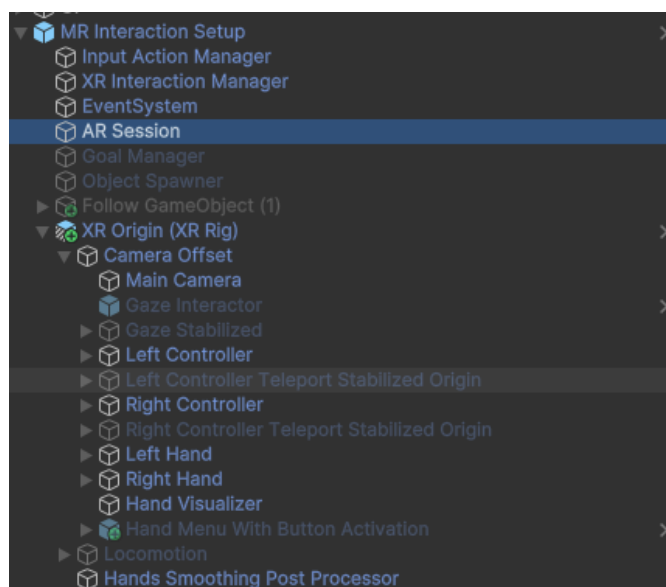


Figura 4.2. Gerarchia del componente xr origin

La configurazione dell'*XR Origin* avviene specificando una serie di gesti e azioni riconosciute (queste ultime dichiarate nel sistema di *input* di *Unity*, come già illustrato). Inoltre, è possibile definire dei *layer* di interazione per limitare l'effetto di determinate azioni a specifici oggetti di scena, come, ad esempio, l'interfaccia utente.

4.4 Gestione delle interazioni all'interno della simulazione

Sebbene lo script per la manipolazione degli oggetti sia unificato, si è reso necessario gestire effetti diversi derivanti dall'interazione a seconda del tipo di entità con cui andiamo ad interagire e dalla fase in cui si trova la simulazione. Un singolo oggetto può inizialmente essere posizionato nello spazio, richiedendo quindi traslazioni e rotazioni; successivamente,

in fase di storyboarding, potrebbe essere selezionato per interagire con ulteriori funzionalità. Inoltre, per le bounding box è prevista anche la possibilità di selezione e modificare l'etichetta dell'oggetto, oltre alle altre opzioni. Come si possono gestire, dunque, le differenti interazioni per i vari tipi di oggetti? Il componente "SnapToPlane" descritto nel paragrafo precedente permette, ogni volta che un'oggetto viene selezionato, di eseguire a sua volta una o più funzioni. Grazie a questa strategia, ad ogni tipologia di oggetto è stato assegnato uno *script* specifico per la gestione dell'interazione. Questi *script* contengono una serie di funzioni che permettono di differenziare il risultato della selezione a seconda dell'oggetto scelto. Sono stati sviluppati due *script* principali a questo proposito: *BoundingBoxInteractionManager* e *InteractionManagerAddOn*

Entrambi gli script condividono una logica basata su eventi o variabili booleane, che consente di identificare lo stato corrente della simulazione e di adattare il comportamento dell'oggetto in modo appropriato. Gli eventi in questione sono quelli di *onSelectionEnter* e *onSelectionExit* che gestiscono il comportamento di un oggetto quando viene preso o rilasciato.

Nella fase di storyboarding, ad esempio, le logiche di spostamento del componente SnapToPlane vengono disattivate per consentire la selezione dell'oggetto senza rischiare movimenti involontari.

Il *BoundingBoxInteractionManager* gestisce specificamente la *bounding box*, permettendo di selezionare la label o spostarla nello spazio se la sua posizione non è corretta.

Listing 4.2. Example C# Code

```
public void onSelectionEnter(SelectEnterEventArgs args)
{
    _currSelectedObj = args.interactableObject.transform.gameObject;
    if (_currSelectedObj == null)
    {
        Debug.LogError("On_Selection_Enter_" + this.name + "null_referen");
        return;
    }
    if (interactionEnabled)
    {
        onObjectSelected.Invoke(this, EventArgs.Empty);
        _SimulationManager.SetActiveCharacter(_currSelectedObj);
    }
}
```

L'*InteractionManagerAddOn*, invece, è uno script più complesso: oltre a gestire il movimento dell'oggetto, implementa le funzionalità necessarie per il corretto svolgimento della logica di storyboarding. Questo script è collegato a tutti gli oggetti di scena digitali (personaggi e oggetti) che l'utente può posizionare. Durante la fase di storyboarding, la selezione di un oggetto attiva la funzione *OnSelectionEnter*, che a sua volta richiama la funzione *SetActiveCharacter* del componente *SimulationManager* (descritta nei prossimi paragrafi). Questa funzione identifica se l'elemento selezionato è un personaggio o un

oggetto di scena, consentendo così un'interazione mirata e coerente con il contesto della simulazione.

4.5 Utilizzo delle funzionalità di ancoraggio

Sviluppare un'applicazione in realtà aumentata implica l'integrazione di oggetti virtuali in un ambiente reale. Questa caratteristica comporta una criticità fondamentale: gli oggetti sintetici devono essere posizionati correttamente non solo rispetto alla simulazione ma anche, e soprattutto, rispetto al mondo reale in cui sono immersi. A differenza degli ambienti virtuali, il mondo reale manca di informazioni strutturate necessarie per gestire coerentemente il posizionamento degli oggetti virtuali. Durante lo sviluppo dell'applicazione, è emerso un problema legato alla deriva dei sensori del visore, che causava cambiamenti improvvisi nel sistema di coordinate relativo della simulazione. Questo fenomeno portava allo spostamento indesiderato degli oggetti virtuali, spesso posizionandoli in luoghi incoerenti o persino fuori dal campo visivo dell'utente, compromettendo significativamente l'esperienza. Per risolvere questa criticità, è stato adottato l'uso delle ancore, strumenti virtuali utilizzati per fissare o stabilizzare gli oggetti in punti specifici dello spazio. In Unity, un'ancora è implementata attraverso un oggetto della classe "Anchor", che definisce una posa per legare una posizione nello spazio fisico con il mondo virtuale. La gestione delle ancore nell'applicazione è stata organizzata attraverso una struttura a manager. In particolare, il sistema include il `CharacterAnchorManager`, responsabile della richiesta di associazione tra un personaggio/oggetto digitale ed un'ancora, e lo `SceneAnchorManager`, che coordina la loro creazione e associazione. Ogni personaggio o oggetto mobile nella scena è associato a uno script `CharacterAnchorManager`, che memorizza l'ancora dell'oggetto e interagisce con lo `SceneAnchorManager`. Quando un oggetto viene istanziato, lo script `CharacterAnchorManager` utilizza la funzione "AttachObjectToAnchor" per richiedere un'ancora. Lo `SceneAnchorManager` cerca un'ancora esistente sufficientemente vicina e, se disponibile, la assegna; altrimenti, ne crea una nuova. Questo approccio garantisce un utilizzo efficiente delle risorse, mantenendo un numero minimo di ancore per evitare conflitti, pur controllando la distanza massima tra di esse. Quando un oggetto viene spostato, si sgancia dall'ancora associata, viene posizionato liberamente nello spazio e, al momento del rilascio, si associa a una nuova ancora in base alla sua posizione finale. Lo `SceneAnchorManager` monitora costantemente le ancore e rimuove quelle inutilizzate per ottimizzare le risorse. Questo sistema garantisce la stabilità spaziale degli oggetti, migliorando la coerenza tra il mondo virtuale e reale e ottimizzando l'esperienza utente, superando così una delle principali criticità dello sviluppo di applicazioni in realtà aumentata.

4.6 Gestione del piano di seduta e piazzamento oggetti

Per implementare queste funzionalità, è stato scelto un sistema di *raycasting*, al fine di evitare l'uso di pulsanti fisici, che potrebbero risultare poco intuitivi per i nuovi utenti, e

per fornire un *feedback* visivo sull'oggetto con cui si sta interagendo.

A tal fine, è stato sviluppato lo script *Controller Manager*, il quale, in base agli eventi generati durante la simulazione, attiva o disattiva la funzionalità di *raycasting*. Il sistema genera dei raggi che partono dal visore: questi raggi si colorano di verde se colpiscono un oggetto o una superficie *interactable*, mentre mantengono un colore rosso in caso contrario.

Questa modalità viene utilizzata per generare oggetti che vengono istanziati esattamente nel punto di intersezione tra il raggio e la superficie. Inoltre, la stessa modalità di *raycasting* è impiegata per regolare l'altezza del piano di seduta/interazione all'interno delle bounding box. In questo caso specifico, l'utente tiene premuto il pulsante di Grip sul controller per regolare l'altezza.

4.7 Animazione dei personaggi

Come verrà approfondito nei paragrafi successivi, la gestione delle animazioni dei personaggi è affidata agli script *Simulation Manager* e *CharacterAnimationManager*. Questi script sono incaricati di attivare i cambi di stato e di gestire esternamente le azioni dei personaggi. Ciascun personaggio, a livello interno, è dotato di due script principali: *CharacterManager* e *State*.

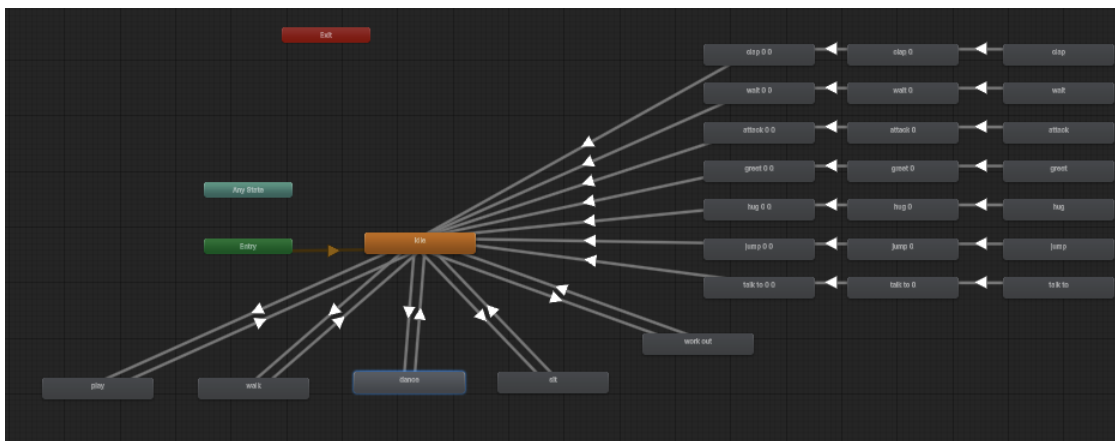


Figura 4.3. Albero delle animazioni di un personaggio

Lo script *State* traccia costantemente lo stato attuale del personaggio, mentre il *CharacterManager* attiva i cambi di stato necessari per eseguire le animazioni. Ogni personaggio è supportato da un albero di animazioni che si aggiorna in tempo reale, consentendo una gestione dinamica delle transizioni di stato.

Le informazioni sulle transizioni tra i vari possibili stati sono contenute in un file di configurazione, che il *Simulation Manager* e il *CharacterAnimationManager* leggono e interpretano per determinare le azioni disponibili per ciascun personaggio in un determinato

momento. Quando un’azione viene selezionata, il sistema raccoglie l’insieme di transizioni che costituiscono l’animazione e la riproduce.

È inoltre importante notare che ogni animazione può essere ripetuta più volte per consentire all’utente di catturare agevolmente il momento desiderato.

4.8 Scansione dell’ambiente automatica effettuata all’avvio dell’applicazione e configurazione delle bounding box

Per il riconoscimento, il calcolo e il posizionamento delle bounding box relative agli oggetti reali è stato utilizzato il componente *ARBoundingBoxManager*, messo a disposizione dalla libreria *AR Foundation*. Questo componente sfrutta le informazioni spaziali acquisite nella fase iniziale per generare una lista di *bounding box*, ciascuna caratterizzata da posa, posizione e scala. Le informazioni contenute in questa lista vengono costantemente aggiornate e modificate nel corso della simulazione, garantendo una rappresentazione coerente con la realtà. Per inserire le bounding box nella scena, è stato creato uno script denominato *BoundingBoxManager*.

Lo script *BoundingBoxManager* si riferisce all’*ARBoundingBoxManager* e utilizza le informazioni fornite da quest’ultimo per generare una serie di bounding box. Inoltre, mantiene costantemente aggiornate le bounding box per tutta la durata della simulazione, assicurando una coerenza visiva e funzionale.

Le *bounding box* sono state inizialmente realizzate a partire da un cubo di dimensioni 1x1x1 metri con scala unitaria. Sarà poi il *BoundingBoxManager* a calcolare e applicare le unità di misura corrette in base alle specifiche esigenze della simulazione. Ogni *bounding box* è dotata di un materiale trasparente per evitare interferenze visive con gli oggetti che contiene. Inoltre, ciascuna *bounding box* include un piano di seduta o interazione che può essere abilitato quando si assegna un’etichetta (label) che ne giustifichi l’uso. A seconda della label scelta è possibile configurare un piano che sarà usato per l’interazione. Ad esempio se si sceglie di etichettare la bounding box come una sedia, il piano in questione sarà utilizzato per inferire l’altezza della sedia. il piano è poi dotato di una freccia che ne indica la direzione per permettere di determinare un verso di seduta coerente.

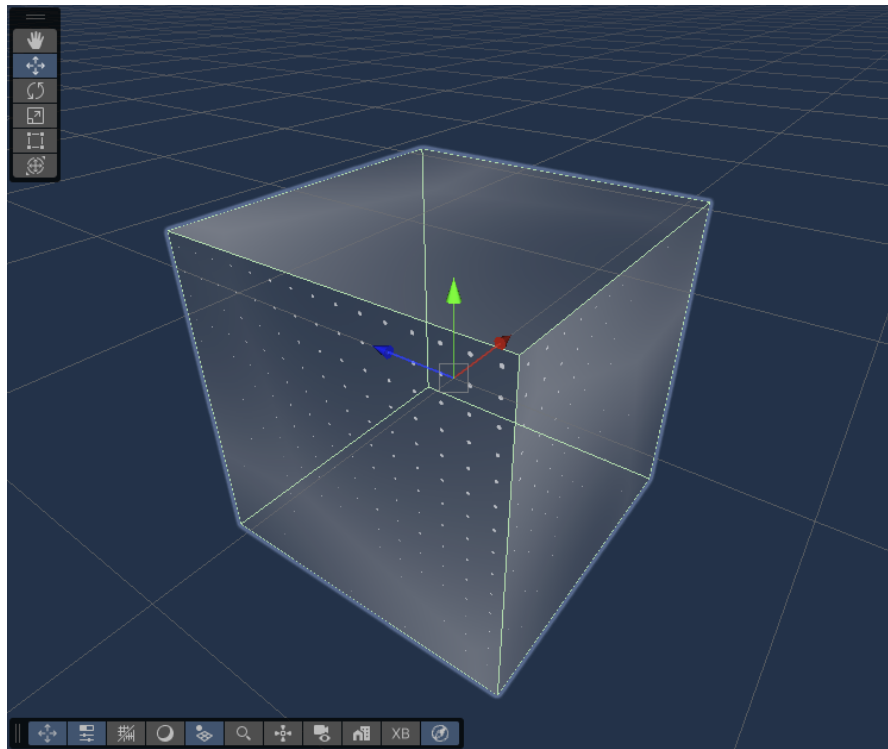


Figura 4.4. Esempio di Bounding box generica

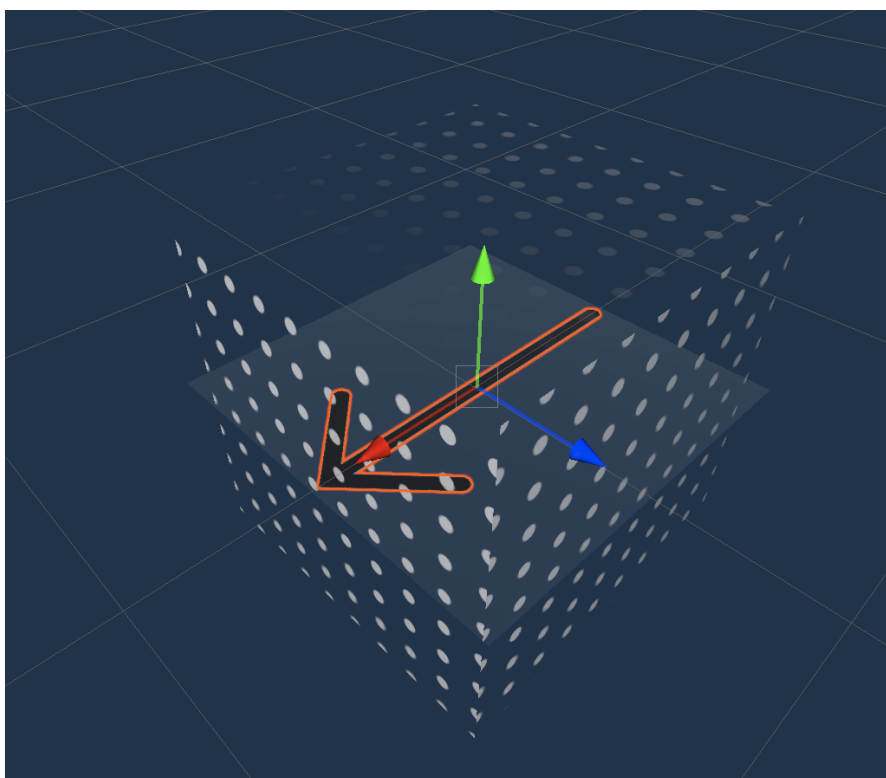


Figura 4.5. Bounding box con piano di seduta/interazione abilitato

4.9 Simulation manager

L'applicazione, nel suo complesso, è gestita da una serie di script che si occupano di specifiche aree funzionali, coordinando in modo efficiente le diverse attività. Questa struttura di gestione è stata ereditata dalla precedente impostazione del progetto e si basa sull'uso di diversi "Manager": il *Simulation Manager*, il *CharacterAnimationManager*, lo *ScreenshotManager* e, infine, l'*OutputGenerator*. Nei paragrafi successivi, verranno descritte in dettaglio le funzionalità di ciascun *Manager*. In questa sezione, ci concentreremo sul *Simulation Manager*, che rappresenta il fulcro del coordinamento delle attività dell'intero software applicativo.

Inizialmente, la simulazione era gestita tramite variabili booleane, le quali venivano modificate da varie funzioni in base alle azioni dell'utente. Sebbene questo approccio fosse corretto, presentava alcune problematiche, tra cui la difficoltà nel tenere traccia delle modifiche apportate alle variabili stesse. Inoltre, dato il miglioramento delle interazioni con gli oggetti, è stato considerato più efficiente adottare una gestione dello stato del sistema tramite l'introduzione di eventi. A tal fine, sono stati definiti diversi eventi, come ad esempio:

Listing 4.3. Example C# Code

```
public static EventHandler<EventArgs> startStoryboarding;  
public static EventHandler<EventArgs> pauseStoryboarding;  
public static EventHandler<EventArgs> setUpMovement;  
public static EventHandler<EventArgs> stopAnimation;
```

Questi eventi vengono utilizzati per gestire la simulazione e sono raccolti dagli altri sistemi, che adattano di conseguenza il loro comportamento.

Oltre alla gestione degli eventi, il *Simulation Manager* si occupa di mantenere traccia del personaggio attualmente selezionato e di richiamare le funzioni necessarie per gestire il *CharacterAnimationManager*, ossia l'interazione tra il personaggio principale e gli eventuali oggetti interagibili presenti nella scena.

In aggiunta alla gestione della simulazione, il *Simulation Manager* contiene la logica necessaria per implementare la funzionalità di "Undo", ovvero la possibilità per l'utente di correggere la descrizione dello storyboard nel caso in cui si accorga di un errore. La funzione associata a questa funzionalità è denominata *UndoLastAction*. Ogni volta che l'utente effettua una cattura per salvare una vignetta, le informazioni relative alla posizione, alla rotazione e allo stato degli oggetti presenti nella scena vengono salvate in un file in formato .csv. Queste informazioni vengono poi utilizzate per ripristinare lo stato precedente degli oggetti nel caso in cui venga invocata la funzione *UndoLastAction*, riportando la scena alle condizioni memorizzate e rimuovendo l'ultima riga del buffer contenente la descrizione testuale.

Infine, il *Simulation Manager* include una serie di funzioni che si occupano di gestire il numero di screenshot effettuati, il tempo di cattura e l'interruzione delle animazioni.

4.10 Character animation manager

Il *CharacterAnimationManager* svolge un ruolo fondamentale nell'interazione tra il personaggio attivo e gli oggetti presenti nella scena, interfacciandosi direttamente con il *Simulation Manager*. Quando un utente seleziona un personaggio cliccando su di esso, il sistema lo imposta come personaggio attivo. Successivamente, selezionando un oggetto nella scena, è possibile programmare il personaggio per eseguire una serie di azioni specifiche associate all'oggetto selezionato.

Le possibili animazioni e azioni che i personaggi possono eseguire sono definite attraverso due file di configurazione testuali. Il primo file, denominato "states.txt", contiene informazioni relative agli stati del personaggio, specificando le azioni eseguibili in ogni stato e le transizioni di stato che queste azioni comportano. Il secondo file, denominato "action.csv", è memorizzato nella directory condivisa "StreamingAssets" e descrive in dettaglio tutte le azioni che un personaggio può eseguire, sia autonomamente sia interagendo con altri oggetti. Il formato adottato per popolare questo file è strutturato nel seguente modo: "nome personaggio"- "nome oggetto"- "azione1", "azione2", e così via.

Per garantire il corretto funzionamento del sistema, è necessario popolare i file con i nomi dei personaggi e le relative azioni, quindi configurare accuratamente il componente *CharacterManager*. In particolare, il tipo di *CharacterManager* deve essere definito in modo coerente con il nome del personaggio, assicurando così la corretta associazione tra personaggio, azioni e oggetti presenti nella scena. Questo sistema garantisce flessibilità

e scalabilità nella gestione delle animazioni, permettendo di definire con precisione le interazioni tra personaggi e oggetti in scenari complessi.

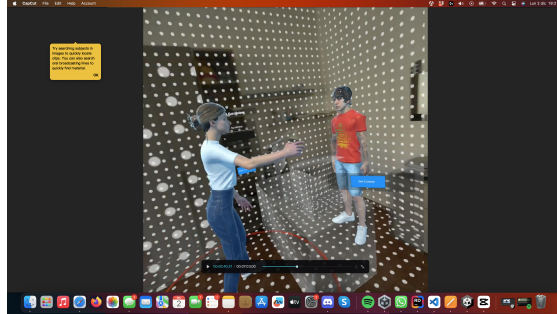


Figura 4.6. Pannello di preview del frame dello storyboard

4.11 Descrizione testuale delle azioni

Lo script *Phrase Generator* fornisce una serie di funzioni per la scrittura su un buffer di stringhe, il quale verrà successivamente utilizzato in fase di salvataggio dello storyboard per generare la descrizione di ciascun frame. La funzione principale di questo script è *GenerateSimplePhrase*, che viene richiamata ogni volta che viene selezionata un'azione e si occupa di recuperare le informazioni rilevanti per comporre una descrizione testuale appropriata.

Come descritto nei capitoli precedenti, per elaborare una descrizione chiara e variata, è stato utilizzato il database lessicale WordNet [15], sviluppato e messo a disposizione dall'Università di Princeton. Questo database consente di generare una serie di sinonimi per creare frasi semanticamente corrette, ma leggermente differenti tra loro, garantendo così varietà nella descrizione delle azioni.

4.12 Funzionalità di cattura foto

Per implementare la funzionalità di cattura foto per realizzare le vignette dello storyboard, è stato necessario affrontare il problema legato alla limitata accessibilità, da parte di Unity, ai dati delle catture video, poiché Meta Quest (che adotta come sistema operativo Android) non consente l'accesso diretto. Per superare questa limitazione, è stata effettuata una chiamata diretta alle API di Android, il che ha permesso di accedere alla funzione di videocamera ad un livello più basso. Questo aspetto è gestito dallo script *QuestScreenCaptureTextureManager*.

Lo script *ScreenshotManager* richiama la funzione *TakeScreenShot()* di *QuestScreenCaptureTextureManager* e visualizza gli screenshot nel pannello UI dedicato. Le immagini vengono salvate in una lista di componenti *RawImage* e *Texture2D*, per semplificare la fase successiva di esportazione. Il numero massimo di screenshot è limitato dal numero di immagini vuote presenti nel pannello di visualizzazione.

4.13 Generazione storyboarding

La generazione dello storyboard è gestita dallo script OutputGenerator, che raccoglie tutti gli screenshot catturati insieme alle descrizioni allegate per creare un file HTML consultabile e modificabile, qualora fossero necessarie migliorie. Inoltre, viene generata una cartella dedicata per archiviare e visualizzare gli screenshot.

Per facilitare l'accesso a questi file, è stata installata l'applicazione File Media Explorer CX, che consente una visibilità completa dello storage su dispositivi Android.

4.14 Interfaccia utente

Per garantire un'esperienza utente intuitiva e funzionale, l'interfaccia dell'applicazione è stata progettata in modo da semplificare l'apprendimento e rendere più rapide ed efficienti le operazioni disponibili. Le informazioni necessarie durante la fase di storyboarding hanno guidato la progettazione dei pannelli personalizzati, strumenti attraverso cui l'utente può monitorare lo stato dello storyboard e selezionare azioni specifiche durante la simulazione.

Un elemento comune a tutti i pannelli è la possibilità di nascondarli in qualsiasi momento, permettendo all'utente di ridurre l'ingombro visivo e mantenere la scena ordinata.



Figura 4.7. menu istanziazione dei personaggi

- Pannello per l'istanziamento dei personaggi e la consultazione dello script: Questo pannello consente di accedere ai due script utilizzati come base per la progettazione

dello storyboard. L'utente può leggere il contenuto dello script in corso. È anche possibile utilizzare un menu laterale per selezionare gli oggetti necessari alla rappresentazione dello script e alla costruzione dello storyboard. Questo approccio garantisce una transizione fluida tra la lettura dello script e la selezione degli elementi per la scena.

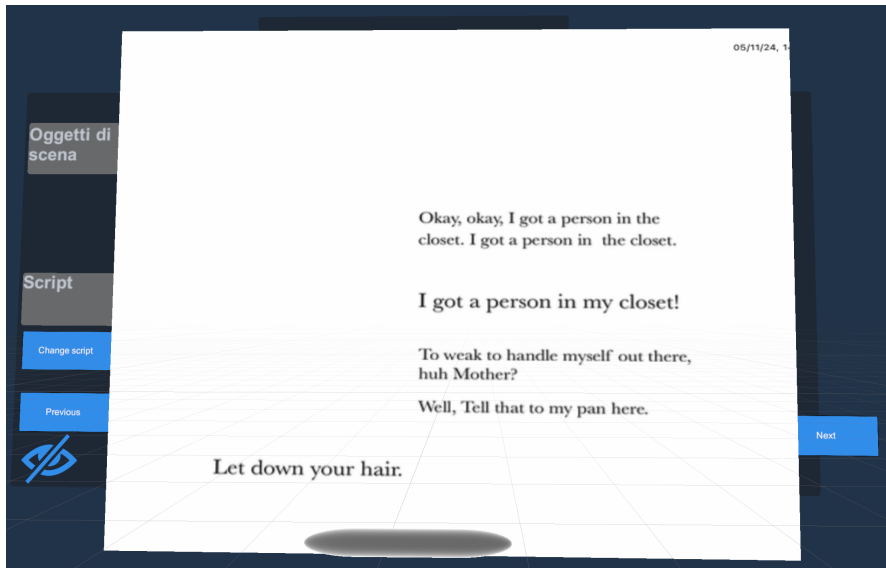


Figura 4.8. Menu consultazione dello script

- Pannello "Frame Preview": Questo pannello gestisce le informazioni relative alla fase di storyboarding e fornisce una preview dello screenshot che verrà generato, mostrando le azioni completate fino a quel momento. Include diverse funzionalità tra cui:
 1. Bottone di azione contemporanea: permette di gestire azioni svolte in parallelo.
 2. Slider temporale: regola la durata delle singole azioni, informazioni che verranno salvate nel file di output.
 3. Take Screenshot: l'utente può scattare uno screenshot tramite pulsante o direttamente utilizzando il controller.

Undo: consente di ripristinare l'ultima operazione eseguita.

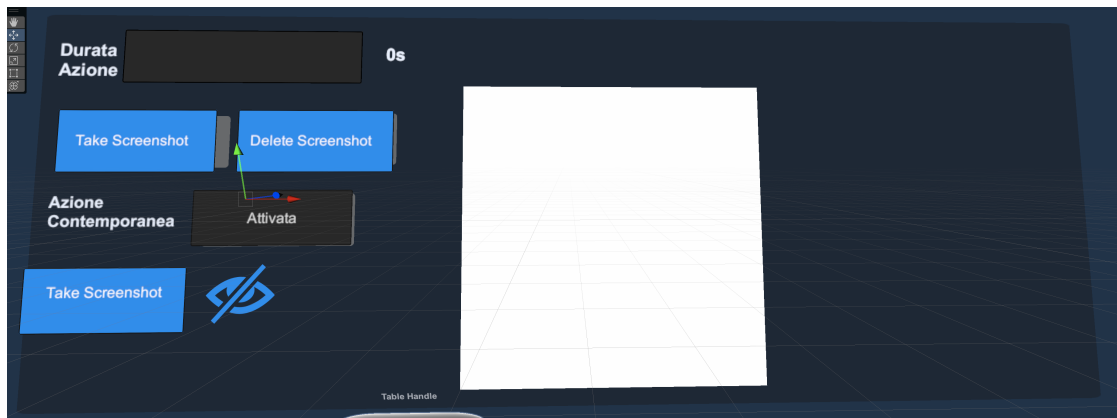


Figura 4.9. Pannello di preview del frame dello storyboard

Questa combinazione di strumenti facilita una gestione precisa e personalizzata delle sequenze di storyboarding.

- Pannello di collezione degli screenshot: questo pannello consente di visualizzare gli screenshot già scattati, offrendo all'utente un riferimento immediato per verificare lo stato di avanzamento del lavoro. È particolarmente utile nel caso in cui ci si dimentichi il punto esatto dell'esecuzione dello script.



Figura 4.10. Pannello collezione degli screenshot

- Pannello impostazioni generali Questo pannello è accessibile esclusivamente osservando il palmo della mano dell'utente, e racchiude le impostazioni di carattere generale:
 1. pulsante per mostrare il tutorial: per consultare i passaggi poco chiari o revisionare i comandi.
 2. Salvataggio dello storyboard: genera un file di output in formato HTML.
 3. Visualizzazione dei pannelli nascosti: permette di riattivare pannelli temporaneamente rimossi dalla scena.
 4. Avvio della fase di storyboarding: attiva l'animazione dei personaggi.
 5. Modalità passthrough: consente di passare dalla realtà sintetica alla realtà mista.



Figura 4.11. Pannello di preview del frame dello storyboard

L'integrazione di queste impostazioni rende l'applicazione versatile e adatta a esigenze diverse durante l'utilizzo.

4.15 Funzioni di accessibilità

L'applicazione offre diverse funzionalità progettate per semplificare la creazione di storyboard. Per garantire che l'utente possa apprendere il funzionamento senza essere sovraccaricato di informazioni, si è deciso di implementare un tutorial interattivo che lo guida

passo passo attraverso l'utilizzo dell'applicazione. Questo tutorial è stato concepito per essere consultabile in qualsiasi momento, in modo da fornire supporto costante durante le varie fasi di configurazione dell'ambiente e di etichettatura delle bounding box. Al fine di migliorare ulteriormente la facilità d'uso, l'applicazione è stata sottoposta, nell'ultima fase dello sviluppo, a un processo di affinamento iterativo mirato a rendere il tutorial più intuitivo e accessibile. Tra le migliori implementate vi è l'introduzione di un'opzione per selezionare la mano dominante dell'utente, la possibilità di navigare il tutorial sia in avanti che all'indietro, e l'aggiunta di una schermata dei controlli, sempre consultabile, per un accesso rapido e immediato alle informazioni principali. In aggiunta sono state inseriti dei bottoni per nascondere i pannelli. Questo approccio ha permesso di ottimizzare l'esperienza d'uso, rendendo l'applicazione più user-friendly e adatta a un ampio spettro di utilizzatori.

Capitolo 5

Test

Per effettuare una valutazione il più oggettiva possibile, si è deciso di testare il comportamento dell'applicazione in uno scenario il più realistico possibile. A tale scopo, sono stati selezionati due script tratti da una raccolta di serie televisive o film di successo dell'ultimo decennio. Da questi script è stato estratto un passaggio significativo, scelto in modo da poter essere utilizzato per verificare l'efficienza e le funzionalità dell'applicazione.

Una volta selezionato lo script, sono stati inseriti i personaggi e gli oggetti necessari alla realizzazione di uno storyboard accurato. Successivamente, sono state definite tutte le azioni e le animazioni che i personaggi devono essere in grado di compiere per rappresentare lo script scelto. Questo approccio ha permesso di testare non solo l'affidabilità dell'applicazione, ma anche la sua capacità di gestire scene complesse e articolate, riproducendo fedelmente le dinamiche previste dallo storyboard.

5.1 Casi di studio

- Il primo caso di studio è tratto dal film d'animazione Rapunzel - L'intreccio della torre (2010). La scena scelta è l'incontro tra i due protagonisti, Rapunzel e Flynn Rider. Questo segmento si è rivelato particolarmente adatto poiché contiene una varietà di azioni che coinvolgono oggetti facilmente reperibili nello spazio circostante e non è appesantito da dialoghi complessi, rendendolo ideale per la rappresentazione attraverso uno storyboard. Per questa scena sono stati inseriti i seguenti elementi: Oggetti utilizzati nella sequenza (es. padella, specchio). Azioni principali che i personaggi devono eseguire. Animazioni necessarie a rappresentare le interazioni.

Rapunzel Script:
F: [enters tower, pants.] Alone at last.
[hit by frying pan]
R: Agh! [fearfully hides, approaches. F: no response]
[Pascal motions head]
[R moves F's head.]
[Pascal turns brown at pointy-teeth drawing]
[R finds F's teeth straight.]
R: Oh. [Flicks F's hair, uncovering his face. Approaches.]
F: Huh?
[R hits F with frying pan. R pants.]

[After some unsuccessful attempts, Rapunzel eventually hides Flynn in closet]
R: Okay, okay, okay, I've got a person in my closet.
I've got a person in my closet.
I've got a person in my closet! Haha! Uh!
Too weak to handle myself out there, huh, mother? Well... hmm-hmm...tell that to my frying
pan-[accidentally hits herself with pan]
[discovers satchel & crown] Huh?
[R takes out crown. R hangs on forearm, Pascal shakes head]
R: [deep breath] Okay.
[opens closet]
Uh?
[examines F]
Hmm.
[R pulls chair and wakes up F.]
F: Agh! Huh? Uh? [struggles] Is this hair?
R: Struggling... struggling is pointless.
F: Huh?
R: I know why you're here, and I'm not afraid of you.

Figura 5.1. Estratto dello script del film Rapunzel

- Il secondo caso di studio si basa su una scena della serie televisiva Mercoledì, prodotta da Netflix e diretta da Tim Burton. È stata selezionata una scena particolarmente dinamica, in cui la protagonista interagisce con diversi oggetti comuni, come una macchina del caffè e delle sedie, mostrando la capacità dell'applicazione di rappresentare azioni dettagliate. In questa scena, sono stati creati digitalmente: I Personaggi per impersonare gli attori coinvolti, oggetti utilizzati nella scena, sequenze animate per rappresentare la riparazione della macchina del caffè da parte di Mercoledì e altre interazioni.

Wednesday script:

- INT. WEATHERVANE CAFE & BAKERY - JERICHO - DAY 35
A vintage industrial ESPRESSO MACHINE GROANS, SHOOTING
STEAM from various orifices. TYLER GALPIN (16) is behind the
counter, futilely trying to tame the mechanical beast. He's the kind
of kid who's constantly overwhelmed by his underwhelming life.
- TYLER
Holy crap! How long have you
been...? Do you make a habit of scaring the hell out of people?
WEDNESDAY It's more of a hobby.
TYLER (re: uniform)
You go to Nevermore? Didn't realize they'd changed up the
uniform.
WEDNESDAY (ignoring his question)
I need a quad over ice. It's an
emergency. (MORE)

WEDNESDAY - #101 - FULL BLUE - 08/31/2021
WEDNESDAY (CONT'D) (off his look)
That's four shots of espresso.
TYLER
I know what a quad is. But --
spoiler alert -- the espresso machine's having a seizure. (nodding to
coffee pot)
So all we have is drip.
WEDNESDAY
What's wrong with your machine?

Figura 5.2. Estratto dello script del film Rapunzel

5.2 Struttura dei test

Il test è stato progettato seguendo una struttura articolata in diverse fasi, al fine di garantire un'analisi completa delle funzionalità e dell'usabilità dell'applicazione. Le fasi principali sono descritte di seguito:

1. Selezione e Preparazione dell'Ambiente. In primo luogo, è stato scelto un ambiente adatto, dotato delle seguenti caratteristiche: Spazio sufficiente per ospitare sia oggetti fisici rilevanti per i casi di studio selezionati, sia elementi digitali creati all'interno dell'applicazione. Assenza di problemi di occlusione e condizioni di illuminazione adeguate, per evitare interferenze durante l'utilizzo dell'applicazione. Successivamente, è stata eseguita una scansione accurata dell'ambiente, operazione necessaria per configurare il sistema prima di avviare l'applicazione. Questa fase di setup assicura che l'ambiente sia pronto per il test senza ostacoli tecnici.
2. Fase Preparatoria: Avvio e Familiarizzazione Dopo il setup iniziale, l'utente avvia un'applicazione di supporto progettata per agevolare la comprensione delle interazioni di base. Questo strumento di apprendimento ha una durata indicativa di circa cinque minuti e consente all'utente di acquisire familiarità con le funzionalità principali del visore in uso.
3. Utilizzo dell'Applicazione di Storyboarding Una volta completata la fase preliminare, l'utente procede con l'avvio dell'applicazione principale. In questa fase, il sistema

guida l'utente attraverso un tutorial interattivo, che illustra passo dopo passo tutte le funzionalità disponibili. Questo approccio garantisce un apprendimento graduale e completo delle feature offerte dall'applicazione. Al termine del tutorial, l'utente può scegliere tra i due script proposti (casi di studio) e procedere con la loro implementazione, utilizzando l'applicazione in modo personalizzato. Dopo aver completato il lavoro di storyboarding, l'applicazione genera un file HTML contenente i risultati finali, offrendo una visualizzazione chiara e accessibile degli storyboard prodotti.

4. Valutazione Post-Test Al termine dell'utilizzo dell'applicazione, l'utente compila una serie di questionari strutturati per valutare: L'efficacia complessiva dell'applicazione. La facilità d'uso percepita durante tutte le fasi del test. Questi feedback sono essenziali per misurare la soddisfazione degli utenti e identificare eventuali aree di miglioramento.

5.3 Tipologia dei questionari

Per valutare l'efficacia e la facilità d'uso dell'applicazione sviluppata, sono stati adottati due strumenti di valutazione riconosciuti come standard nello stato dell'arte: il NASA Task Load Index (NASA-TLX) e il System Usability Scale (SUS). Questi questionari rappresentano un metodo consolidato per l'analisi dell'usabilità e del carico di lavoro mentale associato all'utilizzo di applicazioni software.

5.3.1 NASA-TLX

Il NASA-TLX [2] è uno strumento sviluppato dalla NASA per misurare il carico di lavoro mentale percepito durante l'esecuzione di un compito. È ampiamente utilizzato in discipline come ergonomia, ingegneria dei fattori umani e psicologia del lavoro. Il questionario valuta sei dimensioni del carico di lavoro:

Domanda Mentale (Mental Demand): Quanto sforzo mentale e percettivo è stato richiesto? Domanda Fisica (Physical Demand): Quanto sforzo fisico è stato necessario? Domanda Temporale (Temporal Demand): Quanto eri sotto pressione per il tempo? Prestazione (Performance): Quanto bene pensi di aver svolto il compito? Sforzo (Effort): Quanto duramente hai dovuto lavorare per raggiungere il livello di prestazione desiderato? Frustrazione (Frustration): Quanto ti sei sentito stressato, irritato o frustrato? I partecipanti valutano ogni dimensione su una scala continua (ad esempio, da 0 a 100), e i punteggi vengono combinati per ottenere un indice complessivo del carico di lavoro percepito. Il NASA-TLX consente di individuare quali aspetti specifici del compito contribuiscono maggiormente al carico di lavoro, facilitando interventi mirati per ottimizzare l'esperienza utente.

5.3.2 SUS

Il SUS [7] è un questionario costituito da 10 affermazioni a cui gli utenti rispondono utilizzando una scala Likert a 5 punti, che va da "Fortemente in disaccordo" a "Fortemente d'accordo". Le affermazioni valutano vari aspetti dell'usabilità del sistema, tra cui:

Percezione della complessità: "Trovo il sistema inutilmente complesso." Facilità d'uso: "Penso che il sistema sia facile da usare." Sicurezza e fiducia: "Mi sentirei sicuro nell'usare il sistema da solo." Il punteggio complessivo del SUS, derivato da queste affermazioni, fornisce una misura quantitativa dell'usabilità del sistema. Valori elevati indicano un sistema intuitivo e facile da utilizzare, mentre punteggi più bassi suggeriscono aree di miglioramento.

Capitolo 6

Sviluppi Futuri

Il presente progetto di tesi si è focalizzato sull'aggiunta di funzionalità avanzate per il riconoscimento degli oggetti e l'identificazione dello spazio in realtà aumentata, con l'obiettivo di migliorare l'esperienza di storyboarding. Tale approccio ha aperto diverse possibilità di sviluppo, molte delle quali possono essere ulteriormente ampliate ed esplorate al fine di incrementare la velocità e l'immersione che l'applicazione fornisce.

In questo capitolo verranno analizzate alcune delle direzioni più promettenti per il proseguimento della ricerca. In particolare, si porrà l'attenzione sul progetto ARIA, una piattaforma di ricerca finanziata e supportata da Meta, dedicata allo sviluppo di algoritmi egocentrici basati sull'intelligenza artificiale [3].

Tra le prospettive future si potrebbe considerare un miglioramento dell'interfaccia di interazione dell'applicazione, con l'obiettivo di introdurre rotazioni più granulari e al contempo aumentare l'accessibilità per gli utenti. Inoltre, si potrebbe estendere le funzionalità dell'applicazione, passando dalla semplice creazione di uno storyboard alla generazione di un animatic, offrendo così un maggiore supporto alla fase di pre-visualizzazione nei processi creativi.

6.1 progetto ARIA e posa dei personaggi

Come accennato nell'introduzione al capitolo, il progetto ARIA è un'iniziativa di ricerca dedicata allo sviluppo di algoritmi egocentrici basati sull'intelligenza artificiale. Si tratta di un progetto finanziato da Meta, che coinvolge sia gruppi di ricerca interni all'azienda sia comunità di ricerca esterne. Per favorire l'innovazione e il progresso in questo ambito, il progetto ha reso disponibili una serie di dataset e strumenti open source, consentendo così la sperimentazione e lo sviluppo di soluzioni avanzate. Inoltre, vengono periodicamente organizzate sfide finanziate per stimolare la risoluzione di problemi comuni legati agli algoritmi egocentrici.

Una delle risorse principali messe a disposizione dal progetto è l'ARIA Research Kit, un toolkit progettato per la raccolta di dati egocentrici. Questo kit include gli occhiali ARIA, dispositivi di realtà aumentata che consentono di acquisire dati egocentrici,



Figura 6.1. Occhiali virtuali progetto ARIA

un'applicazione per la revisione dei dati raccolti e una piattaforma per lo sviluppo di applicazioni basate sulle funzionalità offerte dagli occhialini.

Attualmente, il progetto ARIA comprende una serie di dataset ampiamente riconosciuti nel panorama della ricerca sull'intelligenza artificiale, tra cui:

Aria Everyday Activities (AEA): una versione aggiornata del primo Pilot Dataset di ARIA, integrata con nuovi strumenti e dati di localizzazione, pensata per accelerare i progressi nella percezione artificiale e nell'intelligenza artificiale. Aria Digital Twin (ADT): un dataset che combina dati del mondo reale con una controparte digitale altamente accurata e completamente annotata. Aria Synthetic Environments (ASE): un dataset sintetico generato proceduralmente per favorire la ricerca su larga scala nell'ambito del machine learning. HOT3D: un nuovo dataset di riferimento per lo studio delle interazioni mano-oggetto in 3D basate sulla visione. Nymeria: un dataset egocentrico multimodale su larga scala, progettato per l'analisi del movimento del corpo umano nella sua interezza.

Lo sviluppo di algoritmi egocentrici rappresenta un settore di ricerca di grande interesse perchè può essere facilmente utilizzato anche nel progetto di tesi per ampliare le capacità di scene understanding offerte. Attualmente infatti, il dispositivo consente la cattura di informazioni relative allo spazio circostante e l'etichettatura di oggetti di dimensioni medie e grandi. Tuttavia, si intravedono ulteriori possibilità di sviluppo, come l'identificazione di oggetti di dimensioni più ridotte o l'aggiornamento in tempo reale della posizione degli

oggetti, migliorando significativamente le funzionalità di rilevamento e interazione possibili all'interno dell'applicazione.

6.2 Miglioramento delle interfacce per la manipolazione degli oggetti

Per migliorare la facilità d'uso dell'applicazione, si potrebbe intervenire sull'intuitività delle interfacce utente, con l'obiettivo di consentire anche agli utenti meno esperti di comprendere rapidamente il funzionamento e la struttura del sistema. Parallelamente, tale intervento avrebbe lo scopo di rendere le interazioni più semplici e controllabili.

Attualmente, il progetto si basa sull'utilizzo del componente XR Grab Interactable, che replica i controlli manuali che un utente avrebbe nel manipolare un oggetto reale. Sebbene questo approccio sia efficace per simulare un'interazione naturale, gli utenti alle prime armi potrebbero incontrare difficoltà nel manipolare gli oggetti in modo preciso e dettagliato.

Una possibile direzione di sviluppo consiste nell'introduzione di oggetti manipolabili con maggiore facilità, ad esempio attraverso l'utilizzo di interazioni semplificate o metafore di interazione [citazione]. Tali soluzioni potrebbero ridurre significativamente la curva di apprendimento e rendere l'applicazione più accessibile a una platea di utenti più ampia.

Questi interventi permetterebbero non solo di migliorare l'usabilità, ma anche di aumentare l'efficienza e la precisione delle interazioni, favorendo un'esperienza utente più fluida e soddisfacente.

6.3 Animatic

Attualmente, il sistema in realtà aumentata sviluppato consente la cattura di immagini utili per la creazione dello storyboard. Tale sistema potrebbe essere ulteriormente potenziato, integrando la possibilità di sfruttare animazioni e azioni dei personaggi per generare un animatic.

L'animatic è definito come una trasposizione animata dello storyboard. Questo strumento permette di previsualizzare graficamente la sceneggiatura attraverso una sequenza di immagini temporizzate o di video, fornendo una rappresentazione dinamica del flusso narrativo. È particolarmente utile per valutare se le temporizzazioni delle scene chiave di un film risultano coerenti e se il messaggio che si intende comunicare è trasmesso in maniera ottimale. L'animatic offre inoltre il vantaggio di integrare la durata delle singole scene e il ritmo delle azioni direttamente al suo interno, facilitando una pianificazione precisa.

Attualmente, l'applicativo sviluppato include uno slider che consente di impostare la durata delle singole azioni, rendendo possibile la creazione di un animatic che visualizzi ogni vignetta per il tempo prestabilito. Tuttavia, questo approccio non fornisce una rappresentazione completa delle azioni in corso. Una potenziale direzione di sviluppo consiste nel permettere, oltre alla creazione di immagini, anche quella di video attraverso l'uso di camere virtuali.

Al momento, Meta limita l'accesso alle registrazioni video dell'ambiente circostante per le applicazioni di terze parti, rendendo non praticabile l'implementazione di un sistema di camere virtuali. Tuttavia, data la possibilità che in futuro tali restrizioni vengano rimosse, sarà plausibile accedere a questi dati per generare animatic che rappresentino l'ambiente circostante tramite un sistema di camere virtuali configurabili dall'utente.

Un ulteriore passo avanti potrebbe essere l'introduzione di uno strumento di editing video all'interno dell'applicativo, consentendo agli utenti di modificare e montare l'animatic secondo le loro esigenze. Questo tipo di funzionalità aumenterebbe significativamente la flessibilità e l'utilità del sistema.

La possibilità di creare un animatic, in aggiunta allo storyboard, offrirebbe un notevole risparmio di tempo nella fase di pre-produzione, fornendo strumenti essenziali per pianificare in modo più efficace le riprese e migliorare la qualità del prodotto finale.

Bibliografia

- [1] Cos'è la segmentazione delle istanze?
- [2] andra G. Hart. *P. A. Hancock N. Meshkati (Eds.), Human Mental Workload*. NASA-Ames Research Center, 1988.
- [3] Aria. Project aria research. URL https://facebookresearch.github.io/projectaria_tools/docs/intro.
- [4] JOHN HART. «*Basic Components and Principles of the Storyboard*». Focal Press, 2007.
- [5] JOHN HART. «*Basic Components and Principles of the Storyboard*». Focal Press, 2008.
- [6] WANCHAO SU HUI YE, KIN CHUNG KWAN and HONGBO FU. Aranimator: In-situ character animation in mobile ar with user-defined motion gestures. 2020.
- [7] James R. Lewis. *The Factor Structure of the System Usability Scale*. Springer, 2009.
- [8] Satyam Maitrai. The history of storyboarding, 7/04/24.
- [9] Masahito Hirakawa Manabu Eitsuka. Authoring animations of virtual objects in augmented reality-based 3d space. 2013.
- [10] Jun Park Midieum Shin, Byung-soo Kim. Ar storyboard: An augmented reality based interactive storyboard authoring tool. 2013.
- [11] Mixamo. Mixamo website. URL <https://www.mixamo.com/#/m>.
- [12] Online. Le fasi della produzione cinematografica, 2024.
- [13] Online. «pre-production – everything you need to know», 2024.
- [14] Online, 2024.
- [15] Princeton. A lexical database for english. URL <https://wordnet.princeton.edu>.
- [16] Mar Gonzalez-Franco Ramanpreet Kaur. Story creatar: a toolkit for spatially-adaptive augmented reality storytelling. 2021.
- [17] Sketchfab. Sketchfab website. URL <https://sketchfab.com/feed>.
- [18] Unity. Unity website. URL <https://unity.com>.